Universidade do Minho
Escola de Engenharia
Departamento de Informática

Daniel Almeida Carvalho

On Conditional Quantum Control

January 2022

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Daniel Almeida Carvalho

On Conditional Quantum Control

Master dissertation
Integrated Master's In Physics Engineering

Dissertation supervised by
José Nuno Oliveira (U. Minho)
Rui Soares Barbosa (INL)

January 2022

ACKNOWLEDGEMENTS

First and foremost I am thankful to my parents for the constant encouragement and patience.
I am also grateful for the patience and understanding of my supervisor José Nuno Oliveira and the joviality and lightheartedness with which I was met in all our interactions.
Likewise I am in debt to my co-supervisor Rui Soares Barbosa for the many clarifications and corrections to my mistakes.
Lastly I thank and acknowledge the financial support of INESC TEC.

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.

ABSTRACT

The purpose of this dissertation is to ascertain the feasibility of *quantum control*. This is a rather informal concept, however in the context of this work it simply means the control flow of a quantum program being performed without resorting to measurements and without being mediated by an external classical computer.

The approach consists in providing a definition of a conditional like statement which encapsulates the intended behaviour. The definition is given as a specific morphism in a *Biproduct Dagger Compact Closed Category*. These were introduced by Abramsky and Coecke (2004) as a semantic framework capable of expressing the axioms of closed system finite dimensional Quantum Mechanics. Later this framework was extended to capture Open System Quantum Mechanics (Selinger, 2007). As a consequence, and as it pertains to this dissertation, the construct presented in this work has an interpretation for both closed quantum systems (category of finite-dimensional Hilbert spaces and Linear maps) and open quantum systems (category of finite-dimensional Hilbert spaces and Completely Positive maps).

What was found is that in closed quantum systems the proposed construct transpires the idea of *superposition of programs* as conceptualized in previous works on the matter (Bădescu and Panangaden, 2015) (Ying et al., 2014), which gives validity to the notion of quantum control at least in this context. On the other hand, in open quantum systems the meaning of the conditional statement proposed takes the form of *probabilistic branching* dependent on the probability distribution of a *bit*.

Finally a comparison is made between this work and the one carried out by Bădescu and Panangaden (2015) in the context of the QPL programming language (Selinger, 2004), which concludes with a discussion about the incompatibility of quantum control in programming languages whose semantics are based on the open quantum system formalism.

KEYWORDS: QUANTUM CONTROL, BIPRODUCT DAGGER COMPACT CLOSED CATEGORIES, CLOSED QUANTUM SYSTEMS, OPEN QUANTUM SYSTEMS.

RESUMO

Esta dissertação tem como propósito averiguar a viabilidade de *controle quântico*. Sendo este de um certo modo um conceito informal, no contexto desta dissertação significa simplesmente o controle do fluxo da execução de um programa quântico sem recorrer a medições e sem ser mediado por um computador clássico externo.

A abordagem consiste em especificar a definição de uma expressão condicional que encapsule o comportamento desejado. Esta definição é dada como um determinado morfismo numa categoria *Biproduct Dagger Compact Closed*. Estas foram introduzidas por Abramsky and Coecke (2004) como modelo semântico capaz de expressar os axiomas da Mecânica Quântica para sistemas fechados finitos. Mais tarde este modelo foi expandido de forma a poder encorporar sistemas abertos (Selinger, 2007). Consequentemente, e no que diz respeito a esta dissertação, a construção aqui apresentada terá uma interpretação tanto para sistemas fechados (categoria de espaços de Hilbert de dimensão finita e transformações lineares) como para sistemas abertos (espaços de Hilbert de dimensão finita e mapeamentos completamente positivos).

Para sistemas quânticos fechados foi concluído que a construção proposta transparece a idéia de sobreposição de programas conceitualizado em investigações anteriores acerca do assunto (Bădescu and Panangaden, 2015) (Ying et al., 2014), o que valida a noção de controle quântico pelo menos neste contexto. Por outro lado, para sistemas quânticos abertos a expressão condicional sugerida toma a forma de "braching" probabilistico dependente da distribuição probabilistica de um bit.

Por fim, é feita uma comparação entre este e o trabalho realizado por Bădescu and Panangaden (2015) no contexto da linguagem de programação QPL (Selinger, 2004), que conclui numa discussão sobre a imcompatibilidade de controle quântico em linguagens de programação cuja semântica é baseada em sistemas quânticos abertos.

PALAVRAS-CHAVE: CONTROLE QUÂNTICO, CATEGORIAS BIPRODUCT DAGGER COMPACT CLOSED, SISTEMAS QUÂNTICOS FECHADOS, SISTEMAS QUÂNTICOS ABERTOS.

CONTENTS

1

INTRODUCTION

1.1 CONTEXT

In recent history the most consistent factor shaping societies has been technological development. In turn, at the most fundamental level technological development has been the byproduct of the success of natural sciences. This success rests on a century-long agreement about methodology. Based on rationalism and empiricism, the scientific method can naively be described by a simple recursive procedure: (1) Creation of a consistent predictive model of phenomena in the world; (2) Observation of phenomena; (3) Does the observation match the predictions resulting from the model? Yes (2), No (1).

Naturally, this never-ending process carries itself out differently within the natural sciences. But it seems that the rigor and quantitative predictive capacity of a description of natural phenomena are highly dependent on its ability to escape the ambiguity of natural language and maximize the use of formal languages, i.e., Mathematics. As a language for scientific models, Mathematics serves two fundamental purposes. The first is to test the internal consistency of some proposed model of natural phenomena, and the second is to determine whether new observations contradict the current accepted model.

Among the natural sciences, Physics has arguably the strongest mathematical basis, hence its unmatched ability to make rigorous and quantitative models of reality with remarkable predictive power. These models - Newtonian Mechanics, Thermodynamics, Electrodynamics - eventually drove the first and second industrial revolutions. Then, with the subsequent development of Quantum Mechanics and Semiconductor Physics, the computer revolution was made possible. Following the arrival of the computer age, technological advancements have not come in general as the result of fundamental discoveries in Physics but mainly through the evolution of software and the continuing sophistication of computer systems, stimulating consequently a considerable amount of research directed towards understanding computation.

Computation, although intrinsically tied to physical realizability, was born purely out of deeply foundational mathematical motivations, namely the desire to formalize the notion of what was at the time called *effectively calculable procedure* (what is now termed an algorithm), in order to solve the *Entscheidungsproblem*. The problem posed by Hilbert asks, roughly, whether for a given first order logic statement one can find *such* a procedure determining if the statement follows from the axioms, i.e., if the statement is a theorem of the system. This pursuit drove Turing to formalize effectively calculable procedures as those which can be performed by Turing Machines (Turing, 1937), which he then termed as computable functions. Around the same time Church

developed the lambda calculus (Church, 1936b) with much the same purpose. Both of these models turned out to be equivalent, in other words, they defined the same set of (computable) functions. These equivalence seemed to provide sufficient robustness to the concept of computable function so that an overall agreement over the meaning of effectively calculable procedure was reached. Consequently both Turing and Church answered the *Entscheidungsproblem* in the negative. Turing's approach, in turn, based on posing the *Entscheidungsproblem* in terms of the Halting problem, led him to define Universal Turing Machines, which were able to take as input instruction sets encoding other Turing Machines, thus laying the fundamental conceptual ground for the modern programmable computer.

Since those early days, Computer Science has branched out to include a large range of topics such as the theory of computation, programming language theory, algorithms and data structures, complexity theory, cryptography, artificial intelligence and countless others. These aim at developing abstract mathematical models of computational processes independent of machine implementation.

A simultaniously fundamental and practical problem in Computer Science is of ensuring that a program meets the specification of the problem it purports to solve. One possible approach in dealing with this question would be by testing the algorithm on inputs to which the corresponding outputs are known beforehand. There are technical reasons (cf. computational complexity) attesting the inadequacy of this approach, yet fundamentally the problem lies in confusing the correction of a program with a scientific problem, rather than a mathematical one. For, even though a program must ultimately be implemented by a physical process in a physical computer, a program is a transformation of abstract representations (Integers, Strings, Lists, matrices, ...), and thus its correction can only be attested by mathematical proof. As a result, one must develop mathematical semantic frameworks able to capture the meaning of both the program and its intended behaviour.

In most cases programs are expressed through programming languages whose expressions make up high level representations of composite instructions to be executed as machine code. These expressions should hopefully represent data/data manipulation abstractions relevant to the problem to which the programmer sets out to solve. Therefore the meaning of a program uniquely corresponds to the meaning of the expressions comprising it. This meaning can either be, as in most programming languages, informal and as such unconsciously intuited by the programmer, or formal, in which case to every expression of the language one attributes a mathematical object representing it. In practice, this might be too radical a distinction, given that most languages occupy a position on the spectrum of semantic formality. This spectrum is largely determined by the strength of what are called *type systems*.

A type system indirectly imparts meaning on the expressions of a programming language by imposing additional syntactic restrictions on them. This process unfolds by attributing to each expression a type dictating the set/form of computations this expression is subject to (and conversely those to which it is not). Moreover, a type system also determines a set of rules governing the interaction between all of its types. Therefore a type system imposes on the set of all programs in a language a logical structure determining the permissible ways in which smaller and smaller programs can be composed to form larger ones. In the case of statically-typed languages, the adherence to this logical structure can be checked by a decision procedure before runtime. This in itself can be considered a weak form of program verification, in the sense that it reduces the possibility of writing wrong

programs. A program's meaning is in large part implicit in the data structures (types) it employs, therefore, the richer the type system, that is the bigger number of type restrictions imposed on a language's expressions, the more resolution one has over the meanings of programs.

A very influential type system is the typed lambda calculus (Curry et al., 1958), which is a version of the lambda calculus where every term has a type. The typed lambda calculus served as inspiration for the type system of a number functional programming languages. A program in said languages are based on function application, rather than updating states. Moreover, every piece of data has a type, and by composing functions with matching input and output types one constructs larger and larger programs. In this paradigm, if one wishes to prove that a program behaves in a certain away, it suffices to prove some property about functions. Furthermore, compositionality implies that properties of larger programs can be deduced by the properties of its constituents.

Much of the developments of functional programming have been directly tied to concepts which stem from Category Theory (Mac Lane, 2013). Category theory is the theory of abstract structure. It deals with categories and the relations between categories. Sets (and functions) constitute a category, as do groups (and group homomorphisms), vector fields (and linear transformations), topological spaces (and continuous maps), sets (and relations), data types (and programs), and countless others. Category theory can be viewed as a meta mathematical theory, as it establishes a variety of patterns common to a variety of mathematical fields. A theory that supports this level of generality must be stated in very abstract terms, thus it should be, at least at first glance, very far away from practical considerations. Yet it has contributed with significant insights into the semantics of programming languages. This fact comes about due to a remarkable correspondence between a certain class of categories (Cartesian Closed Categories) and the typed lambda calculus (Lambek, 1980). This correspondence allows one to reason about a type system (and thus a programming language) by reasoning about an appropriately chosen category.

1.2 QUANTUM COMPUTING

As already mentioned the theoretical foundations of computation were laid in response to a fundamental mathematical problem. On the other hand, quantum computation was first considered in 1981 by Richard P. Feynman (2018) as an attempt to determine whether Physics, namely Quantum Mechanics, can be efficiently simulated. In this context, the meaning of efficiently can be inferred by the following remark made by Feynman: "if doubling the volume of space and time means I'll need an exponentially larger computer, I consider that against the rules". Classical computing did not meet this requirement, leading Feynman to conjecture the existence of a computer of "quantum mechanical elements" which would scale proportionally with an increase in size of the system to be simulated. Such elements would constitute a supposed universal quantum system able to (efficiently) simulate any other quantum system.

Building on this idea, David Deutsch eventually put quantum computing on more formal grounds with the definition of a Quantum Turing Machine (Deutsch, 1985). In the subsequent years, a number of quantum algorithms were developed, exploiting quantum mechanical properties such as superposition, quantum interference, and entanglement. These algorithms have been shown to offer an advantage over their classical counterparts. For

instance, Grover's search algorithm (Grover, 1996) provides a quadratic speed up as compared to the best available classic search algorithms. A more significant example is Shor's polynomial-time prime factoring algorithm (Shor, 1994), which has serious implications as it relates to the breaking of some cryptographic protocols, since they are designed with the assumption that prime factorization is a difficult problem (solved in exponential-time). These and other developments fostered a considerable amount of research in quantum computing that continues to this day, both at the experimental level, where attempts to build a working quantum computer are multiplying, and at the theoretical level, where the pursuit for clearer understanding of this new computing paradigm has been continually intensifying.

From the time of its advent to the present day, the field of quantum computing has branched out into several sub-fields – Quantum Simulation, Quantum Cryptography, Quantum Machine Learning, Quantum Error Correction, Quantum Complexity Theory, to name just a few – bringing to the forefront new challenges, some of which owe their novelty to this unique paradigm, while others are inherited from classical computing with a new light cast on them. One of the latter is the problem of program correction, which becomes even more of an issue given the fact that the measurement problem makes it impossible to test the state of a program.

Hoping to replicate the successes that category theory has had on classical computation, one could aspire to apply it to this not-so-recent discipline. Albeit it could be argued that one can already reason about quantum computing in the Hilbert space formalism. But it would seem unlikely that, if the day comes in which quantum computational systems are widespread and of large scale, one would reason about them in such a low-level representation. Instead, the development of a semantic framework could be foreseen whose main compenents would be key quantum informational concepts such as resource sensitivity, superposition, entanglement, flow of classical and quantum information and measurement. Started by Abramsky and Coecke (2004), the field of categorical quantum mechanics is an effort in doing precisely that. The approach is to reformulate finite dimensional quantum mechanics as typed process theory by defining a general class of category, called *Biproduct Dagger Compact Closed Categories*, in which it is possible to state the postulates of quantum mechanics. The hope is that this reformulation is more transparent to the computational aspects of quantum mechanics.

1.3 AIMS OF THE DISSERTATION

One of the most essential building blocks of any programming language are control structures. These constitute a set of instructions which allow a program to perform decision making at run time, enabling a program to bifurcate, bypass sections of code or even repeat them. The most common of these structures is the conditional statement

$$\textbf{if } p\ x \textbf{ then } g\ x \textbf{ else } h\ x \tag{1.1}$$

which executes some test on some data and performs one of two instructions according to the test result. In a functional programming language conditional statements can be modeled categorically in the context of *distributive categories* (Bird and De Moor, 1996) (Gibbons, 1997).

In the design of quantum programming languages the predominant consensus has been to bestow the control of execution on a classical computer. This classical computer has access to a set of quantum data and

instructions. A program usually unfolds by repeating a sequence of quantum instructions acting on quantum data an amount of times which can either be predetermined by the controller or influenced only by means of a measurement. A prime example of this paradigm is the QPL programming language (Selinger, 2004).

One may however ponder a paradigm in which control might be quantum. An early effort in this direction was made by Altenkirch and Grattage (2005). In it, the authors describe a functional quantum programming language which includes a conditional statement akin to 1.1 which abstains from measurement, where x now refers to a quantum state. This construct is subject to a restraint which states that $g\ x$ and $h\ x$ should be orthogonal. This condition serves to ensure reversibility, since if it were otherwise the case that the space representing the branches of the conditional had a non-empty overlap then one could not in general determine which branch actually took place, hence breaking reversibility.

This restriction can be avoided if both g and h are conditioned by x but do not act on x, and instead act on an additional system. This was the approach taken by Bădescu and Panangaden (2015) and Ying et al. (2014). The resulting construct captures the notion of "superposition of programs".

The main objective of this dissertation is to inspect the viability of such a notion. As such, it is heavily based one both of these works. The difference lies mainly in the approach. Here a definition of a general conditional statement is provided at the level of Biproduct Dagger Compact Closed Categories, which will result in a subsequent definition in the specific categories of Hilbert spaces with either linear transformations or completely positive maps. The hope is that this will provide structural insights into the differences between the standard formalism of Quantum Mechanics and the Density Matrix formalism of Quantum Mechanics as semantic models for quantum programming languages.

1.4 OUTLINE

This dissertation is divided in essentially two parts. The first, which includes chapter 2 through chapter 4, contains a review of the relevant background material for later chapters. Ultimately, the objective of this review is twofold:

- Describe the standard formalism of Quantum Mechanics in both the closed and open system interpretations.
- Review the recasting of Quantum Mechanics - in the closed system framework (Abramsky and Coecke, 2004) as well as in the open system framework (Selinger, 2007) - in the language of *Biproduct Dagger Compact Closed Categories*.

With this as specification, Chapter 2 is a basic overview of the axioms of Quantum Mechanics. The focus is set on the mathematical structures governing quantum systems and their dynamics. Moreover, the fundamental distinctions between the closed and open system frameworks are highlighted.

Chapter 3 consists of an overview of Category Theory. This chapter summarizes almost the whole of this dissertation's mathematical background. Its presentation is directed towards the concept of *Cartesian Closed Categories* and its connection with the *Typed Lambda Calculus*, so as to explain to some extent the importance of Category Theory in the semantics of Programming Languages.

Chapter 4 builds on the previous two chapters to give a review of the field of *Categorical Quantum Mechanics*. Its central concept is that of *Biproduct Dagger Compact Closed Categories* (Abramsky and Coecke, 2004). Additionally, the categorical construct (Selinger, 2007) which represents the passage from closed quantum systems to open quantum systems is presented.

The second part of this thesis is found in chapter 5. It starts with an important part of the inspiration for this work, namely the modelling of classical conditional statements in *distributive categories* (Gibbons, 1997). It proceeds by describing measurement-based control. Then finally a conditional-like statement aimed at instantiating quantum control is defined as a morphism in an Biproduct Dagger Compact Closed Category, which will then be interpreted in closed and open quantum systems. To conclude the chapter the comparison between this work and previous related work by Bădescu and Panangaden (2015) is discussed in the context of the QPL programming language (Selinger, 2004).

Finally, chapter 6 summarizes the main conclusions taken from this dissertation and indicates possible directions for future work.

2

QUANTUM THEORY: BASICS

As a start of discussion, the most appropriate course of action is to state precisely what is meant by quantum computation. In order to do so, the concepts of data, data transformation and the retrieval of the computational result must be formally defined. The Hilbert space formalism provides exactly a framework for this purpose. In this framework the data, represented by a ray in a Hilbert space is assumed to be isolated from its environment and thus the transformations it can undergo amount to no loss of information. As a result, data transformation is reversible. On the other hand, the retrieval of the computational result is achieved through an interaction between the data and a measuring system. This process is fundamentally indeterministic.

One can then informally describe quantum computation as a process starting with a preparation of a state in a Hilbert space, followed by reversible transformation, and ending with a measurement.[1]

The purpose of this chapter is to formalize this rather vague description. It is important to point out, right from the outset, that the approach to the subject matter will not be from a physical perspective but instead a mathematical one. Hence there will not be any mention of concepts such as the Schrödinger equation, energy levels, momentum and position operators, spin and so on. The focus will be instead on the mathematical structure of a quantum system and its dynamics.

2.1 CLOSED QUANTUM SYSTEMS

The validity of assuming the total isolation of the system is dependent on the circumstances. Systems for which this assumption is reasonable are called *Closed systems*. The mathematical description of closed quantum mechanics can be captured by four simple postulates that provide the answers to the following questions:

1. What is the state of a quantum mechanical system?
2. How do smaller systems combine to form larger ones?
3. How does a system evolve in time?
4. What is a quantum measurement and what is its effect on the system measured?

Before stating what a quantum system is, one first needs to introduce the concept of Hilbert space.

1 Obviously, nothing impedes the use of measurements in intermediate stages of the computation. But the principle of deferred measurement (Nielsen and Chuang, 2002) guarantees that these can always be moved to the end of the computation.

Definition 2.1. *A finite-dimensional Hilbert space H is a finite-dimensional vector space over complex numbers, which is equipped with an inner product operation $\langle\ |\ \rangle : H \times H \to \mathbb{C},\ (x,y) \mapsto \langle x|y\rangle$ which is required to satisfy*[2]

- $\langle \boldsymbol{x}|\boldsymbol{y}\rangle = \langle \boldsymbol{y}|\boldsymbol{x}\rangle^*$.
- $\langle \boldsymbol{x}|\boldsymbol{x}\rangle \geq 0$ *and* $\langle \boldsymbol{x}|\boldsymbol{x}\rangle = 0$ *iff* $\boldsymbol{x} = 0$.
- $\langle \boldsymbol{x}|c_1\boldsymbol{y} + c_2\boldsymbol{z}\rangle = c_1\langle \boldsymbol{x}|\boldsymbol{y}\rangle + c_2\langle \boldsymbol{x}|\boldsymbol{z}\rangle$.

for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in H$ and $c_1, c_2 \in \mathbb{C}$, and where $(\ )^*$ denotes the complex conjugate. The inner product induces the norm $||\boldsymbol{x}|| = \sqrt{\langle \boldsymbol{x}|\boldsymbol{x}\rangle}$.

A *ray* in a Hilbert space is an equivalence class of unit vectors which differ by a phase factor ($|x\rangle \sim e^{i\theta}|x\rangle$). The concept of ray captures the fact that $|x\rangle$ and $e^{i\theta}|x\rangle$ describe the same physical system, i.e., all physical quantities obtained by measuring one coincide with those obtained by measuring the other.

Postulate 2.1. *The state of a quantum system is given by a ray in a Hilbert space.*

A n-dimensional quantum state $|\psi\rangle \in \mathbb{C}^n$ has the general form

$$|\psi\rangle = \sum_{i=0}^{n-1} \alpha_i |i\rangle, \quad \sum_{i=0}^{n-1} |\alpha_i|^2 = 1$$

for an *orthonormal* $\{|i\rangle\}_0^{n-1}$ of $\mathbb{C}^n$. For $n = 2$, we have the simplest quantum system - the qubit - which has the form

$$|q\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle, \quad |\alpha_0|^2 + |\alpha_1|^2 = 1$$

This is interpreted as follows: If a measurement - in the $\{|0\rangle, |1\rangle\}$ basis - is performed on $|q\rangle$ the outcome will be either $|0\rangle$ with probability $|\alpha_0|^2$ or $|1\rangle$ with probability $|\alpha_1|^2$.

Given that a quantum system is described by unit vectors in a Hilbert space, the dynamics of quantum systems must be described in turn by a mappings between such vectors. These mappings are called isometries (A.19) and the subclass of those that are deterministic are called unitary (A.19).

Postulate 2.2. *The deterministic evolution of a closed quantum system is described by a unitary transformation. That is, the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2,*

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

The next logical step is to answer the question of how individual quantum systems are mathematically described when considered as a single system.

2 We adopt the *Dirac* notation in which the ket $|x\rangle$ is a vector in H and the bra $\langle x|$ is a linear functional $H \to \mathbb{C}$.

Postulate 2.3. *Let H_0 and H_1 be Hilbert spaces, by A.25 $H_0 \otimes H_1$ is also a Hilbert space. Then given two quantum systems described by their respective Hilbert spaces H_0 and H_1, their compound system is described by $H_0 \otimes H_1$. Therefore a state of the compound system is given by a ray in $H_0 \otimes H_1$.*

Thus, a general state of the compound system has the form

$$|\psi\rangle = \sum_{i,j=0}^{|H_0|,|H_1|} c_{ij}|i\rangle \otimes |j\rangle. \quad |\psi\rangle = \sum_{i,j=0}^{|H_0|,|H_1|} |c_{ij}^2| = 1$$

Upon considering the last equation, it becomes immediately clear that the state of a compound quantum system cannot, in general, be expressed as a (tensor) product of states representing their corresponding parts of the system. This fact allows for the existence of states whose parts are correlated. An instance of this is the maximally entangled state,

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle).$$

This state possesses the property that once a measurement is performed locally on the first system, the result of a subsequent measurement - in the same basis as the previous one - on the second system is completely determined. This property can be exploited in order to indirectly measure a quantum system. Usually this proceeds by entangling the system with an auxiliary system and then performing a measurement on the auxiliary system, allowing one to infer the state of the system.

Postulate 2.4. . *A measurement on a quantum system is defined by a self-adjoint operator $A = \sum_i a_i P_i$, where the set of eigenvalues $\{a_i\}$ represents the possible measurement outcomes, while the operators $\{P_i\}$, called projectors[3], represent the state change due to the measurement. A quantum measurement unveils as follows*

1. *the initial state change undergoes one of the transitions*

$$|\psi\rangle \mapsto \frac{P_i|\psi\rangle}{\sqrt{p_i}}$$

and the probability of this transition occurring is

$$p_k = \langle\psi|P_i|\psi\rangle$$

2. *The observer receives the value a_i as a token witness of the measurement.*

This four postulates are only viable in very few practical situations and in theoretical considerations. They became unreasonable when interaction between the quantum system and its environment is inevitable. In such cases, this interaction produces uncertainty in the description of the system. Hence it does not take into account decoherence, therefore situations in which there is noise and complete knowledge of the systems state is impossible are not captured. A new state description is therefore needed. One in which, the state is no longer a ray in

3 Projectors are bounded linear maps $P : H \to H$ such that $P_i \circ P_i = P_i$, $P_i = P_i^\dagger$ *and* $P_i \circ P_j = \delta_{ij}$.

a Hilbert space (*pure state*), but some probability distribution dependent one the environments state, reflecting the possible/likely set of pure states the quantum system might be in. These are called *mixed states*.

2.2 OPEN QUANTUM SYSTEMS

Quantum systems susceptible to noise are called *Open Quantum Systems*. The description of the state of a open quantum system takes into account the uncertainty born out of the interaction between the state under observation S and the environment E. This interaction introduces correlations between the two systems, making it impossible to arrive at a complete description of the system of interest S without a reference to the environments state E. Thus any description of S must be a probability distribution of pure states of S, dependent on description of E and its interaction with S. This description is encapsulated mathematically in the *density operator*.

Definition 2.2. *A density operator $\rho \in L(H)$ is a linear bounded operator such that:*

- *unit trace: $Tr\ \rho = 1$*
- *self-adjoint: $\rho = \rho^\dagger$*
- *positive: $\langle x|\rho x\rangle \geq 0$ for all $x \in H$.*

Positivity implies that the eigenvalues of the density operator should be positive and the trace condition ensures that these sum up to the unity. This suggests an interpretation of the density operator as a probability distribution of pure states. Specifically, each of its eigenvalues are associated with a probability, and their corresponding eigenvectors with a pure state. Thus, for a set of normalized (not necessarily orthogonal) states $\{|\psi\rangle\}_i \subset H$, a density operator has the general form

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

subject to

$$Tr\ \rho = \sum_i p_i = 1.$$

This representation, however, is not unique. In fact, for any given density operator there is an infinite number of *ensembles* $\{p_i, |\psi_i\rangle\}$ giving rise to it. The following example illustrates this point: Consider the states

$$|\psi\rangle_0 = |0\rangle, \qquad |\psi\rangle_1 = |1\rangle, \qquad |\psi\rangle'_0 = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \qquad |\psi\rangle'_1 \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle),$$

and the ensembles

$$\rho = \frac{1}{2}(|\psi\rangle_0\langle\psi| + |\psi\rangle_1\langle\psi|) \qquad\qquad \rho' = \frac{1}{2}(|\psi\rangle'_0\langle\psi| + |\psi\rangle'_1\langle\psi|)$$

After a bit of algebra one concludes that these two ensembles represent the same density operator, i.e, $\rho = \rho'$.

Postulate 2.5. *The state of an open quantum system is given by a density operator in* $L(H)$.

As for combined systems, their description in open quantum systems are constructed similarly to in closed quantum systems.

Postulate 2.6. *Given two open quantum systems described by their respective Hilbert spaces* H_0 *and* H_1*, a state in their compound system is given by a density operator* $\rho \in L(H_0 \otimes H_1)$.

To accommodate this new definition of quantum state it is necessary to revise the notion of quantum state dynamics. One may consider this evolution as a map taking density operators to density operators. As such this evolution must be described by a map between linear bounded operators that preserve the probability distribution interpretation. This map is referred to as a $superoperator$.

Definition 2.3. *A linear map* $F : L(H) \rightarrow L(H)$ *is a superoperator if for all density operators* $\rho \in L(H)$ *the following hold.*

- *Convex-linear :* $F(\sum_i p_i P_i) = \sum_i p_i F(P_i)$.
- *Trace-preserving :* $Tr(F(\rho) = Tr(\rho)$
- *Completely-positive : Whenever* $\rho \in L(H \otimes K)$ *then* $(F \otimes id_{L(K)})(\rho)$ *is positive.*

Intuitively this last condition states that F *must not only be positive but also, if we extend the system to include an additional subsystem* E*, and if the combined system remains separate then the operation should simply extend to a new superoperator* $F \otimes id_{L(K)}$*, which must also be positive.*

Postulate 2.7. *The most general evolution an open quantum system can undergo is given by a superoperator.*

It is noticeable that this description of quantum dynamics apparently does not differentiate deterministic and non-deterministic behaviour. That is due to both unitary evolution and measurements being instances of the general structure. Obviously these are subclasses and as such can be differentiated by providing further restrictions. This is best understood, however, in light of what is referred to as the *Kraus operator-sum representation* of superoperators.

Theorem 2.1. *(Kraus) Linear mapping* $\phi : L(H_n) \rightarrow L(H_n)$ *is completely positive if and only if there exists* n^2 *mappings* $K_i \in L(H_n)$ *such that*

$$\phi(\rho) = \sum_{i=1}^{n^2} K_i \rho K_i^*$$

for each $\rho \in L(H_n)$*. Moreover if*

$$tr(\phi(\rho)) = 1 \equiv \sum_i K_i^\dagger K_i = I$$

holds then it also a superoperator.

For the case of unitary evolution the set of (Kraus operators) is given by the singleton set $\{U\}$, reducing 2.1 to

$$\phi(\rho) = U\rho U^{\dagger}$$

where U is some unitary operator. In the case of a measurement, for instance in the computational basis, the set of Kraus operators is $\{P_0, P_1\}$, reducing 2.1 to

$$\phi(\rho) = P_0\rho P_0^{\dagger} + P_1\rho P_1^{\dagger}$$

where $P_0 = |0\rangle\langle 0|$ and $P_1 = |1\rangle\langle 1|$.

It is important to point out that the set of operators $\{K_i\}$ is not unique, i.e., different sets of Kraus operators may describe the same dynamical behaviour, the same superoperator. Whenever this is the case for any two Kraus representations they are said to be *extensionally equal*.

Theorem 2.2. *(Extensional equality of Kraus operators) (Nielsen and Chuang, 2002) Suppose* $\{E_1, ..., E_m\}$ *and* $\{F_1, ..., E_n\}$ *are sets of operators giving rise to superoperatos* $\mathcal{E}$ *and* $\mathcal{F}$*. By appending zero operators to the shorter set one may ensure that* $m = n$*. Then* $\mathcal{E} = \mathcal{F}$ *if and only if there exists complex numbers* u_{ij} *such that* $E_i = \sum_j u_{ij}F_j$*, where* u_{ij} *is an* m *by* m *unitary matrix.*

This liberty in the choice of Kraus operators reflects the variety of measurement approaches an experimenter may take to gather the same measurement outcomes.

2.3 SUMMARY

As summary it is important to contrast the different interpretation of quantum systems inherent in both models. In *Closed Quantum Systems*:

- A state is represented by unit vectors in a Hilbert space given generally by

$$|\psi\rangle = \sum_{i=0}^{n-1} \alpha_i |i\rangle,$$

  subject to the normalization condition

$$\sum_{i=0}^{n-1} |\alpha_i|^2 = 1$$

- Deterministic state dynamics occur by means of unitary operators.
- Measurement is described by a self-adjoint operator.

In *Open Quantum Systems*:

- A state is represented by density matrix. A probability distribution of Hilbert space unit vectors

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

subject to

$$Tr\,\rho = \sum_i p_i = 1.$$

- The general dynamics are described by a superoperator, which usually is dependent on the environment state.

An intuitive connection between these formalisms can be made by considering the following scenario:

- Two systems, one called the *principal* system, the other the *environment* system, come into contact with each other. It is assumed that at first they constitute a separate system i.e., $\rho = \rho_p \otimes \rho_e$.
- From then on they evolve unitarily. In other words as a closed system.
- Then the interaction between the two systems ceases. At which point the state of the environment is traced over[4]. Giving

  $$\rho_p(t) = tr_e[U(\rho_p \otimes \rho_e)U^\dagger],$$

  which is a valid description of the evolution of the principal system as an open quantum system.

The process just described elucidates two fundamental aspect pertaining to the relationship between the two models:

1. The dynamics of an open quantum system can be viewed as the dynamics of a larger closed system containing the environment.
2. If one's interest is to investigate one of the constituents of a compound system, one is in general forced, due to the nature of entangled states, to step into the open system framework. In other words, the partial trace of a compound system is in general a mixed state.

As for the fundamental differences of these two frameworks, they lie in their generality. As opposed to Closed Quantum Systems, Open Quantum Systems takes into account decoherence, hence scenarios where complete knowledge of the systems state is impossible can be dealt with. A crucial example of this is Quantum Error Correction, where noise must be taken into consideration. Although this will be immensely important in the eventual construction of a working quantum computer, in this dissertation the concern lies elsewhere. One is instead interested in these two formalism as semantic frameworks for quantum programming languages. In order to make this clear one first needs to introduce a bit of Category Theory.

4 This comes about through the application of the partial trace A.26.

3

CATEGORY THEORY: BASICS

Apart from the bit of Linear Algebra used in the previous chapter, this dissertations mathematical basis is grounded almost exclusively within the confines of Category Theory. Therefore, a review of its most basic concepts is due. This chapter consists of a general overview of the fundamental categorical constructs, as they would appear in any textbook on Category theory. It starts with the very definition of category, proceeding then to universal constructions such as *initial/terminal* objects, *products/coproducts* and *exponentials*, and eventually the integral concepts of *functors*, *natural transformations* and *adjunctions* are introduced. This summary of the basics of category theory culminates in the definition of *Cartesian Closed Categories* and how these are relate to typed lambda calculus.

3.1 CATEGORIES

A category is a mathematical concept that captures in the shortest away possible what is at the heart of any mathematical theory: the objects of the theory, the possible transformations between these objects, and the law expressing the manner in which transformations with matching domain and co-domain can be composed. The categorical approach differs from the reductionist way in which mathematics is usually done by asking not what an object is but rather how does it interact with objects of the same type.

Definition 3.1. *A category $\mathcal{C}$ consists of:*

- *A collection $\boldsymbol{Ob}(\mathcal{C})$ of objects.*
- *A collection of $\boldsymbol{Ar}(\mathcal{C})$ of morphisms.*
- *Mappings dom, cod : $\boldsymbol{Ar}(\mathcal{C}) \to \boldsymbol{Ob}(\mathcal{C})$, which assign to each morphism f its domain $dom(f)$ and its co-domain $cod(f)$.*
- *A set[1] for each pair of objects A and B,*

$$C(A,B) := \{f \in \boldsymbol{Ar}(C) | f : A \to B\}$$

1 This actually makes this into the definition of *locally small* category. In any case, all the categories discussed in this thesis are locally small.

called the ***hom-set****.*

- *An* ***associative composition*** *map, i.e., a map*

$$(\_ \circ \_)_{A,B,C} : \mathcal{C}(B,C) \times \mathcal{C}(A,B) \to \mathcal{C}(A,C)$$

such that for any $f \in \mathcal{C}(A,B)$, $g \in \mathcal{C}(B,C)$ *and* $h \in \mathcal{C}(C,D)$ *the following holds*

$$h \circ (g \circ f) = (h \circ g) \circ f$$

- *A unique morphism* $id_X \in \mathcal{C}(X,X)$ *for all objects X, called the* ***identity*** *morphism, which is the neutral element of composition, i.e., for all* $f \in \mathcal{C}(A,B)$ *the following holds*

$$f \circ id_A = f = id_B \circ f$$

As this definition suggest the relationships between objects in a category are established through the morphisms between them. Contrary to common intuition, equality is generally not one such relationship in Category theory. In this context the appropriate notion is *equivalence*, which is expressed through an *isomorphism*.

Definition 3.2. *Two objects* A *and* B *in a category are said to be isomorphic if there are morphisms* $f : A \to B$ *and* $f^{-1} : B \to A$*, such that*

$$f \circ f^{-1} = id_B$$
$$f^{-1} \circ f = id_A.$$

In general there could be many isomorphism between two objects. If there is only one then this is called a unique isomorphism.

There are an infinite number of categories, thus it is better to mention those that are consequential to the present matter.

- **Set** - This is the category that has as objects sets and whose morphisms are functions. Composition is the usual function composition $(f \circ g)x = f(g\ x)$. Isomorphisms in **Set** are bijections.
- **FdVec**$_{\mathcal{K}}$ - In this case objects are finite dimensional vector spaces over some field $\mathcal{K}$, while the morphisms are linear mappings. Composition is again function composition. Linear bijections are the isomorphisms in this instance.
- **FHilb** - Category of finite dimensional Hilbert spaces and linear maps. It inherits most of the structure of **FdVec**$_{\mathcal{K}}$, in fact it shares its composition and isomorphisms. It is worthy of mention, however, that in the context of **FHilb** a stronger notion of isomorphism is usually more relevant, namely one that preserves the inner product - unitary isomorphism.

- CPM(**FHilb**) - This category as the same objects as **FHilb**, while the morphisms are completely positive maps $L(V) \to L(W)$. Composition is the same as in **FHilb**.
- **Sup** - This category as finite Hilbert spaces as objects,the morphisms are superoperators $L(V) \to L(W)$. Composition is inherited from **FHilb**.
- $\mathbf{Mat}_{\mathbb{K}}$ - Category whose objects are natural numbers and whose morphisms are $m \times n$ matrices. Composition is given by matrix multiplication and the identity matrix is the identity morphism.

One very common procedure in Category theory is the process of constructing a category given another. The simplest example of this is the notion of subcategory.

Definition 3.3. *Let $\mathcal{C}$ be a category, and suppose one is given the collections*

$$\boldsymbol{Ob}(\mathcal{D}) \subseteq \boldsymbol{Ob}(\mathcal{C}), \quad \forall A, B \in \boldsymbol{Ob}(\mathcal{D}).\, \mathcal{D}(A,B) \subseteq \mathcal{C}(A,B)$$

then $\mathcal{D}$ is said to be a category of $\mathcal{C}$ if

$$A \in \boldsymbol{Ob}(\mathcal{D}) \implies id_A \in \mathcal{D}(A,A), \quad f \in \mathcal{D}(A,B),\, g \in \mathcal{D}(B,C) \implies g \circ f \in \mathcal{D}(A,C)$$

holds.

The category **Sup** is a subcategory of $CPM(\mathbf{FHilb})$, which in turn is a subcategory of **FHilb**. the relationships between these three category will play a significant role in the later stages of this dissertation.

Another crucial construction in category theory is the *opposite* category. This construction is made possible by the fact that if the direction of the morphisms in a category is reversed, the conditions in 3.1 still hold.

Definition 3.4. *Let $\mathcal{C}$ be a category. One can construct the **opposite** category - written $\mathcal{C}^{op}$ - by reversing all morphisms of $\mathcal{C}$, i.e., interchanging the domain and co-domain of each morphism.*

A consequence of this is the so-called *duality principle* which states that given a valid statement about a category $\mathcal{C}$, the statement constructed by reversing all morphisms and composites in C, is also valid about C^{op}.

3.2 UNIVERSAL CONSTRUCTIONS

Objects in a category have two types of properties: those that are defined within the category in question, and those that can be stated in purely categorical terms. The latter are expressed by *universal properties*. These classify an object through the relationship it bares to all other objects in the category. Objects classified through this procedure are said to be *unique up to unique isomorphims*, meaning: if there is another object which satisfies the same property then there is a unique isomorphism between the two.

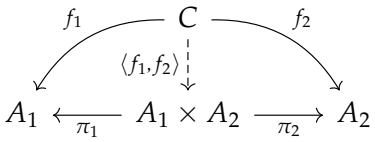
The most basic of *universal properties* define ***initial*** and ***terminal*** objects.

Definition 3.5. *An object **i** is initial in a category $\mathcal{C}$ if there is a unique morphism $?_A : I \to A$ from it to every other object $A \in Ob(\mathcal{C})$. Dually, an object **T** is Terminal in a category $\mathcal{C}$ (initial in $\mathcal{C}^{op}$) if there is a unique morphism $!_A : A \to f$ from it to every other object $A \in Ob(\mathcal{C})$.*

The initial and terminal objects in **Set** are respectively the empty set and the singleton set. As for $\mathbf{FVec}_{\mathcal{K}}$ (and **FHilb**) both the initial and terminal objects are given by the zero dimensional vector space (Hilbert space). When an object is both initial and terminal, it is called the ***zero object***.

Initality and terminality are fairly simple properties, and as is very common in category theory, they are instances of more general patterns. To see this, the concepts of ***product*** and ***coproduct*** need to be introduced.

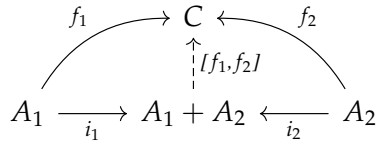
Definition 3.6. *Let A_1, A_2 be objects in a category $\mathcal{C}$. A product of A_1 and A_2 is an object $A_1 \times A_2$ together with a pair of morphisms $\pi_i : A_1 \times A_2 \to A_i$, such that for every other object C together with $f_i : C \to A_i$ there is a unique morphism $\langle f_1, f_2 \rangle : C \to A_1 \times A_2$ making the following diagram commute*



$$\pi_i \circ \langle f_1, f_2 \rangle = f_i$$

Once again by reversing the direction of all morphisms one arrives at the dual concept, in this case the coproduct.

Let A_1, A_2 be objects in a category $\mathcal{C}$. A coproduct of A_1 and A_2 is an object $A_1 + A_2$ together with a pair of morphisms $i_i : A_i \to A_1 + A_2$, such that for every other object C together with $f_i : A_i \to C$ there is a unique morphism $[f_1, f_2] : A_1 + A_2 \to C$ making the following diagram commute



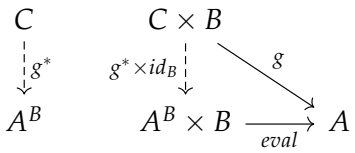
$$[f_1, f_2] \circ ß_i = f_i$$

To notice the connection between initial/terminal objects and products/coproducts one must only wonder what a nullary product/coproduct might be. Quickly one would realize that an initial object is just a nullary coproduct. The same goes for the terminal object. So it should not come as a surprise that the direct sum A.9 is both the product and coproduct in $\mathbf{FVec}_{\mathcal{K}}$. In the case of **Set** the cartesian product is the product and the disjoint union the coproduct.

Importantly, not every category contains these universally defined objects. These are further structure which certain categories have and others do not. These properties serve as a away to group and classify categories. For instance, there are categories which contain objects that in a certain way "represent" the morphisms between two given objects. These are called **exponentials**.

Definition 3.7. *Let A and B be objects in a category $\mathcal{C}$, and let $\mathcal{C}$ have all binary products with B. An object A^B together with a morphism $eval : A^B \times B \to A$ is an **exponential** object if for any object*

C and morphism $g : C \times B \to A$ there is a unique morphism $g^* : C \to A^B$ such that the diagram commute



commutes.

The standard example of exponentials is in the category of sets, in which one can think of it as lossless translation between a two argument function $f(x, y)$ and a one argument function $f^*(x)(y)$. This procedure is referred to as *currying*.

FVec$_\mathcal{K}$ and **FHilb**, on the other hand, do not have exponentials, but they abide by a similar structure mediated by a categorical abstraction of the tensor product, which is not a product.

3.3 FUNCTORS, NATURAL TRANSFORMATIONS AND ADJUNCTIONS

So far, definitions have been made by establishing relationships between objects in a category. Yet what is most remarkable in category theory are the relationships it provides between categories themselves. To do this one must merely consider categories as objects in some category. This is the category of categories **CAT**. One is only left with the problem of finding an appropriate definition of morphism in **CAT**. Since they have to preserve the structure in 3.1 they are essentially category homomorphisms, usually called functors.

Definition 3.8. *Let $\mathcal{C}$ and $\mathcal{D}$ be categories. A mapping $F : \mathcal{C} \to \mathcal{D}$ is a **covariant functor** if:*

- *for all objects $A \in \boldsymbol{Ob}(\mathcal{C})$ there is a corresponding object $FA \in \boldsymbol{Ob}(\mathcal{D})$ and*
- *for all morphisms $f \in \mathcal{C}(A, B)$ there is a corresponding arrow $Ff \in \mathcal{D}(FA, FB)$ such that composition and identities are preserved, i.e., for all $f \in \mathcal{C}(A, B)$ and $g \in \mathcal{C}(B, C)$*

$$F(g \circ f) = Fg \circ Ff, \quad F(id_A) = id_{FA}$$

holds in $\mathcal{D}$.

*Similarly a mapping $F : \mathcal{C} \to \mathcal{D}$ is a **contravariant functor** if:*

- *for all objects $A \in \boldsymbol{Ob}(\mathcal{C})$ there is a corresponding object $FA \in \boldsymbol{Ob}(\mathcal{D})$ and*
- *for all morphisms $f \in \mathcal{C}(A, B)$ there is a corresponding arrow $Ff \in \mathcal{D}(FB, FA)$ such that composition and identities are preserved, i.e., for all $f \in \mathcal{C}(A, B)$ and $g \in \mathcal{C}(B, C)$*

$$F(g \circ f) = Ff \circ Fg, \quad F(id_A) = id_{FA}$$

holds in $\mathcal{D}$. In this case the direction of composition is reversed. For that reason a contravariant functor $F : \mathcal{C} \to \mathcal{D}$ is just an instance of a covariant functor of signature $F : \mathcal{C}^{op} \to \mathcal{D}$.

An important property of **CAT** is that it has products. Thus functors can be generalized to have more than one argument.

Definition 3.9. *A bifunctor is merely a functor whose domain is the product of two categories, in that, it is a mapping* $- \otimes - : \mathcal{C} \times \mathcal{C} \to \mathcal{D}$[2] *such that*

- *for all objects* $(A, B) \in Ob(\mathcal{C} \times \mathcal{C})$ *there is a corresponding object* $A \otimes B \in \boldsymbol{Ob}(\mathcal{D})$ *and*
- *for all morphisms* $f \in \mathcal{C}(A, B)$ *and* $g \in \mathcal{C}(C, D)$ *there is a corresponding morphism* $f \otimes g \in \mathcal{D}(A \otimes C, B \otimes D)$ *such that for all* $A, B \in \boldsymbol{Ob}(\mathcal{C})$ *and* f, g, h, k *of the appropriate type it is the case that*

$$(g \circ f) \otimes (k \circ h) = (g \otimes k) \circ (f \otimes h) \text{ and } id_A \otimes id_B = id_{A \otimes B} \tag{3.1}$$

Functors are everywhere. In fact, two of them have already made an appearance. The operators $\times$ and $+$ in the definitions of product and coproduct are functors, more specifically bifunctors. They can be defined as

$$\begin{aligned} &\times : \mathcal{C} \times \mathcal{C} \to \mathcal{C} && + : \mathcal{C} \times \mathcal{C} \to \mathcal{C} \\ &\times (A, B) \mapsto A \times B && + (A, B) \mapsto A + B \\ &\times (f, g) \mapsto \langle f \circ \pi_1, g \circ \pi_2 \rangle && + (f, g) \mapsto [i_1 \circ g, i_2 \circ g]. \end{aligned}$$

A disguised example of bifunctor appeared in the previous chapter. Namely the tensor product. One can easily see in A.25 that the tensor product satisfies the functor criteria. There is a particularly important functor that combines covarience, contravarience and bifunctoriality.

Definition 3.10. *Given any category* $\mathcal{C}$*, the* ***hom functor***

$$hom : \mathcal{C}^{op} \times \mathcal{C} \to Set$$

maps any object $(X, Y) \in \mathcal{C}^{op} \times \mathcal{C}$ *to the set* $\mathcal{C}(X, Y)$*, and any morphism* $(f, g) \in \mathcal{C}^{op} \times \mathcal{C}$ *to the function*

$$\begin{aligned} hom(f, g) : \mathcal{C}(X, Y) &\to \mathcal{C}(X', Y') \\ h &\mapsto g \circ h \circ f \end{aligned}$$

for morphisms $f : X' \to X$ *and* $g : Y \to Y'$ *in* $\mathcal{C}$*.*

In general, there may be more than one functor between two categories. It would then be helpful if one could somehow establish connections between such functors. As it happens functors between categories $\mathcal{C}$ and $\mathcal{D}$ are objects in what is called the functor category $[\mathcal{C}, \mathcal{D}]$. The morphisms of this category are called **natural transformations**.

2 The product category $\mathcal{C} \times \mathcal{C}$ is simply the category whose objects are pairs of objects in $\mathcal{C}$ and morphisms are pairs of morphisms in $\mathcal{C}$

Definition 3.11. *Let $\mathcal{C}$ and $\mathcal{D}$ be categories and $F, G : \mathcal{C} \to \mathcal{D}$ be functors. A **natural transformation** $\tau : F \Rightarrow G$ is a collection of morphisms in $\mathcal{D}$ indexed by objects of $\mathcal{C}$, i.e.,*

$$\{\tau_A : FA \to GA\}_{A \in \boldsymbol{Ob}(\mathcal{C})}$$

such that, for all $f \in \mathcal{C}(A, B)$, the following diagram commutes

$$\begin{array}{ccc} FA & \xrightarrow{Ff} & FB \\ {\scriptstyle \tau_A}\downarrow & & \downarrow{\scriptstyle \tau_B} \\ GA & \xrightarrow{Gf} & GB \end{array} \qquad \tau_B \circ Ff = Gf \circ \tau_A$$

in $\mathcal{D}$.

Definition 3.12. *A **natural isomorphism** is an isomorphism in the functor category.*

Earlier, allusion was made to the fact that the relationships between objects in a category are codified in the morphisms between them. When considering categories as objects this motto gains another dimension, that is, the relationships between two categories are codified in the functors between them and the relationships between these functors which are expressed through natural transformations. Two important such relationships are **equivalences** and **adjunctions**.

Definition 3.13. *Two categories $\mathcal{C}$ and $\mathcal{D}$ are equivalent if there are functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$, and natural isomorphism $G \circ F \Rightarrow id_{\mathcal{C}}$ and $F \circ G \Rightarrow id_{\mathcal{D}}$.*

The usual habit of representing linear maps as matrices stems from an equivalence between $\mathbf{FdVec}_{\mathbb{K}}$ and $\mathbf{Mat}_{\mathbb{K}}$. In this case the functor $F : \mathbf{FdVec}_{\mathbb{K}} \to \mathbf{Mat}_{\mathbb{K}}$ maps a vector space to its dimension, and linear maps to its matrix in a given basis. While the functor $G : \mathbf{Mat}_{\mathbb{K}} \to \mathbf{FdVec}_{\mathbb{K}}$ maps a natural number n to the vector space $\mathbb{K}^n$, and maps a matrix $n \to m$ to the linear map $\mathbb{K}^n \to \mathbb{K}^m$ represented by the matrix.

Equivalence is a very restrictive condition, and therefore it is seldom the case one finds an equivalence between categories. A more liberal, and thus more interesting notion is of an adjunction.

Definition 3.14. *Let $\mathcal{C}$ and $\mathcal{D}$ be categories. An **adjunction** from $\mathcal{D}$ to $\mathcal{C}$ is a triple (F,G,θ), where F and G are functors*

$$\mathcal{C} \underset{G}{\overset{F}{\rightleftarrows}} \mathcal{D}$$

and θ is a collection of bijections

$$\theta_{A,B} : \mathcal{C}(A, G(B)) \cong \mathcal{D}(F(A), B)$$

*for each $A \in Ob(\mathcal{C})$ and $B \in Ob(\mathcal{D})$ natural in A and B. It is said that F is **left adjoint** to G, and G is **right adjoint** to F, denoted $F \dashv G$.*

There are alternative definitions of adjunction. One such alternative is given in terms of what are called unit and counit (definition A.29).

It can be straightforwardly shown that the global existence of a product/copruduct in a category $\mathcal{C}$ is due to the existence of the following isomorphisms

$$\mathcal{C}(C, A \times B) \cong \mathcal{C} \times \mathcal{C}(\Delta\, C, (A, B))$$
$$\mathcal{C} \times \mathcal{C}((A, B), \Delta\, C) \cong \mathcal{C}(A + B, C)$$

natural in A, B and C. That is the existence of, respectively, a right ($\times$) and left ($+$) adjoint to the diagonal functor $\Delta : \mathcal{C} \to \mathcal{C} \times \mathcal{C}$. Likewise, the existence of a right adjoint $(-)^A$ to the the product functor $- \times A : \mathcal{C} \to \mathcal{C}$, i.e, the natural isomorphism

$$\mathcal{C}(C, B^A) \cong \mathcal{C}(C \times A, B)$$

determines the existence of exponentials in a category $\mathcal{C}$. At this point one can define a class of categories of great interest to computer scientists.

Definition 3.15. *A category $\mathcal{C}$ with terminal object, finite products and exponentials is called a* ***Cartesian Closed Category (CCC)****.*

3.4 SIMPLY TYPED LAMBDA CALCULUS AND CARTESIAN CLOSED CATEGORIES

So far in this chapter a general review of the basics of category theory has been presented. Yet, the connection between the field and computer science has not been made concrete. And although this is not the objective of this dissertation, one should nevertheless make an attempt at a brief explanation.

The lambda calculus (Church, 1936a), similarly to Turing's universal machine, was an effort to answer Hilbert's $Entscheidungsproblem$. Therefore it was developed as formal system intended as a model for computation. Its fundamental goal is to represent functions through rules of computation, based on two core ideas: $abstraction$ and $function\ application$. Roughly, the system is built solely of what are called λ-**terms**, which are inductively defined as follows:

$$\begin{aligned} term \ ::= & \ x \mid y \mid x \mid ... && variable\ names \\ & \mid \lambda x.\, term && abstraction \\ & \mid (term\ term) && application \end{aligned}$$

The one[3] rule of computation is given by

$$(\lambda x.t)(s) =_\beta t[s/x],$$

3 There are more rewriting rules such as $\alpha - equivalence$, but for brevity concerns the reader is referred to (Selinger, 2008) for a more thorough review of these and the lambda calculus in general.

which translates to: in term t, substitute every free occurrence of x with s.

This is a very simple system, yet it is powerful enough to be able to define all computable function $f : \mathbb{N} \to \mathbb{N}$.

There are, however, inconsistencies in the system resulting from the lack of restrictions on term application - the Kleene-Rosser paradox (Curry, 1942). [4]

To avoid this problem, Church introduced the simply typed lambda calculus (Church, 1940), in which every term has a type and term application is subject to typing rules.

This new computational system served subsequently as inspiration for the type systems of modern functional functional programming languages such as Haskell and ML[5], in the same fashion as Turing's computational model served as the conceptual framework for imperative programming languages.

The connection between category theory and computer science came through a remarkable relationship between the simply typed lambda calculus and cartesian closed categories. The fundamental insight due to Lambek (1980) is that for any typed lambda calculus a category can be constructed by choosing its types as objects and the equivalence class of terms determined by the rewriting rules as the morphisms. Moreover, the class of categories constructed in such a way form themselves a category which is *equivalent* to the category of Cartesian Closed Categories (Lambek and Scott, 1988). The crucial point of this equivalence is the role of the universal property of exponential objects defining function types which model λ-abstraction and application.

The consequence of this result is the fact that Cartesian Closed Categories are models for the typed lambda calculus. Hence the key role category theory plays in the semantics of functional programming languages. It is thus not surprising that many programming constructs in functional programming languages have elegant categorical interpretations. Some examples are: polymorphic types as functors, polymorphic functions as natural transformations, inductive data types as initial algebras (Adámek et al., 2018) (Meijer et al., 1991), and perhaps of most practical significance is the monad (Moggi, 1989) (Moggi, 1991) which solved the problem, which for a long time eluded computer scientist, of providing a compositional semantic description of computations which can not be described by pure functions.

3.5 SUMMARY

This chapter's objective was to establish the mathematical framework which will serve has the basis for the following chapters. In doing so, the basic categorical concepts were introduced such as universal properties, functors, natural transformations, adjunction and a few others. These were interspersed with a few examples which will be relevant later on. This overview, although abstract due to the inherent nature of the subject, was directed towards the notion of Cartesian Closed Categories and their connection with the typed lambda calculus.

4 Informally it can be stated as follows. Assume the system is capable of expressing logical negation $\neg$, then one can define the term $k = \lambda x.\neg(x\ x)$, which then according to the rules of the system can be applied to itself

$$k\ k = (\lambda x.\neg(x\ x)\ k = \neg(k\ k),$$

resulting in a contradiction.

5 Both of these languages are based on System F, an extension of the simply typed lambda calculus which allows for variables to also range over types, allowing for parametric polimorphism.

The next chapter is a review of the field of categorical quantum mechanics as developed by Abramsky and Coecke (2004). This field is aimed at providing a categorical model for quantum computation.

4

QUANTUM MECHANICS: CATEGORICALLY

This chapter serves to explicitly state the semantic framework in which the subsequent chapters will be expressed. As such, it is purely a brief review of the field of Categorical Quantum Mechanics as presented by Abramsky and Coecke (2004). Essentially, the objective of the field is to capture the fundamental structural features of the (finite-dimensional) Hilbert space in an abstract categorical fashion. To be more precise, the goal is to find a moderately general class of category of which the category of finite-dimensional Hilbert spaces is an instance, and in which the postulates of Quantum Mechanics can be expressed. This is accomplished by providing step by step bits of structure to a category which are supposed to mimic features of Hilbert spaces at the categorical level of generality. When this process is finished one is left with *biproduct dagger compact closed categories*.

Regardless of one's appreciation for the categorical approach, the use of this formalism is justified by the fact that equations derived in the formalism hold if and only if they are derivable in finite dimensional Hilbert spaces (Selinger, 2012). Thus one assumes that a morphism in a biproduct dagger compact closed, which moreover does not violate a generalized version of the quantum mechanical axioms, is a valid quantum mechanical process.

This chapter concludes with a construction introduced by Selinger (2007) which categorically symbolizes the passage from closed to open quantum systems.

4.1 MONOIDAL CATEGORIES

By considering a categorical model for quantum mechanics one determines from the get-go, and without assuming any further structure, two properties of the model:

- Processes must be typed;
- The flow of time is reflected by sequential composition of processes.

The next bit of structure one must introduce deals with the manner in which systems are to be combined, and furthermore, how processes can be applied independently on the parts of a compound system. This structure comes about as a generalization of a monoid in categorical terms. A monoid is a set M equipped with an associative operation $\Box : M \times M \to M$, where $(\times)$ is the cartesian product, and an element $m \in M$, which is the unit of such operation. This generalizes to the concept of *monoidal category*.

Definition 4.1. *(Mac Lane, 2013) A monoidal category consists of a category $\mathcal{C}$, together with*

- *a bifunctor $- \otimes - : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$.*
- *a unit object I and natural isomorphisms*[1]

$$\lambda_A : A \simeq I \otimes A$$
$$\rho_A : A \simeq A \otimes I$$
$$\alpha_{A,B,C} : A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C$$

such that

$$\begin{array}{ccccc} A \otimes (B \otimes (C \otimes D)) & \xrightarrow{\alpha_{A,B,C\otimes D}} & (A \otimes B) \otimes (C \otimes D) & \xrightarrow{\alpha_{A\otimes B,C,D}} & ((A \otimes B) \otimes C) \otimes D \\ \downarrow{\scriptstyle id_A \otimes \alpha_{B,C,D}} & & & & \uparrow{\scriptstyle \alpha_{A,B,C} \otimes id_D} \\ A \otimes ((B \otimes C) \otimes D) & & \xrightarrow{\alpha_{A,B\otimes C,D}} & & (A \otimes (B \otimes C)) \otimes D \end{array}$$

commutes for all A, B, C, $D \in \mathcal{C}$,

$$\begin{array}{ccc} & A \otimes B & \\ {\scriptstyle \rho_A \otimes id_B} \swarrow & & \searrow {\scriptstyle id_A \otimes \lambda_B} \\ (A \otimes I) \otimes B & \xleftarrow{\alpha_{A,I,B}} & A \otimes (I \otimes B) \end{array}$$

commutes for all A, $B \in \mathcal{C}$, and

$$\lambda_I = \rho_I.$$

Furthermore if there is the natural isomorphism

$$\sigma_{A,B} : A \otimes B \simeq B \otimes A$$

in $\mathcal{C}$ satisfying

$$\begin{array}{ccccc} A \otimes (B \otimes C) & \xrightarrow{\alpha_{A,B,C}} & (A \otimes B) \otimes C & \xrightarrow{\sigma_{A\otimes B,C}} & C \otimes (A \otimes B) \\ \downarrow{\scriptstyle id_A \otimes \sigma_{B,C}} & & & & \downarrow{\scriptstyle \alpha_{C,A,B}} \\ A \otimes (C \otimes B) & \xrightarrow{\alpha_{A,C,B}} & (A \otimes C) \otimes B & \xrightarrow{\sigma_{A,C}\otimes id_B} & (C \otimes A) \otimes B \end{array}$$

and

$$\begin{array}{ccccc} A \otimes B & \xrightarrow{\sigma_{A,B}} & B \otimes A & & I \otimes A \\ \downarrow{\scriptstyle id_{A\otimes B}} & \swarrow {\scriptstyle \sigma_{B,A}} & & \nearrow {\scriptstyle \lambda_A} & \downarrow{\scriptstyle \sigma_{I,A}} \\ A \otimes B & & & A \xrightarrow[\rho_A]{} & A \otimes I \end{array}$$

for all A, B, $C \in \mathcal{C}$, then $\mathcal{C}$ is called symmetric monoidal.

1 Whenever the above natural isomorphisms are identities, the category is called strict (symmetric) monoidal.

The significance of the commuting diagrams above lies in the fact that they constitute the minimal requirements to prove the coherence theorem for symmetric monoidal categories (Mac Lane, 2013, p. 253-255). Roughly, the theorem states that if the above diagrams commute then for any two objects which only differ by composing α, σ, λ, ρ with $- \otimes -$ and/or $- \circ -$ there is an unique natural isomorphism between the two. For example, in a symmetric monoidal category there is a unique natural isomorphism between $(A \otimes B) \otimes (C \otimes (D \otimes E))$ and $(B \otimes A) \otimes ((C \otimes E) \otimes D)$. As a consequence one needs not to worry about cumbersome parenthesis manipulations.

Proposition 4.1. *The category of finite dimensional Hilbert spaces* $\mathbb{K}$ *together with the tensor product constitutes a symmetric monoidal category.*

Proof. Given appropriate domain and codomain for h and k, bifunctoriality follows.

$$\begin{aligned}(f \circ h) \otimes (g \circ k)(v \otimes w) &= (f \circ h)v \otimes (g \circ k)w \\ &= f(h(v)) \otimes g(k(w) \\ &= (f \otimes g)(h(v) \otimes k(w)) \\ &= (f \otimes g) \circ (h \otimes k)(v \otimes w)\end{aligned}$$

The unit object is given by the set of complex numbers $\mathbb{C}$. The rest of conditions on 4.1 can be trivially verified. □

The Cartesian Closed Categories discussed in the previous chapter are also examples of monoidal categories, where the product is the monoidal bifunctor and the terminal object the monoidal unit. However, in general the monoidal operation is not a product, thus the copy and delete operations

$$\Delta_X : X \to X \otimes X, \quad !_X : X \to 1$$

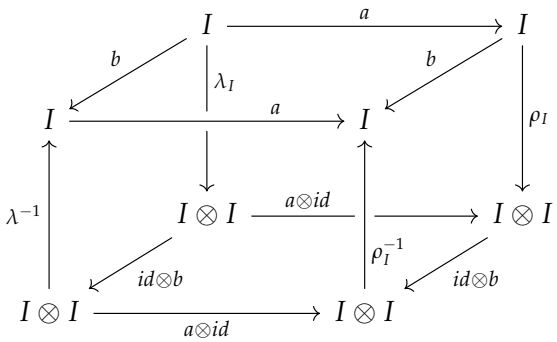
are not generally available for every object in a monoidal category. This restriction already captures the no-cloning of quantum states principle (Wootters and Zurek, 1982). In fact, one can within the class of monoidal categories identify those categories with products by those which possess the above morphisms for all objects and satisfy the conditions one would expect for copying and deleting operations.

Symmetric monoidal categories classify a very general class of process theories, therefore further structure has to be incorporated in order to for one to model quantum mechanics. Nevertheless, two types of morphisms are worthy of mention even at this level of generality. Morphisms of type $\psi : I \to A$ are called ***states***, while morphisms of type $s : I \to I$ are called ***scalars***. The states $\psi : \mathbb{C} \to H$ in **FHilb** are vectors in H, while the scalars $s : \mathbb{C} \to \mathbb{C}$ are in one to one correspondence with complex numbers. This is because any linear map $s : \mathbb{C} \to \mathbb{C}$ is completely determined by $s(1)$ and thus for any $c \in \mathbb{C}$ it follows that $s(c) = c.s(1)$. Subsequently it can be concluded that composition corresponds to multiplication of complex numbers, in this case.

The set of scalars determine to a large extent the overall structure of a monoidal category. However they share algebraic properties, some of which can be determined already at this level of generality.

Lemma 4.1. *(Kelly and Laplaza, 1980) In a monoidal category $\mathcal{C}$ the monoid $\mathcal{C}(I,I)$ is commutative.*

Proof. For any two scalars $a, b \in \mathcal{C}(I,I)$ the following diagram commutes



The four lateral sides of the parallelogram commute by naturality of λ_I and ρ_I, and $\lambda_I = \rho_I$. The bottom side commutes by bifunturiality of $\otimes$. □

An abstract scalar multiplication $s \bullet f$ for some morphism $f : A \to B$ emerges by noticing that every scalar $s : I \to I$ induces a natural transformation (Abramsky and Coecke, 2009)

$$s_A : A \xrightarrow{\lambda_A} I \otimes A \xrightarrow{s \otimes id_A} I \otimes A \xrightarrow{\lambda_A^{-1}} A$$

Naturality implies

$$f \circ s_A = s_B \circ f,$$

Hence one can define scalar multiplication b

$$s \bullet f := f \circ s_A = s_B \circ f.$$

Additionally the following

$$\begin{aligned} 1 \bullet f &= f \\ s \bullet (t \bullet f) &= (s \circ t) \bullet f \\ (s \bullet g) \circ (t \bullet f) &= (s \circ t) \bullet (g \circ f) \\ (s \bullet f) \otimes (t \circ g) &= (s \circ t) \bullet (f \otimes g) \end{aligned}$$

generalize the usual properties of scalar multiplication.

4.2 DAGGER CATEGORIES

One fundamental feature of Hilbert spaces is that every map between Hilbert spaces has an adjoint. This global transformation of maps points to the existence of a functor: the dagger.

Definition 4.2. *(Selinger, 2007) A dagger symmetric monoidal category is a symmetric monoidal category $\mathcal{C}$ equipped with a contravariant functor $\dagger : \mathcal{C}^{op} \to C$ which acts as an identity on objects and for all morphisms $f \in \mathcal{C}$. $f^{\dagger\dagger} = f$ (involutive). Moreover it interacts coherently with the symmetric monoidal structure, i.e.,*

$$\begin{aligned}(f \otimes g)^\dagger = f^\dagger \otimes g^\dagger &: B \otimes D \to A \otimes C,\\ \alpha^\dagger_{A,B,C} = \alpha^{-1}_{A,B,C} &: (A \otimes B) \otimes C \to A \otimes (B \otimes C),\\ \lambda^\dagger_A = \lambda^{-1}_A &: I \otimes A \to A,\\ \sigma^\dagger_{A,B} = \sigma^{-1}_{A,B} &: B \otimes A \to A \otimes B.\end{aligned}$$

for all $f : A \to B$ and $g : C \to D$.

Proposition 4.2. *The category of finite dimensional Hilbert spaces **FHilb** is dagger symmetric monoidal.*[2]

Proof. The dagger structure in FHilb is given by the adjoint, which is defined in terms of the inner product. Namely, the adjoint of $f : A \to B$ is the unique linear map $f^\dagger : B \to A$ satisfying

$$\langle fv|w\rangle = \langle v|f^\dagger w\rangle,$$

for all $v \in A$ and $w \in B$. It follows

$$\begin{aligned}\langle u_1 \otimes v_1|(f \otimes g)^\dagger u_2 \otimes v_2\rangle &= \langle (f \otimes g)u_1 \otimes v_1|u_2 \otimes v_2\rangle\\ &= \langle f(u_1) \otimes g(v_1)|u_2 \otimes v_2\rangle\\ &= \langle f(u_1)|u_2\rangle\langle g(v_1)|v_2\rangle\\ &= \langle u_1|f^\dagger(u_2)\rangle\langle v_1|g^\dagger(v_2)\rangle\\ &= \langle u_1 \otimes v_1|f^\dagger(u_2) \otimes g^\dagger(v_2)\rangle\\ &= \langle u_1 \otimes v_1|(f^\dagger \otimes g^\dagger)u_2 \otimes v_2\rangle\end{aligned}$$

The other conditions in 4.2 follow similarly. □

This structure allows one to capture at the categorical level many features of Hilbert spaces. A morphism $f : A \to B$ in a dagger category is:

- the adjoint of $g : B \to A$ when $g = f^\dagger$,

2 This also holds for the category **Hilb** of Hilbert spaces (not necessarily finite-dimensional) and bounded linear maps.

- unitary when $f \circ f^\dagger = id_B$ and $f^\dagger \circ f = id_A$,
- an isometry when $f = f^\dagger$,
- self-adjoint when $A = B$ and $f = f^\dagger$,
- positive when $A = B$ and $f = g^\dagger \circ g$ for some $g : A \to C$.

Finally one can introduce the inner product of two states $\psi, \phi : I \to B$ as

$$\langle \phi | \psi \rangle := \phi^\dagger \circ \psi$$

which, as one would expect, is a scalar.

4.3 COMPACT CLOSED CATEGORIES

The defining characteristic of Cartesian Closed Categories is the existence of objects which represent the morphisms between two other objects. The category of finite dimensional Hilbert Spaces is not Cartesian Closed, nonetheless it satisfies an analogous behaviour mediated by the monoidal structure instead of the product. This property of Hilbert spaces is usually referred to as the map-state duality, and identifies much of the algebraic properties of entanglement.

Much of these properties can be seen instantiated in the context of the quantum teleportation protocol (Bennett et al., 1993). The quantum teleportation protocol is designed to transmit the state of an unknown qubit q_C from a source which one calls Alice to a target which one call Bob. For this to be achieved Alice and Bob share any one of the Bell states

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B) \qquad |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A|0\rangle_B - |1\rangle_A|1\rangle_B)$$
$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A|1\rangle_B + |1\rangle_A|0\rangle_B) \qquad |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A|1\rangle_B - |1\rangle_A|0\rangle_B)$$

purposed to function as a communication channel.

At the start of the protocol both q_C and q_A are present at Alice's location, while q_B is present at Bob's location. Picking $|\Phi^+\rangle$ as the communication channel[3], the three qubit system is described by the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(\alpha|0\rangle + \beta|1\rangle) \otimes [\frac{1}{\sqrt{2}}(|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B)].$$

Expanding this in the Bell basis one gets

$$|\psi\rangle = \frac{1}{2}\Big[|\Phi^+\rangle_{CA} \otimes (\alpha|0\rangle_B + \beta|1\rangle_B) + |\Phi^-\rangle_{CA} \otimes (\alpha|0\rangle_B - \beta|1\rangle_B)$$

3 The protocol is independent of the choice of Bell state. A different choice of Bell state simply determines a different but likewise valid choice of unitary corrections at the end of the protocol.

$$+|\Psi^+\rangle_{CA} \otimes (\alpha|1\rangle_B + \beta|0\rangle_B) + |\Psi^-\rangle_{CA} \otimes (\alpha|1\rangle_B - \beta|0\rangle_B)\Big]$$

At this point a measurement in the bell basis is performed by Alice on q_C and q_A, yielding one of the states

$$|\Phi^+\rangle_{CA} \otimes (\alpha|0\rangle_B + \beta|1\rangle_B) \qquad |\Phi^-\rangle_{CA} \otimes (\alpha|0\rangle_B - \beta|1\rangle_B)$$
$$|\Psi^+\rangle_{CA} \otimes (\alpha|1\rangle_B + \beta|0\rangle_B) \qquad |\Psi^-\rangle_{CA} \otimes (\alpha|1\rangle_B - \beta|0\rangle_B)$$

Then, depending on the result, Alice sends to Bob two bits encoding one of the following unitary transformations

$$\beta_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \beta_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \beta_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \beta_4 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Upon receiving this information Bob performs the corresponding transformation on q_B. It can be easily verified that the state of q_C initially in Alice's possession is reproduced at Bob's location.

Implicit in this protocol is the role of entanglement states in the flow of quantum information within compound systems.

One can observe this fact through the lenses of the isomorphism

$$k := \sum_{i,j=0}^{n} \alpha_{ij}|i\rangle \otimes |j\rangle \mapsto \sum_{i,j=0}^{n} \alpha_{ij}\sqrt{n}|i\rangle\langle j| \tag{4.1}$$

with inverse

$$k^{-1} := \sum_{i,j=0}^{n} \alpha_{ij}|i\rangle\langle j| \mapsto \sum_{i,j=0}^{n} \frac{\alpha_{ij}}{\sqrt{n}}|i\rangle \otimes |j\rangle, \tag{4.2}$$

which states that single state transformations can be fully encoded in bipartite states. Under this correspondence one can equate the Bell state $|\Phi^+\rangle$ with the identity operator, or more precisely as the identity channel between systems A and B. Moreover for every bipartite state there is a unique observational branch

$$l := \sum_{i,j=0}^{n} \alpha_{ij}|i\rangle \otimes |j\rangle \mapsto \sum_{i,j=0}^{n} \alpha_{ij}^*\langle i| \otimes \langle j|$$

which implies by transitivity of isomorphisms that for every (bipartite) observational branch there is a unique single state transformation.

Thus one can reinterpret the quantum teleportation protocol as follows: Alice is in possession of system q_C with unknown state $|\phi\rangle$ and a channel (Bell state) which she shares with Bob. By performing a Bell measurement on q_C and q_A (Alice's part of the channel), a correlation is established between q_C and Alice's part of the channel. This correlation is encoded in a unitary transformation relating q_C and q_A, and is completely determined by the measurement outcome. This correlation is then propagated unchanged by the channel to Bob. At this stage there is a correlation between Bob's state and $|\phi\rangle$. Thus, once Alice signals to Bob the outcome of her measurement, Bob can subsequently apply U^{-1} to its state and recover $|\phi\rangle$.

It turns out that much of the interesting properties of entanglement states is a consequence of this interpretation, or rather the isomorphism 4.1. As it happens this isomorphism is an instance of a more general categorical structure.

Definition 4.3. *A compact closed category $\mathcal{C}$ is a symmetric monoidal category in which to each object A in $\mathcal{C}$ a dual object A^*, a unit $\eta_A : I \to A^* \otimes A$ and a counit $\epsilon_A : A \otimes A^* \to I$ are assigned such that the diagrams*

$$\begin{array}{ccccc} A & \xrightarrow{\rho_A} & A \otimes I & \xrightarrow{id_A \otimes \eta_A} & A \otimes (A^* \otimes A) \\ \downarrow{\scriptstyle id_A} & & & & \downarrow{\scriptstyle \alpha_{A,A^*,A}} \\ A & \xleftarrow[\lambda_A^{-1}]{} & I \otimes A & \xleftarrow[\epsilon_A \otimes id_A]{} & (A \otimes A^*) \otimes A \end{array} \qquad \begin{array}{ccccc} A^* & \xrightarrow{\lambda_{A^*}} & I \otimes A^* & \xrightarrow{\eta_A \otimes id_{A^*}} & (A^* \otimes A) \otimes A^* \\ \downarrow{\scriptstyle id_{A^*}} & & & & \downarrow{\scriptstyle \alpha^{-1}_{A^*,A,A^*}} \\ A^* & \xleftarrow[\rho_{A^*}^{-1}]{} & A^* \otimes I & \xleftarrow[id_{A^*} \otimes \epsilon_A]{} & A^* \otimes (A \otimes A^*) \end{array}$$

commute. The operation $(\ )^*$ extends to a functor $(\ )^* : \mathcal{C}^{op} \to \mathcal{C}$ by mapping $f : A \to B$ to

$$\rho_{A^*}^{-1} \circ (id_{A^*} \otimes \epsilon_B) \circ (id_{A^*} \otimes f \otimes id_{B^*}) \circ (\eta_A \otimes id_{B^*}) \circ \lambda_{B^*}$$

where $f^* : B^* \to A^*$. Additionally in a compact closed category there are natural isomorphisms

$$u_{A,B} : (A \otimes B)^* \cong A^* \otimes B^*, \quad A^{**} \cong A, \quad I^* \cong I \tag{4.3}$$

making the monoidal operation self-dual.

Closedness comes from the fact that the categories here defined are instances of monoidal closed categories (definition A.30), a generalization of Cartesian Closed Categories. The unit and counit are indeed a reference to the adjunction implicit in the closed monoidal structure. Namely, in a compact closed category the functor $- \otimes A : \mathcal{C} \to \mathcal{C}$ is both left and right adjoint to the functor $- \otimes A^* : \mathcal{C} \to \mathcal{C}$.

Before attending further matters it is important to emphasize that this last isomorphisms will be treated for ease of calculation as identities. This is possible since for any monoidal category one can always find an equivalent category which is strict monoidal (Mac Lane, 2013, p. 257).

Proposition 4.3. *The category of finite dimensional Hilbert spaces is compact closed.*

Proof. For any finite dimensional vector space V one can define V^* to be its linear algebraic dual space: the space of linear functionals $f : V \to \mathbb{K}$. The unit and counit thus can be written as

$$\begin{aligned} \eta_V : \mathcal{K} &\to V^* \otimes V & \epsilon_V : V \otimes V^* &\to \mathbb{K} \\ := 1 &\mapsto \sum_{i=1}^{n} e^i \otimes e_i & := e_i \otimes e^j &\mapsto e^j(e_i) \end{aligned}$$

where $\{e_i\}_{i=0}^n$ is a basis of V and e_j is the linear functional of V^* determined by $e^j(e_i) = \delta_{ij}$. The same construction does not hold for Hilbert spaces, since the map between $\psi \in H$ and $\langle \psi | - \rangle \in H \to \mathbb{C}$ is not linear (it is conjugate-linear), hence it does not live in **FHilb**. One can get around this issue by

defining the dual object H^* as the conjugate Hilbert space $\overline{H}$. The vectors of $\overline{H}$ coincide with those of H, however complex conjugation demands scalar multiplication and and the inner product to be

$$\alpha \bullet_{\overline{H}} \phi := \overline{\alpha} \bullet_H \phi \qquad\qquad \langle \phi | \psi \rangle_{\overline{H}} = \langle \psi | \phi \rangle_H$$

At this point one is in the position to define unit and counit as

$$\begin{aligned} \eta_H : \mathbb{C} &\to \overline{H} \otimes H & \epsilon_H : H \otimes \overline{H} &\to \mathbb{C} \\ := 1 &\mapsto \sum_{i=1}^{n} e_i \otimes e_i & := \phi \otimes \psi &\mapsto \langle \psi | \phi \rangle_H \end{aligned}$$

for $\{e_i\}_{i=0}^n$ a basis for both H and $\overline{H}$. It then follows for $v \in H$

$$\begin{aligned} \lambda^{-1} \circ (\epsilon \otimes id) \circ \alpha \circ (id \otimes \eta) \circ \rho\ (v) &= \lambda^{-1} \circ (\epsilon \otimes id) \circ \alpha \circ (id \otimes \eta)(v \otimes 1) \\ &= \lambda^{-1} \circ (\epsilon \otimes id) \circ \alpha((v \otimes (\sum_{i=0}^{n} e_i \otimes e_i)) \\ &= \lambda^{-1} \circ (\epsilon \otimes id)(\sum_{i=1}^{n} (v \otimes e_i) \otimes e_i) \\ &= \lambda^{-1}(\sum_{i=1}^{n} \langle e_i | v \rangle \otimes e_i) \\ &= \sum_{i=1}^{n} \langle e_i | v \rangle e_i = v \end{aligned}$$

The other diagram of definition 4.3 follows similarly. □

By propositions 4.2 and 4.3 **FHilb** is both a dagger symmetric monoidal and compact closed category. But the interaction between these two structures is yet to be defined.

Definition 4.4. *(Selinger, 2007) A dagger compact closed category $\mathcal{C}$ is a dagger symmetric monoidal category that is also compact closed and such that the following diagram commutes*

$$\begin{array}{ccc} I & \xrightarrow{\epsilon_A^\dagger} & A \otimes A^* \\ & \searrow^{\eta_A} & \downarrow{\scriptstyle \sigma_{A,A^*}} \\ & & A^* \otimes A \end{array}$$

for all A in $\mathcal{C}$.

If $\mathcal{C}$ is dagger compact closed, then the composition of the functors $(\ )^*$ and $(\ )^\dagger$ yields a covariant functor $(\ )_* : \mathcal{C} \to \mathcal{C}$. This functor can be thought of as the abstract counterpart to complex conjugation, while $(\ )^*$ can be conceptualized as the matrix transpose. Combining these two one obtains

$$f^\dagger = (f^*)_* = (f_*)^*$$

the adjoint once again.

Proposition 4.4. *The category of finite dimensional Hilbert spaces is dagger compact closed.*

Proof. Let $v = \phi \otimes \psi \in \overline{H} \otimes H$, $1 \in \mathbb{C}$ and $\{e_i\}_{i=1}$ is an orthonormal basis of H and $\overline{H}$. Then

$$\begin{aligned}\langle \phi \otimes \psi | \eta(1) \rangle_{\overline{H}\otimes H} &= \langle \phi \otimes \psi | \sum_{i=1}^{n} e_i \otimes e_i \rangle_{\overline{H}\otimes H} = \sum_{i=1}^{n} \langle \phi \otimes \psi | e_i \otimes e_i \rangle_{\overline{H}\otimes H} = \sum_{i=1}^{n} \langle \phi | e_i \rangle_{\overline{H}} \langle \psi | e_i \rangle_H \\ &= \sum_{i=1}^{n} \langle e_i | \phi \rangle_H \langle \psi | e_i \rangle_H = \sum_{i=1}^{n} \langle \psi | e_i \rangle_H \langle e_i | \phi \rangle_H = \langle \psi \sum_{i=1}^{n} | e_i \rangle \langle e_i | \phi \rangle_H = \langle \psi | \phi \rangle_H\end{aligned}$$

and

$$\begin{aligned}\langle \phi \otimes \psi | (\sigma \circ \epsilon^\dagger)(1) \rangle_{\overline{H}\otimes H} &= \langle \phi \otimes \psi | (\epsilon \circ \sigma^\dagger)^\dagger(1) \rangle_{\overline{H}\otimes H} = \langle (\epsilon \circ \sigma^\dagger)\phi \otimes \psi | 1 \rangle_{\mathbb{C}} = \langle \epsilon\, \psi \otimes \phi | 1 \rangle_{\mathbb{C}} \\ &= \langle \langle \phi | \psi \rangle_H | 1 \rangle_{\mathbb{C}} = \langle \psi | \phi \rangle_H\end{aligned}$$

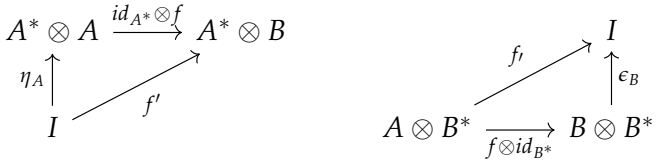
Note that $\overline{H \otimes \overline{H}} = \overline{H} \otimes \overline{\overline{H}} = \overline{H} \otimes H$ follows from 4.3. Putting the above together

$$\langle \phi \otimes \psi | \eta(1) \rangle_{\overline{H}\otimes H} = \langle \phi \otimes \psi | (\sigma \circ \epsilon^\dagger)(1) \rangle_{\overline{H}\otimes H}$$

which is equivalent to the diagram in definition 4.4. □

One can now formalize the correspondence between bipartite states and single state transformations, by noticing that for every morphism $f : A \to B$ in a compact closed category one can define its ***name*** and ***coname***.

Definition 4.5. *The name f' and coname $f_\prime$ of a morphism $f : A \to B$ in a compact closed category are defined by the following commuting diagrams.*



As consequence, in a compact closed category there are natural isomorphisms

$$\mathcal{C}(A \otimes B^*, I) \cong \mathcal{C}(A, B) \cong \mathcal{C}(I, A^* \otimes B)$$

for any A and B. This isomorphism is exactly the (abstract) categorical version of 4.1. Furthermore, the correction of the quantum teleportation protocol can be proved solely by means of the algebraic structure of compact closed categories and an additional structure which formalizes the implicit classical communication and the indeterministic nature of measurements implicit in the protocol.

4.4 BIPRODUCTS

Dagger compact closure has almost all the structure necessary to prove the correctness of protocols such as teleportation, logic-gate teleportation and entanglement swapping (Abramsky and Coecke, 2004). One only needs a way to describe measurements and the flow of classical information. Both of these are accommodated by **biproducts**. Biproducts in a category $\mathcal{C}$ establish an addition operation on each hom-set $\mathcal{C}(A,B)$, which is what gives rise to superposition. Additionally it is in the context of biproducts that measurements can be modelled.

Definition 4.6. *If A_1 and A_2 are objects in a category $\mathcal{C}$, their biproduct is an object $A_1 \oplus A_2 \in \mathcal{C}$, together with morphisms*

$$q_i : A_i \to A_1 \oplus A_2$$
$$\pi_i : A_1 \oplus A_2 \to A_i$$

for $i = 1,2$, such that $(A_1 \oplus A_2, \pi_i)$ is a product, $(A_1 \oplus A_2, q_i)$ is a coproduct and $\pi_i \circ q_j = \delta_{i,j}$.

A category $\mathcal{C}$ is said to have finite biproducts if it has a zero object and a biproduct for any two pairs of objects. Furthermore, in a category $\mathcal{C}$ which admits biproducts, projections π_i and coprojections q_i can be chosen in a fashion such that for each $\bigoplus_{k=1}^{k^n} A_k$

$$\pi_j \circ q_i = \delta_{i,j} \quad \textit{and} \quad \sum_{k=1}^{k=n} q_k \circ \pi_k = id_{\bigoplus A_k}$$

holds.

Definition 4.7. *(Mitchell, 1965) Given a category $\mathcal{C}$ with biproducts, an addition operation can be defined on every hom-set $\mathcal{C}(A,B)$ as*

$$\begin{array}{ccc} A & \xrightarrow{f+g} & B \\ \downarrow{\scriptstyle \Delta} & & \uparrow{\scriptstyle \nabla} \\ A \oplus A & \xrightarrow{f \oplus g} & B \oplus B \end{array}$$

where Δ and ∇ are respectively the diagonal and codiagonal maps. The operation $(+)$ is associative and commutative with the zero morphism $0_{A,B} : A \to 0 \to B$ as the unit. Moreover, composition is linear $h \circ (f+g) = h \circ f + h \circ g$, therefore $\mathcal{C}$ is enriched in commutative monoids[4].

In the special case where $f, g : I \to A$ one gets the superposition state $f + g : I \to A$. As was the case for compact closure and dagger structures, for a category $\mathcal{C}$ to be a biproduct dagger compact closed category it is not sufficient for it to be both dagger compact closed and have biproducts.

4 Sometimes the hom-sets in a category carry additional structure (monoid, group, poset ...), and in those instances the category is said to be *enriched* in that given structure.

Definition 4.8. *(Selinger, 2007) A biproduct dagger compact closed category $\mathcal{C}$ is a dagger compact closed category with biproducts, such that $\pi_i^\dagger = q_i : A_i \to A_1 \oplus A_2$, for all objects A_1, A_2 in $\mathcal{C}$ and $i = 1, 2$. Each of the following constitute equivalent criteria for biproduct dagger compact closeness*

1. $(f \oplus g)^\dagger = f^\dagger \oplus g^\dagger$,
2. $\langle f, g \rangle^\dagger = [f^\dagger, g^\dagger]$,
3. *The following diagram commutes*

$$\begin{array}{ccc} A^\dagger \oplus B^\dagger & & \\ \downarrow{\scriptstyle id} & \searrow^{[\pi_1^\dagger, \pi_2^\dagger]} & \\ A \oplus B & \xrightarrow{id} & (A \oplus B)^\dagger \end{array}$$

From this last diagram one can easely conclude that in a biproduct dagger compact closed category there are natural isomorphisms

$$v_{A,B} : (A \oplus B)^* \cong A^* \oplus B^*, \;\; v_0 : 0 \cong 0^*$$

and since the dagger preserves biproducts, the equations

$$(f + g)^\dagger = f^\dagger + g^\dagger, \;\; 0_{A,B}^\dagger = 0_{B,A}$$

hold. In a compact closed category with biproducts the monoidal structure interacts nicely with biproducts. By this one means there are natural isomorphims

$$distr : A \otimes (B \oplus C) \cong (A \otimes B) \oplus (A \otimes B) \tag{4.4}$$

$$distl : (A \oplus B) \otimes C \cong (A \otimes C) \oplus (B \otimes C) \tag{4.5}$$

given respectively by $\langle id \otimes \pi_1, id \otimes \pi_2 \rangle$ and $\langle \pi_1 \otimes id, \pi_2 \otimes id \rangle$. This is a consequence of monoidal closure, theorem A.4 and self duality of the monoidal structure.

Proposition 4.5. *The category of finite dimensional Hilbert spaces **FHilb** is biproduct dagger compact closed category.*

Proof. The zero object **0** is given by the 0-dimensional Hilbert space. The biproduct is given by the direct sum A.9. Also let $(x_1, x_2) \in H_1 \oplus H_2$ and $v \in H_i$ then

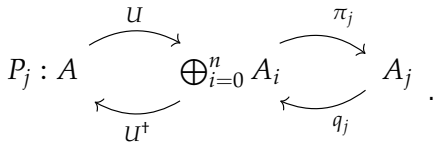
$$\begin{aligned} \langle (x_1, x_2) | \pi_1^\dagger v \rangle &= \langle \pi_1 (x_1, x_2) | v \rangle = \langle x_1 | v \rangle \\ \langle (x_1, x_2) | q_1 v \rangle &= \langle (x_1, x_2) | (v, 0) \rangle = \langle x_1 | v \rangle + \langle x_2 | 0 \rangle = \langle x_1 | v \rangle \end{aligned}$$

follows, and similarly for π_2 and q_2. □

One can now define a measurement in a biproduct dagger compact closed category by associating the unitary (decomposition)

$$U : A \to \bigoplus_{i=0}^{n} A_i$$

with the endomorphisms $P_j : A \to A$, defined by the clock-wise composition



These can be shown to posses the expected properties.

$$P_j^\dagger = (U^\dagger \circ q_j \circ \pi_j \circ U)^\dagger = (\pi_j \circ U)^\dagger \circ (U^\dagger \circ q_j)^\dagger = U^\dagger \circ q_j \circ \pi_j \circ U = P_j$$
$$P_i \circ P_j = U^\dagger \circ q_i \circ \pi_i \circ U \circ U^\dagger \circ q_j \circ \pi_j \circ U = U^\dagger \circ q_i \circ \pi_i \circ q_j \circ \pi_j \circ U = U^\dagger \circ q_i \circ \delta_{ij}^A \circ \pi_j \circ U = \delta_{ij} \circ P_i$$
$$\sum_{i=0}^{n} P_i = \sum_{i=0}^{n} U^\dagger \circ q_i \circ \pi_i \circ U \circ = U^\dagger \circ (\sum_{i=0}^{n} q_i \circ \pi_i) \circ U \circ = U^\dagger \circ id_{\oplus_i A_i} \circ U = id_A$$

The first and second properties just listed of these morphisms affords them their name - *projectors*.It is now possible to describe a measurement of some observable determined by a decomposition

$$U : A \to \bigoplus_{i=0}^{n} A_i.$$

Associated with this decomposition are the projectors P_i, which give rise to a non-destructive measurement

$$\langle P_i \rangle_i^n : A \to \bigoplus_{i=0}^{n} A$$

One refers to each $P_i : A \to A$ as a measurement branch. The morphisms

$$p_i = \pi_i \circ U : A \to I$$

are the observational branches when $A_i = I$.

Reverting back to the quantum teleportation example, one can see that the unitary correction dependent on the signal sent to Bob performed in the last step of the protocol can be given by the morphism

$$U = \beta_1 \oplus \beta_2 \oplus \beta_3 \oplus \beta_4$$

completely determined by the Bell measurement

$$\langle s \bullet \beta_{i'} \rangle_{i^n},$$

where s is a scalar chosen to ensure normalization.

4.5 CATEGORICAL QUANTUM MECHANICS

So far in this chapter the class of biproduct dagger compact closed categories was defined as a generalization of the structure of finite dimensional Hilbert spaces. It is then possible to restate the axioms of quantum Mechanics in this language (Abramsky and Coecke, 2004):

1. The state space of a quantum system is represented by an object A. A basic variable is a state space Q with a given unitary isomorphism

   $$base_Q : I \oplus I \to Q$$

   A preparation in a state space A is a morphism $\psi : I \to A$ for which there is a unitary isomorphism $U : I \oplus B \to A$ such that

   $$\begin{array}{ccc} I & \xrightarrow{\psi} & A \\ \downarrow{\scriptstyle q_1} & \nearrow{\scriptstyle U} & \\ I \oplus B & & \end{array}$$

   commutes. One can think of the preparation of a qubit as assigning by convention the state $|0\rangle \in \mathbb{C}^2$ to q_1, and then any other qubit state is reachable through a rotation U on the Block surface.

2. A compound system for which the subsystems are described respectively by A and B is described by $A \otimes B$.

3. Basic data transformations are given by unitary isomorphisms.

4. A non-destructive measurement is described by

   $$\langle P_i \rangle_i^n : A \to \bigoplus_{i=0}^{n} A,$$

   arising from a decomposition

   $$U : A \to \bigoplus_{i=0}^{n} A_i.$$

   describing a destructive measurement.

This formalism is applicable only to closed quantum system. However it can be generalized to open quantum system through the use of a functor (Selinger, 2007) which symbolizes the passage from one framework to the other. An overview of this construction will be done in 4.6.

4.6 CPM CONSTRUCTION

In order to make the jump from closed to open quantum mechanics in the context of dagger compact closed categories it is necessary to define abstract counterparts for density operator, completely positive maps and trace.

A density operator is a positive linear map. In a dagger compact closed category a morphism $f : A \to A$ is positive if there is an object B and a morphism $g : A \to B$ such that $f = g^\dagger \circ g$. One can turn such a morphism into a state $f : I \to A^* \otimes A$ by means of the isomorphism $\mathcal{C}(A, B) \cong C(I, A^* \otimes B)$.

Definition 4.9. *A positive name is a morphism $f' : I \to A^* \otimes A$ that is the name of a positive morphism $f : A \to A$.*

A positive name can be seen to capture abstractly the notion of density operator[5]. A similar type of reasoning leads to the definition of completely positive map.

Definition 4.10. *(Selinger, 2007) Let A, B be objects in a dagger compact closed category $\mathcal{C}$. A morphism $f : A^* \otimes A \to B^* \otimes B$ is completely positive if all objects $C \in \mathcal{C}$ and all positive names $g : I \to C^* \otimes A^* \otimes A \otimes C$, the morphism $(id_{C^*} \otimes f \otimes id_C) \circ g$ is a positive name.*

Proposition 4.6. *If $f : A \to B$ is any morphism in a dagger compact closed category $\mathcal{C}$, then $f_* \otimes f : A^* \otimes A \to B^* \otimes B$ is completely positive. If $f : A^* \otimes A \to B^* \otimes B$ and $g : B^* \otimes B \to C^* \otimes C$ are completely positive, then $g \circ f : A^* \otimes B \to C^* \otimes C$ is as well. Additionally the map $id_{A^*} \otimes id_A : A^* \otimes A \to A^* \otimes A$ acts as an identity on completely positive morphisms in $\mathcal{C}$.*

Proof. (Selinger, 2007) □

As a result the set of completely positive morphisms in a dagger compact closed category $\mathcal{C}$ forms a category, denoted $CPM(\mathcal{C})$. Hence there is a functor $F : \mathcal{C} \to CPM(\mathcal{C})$ defined by

$$F(A) = A, \qquad F(f) = f_* \otimes f$$

for $A \in \mathcal{C}$ and $f \in \mathcal{C}(A, B)$. The states of $CPM(\mathbf{FHilb})$ are exactly the positive operators on **FHilb**, and as one would expect the scalars of $CPM(\mathbf{FHilb})$ are positive real numbers.

Ideally, one would hope that the category $CPM(\mathcal{C})$ would be dagger compact closed as well.

Proposition 4.7. *If $\mathcal{C}$ is a dagger compact closed category then so is $CPM(\mathcal{C})$.*

Proof. (Selinger, 2007) □

Unfortunately it is not all roses and unicorns. If $\mathcal{C}$ has biproducts the functor $F : \mathcal{C} \to CPM(\mathcal{C})$ does not preserve them. Yet all is not lost, in fact in certain circumstances it is possible to canonically endow a category with biproducts.

5 without the trace condition.

Lemma 4.2. *(Coecke et al., 2017) Any category $\mathcal{C}$ that is enriched in commutative monoids may be embedded universally into one with biproducts $\mathcal{C}^{\oplus}$. The objects of $\mathcal{C}^{\oplus}$ are finite tuples $\langle A_1, ..., A_n \rangle$ of objects of $\mathcal{C}$, with the empty tuple as the zero object. Morphisms $M : \langle A_1, ..., A_n \rangle \to \langle B_1, ..., B_m \rangle$ are matrices of morphisms $M_{i,j} : A_i \to B_j$ and composition is matrix multiplication, where the addition of matrix entrances is given by the commutative monoid and their multiplication by the composition in $\mathcal{C}$. Identity is the matrix with identities on the diagonal and zeros everywhere else.*

Proposition 4.8. *The category $CPM(\mathcal{C})$ can be attributed the biproducts just described.*

Proof. By definition 4.7 the category $\mathcal{C}$ is enriched in commutative monoids. Furthermore, in a biproduct dagger compact closed category, if $f, g : A^* \otimes A \to B^* \otimes B$ are completely positive, then so is $f + g$ and $0 : A^* \otimes A \to B^* \otimes B$ (Selinger, 2007), which makes $CPM(\mathcal{C})$ enriched in commutative monoids, and by lemma 4.2 a category which can be canonically embedded into one with biproducts. □

Lastly if $CPM(\mathcal{C})$ is dagger compact closed then so is $CPM(\mathcal{C})^{\oplus}$, where the dagger on a morphism $f \in CPM(\mathcal{C})^{\oplus}(A, B)$ is given by matrix transposition followed by taking the dagger of each of the matrix elements.

To complete the description of open quantum systems in this framework one needs to introduce the trace of a morphism and what it means for a morphism to be trace preserving.

Definition 4.11. *(Selinger, 2007) Given a morphism $f : A \to A$ in a compact closed category, its trace $tr\, f : I \to I$ is defined as*

$$tr\, f = \; I \xrightarrow{\eta_A} A^* \otimes A \xrightarrow{\sigma} A \otimes A^* \xrightarrow{f \otimes id_{A^*}} A \otimes A^* \xrightarrow{\epsilon_A} I$$

The trace is an operation $Tr : \mathcal{C}(A, A) \to \mathcal{C}(I, I)$.

Definition 4.12. *In a compact closed category a morphism of type $f_* \otimes f : A^* \otimes A \to A^* \otimes A$ is trace preserving if for all states of type $\psi_* \otimes \psi : I^* \otimes I \to A^* \otimes A$*

$$Tr(\alpha[(f_* \otimes f) \circ (\psi_* \otimes \psi)]) = Tr(\alpha(\psi_* \otimes \psi))$$

holds, where $\alpha : \mathcal{C}(I^* \otimes I, A^* \otimes A) \cong \mathcal{C}(A, A)$.

Definition 4.13. *A morphism $M \in CPM(\mathcal{C})^{\oplus}(A, A)$ is trace preserving if all $M_{ij} \in CPM(\mathcal{C})(A, A)$ are trace-preserving. The trace of $M \in CPM(\mathcal{C})^{\oplus}(A, A)$ is the usual trace of a matrix, thus is the sum of the traces of the elements M_{ii}.*

Lemma 4.3. *A morphism $(f \oplus g) \in CPM(\mathcal{C})^{\oplus}(A, A)$ is trace preserving if both $f, g \in CPM(\mathcal{C})(A, A)$ are trace preserving.*

4.7 SUMMARY

This chapter constituted a review of the fundamentals of Categorical Quantum Mechanics (Abramsky and Coecke, 2004). This field is based on iteratively providing to a category structures which encompass the overall algebraic features of finite dimensional Hilbert spaces, and by proxy formulating a semantic framework for reasoning about quantum systems. This process begins by assigning a symmetric monoidal structure to a category as a model of compound systems. Implicit in this treatment of compound systems are computationally relevant properties of quantum systems such as no-cloning and no-deleting. On top of this, in order to capture essential quantum mechanical related concepts such as the inner product and unitary transformations, a dagger structure is imposed. Additionally, a compact closed structure compatible with the dagger is introduced in order to posit the existence of abstract entanglement states. These states are defined by their primary role in propagating information within a compound systems. And finally, biproducts are put forward with the aim of describing both measurements and the flow of classical information resulting from the indeterminacy of measurements.

At the end of this process one can restate the axioms of closed system Quantum Mechanics. Moreover through the CPM construction (Selinger, 2007) one can capture the passage from closed to open Quantum Mechanics. This passage is embodied by a functor which takes a dagger compact closed category and picks its (abstract) completely positive maps. The resulting category is again dagger compact closed. However, if the input category has biproducts then these will not be preserved by the functor. Nonetheless one can canonically embed, under certain circumstances, a category into a category endowed with biproducts. This happens to be the case for the category of completely positive maps, even though its biproducts are not related to the "mother" category.

In the next chapter a morphism in a biproduct dagger compact closed category will be defined as an instantiation of a conditional structure. Thus two definitions will emerge, one for closed systems and one for closed systems. The disparities between these two, will essentially be due to the difference in biproducts.

5

QUANTUM CONTROL

Any machine to which the name computer can be appropriately attributed has at least, and whatever the architectural implementation, a finite set of instructions and finite data. A program is a deliberate arrangement of instructions in an ordered process. To avoid confusion, one must differentiate between the order in which the programmer puts the instructions together and the order these instructions are later executed. The former, once established never changes while the latter may be unique to each possible value of the input. The order of execution is determined by a register, the program counter, which indicates at the present moment which instruction is to be executed. By default, the program counter evolves sequentially[1], but exceptionally its value can be altered by a subset of instructions.

This subset of instructions are said to control the flow of execution. It could be expected that these instructions would vary unpredictably in their application. Yet, their use can usually be systematized in a reduced number of structures which are available in almost all high-level programming languages. The most basic example of these so-called control structures are conditional statements.

$$\textbf{if } p\ x \textbf{ then } g\ x \textbf{ else } h\ x \tag{5.1}$$

Conditional statements represent a process by which, depending on some binary classification $p : A \to Bool$ of the input $x \in A$, control is referred back to either g or h.

In Classical computing, control is a very well understood topic, both operationally and semantically. In Quantum computing, however, there is still room for debate. Nevertheless, the majority of proposals for quantum programming languages converge on a paradigm where a classical computer controls the execution of (quantum) instructions which act on quantum data (Selinger, 2004). This paradigm has the name "*Classical Control - Quantum Data*".

Yet the possibility of quantum control remains open, in part by the fact that in the first formulation of quantum computing, the Quantum Turing Machine (Deutsch, 1985), control of execution is described by a unitary operator.

The objective of this chapter, and ultimately of this dissertation is to explore this concept. The approach is based on the modelling of conditional statements such as 5.1 in *distributive* categories and can be stated as follows:

- Formulation of a notion of control in the general class of *monoidal distributive categories* which is general enough to be applicable to the category **FHilb**, and additionally, when instantiated in **FHilb** it should

1 According to the first type of order.

capture the notion of *superposition of programs* introduced by Bădescu and Panangaden (2015) and Ying et al. (2014).

5.1 CLASSICAL CONTROL

Before attempting to formalize some type of conditional statement in the quantum setting it is important for contextualization purposes to review how constructs of the form

$$\textbf{if } p\ x \textbf{ then } g\ x \textbf{ else } h\ x \tag{5.2}$$

are modelled in classical computing, specifically in functional programming languages. The meaning of the program 5.2 is given as the function - let it be called $cond_p$ - defined by

$$cond_p(x) = \begin{cases} p\ x & \Longrightarrow\ g\ x \\ \neg(p\ x) & \Longrightarrow\ h\ x. \end{cases}$$

What is intended is to define this function in purely categorical terms, that is to say without any mention to a specific element x.

To begin it is necessary to establish the type of the morphism $cond_p$, and of its constituents. So as to be as encompassing as possible, $cond_p$ has the general type $cond_p : A \to B$, and consequently, since the category in question is **Set**, x is an element of A, $x \in A$. The morphims g and h have both $x \in A$ as input and thus have the type $g, h : A \to B$. Lastly p is a predicate and has the type $p : A \to Bool$.

In order for 5.2 to be expressible solely through a composite process, it is expected that $p : A \to Bool$, despite determining whether g or h are executed, does not have an effect on the input of g and h. Therefore $x \in A$ must be copied so that it serves as input to both $p : A \to Bool$ and either g or h. Only then can $p : A \to Bool$ be applied to one of the copies of $x \in A$. This two step process yields the following composition of morphisms:

$$A \xrightarrow{copy} A \times A \xrightarrow{id \times p} A \times Bool,$$

where $copy = \langle id, id \rangle$. At this point the object/type $A \times Bool$ can be interpreted to represent the input to which either g or h will be applied and a token that determines which will be applied. In **Set** the object *Bool* is identical[2] to $1 + 1$, where 1 is the singleton set (terminal object) and $+$ is the disjoint union (coproduct). Thus the above composition can be rewritten as

$$A \xrightarrow{\langle id, id \rangle} A \times A \xrightarrow{id \times p} A \times (1 + 1).$$

The last step of a conditional statement can be deduced from the words "either g or h" to be the unique morphism $[g, h] : A + A \to B$, for $g, h : A \to B$. The key point is the realization that $A + A$ and $A \times (1 + 1)$ are isomorphic. Specifically in **Set** the morphisms $dist : A \times (B + C) \to (A \times B) + (A \times C)$ and $unit : A \times 1 \to 1$ are natural isomorphisms. Thus one can construct the morphism

2 The class of isomorphic sets.

$$p? : A \xrightarrow{\langle id,id\rangle} A \times A \xrightarrow{id\times p} A \times (1+1) \xrightarrow{dist} (A\times 1)+(A\times 1) \xrightarrow{unit+unit} A+A,$$

called a guard. This process can be better visualized by the following diagram:

$$\begin{array}{ccc} A & \xrightarrow{\langle id,p\rangle} & A \times Bool \\ \downarrow{\scriptstyle p?} & & \downarrow{\scriptstyle dist} \\ A+A & \xleftarrow[unit+unit]{} & (A\times 1)+(A\times 1) \end{array}$$

Finally composing $p? : A \to A + A$ it with $[g, h] : A + A \to B$ yields the definition of conditional statement.

$$cond_p(g,h) : A \xrightarrow{p?} A+A \xrightarrow{[g,h]} B$$

What is important to conclude from this exercise is that the structures used in the definition of conditionals are not unique to **Set**. In fact, they can be generalized in a broad class of categories.

Definition 5.1. *A category $\mathcal{C}$ is said to be distributive if it has finite products, finite coproducts and the natural transformation*

$$[\pi_1 \times id, \pi_2 \times id] : Y \times X + Z \times X \to (Y + Z) \times X$$

is a natural isomorphism. We denote its inverse by

$$distl : (Y + Z) \times X \to Y \times X + Z \times X$$

From this definition and the previous discussion follows that the conditional statement usually interpreted in **Set** can be viewed as an instance of a more general class of conditional statements expressed in distributive categories.

5.2 CONTROL BY MEANS OF A MEASUREMENT

A first attempt at providing control structures to the quantum setting would probably arise through measurements, in that the outcome of a measurement on a system would subsequently determine some posterior action. Without much preamble it is foreseeable that control of this nature will not be quantum, given that the evolution of the system is contingent on the extraction of classical information - the set of tokens (eigenvalues) corresponding to the elements of the spectral decomposition of the measurement.

This can be stated in a biproduct dagger compact closed category as follows.

Definition 5.2. *Let $\mathcal{C}$ be a biproduct dagger compact closed category, then non-deterministic branching determined by a non-destructive measurement $\langle P_i\rangle_{i=0}^{i=1}$ on some qubit $Q \cong I + I$ is given by*

$$
\begin{array}{ccccc}
 & & Q \otimes A & & \\
 & & \downarrow{\scriptstyle \langle P_i \rangle_{i=0}^{i=1} \otimes id} & & \\
 & & (Q+Q) \otimes A & & \\
 & & \downarrow{\scriptstyle distl} & & \\
(Q \otimes A) & \xrightarrow{i_1} & (Q \otimes A) + (Q \otimes A) & \xleftarrow[i_2]{} & (Q \otimes A) \\
\downarrow{\scriptstyle id \otimes g} & & \downarrow{\scriptstyle (id \otimes g) + (id \otimes h)} & & \downarrow{\scriptstyle id \otimes h} \\
(Q \otimes B) & \xrightarrow{i_1} & (Q \otimes B) + (Q \otimes B) & \xleftarrow[i_2]{} & (Q \otimes B)
\end{array}
$$

for $g, h : A \to B$.

A construct such as this does not capture what is intended, in the sense that it cannot be called "quantum control". That is because it is not reversible, due to the indeterminism introduced by the use of measurements. Its meaning can be put in the words of Abramsky and Coecke (2004):

> ... suppose e.g. that we have a compound system $Q \otimes A$ and we (non-destructively) measure the qubit in the first component, obtaining $(Q \oplus Q) \otimes A$. At this point, we know 'locally', i.e. at the site of the first component, what the measurement outcome is, but we have not propagated this information to the rest of the system A. However after applying the distributivity isomorphism $(Q \oplus Q) \otimes A \cong (Q \otimes A) \oplus (Q \otimes A)$ the information about the outcome of the measurement propagates throughout the system, and we can perform operations on A depending on the measurement outcome, e.g $(1_Q \otimes U_0) \oplus (1_Q \otimes U_1)$ where U_0, U_1 are the operations we wish to perform on A in the event that the outcome of the measurement we performed on Q was 0 or 1 respectively.

Even though this construct does not pass the informal quantum control requirement, structurally it contains more than enough ingredients to define a construct which does meet the requirement. The structure here referred to is that of a distributive monoidal category.

5.3 QUANTUM CONTROL

In the section before last, the categorical properties needed to model classical conditional statements were deduced from a programmer's intuitive assessment of its behaviour and set-theoretical considerations. Consequently one can reverse this approach, whereby one can infer the particular notion of control in a programming language from the overall structure of the category which models it. Thus, what now follows is a definition of a conditional statement, very much inspired by the modelling of classical conditionals in distributive categories, in terms of structures found in Biproduct Dagger Compact Closed Categories. In the end, two definitions will emerge. One for **FHilb**, which is assumed to give semantics for a programming language based on closed quantum systems, and and another for $CPM(\mathbf{FHilb})^{\oplus}$, which is assumed to give semantics for a programming language based on open quantum systems.

The relevant structure needed in this case is the monoidal analogue to distributive categories.

Definition 5.3. *A monoidal category $(\mathcal{C}, \otimes, I)$ is monoidal (left) distributive if it has coproducts and the natural transformation*

$$[\pi_1 \otimes id, \pi_2 \otimes id] : Y \otimes X + Z \otimes X \to (Y + Z) \otimes X$$

is an isomorphism.[3] *We denote its inverse by*

$$distl : (Y + Z) \otimes X \to Y \otimes X + Z \otimes X$$

One can view the difference between monoidal distributive and distributive categories through the fact that the former does not make the assumption that $copy : A \to A \otimes A$ and $delete : A \otimes B \to A$ operations exist.

Proposition 5.1. *If $\mathcal{C}$ is a biproduct dagger compact closed category then it is distributive monoidal.*

Proof. By definition $\mathcal{C}$ is both symmetric monoidal and has coproducts. Moreover by 4.4 it is distributive monoidal. □

Definition 5.4. *Let $(\mathcal{C}, \otimes, I, +, distl)$ be a monoidal distributive category, then an object $C \cong I + I$ is called the* ***control unit*** *object.*

Definition 5.5. *Let $(\mathcal{C}, \otimes, I, +, distl)$ be a monoidal distributive category and $C \cong I + I$ be its control unit object. Then a conditional on $A \in \mathcal{C}$ with respect to C is a morphism $cond_q(g, h) : C \otimes A \to C \otimes B$ given by the following diagram*

$$\begin{array}{ccccc}
 & & C \otimes A & & \\
 & & \downarrow{\scriptstyle q \otimes id} & & \\
 & & (I + I) \otimes A & & \\
 & & \downarrow{\scriptstyle distl} & & \\
(I \otimes A) & \xrightarrow{i_1} & (I \otimes A) + (I \otimes A) & \xleftarrow[i_2]{} & (I \otimes A) \\
\downarrow{\scriptstyle id \otimes g} & & \downarrow{\scriptstyle (id \otimes g) + (id \otimes h)} & & \downarrow{\scriptstyle id \otimes h} \\
(I \otimes B) & \xrightarrow{i_1} & (I \otimes B) + (I \otimes B) & \xleftarrow[i_2]{} & (I \otimes B) \\
 & & \downarrow{\scriptstyle distl^{-1}} & & \\
 & & (I + I) \otimes B & & \\
 & & \downarrow{\scriptstyle q^{-1} \otimes id} & & \\
 & & C \otimes B & &
\end{array}$$

for $g, h : A \to B$.

This construct must satisfy the obvious requirement of preserving unitarity in a biproduct dagger compact closed category.

3 In the case C is symmetric, it is also right distributive, i.e., there is a natural isomorphism $distr : X \otimes (Y + Z) \to (X \otimes Y) + (X \otimes Z)$.

Proposition 5.2. *If q, g and h are unitary morphisms in a **BDCC** category, then $cond_q(g,h)$ is also unitary.*

Proof. Given that $distl$ and $distl^{-1}$ are given by

$$distl = \langle \pi_1 \otimes id, \pi_2 \otimes id \rangle$$
$$distl^{-1} = [q_1 \otimes id, q_2 \otimes id]$$

and in a **BDCC** $\langle f,g \rangle^\dagger = [f^\dagger, g^\dagger]$ it follows

$$distl^\dagger = [(\pi_1^\dagger \otimes id), (\pi_2 \otimes id)^\dagger] = [q_1 \otimes id, q_2 \otimes id] = distl^{-1}$$

thus both $distl$ and $distl^{-1}$ are unitary. Moreover, both composition, biproduct and monoidal functor preserve unitarity. Making $cond_q(g,h)$ unitary. □

In a **BDCC** category one can start an attempt at representing $cond_q(g,h)$. The biproduct structure provides a matrix calculus where a morphism of the form $f : A \oplus B \to C \oplus D$ is represented by the matrix

$$M_f := \begin{bmatrix} \pi_1^{C,D} \circ f \circ q_1^{A,B} & \pi_1^{C,D} \circ f \circ q_2^{A,B} \\ \pi_2^{C,D} \circ f \circ q_1^{A,B} & \pi_2^{C,D} \circ f \circ q_2^{A,B} \end{bmatrix}$$

thus the matrix corresponding to $l = (id \otimes g) \oplus (id \otimes h)$ can be calculated

$$\pi_1 \circ (id \otimes g) \oplus (id \otimes h) \circ q_1 = \pi_1 \circ [q_1 \circ (id \otimes g), q_2 \circ (id \otimes h)] \circ q_1 = \pi_1 \circ q_1 \circ (id \otimes g) = id \otimes g$$
$$\pi_2 \circ (id \otimes g) \oplus (id \otimes h) \circ q_1 = \pi_2 \circ [q_1 \circ (id \otimes g), q_2 \circ (id \otimes h)] \circ q_1 = \pi_2 \circ q_1 \circ (id \otimes g) = 0$$
$$\pi_1 \circ (id \otimes g) \oplus (id \otimes h) \circ q_2 = \pi_1 \circ [q_1 \circ (id \otimes g), q_2 \circ (id \otimes h)] \circ q_2 = \pi_1 \circ q_2 \circ (id \otimes h) = 0$$
$$\pi_2 \circ (id \otimes g) \oplus (id \otimes h) \circ q_2 = \pi_2 \circ [q_1 \circ (id \otimes g), q_2 \circ (id \otimes h)] \circ q_2 = \pi_2 \circ q_2 \circ (id \otimes h) = id \otimes h$$

Note that here id is really id_I thus $id \otimes g = g$ hence

$$M_l := \begin{bmatrix} g & 0 \\ 0 & h \end{bmatrix} : A \oplus A \to B \oplus B$$

for $g,h : A \to B$. As one would expect the composition of morphisms is given by matrix multiplication and addition by matrix addition.

As for the full interpretation of 5.5 it will need to be done in a specific category since the morphisms $q \otimes id$, $distl$ and $distl^{-1}$ do not have the form $f : A \oplus B \to C \oplus D$, thus a matrix representation of these can not be given at least at this abstract level.

5.4 CONDITIONALS IN **FHILB** (CLOSED QUANTUM SYSTEMS)

In order to instantiate 5.5 in the category of finite dimensional Hilbert spaces, one first needs to know the corresponding definitions of the abstract operations of $cond_q(g,h)$ in **FHilb**. These are given as follows:

- The monoidal bifunctor $- \otimes -$ is given by the tensor product.
- The unit is given by the set of complex numbers $\mathbb{C}$.
- The coproduct (biproduct) $- \oplus -$ is given by the direct sum.
- The control unit object C is $Q = \mathbb{C}^2 \cong \mathbb{C} \oplus \mathbb{C}$.
- An isomorphism $q : Q \to (I \oplus I)$ is defined as follows

$$\alpha|0\rangle + \beta|1\rangle \mapsto (\alpha, \beta) \tag{5.3}$$

where $|0\rangle$ and $|1\rangle$ are orthogonal vectors in Q.

One can now calculate the action of $cond_q(g,h)$ on some $x = (\alpha|0\rangle + \beta|1\rangle) \otimes |a\rangle \in Q \otimes A$, where $\{|0\rangle, |1\rangle\}$ is a basis of Q and $a \in A$.

$$\begin{aligned}
(\alpha|0\rangle + \beta|1\rangle) \otimes |a\rangle &\xrightarrow{q \otimes id} (\alpha, \beta) \otimes |a\rangle \\
&\xrightarrow{distl} (\alpha \otimes |a\rangle, \beta \otimes |a\rangle) \\
&\xrightarrow{id \otimes g \oplus id \otimes h} (\alpha \otimes g|a\rangle, \beta \otimes h|a\rangle) \\
&\xrightarrow{distl^{-1}} (\alpha, 0) \otimes g|a\rangle + (0, \beta) \otimes h|a\rangle \\
&\xrightarrow{q^{-1} \otimes id} \alpha|0\rangle \otimes g|a\rangle +_{\mathbb{C}} \beta|1\rangle \otimes h|a\rangle
\end{aligned}$$

This, in fact captures somewhat the notion of superposition of programs. Obviously superposition of programs is not a formalized concept, and therefore it may be ultimately meaningless. Yet it matches the definition in (Bădescu and Panangaden, 2015), which is written as

$$cond_q(g,h) = |0\rangle\langle 0| \otimes g + |1\rangle\langle 1| \otimes h \tag{5.4}$$

where $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$ are determined by $q_1 : I \to I + I$, which can be set by convention. To see this, consider the above calculation which yielded.

$$cond_q(g,h)[(\alpha|0\rangle + \beta|1\rangle) \otimes a] = \alpha|0\rangle \otimes g\, a + \beta|1\rangle \otimes h\, a.$$

And by

$$\begin{aligned}
&(|0\rangle\langle 0| \otimes g + |1\rangle\langle 1| \otimes h)((\alpha|0\rangle + \beta|1\rangle) \otimes a) \\
&= (|0\rangle\langle 0| \otimes g + |1\rangle\langle 1| \otimes h)(\alpha|0\rangle \otimes a + \beta|1\rangle \otimes a) \\
&= (|0\rangle\langle 0| \otimes g + |1\rangle\langle 1| \otimes h)(\alpha|0\rangle \otimes a) + (|0\rangle\langle 0| \otimes g + |1\rangle\langle 1| \otimes h)(\beta|1\rangle \otimes a) \\
&= (|0\rangle\langle 0| \otimes g)(\alpha|0\rangle \otimes a) + (|1\rangle\langle 1| \otimes h)(\alpha|0\rangle \otimes a) + (|0\rangle\langle 0| \otimes g)(\beta|1\rangle \otimes a) \\
&+ (|1\rangle\langle 1| \otimes h)(\beta|1\rangle \otimes a) \\
&= \alpha|0\rangle \otimes g\, a + \beta|1\rangle \otimes h\, a
\end{aligned}$$

and equality of functions one concludes that 5.4 holds. Also the matrix corresponding to $cond_q(g,h)$ is given as

$$cond_q(g,h) = |0\rangle\langle 0| \otimes g + |1\rangle\langle 1| \otimes h = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \otimes g + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \otimes h = \begin{bmatrix} g & 0 \\ 0 & h \end{bmatrix}, \tag{5.5}$$

which is exactly the matrix given previously, and matches the one given by Ying et al. (2014). It may seem odd that the morphism $distl$, $distl^{-1}$, $q \otimes id$ and $q^{-1} \otimes id$ do not affect the matrix representation. This is explained by the fact that all of this morphisms in $Mat_{\mathbb{C}}$[4] are more than natural isomorphisms, they are identity matrices (assuming the basis chosen is determined by q).

Another point of agreement with Bădescu and Panangaden (2015) is the fact that if the left input to $cond(g,h)$ is in a classical state relative to $q : C \to I + I$, for instance when $\beta = 0$, the result is a local operation on A, in this case $|0\rangle \otimes g\ a$.

Many quantum gates can be expressed as conditionals of this form. For instance, let q be the qubit decomposition associated with the basis $\{|0\rangle, |1\rangle\}$ and X be the NOT gate, then the CNOT gate can be written as follows

$$cond_q(id, X) : \mathbb{C} \otimes \mathbb{C} \to \mathbb{C} \otimes \mathbb{C}$$

$$cond_q(id, X) = |0\rangle\langle 0| \otimes id + |1\rangle\langle 1| \otimes X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Similarly the Toffoli gate can be defined by generalizing 5.5 to handle two 'predicates' A.4, as follows

$$cond_{q\otimes q}(id, id, id, X) : (\mathbb{C} \otimes \mathbb{C}) \otimes \mathbb{C} \to (\mathbb{C} \otimes \mathbb{C}) \otimes \mathbb{C}$$

$$cond_{q\otimes q}(id, id, id, X) = |00\rangle\langle 00| \otimes id + |10\rangle\langle 10| \otimes id + |01\rangle\langle 10| \otimes id + |11\rangle\langle 11| \otimes X$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Or the Franklin gate

$$cond_q(id, \sigma) : \mathbb{C} \otimes (\mathbb{C} \otimes \mathbb{C}) \to \mathbb{C} \otimes (\mathbb{C} \otimes \mathbb{C})$$

4 Category whose objects are natural numbers and morphisms are complex valued matrices.

$$cond_q(id,\sigma) = |0\rangle\langle 0| \otimes id_{\mathbb{C}\otimes\mathbb{C}} + |1\rangle\langle 1| \otimes \sigma_{\mathbb{C}\otimes\mathbb{C}}$$
$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

In the same vain one can write all controled unitaries.

5.5 CONDITIONALS IN $CPM(\mathbf{FHILB})^{\oplus}$ (OPEN QUANTUM SYSTEMS)

In the closed system case $cond_q(g,h)$ had to be unitary for it to constitute an appropriate transformation. In the open system case the criteria changes to the preservation of trace, since the morphisms of $CPM(\mathbf{FHilb})$ are already completely positive.

Proposition 5.3. *Let $\mathcal{C}$ be a **BDCC** category and $CPM(\mathcal{C})^{\oplus}$ be the category resulting from the action of the functor $F : \mathcal{C} \to CPM(\mathcal{C})$ (which filters the completely positive maps of $\mathcal{C}$) and the canonical biproduct embedding 4.2. If $g,\ h \in CPM(\mathcal{C})(A,B)$ are trace preserving then so is $cond_q(g,h)$.*

Proof. By 4.3 $id \otimes g \oplus id \otimes h$ is trace preserving for g and h trace preserving. Moreover $distl$ is unitary and thus preserves trace and since the functor $F : \mathcal{C} \to CPM(\mathcal{C})$ maps unitaries to unitaries $cond_q(g,h)$ is trace preserving. □

Once again, in order to instantiate $cond_q(g,h)$ in CPM(**FHilb**)$^{\oplus}$, the correspondence between the abstract operations of $cond_q(g,h)$ with those of CPM(**FHilb**)$^{\oplus}$ has to be made.

These are given as follows:

- The monoidal functor $- \otimes -$ is again given by the tensor product.
- The unit is given by the set of positive real numbers $\mathbb{R}_+$.
- The coproduct (biproduct) is given as in lemma 4.2.
- The control unit object C such that $q : C \cong I \oplus I$ is the (type of) bit.
- In this case the isomorphism $q : I \oplus I$ is an identity morphism.

One can now, given a (probabilistic) (α,β), density operator ρ, and trace preserving maps g and h calculate the action of $cond_q(g,h)$ on some $x = (\alpha,\beta) \otimes \rho \in Q \otimes A$ as follows

$$(\alpha,\beta) \otimes \rho \xmapsto{distl} (\alpha \otimes \rho, \beta \otimes \rho)$$

$$\xrightarrow{id\otimes g\oplus id\otimes h} (\alpha\otimes g\rho, \beta\otimes h\rho)$$
$$\xrightarrow{distl^{-1}} [(\alpha,0)\otimes g\rho\ +(0,\beta)\otimes h\rho]$$
$$= \alpha.g\rho +_{\mathbb{R}} \beta.h\rho$$

This does not capture the notion of superposition of programs. It is however to be expected since here the control unit is a probabilistic bit. What it does capture is a Markov process where operations g, h are performed on state space A with probabilities α and β respectively. And in the limit case where one of α and β is zero, it results in a classical conditional.

This can be seen more clearly by considering a pure state $P \in A$ and a unitaries g and h, then the above calculation unfolds as follows

$$(\alpha,\beta)\otimes P \mapsto \ldots \mapsto \alpha.gP + \beta.hP \tag{5.6}$$

which states that $cond_q(g,h)$ takes a (probabilistic) bit (α,β) and a pure state P and maps these to a mixed state whose "branches" are given by gP and hp with probabilities α and β respectively. This points to fact that control in the category $CPM(\mathbf{FHilb})^{\oplus}$ is classical, whereas in **FHilb** is quantum, even though the abstract categorical notion is the same. Additionally, it is commensurate with the fact that quantum programming languages which follow the motto of $Classical\ Control\ \&\ Quantum\ Data$ have semantics similar to $CPM(\mathbf{FHilb})^{\oplus}$.

5.6 CONTROL IN QPL

The goal of studying some programming construct is ultimately to integrate it into some programming language. For this to be a possibility such construct must be compatible with the semantics of the language. This section deals with the relation between the control structure defined in the previous section and the QPL programming language (Selinger, 2004). QPL as a language is more expressive than the ubiquitous quantum circuit model, since it explicitly deals with flow control and measurement. On the other hand, it is not as general as the Quantum Turing Machine, where both data and control flow are quantum, whereas in QPL flow control is exclusively classical. QPL is a functional language because every atomic or composite statement is represented by a linear mapping between tuples of density matrices. Furthermore, it supports high-level features including loops, recursion and procedure calls, albeit, due to its finite type system (based on finite dimensional vector spaces), it does not support inductive data types, thus it is incapable of dealing with variable size data.

The essential feature of QPL is its superoperator semantics. It contains two primitive data types: **bit** and **qbit**. The state for a typing context Γ containing n bits and m qubits is then given by a 2^n-tuple $(A_0, ..., A_{2^n-1})$ of density matrices, each of dimension $2^m \times 2^m$. A statement in QPL is interpreted as a morphism in the category **Q** of maps between such tuples.

Definition 5.6. *(Selinger, 2004) A signature is a list of non-zero natural numbers $\sigma = n_1, ..., n_s$. To each signature one associates a complex vector space*

$$V_\sigma = \mathbb{C}^{n_1\times n_1} \times ... \times \mathbb{C}^{n_s\times n_s}$$

The elements of V_σ are tuples of matrices $A = (A_1, ..., A_s)$, where the number and dimensions of each A_i are determined by σ.

Definition 5.7. *(Selinger, 2004) Let $F : V_\sigma \to V_{\sigma'}$ be a linear map. F is positive if $F(A)$ is positive for all $A \in V_\sigma$. Furthermore, F is completely positive if $id_\tau \otimes F : V_{\tau\otimes\sigma} \to V_{\tau\otimes\sigma'}$ is positive for all signatures τ. Finally, F, is a superoperator if it is completely positive and $tr\ F(A) \leq tr\ A$, where $tr\ A := \sum_i tr\ A_i$.*

The set $D_\sigma \subseteq V_\sigma$ is the set of *density matrix tuples*. The The Löwner order A.24 makes this set a complete partial order with the zero matrix 0 as the least element.

Definition 5.8. *The category **Q** is characterised as follows:*

- *Objects: signatures - $\sigma = n_1, ..., n_s$ representing complex vector spaces $V_\sigma = \mathbb{C}^{n_1\times n_1} \times ... \times \mathbb{C}^{n_s\times n_s}$.*
- *Morphisms: Superoperators between the undelying vector spaces of signatures - $F : \sigma \to \sigma'$ represents $F : V_\sigma \to V_{\sigma'}$.*
- *Composition ($\bullet$) : Let $F : \sigma \to \tau$ and $G : \epsilon \to \sigma$ be superoperators and $A = (A_1, ..., A_s)$ be a tuple of density matrices, then*

$$F \bullet G(A) := (F_1 \circ G_1(A_1), ..., F_s \circ G_s(A_s))$$

  for $F_i : \sigma_i \to \tau_i$, $G_i : \epsilon_i \to \sigma_i$, and where ($\circ$) is the composition of linear maps.

*Additionally, **Q** is equipped with the following structure:*

- *Coproduct ($\oplus$): The operation $\oplus$ denoted the concatenation of signatures, and the co-pairing map $[F, G] : \sigma \oplus \sigma' \to \tau$ is $[F, G](A, B) = F(A) + G(B)$.*
- *Tensor product: For signatures $\sigma = n_1, ..., n_s$ and $\tau = m_1, ..., m_t$, their tensor product is defined as*

$$\sigma \otimes \tau = n_1m_1, ..., n_1m_t, ..., n_sm_1, ..., n_sm_t.$$

  *while $F \otimes G : \sigma \otimes \sigma' \to \tau \otimes \tau'$ is given by $(F \otimes G)(A \otimes B) = F(A) \otimes G(A)$. This operation makes **Q** a strict symmetric monoidal category. Moreover the distributivity isomrphism is an identity*

$$(\sigma \oplus \sigma') \otimes \tau = (\sigma \otimes \tau) \oplus (\sigma' \otimes \tau).$$

- *CPO-enrichment[5]: The complete partial order on the set of density matrices can be generalized to every hom-set $\mathbf{Q}(\sigma, \sigma')$ since every morphisms in **Q** is monotone with respect to the Löwner*

5 Categories which are enriched are categories whose hom-sets are not only sets but carry additional structure, in this case each home-set $\mathbf{Q}(\sigma, \sigma')$ is a complete partial order.

order and also preserves least upper bounds of increasing sequences in $\boldsymbol{Q}(\sigma,\sigma')$ (Selinger, 2004). Moreover, all the categorical structures (composition, coproduct, and tensor product) preserve least upper bounds of increasing sequences. A complete partial order allows one to interpret recursive procedures. For instance if procedure Y is defined by $Y = X(Y)$. One can then interpret Y as the least fixed point A.3 of the Scott continuous function $\phi_X : \boldsymbol{Q}(\sigma,\sigma') \to \boldsymbol{Q}(\sigma,\sigma')$.

The category **Q** is very similar to $CPM(\mathbf{FHilb})^{\oplus}$, in fact **Q** is a subcategory of $CPM(\mathbf{FHilb})^{\oplus}$. They nonetheless differ in two aspects. The first is the fact that the concatenation operation $\oplus$ of **Q** is not a product, because the diagonal morphism $\langle id, id\rangle : \tau :\to \tau \oplus \tau$ does not preserve trace. The second aspect pertains to the fact that **Q** is not compact closed, since in general the image of the compact closed isomorphism $CPM(\mathbf{FHilb})^{\oplus}(A,B) \cong CPM(\mathbf{FHilb})^{\oplus}(I, A^* \otimes B)$ on a trace-preserving map is not a density matrix.

Nonetheless, the conditional statement presented in the previous chapter only requires monoidal distributivity, and since it is expressed purely in terms of the categorical structure it is monotone with respect to the order on each hom-set $\mathbf{Q}(\sigma,\sigma')$. Thus $cond_q(g,h)$ is a sensible construct in QPL. Its meaning is essentially the same as in $CPM(\mathbf{FHilb})^{\oplus}$, namely that of probabilistic branching, dependent on the probability distribution of a "coin" q.

Unfortunately the superposition of programs notion seems not to be compatible with the semantics of QPL. This assessment is corroborated by the work of Bădescu and Panangaden (2015). There, the authors define a construct called *quantum alternation* in the spirit of the one described in this work. The approach is to write it in terms of Kraus decompositions, and form the category **C** by replacing the morphisms of **Q** by Kraus decompositions. Lastly they attempt to lift the semantics of **C** to **Q** by extensional equality of Kraus decompositions 2.2.

Definition 5.9. *(Bădescu and Panangaden, 2015) The category **C** is defined as follows:*

- *Objects: signatures,*
- *Morphisms: Kraus decompositions,*
- *Composition: Given Kraus decompositions $\mathcal{S} \subseteq B(K,L)$[6] and $\mathcal{T} \subseteq B(H,K)$, their composition $\mathcal{S} \circ \mathcal{T}$ is defined to be the set obtained from the multiset $\{E \circ F |\ E \in \mathcal{S}, F \in \mathcal{T}\}$ by replacing l occurrences of a bounded K with $\sqrt{l}K$ and removing any occurrence of the zero operator.*

Definition 5.10. *(Bădescu and Panangaden, 2015) The quantum alternation of two morphisms $\mathcal{S},\mathcal{T} : H \to K$ is defined to be the morphism $\mathcal{S} \bullet \mathcal{T} : \textbf{\textit{qbit}} \otimes H \to \textbf{\textit{qbit}} \otimes K$ defined by*

$$\mathcal{S} \bullet \mathcal{T} = \left\{ \Pi_0 \otimes \frac{E}{\sqrt{|\mathcal{T}|}} + \Pi_1 \otimes \frac{F}{\sqrt{|\mathcal{S}|}} |\ E \in \mathcal{S}, F \in \mathcal{T} \right\},$$

where the projections Π_0 and Π_1 are determined by the qubit used in the alternation.

6 $B(K,L)$ stands for the set of bounded linear operators between Hilbert spaces K and L.

If one chooses $\mathcal{S} = \{U_0\}$ and $\mathcal{T} = \{U_1\}$ where both U_0 and U_1 are unitaries, $\mathcal{S} \bullet \mathcal{T}$ reduces to $cond_q(U_0, U_1)$ in **FHilb**.

Anyway, quantum alternation is not compatible with the semantics of QPL since it does not preserve extensional equality, that is, even if $\{U_0\}$ and $\{U_1\}$ are extensionally equal, and $\{V_0\}$ and $\{V_1\}$ are extensionally equal, it does not necessarily follow that $\{U_0\} \bullet \{V_0\}$ and $\{U_1\} \bullet \{V_1\}$ are extensionally equal. Moreover, if one defines an order on the hom-sets of $\mathbf{C}$ by means of the order $\sqsubseteq$ of $\mathbf{Q}$, by setting $\mathcal{S} \sqsubseteq \mathcal{T}$ if the relation holds for the corresponding superoperators, one finds that quantum alternation is not monotone to it (Bădescu and Panangaden, 2015), making it incompatible with recursion in QPL.

This seems to emphasize the incompatibility of quantum control in languages with semantics based on open quantum systems.

5.7 SUMMARY

This chapter was an effort to define a quantum conditional statement in terms of the structure of biproduct dagger compact closed categories. The technique employed was based on the modelling of classical conditional statements in distributive categories. For the purpose of illustrating control structures in the quantum setting, a measurement based conditional statement was presented as a counter example of what quantum control should represent. Finally, a definition of a conditional-like statement was specified within the framework of distributive monoidal categories. Since the biproduct dagger compact closed categories **FHilb** and $CPM(\mathbf{FHilb})^{\oplus}$ are both distributive monoidal, it was then possible to instantiate the definition proposed in closed and open quantum systems.

In closed quantum systems **FHilb** the definition yielded the idea of superposition of programs previously described by Bădescu and Panangaden (2015) and Ying et al. (2014). Using the construct here described one can write th usual controlled operations such as the CNOT, Toffoli, and Franklin gates, and many others. In the case of open quantum systems $CPM(\mathbf{FHilb})^{\oplus}$ the definition resulted in probabilistic branching. This disparity is due essentially to the difference in biproducts and unit object between both categories.

To conclude, a comparison is made with the work of Bădescu and Panangaden (2015) within the context of the QPL programming language (Selinger, 2004). The approach taken in the former was to define quantum control in terms of Kraus decompositions, and then lift it to the superoperators semantics of QPL. This turned out not to be possible. This is comensurate with the results gathered in this dissertation, since the instantiation in QPL of the notion of quantum control here presented (this is possible since the category modelling QPL is distributive monoidal) results not in superposition of programs, but in probabilistic branching (same as in $CPM(\mathbf{FHilb})^{\oplus}$).

6

CONCLUSION

6.1 CONCLUSIONS

The purpose of this dissertation was to study and to some extent an attempt to formalize the ambiguous and slightly controversial notion of quantum control. The approach taken was to provide a definition of a conditional like statement aimed at capturing superposition of programs in the language of Biproduct Dagger Compact Closed Categories. These surged, in the tradition of using category theory as a semantic environment for programming languages, as an axiomatic framework for Quantum Theory, aimed at shining a light to its computational and information processing aspects, such as sequential and parallel composition, entanglement, resource sensitivity, and the flow of classical information.

These framework covers both the closed and open system formalisms of Quantum Mechanics. The categorical passage from the former to the latter is achieved through a functor, which however does not preserve the biproduct structure. Meaning, the biproducts of the categorical description of closed quantum systems are not determined by those of closed quantum systems. It is indeed this shift that causes the disparity in interpretation of the conditional statement provided in this thesis when instantiated in **FHilb** (closed quantum systems) and $CPM(\mathbf{FHilb})$ (open quantum systems). In **FHilb** the observed behaviour of the aforementioned conditional construct is that of what one can intuit to be superposition of programs, which is in concordance with previous work on the subject (Bădescu and Panangaden, 2015), (Ying et al., 2014). On the other hand, in $CPM(\mathbf{FHilb})$ the behaviour witnessed is that of probabilistic branching. This is somewhat to be expected since $CPM(\mathbf{FHilb})$ is very closely related to a category which gives semantics to a programming language (Selinger, 2004) based on the classical control and quantum data paridigm.

In any case, there are two clear problems with quantum control, at least as defined in this dissertation. The first is the already mentioned incompatibility with the open quantum system formalism. The other problem pertains to the nature of the quantum computational advantage. That is, quantum control acts by assigning two quantum programs (unitary maps) to the basis states of a particular decomposition of a qubit (by convention one can set this to be the computational basis), which will then be applied to a second system which together with the qubit constitute a compound system. Consequently, if one wishes to extend the state space of the control qubit in order to increase the number of programs in superposition, the number of programs one would have to specify

would grow exponentially, essentially doing exactly the opposite of what partly affords the quantum advantage. That being, the ability to apply a single program to the whole state space.

6.2 PROSPECT FOR FUTURE WORK

Conditional statements are not the only means of controlling the flow of execution of a program. Recursion is another control structure/paradigm in a programmer's arsenal. In fact, recursion is the principal way of constructing programs in functional programming languages. In the quantum setting, recursion has been predominantly treated in the same way as control flow, namely by leaving recursion calls to be dealt by a classical controller. Recursion in QPL is an instance of this philosophy. Although one might suspect that unrestricted recursion of quantum nature is most likely meaningless, structured recursion might offer a viable alternative.

A restrictive type of recursion can be modelled categorically by what are called initial algebras. These give semantics to the so-called inductive data types. An initial algebra in a category $\mathcal{C}$ is a tuple $(in : FA \to A, A)$, where F is an endofunctor and A is an object of $\mathcal{C}$ such that

$$\begin{array}{ccc} A & \xleftarrow{in} & FA \\ {\scriptstyle k}\downarrow & & \downarrow{\scriptstyle Fk} \\ B & \xleftarrow[\alpha]{} & FB \end{array}$$

commutes for any other tuple $(\alpha : FB \to B, B)$, where k is referred to as a *catamorphism* and in is an isomorphism. The commutativity of the diagram gives the recursive definition:

$$in \circ k = \alpha \circ Fk.$$

For instance, the initial algebra of the functor $FX = 1 + X$ in **Set** yields the type of natural numbers $\mathbb{N}$. Similar functors result in the List datatype, the binary Tree datatype, and others.

Recursive types in quantum programming languages have been the subject of some research. For instance, Selinger (2004) suggests an extension of the type system of QPL to include recursively defined lists, whose structure is however classical. An alternative approach has been devised by Neri (2018) and Neri et al. (2021), in which a reversible version of list catamorphisms is built by a technique called *minimal complementation* which is then generalized to a quantum computation by means of the vector space monad in the Haskell programming language. Another attempt at incorporating recursive types in a quantum computing was put foward by Sabry et al. (2018). In this instance the authors develop a reversible language with lists and fixed points, which is then extended to the closed quantum system setting by permitting linear combination of terms and restricting fixpoints to structurally recursive fixpoints. As a consequence a form of quantum control arises through the list structure.

As future work it would be interesting to relate both works mentioned above with the form of quantum control presented in this dissertation, possibily leading to formulating initial algebras and thus semantics for inductive data types in the biproduct dagger compact closed framework.

BIBLIOGRAPHY

Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004.*, pages 415–425. IEEE, 2004.

Samson Abramsky and Bob Coecke. Categorical quantum mechanics. *Handbook of quantum logic and quantum structures*, 2:261–325, 2009.

Samson Abramsky and Nikos Tzevelekos. Introduction to categories and categorical logic. In *New structures for physics*, pages 3–94. Springer, 2010.

Jiří Adámek, Stefan Milius, and Lawrence S Moss. Fixed points of functors. *Journal of logical and algebraic methods in programming*, 95:41–81, 2018.

Thorsten Altenkirch and Jonathan Grattage. A functional quantum programming language. In *20th Annual IEEE Symposium on Logic in Computer Science (LICS'05)*, pages 249–258. IEEE, 2005.

Costin Bădescu and Prakash Panangaden. Quantum alternation: Prospects and problems. *arXiv preprint arXiv:1511.01567*, 2015.

Charles H Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Physical review letters*, 70(13):1895, 1993.

Richard Bird and Oege De Moor. The algebra of programming. In *NATO ASI DPD*, pages 167–203, 1996.

Church. Alonzo church. an unsolvable problem of elementary number theory. american journal of mathematics, vol. 58 (1936), pp. 345–363. *The Journal of Symbolic Logic*, 1(2):73–74, 1936a.

Alonzo Church. An unsolvable problem of elementary number theory. *American journal of mathematics*, 58(2): 345–363, 1936b.

Alonzo Church. A formulation of the simple theory of types. *The journal of symbolic logic*, 5(2):56–68, 1940.

Bob Coecke, John Selby, and Sean Tull. Two roads to classicality. *arXiv preprint arXiv:1701.07400*, 2017.

Haskell B Curry. The inconsistency of certain formal logics. *The Journal of Symbolic Logic*, 7(3):115–117, 1942.

Haskell Brooks Curry, Robert Feys, William Craig, J Roger Hindley, and Jonathan P Seldin. *Combinatory logic*, volume 1. North-Holland Amsterdam, 1958.

David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

Richard P Feynman. Simulating physics with computers. In *Feynman and computation*, pages 133–153. CRC Press, 2018.

Jeremy Gibbons. Conditionals in distributive categories. Technical report, CMS-TR-97-01, School of Computing and Mathematical Sciences, Oxford Brookes . . . , 1997.

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.

Gregory M Kelly and Miguel L Laplaza. Coherence for compact closed categories. *Journal of pure and applied algebra*, 19:193–213, 1980.

Joachim Lambek. From λ-calculus to cartesian closed categories. *To HB Curry: essays on combinatory logic, lambda calculus and formalism*, pages 375–402, 1980.

Joachim Lambek and Philip J Scott. *Introduction to higher-order categorical logic*, volume 7. Cambridge University Press, 1988.

Saunders Mac Lane. *Categories for the working mathematician*, volume 5. Springer Science & Business Media, 2013.

Erik Meijer, Maarten Fokkinga, and Ross Paterson. Functional programming with bananas, lenses, envelopes and barbed wire. In *Conference on functional programming languages and computer architecture*, pages 124–144. Springer, 1991.

Barry Mitchell. *Theory of categories*. Academic Press, 1965.

E Moggi. Computational lambda-calculus and monads. in 4'th symposium on logic in computer science, 1989.

Eugenio Moggi. Notions of computation and monads. *Information and computation*, 93(1):55–92, 1991.

A. Neri. Towards quantum program calculation. Master's thesis, University of Minho, Informatics Department, October 2018. (PDF).

Ana Neri, Rui Soares Barbosa, and José Nuno Oliveira. Compiling quantamorphisms for the IBM Q Experience. *IEEE Transactions on Software Engineering*, 2021. doi: 10.1109/TSE.2021.3117515.

Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

nLab authors. adjoints preserve (co-)limits. http://ncatlab.org/nlab/show/adjoints%20preserve%20%28co-%29limits, February 2021. Revision 8.

Amr Sabry, Benoît Valiron, and Juliana Kaizer Vizzotto. From symmetric pattern-matching to quantum control. In *International Conference on Foundations of Software Science and Computation Structures*, pages 348–364. Springer, 2018.

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

Peter Selinger. Dagger compact closed categories and completely positive maps. *Electronic Notes in Theoretical computer science*, 170:139–163, 2007.

Peter Selinger. Lecture notes on the lambda calculus. *arXiv preprint arXiv:0804.3434*, 2008.

Peter Selinger. Finite dimensional hilbert spaces are complete for dagger compact closed categories. *arXiv preprint arXiv:1207.6972*, 2012.

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.

Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London mathematical society*, 2(1):230–265, 1937.

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

Mingsheng Ying, Nengkun Yu, and Yuan Feng. Alternation in quantum programming: from superposition of data to superposition of programs. *arXiv preprint arXiv:1402.5172*, 2014.

A

APPENDICES

A.1 LINEAR ALGEBRA

Definition A.1. *A **vector space** over a field $\mathbb{K}$ is a set V, admitting two operations, called addition and multiplication by scalars, subject to the rules*

1. *If $x \in V$ and $y \in V$ then $x + y \in V$.*
2. *If $x \in V$ and $c \in \mathbb{K}$ then $cx \in V$.*
3. *$x + y = y + x$ for all $x, y \in V$.*
4. *$(x + y) + z = x + (y + z)$ for all $x, y, z \in V$.*
5. *There is a vector $\mathbf{0} \in V$ such that $x + \mathbf{0} = \mathbf{0} + x = x$ for all $xinV$.*
6. *For each $x \in V$ there is a unique vector $-x \in V$ such that $x + (-x) = (-x) + x = \mathbf{0}$.*
7. *$c(x + y) = cx + cy$, $(c + d)x = cx + dx$ for all $x, y \in V$ and $c, d \in \mathbb{K}$.*
8. *$(cd)x = (cd)x$ for all $x \in V$ and $c, d \in \mathbb{K}$.*
9. *There is a unique $\mathbf{1} \in \mathbb{K}$ such that $\mathbf{1}x = x$ for all $x \in V$.*

Definition A.2. *Let V and W be vector spaces over the field $\mathbb{K}$. A map $T : V \to W$ is **linear** if*

$$\begin{aligned} T(u + v) &= T(u) + T(v) && \text{for all } u, v \in V \\ T(a \cdot v) &= a \cdot T(v) && \text{for all } a \in \mathbb{K} \text{ and } v \in V \end{aligned}$$

The set of all linear maps from V to W is denoted by $\mathbb{L}(V, W)$. In the case $\mathbb{L}(V, V)$ we abreviate to $\mathbb{L}(V)$.

Definition A.3. *If there is some bijection $i \in \mathbb{L}(V, W)$, then V and W are **isomorphic** vector spaces. The map $i : V \to W$ is called a **vector space isomorphism**.*

Theorem A.1. *The set $L(V,W)$ is a vector space. Addition $S+T : V \to W$ and scalar multiplication $aT : V \to W$ are defined pointwise as*

$$(S+T)x = Sx + Tx \quad (aT)x = a(Tx), \text{ for } S,T \in L(V,W)$$

Proof. Let $S,T \in L(V,V)$

1. $L(V,W)$ is closed under (linear map) addition

$$\begin{aligned}(S+T)c(x+x') &= (S+T)(cx+cx')\\ &= (S(cx+cx') + T(cx+cx')\\ &= (Scx + Tcx + Scx' + Tcx')\\ &= ((S+T)cx + (S+T)cx')\\ &= c((S+T)x + (S+T)x')\end{aligned}$$

2. $L(V,W)$ is closed under (linear map) scalar multiplication

$$\begin{aligned}(aT)c(x+x') &= (aT)(cx+cx')\\ &= (aTcx + aTcx')\\ &= c(aTx + aTx')\end{aligned}$$

3. $S+T = T+S$ holds, since

$$(S+T)x = Sx + Tx = Tx + Sx = (T+S)x$$

4. Addition is shown to be associative

$$\begin{aligned}((S+T)+Z)x &= (S+T)x + Zx = Sx + Tx + Zx\\ &= Sx + (T+S)x\\ &= (S + (T+Z))x\end{aligned}$$

5. There is a linear map denoted $\mathbf{0} \in L(V,W)$ defined as $\mathbf{0}x = 0$ which serves as unit of linear map addition, since it satisfies $T + \mathbf{0} = T$ for all $T \in L(V,W)$.

6. For all linear mappings $T \in L(V,V)$ there is another defined as $-T = (-1)T$ which serves as additive inverse since $T + (-T) = \mathbf{0}$.

The other axioms follow trivially from the previous. □

Definition A.4. *Let U, V, W be vector spaces over the same field. For any pair of linear maps $T \in \mathbb{L}(U,V)$ and $S \in \mathbb{L}(V,W)$ there is a unique linear map given by the composition of S and T, namely*

$$(S \circ T) : U \to W$$
$$(S \circ T)x = S(Tx)$$

for all $x \in U$.

Definition A.5. *For every vector space V over a field $\mathbb{K}$ there is a set $L(V,\mathbb{K})$ of mappings $F : V \to \mathbb{K}$ called linear functionals. Moreover if we consider $\mathbb{K}$ to be a vector space over itself, then $L(V,\mathbb{K})$ is itself a vector space which we call the **dual space** of V denoted V^*.*

Definition A.6. *Let V be a vector space over a field $\mathbb{K}$. $v \in V$ is a **linear combination** of the vectors V_0, V_1, ..., V_{n-1} if V can be written as*

$$v = c_0v_0 + c_1v_1 + ... + c_{n-1}v_{n-1}$$

for some $c_0, c_1, ..., c_{n-1} \in \mathbb{K}$.

Definition A.7. *A set $\{v_0, v_1, ..., v_{n-1}\}$ of vectors in V is called **linearly independent** if*

$$\mathbf{0} = c_0v_0 + c_1v_1 + ... + c_{n-1}v_{n-1}$$

implies that $c_0 = c_1 = ... = c_{n-1} = 0$.

Definition A.8. *A set $B \subseteq V$ of vectors is called a **basis** of a vector space V if both*

1. *every, $v \in V$ can be written as a linear combination of vectors from B and*
2. *B is linearly independent.*

Definition A.9. *Let V and W be vector spaces over the same field $\mathbb{K}$. The **direct sum** $V \oplus W$ is a vector space defined by*

$$V \oplus W = V \times W = \{(v,w) : v \in V.w \in W\} \tag{A.1}$$

and with the vector operations

$$(v_1,w_1) + (v_2,w_2) = (v_1 + v_2, w_1 + w_2)$$
$$c(v,w) = (cv,cw).$$

The direct sum of vector spaces comes equipped with projection maps

$$\pi_i : V_1 \oplus V_2 \to V_i$$

$$\pi_1(v_1, v_2) = v_1$$
$$\pi_2(v_1, v_2) = v_2$$

and coprojection maps

$$q_i : V_i \to V_1 \oplus V_2$$
$$q_1 v = (v, 0)$$
$$q_2 v = (0, v)$$

Definition A.10. *Let V be a vector space over the complex numbers. V is a **complex inner product space** if it is equipped with an operation $\langle -, - \rangle : V \times V \to \mathbb{C}$ called the inner product. The inner product must obey the following properties for all $v, u, w \in V$ and $a, b \in \mathbb{C}$:*

1. *Skew symmetric:*
$$\langle u, v \rangle = \langle v, u \rangle^*$$

2. *Non-degenerate:*
$$\langle v, v \rangle \geq 0, \quad \langle v, v \rangle = 0 \text{ iff } v = \mathbf{0}$$

.

3. *Linear:*
$$\langle v, au + bw \rangle = a\langle v, u \rangle + b\langle u, w \rangle$$

Definition A.11. *For every complex inner product space $(V, \langle -, - \rangle)$ a **norm**, i.e., a function $| \, | : V \to \mathbb{C}$, can be defined as*
$$|v| = \sqrt{\langle v, v \rangle}$$

Definition A.12. *A basis $B = \{v_0, v_1, ..., v_{n-1}\}$ for an inner product space $(V, \langle -, - \rangle)$ is called an **orthonormal basis** if*
$$\langle v_j, v_k \rangle = \delta_{i,j} = \begin{cases} 1, & \text{if } j = k \\ 0, & \text{if } j \neq k \end{cases}$$

Definition A.13. *For every complex inner product space $(V, \langle -, - \rangle)$, we can define a distance function $d(\, , \,) : V \times V \to \mathbb{R}$ as*

$$d(u, v) = |v - u| = \sqrt{\langle u - v, u - v \rangle}.$$

Definition A.14. *Within an inner product space $(V, \langle -, - \rangle)$, a sequence of vectors $v_0, v_1, v_2, ...$ is called a **Cauchy sequence** if for every $\epsilon \geq 0$, there exists an $N_0 \in \mathbb{N}$ such that*

$$d(v_m, v_m) \leq e, \quad \forall m, n \geq N_0$$

where $d(\ ,\ )$ is the distance function defined on $(V, \langle -, - \rangle)$

Theorem A.2. *Every Euclidean space, i.e., a finite-dimensional inner product space, is isomorphic to its dual.*

Definition A.15. *A complex inner product space is called **complete** if for any Cauchy sequence of vectors $v_0, v_1, v_2, ...$ there exists a vector $w \in V$ such that*

$$\lim_{n\to\infty} |v_n - w| = 0$$

Definition A.16. *A **Hilbert space** is a complex inner product space that is complete.*

Definition A.17. *A linear map $f : H \to K$ between Hilbert spaces is **bounded** when there is a number $b \in \mathbb{R}$ such that $||f(v)|| \leq b.||v||$ for all $v \in H$.*

Definition A.18. *For a bounded linear map $f : H \to K$, its **adjoint** $f^\dagger : K \to H$ is the unique linear map, such that for all $|\phi\rangle \in H$ and $\psi \in K$*

$$\langle f(\phi)|\psi\rangle = \langle \phi|f^\dagger(\psi)\rangle \tag{A.2}$$

Definition A.19. *A bounded linear map $U : H \to H$ is*

- *an **isometry** if: $U^\dagger U = I$,*
- ***unitary** if: $U^\dagger U = I = UU^\dagger$,*
- ***self-adjoint** if: $U^\dagger = U$.*

Theorem A.3. *(Spectral) If a bounded linear map $A : H \to H$ is an isometry, i.e., then there is a set $\{a_i\}$ and a set $\{|i\rangle\}$ such that*

$$A = \sum_{i=0}^{|H|} a_i |i\rangle\langle i|.$$

*The elements $\{a_i\}$ are the **eigenvalues** of A and $\{|i\rangle\}$ is an orthonormal basis of H and also the set of **eigenvectors** of A.*

Definition A.20. *The **trace** of an operator T is a mapping $Tr : H \to \mathcal{K}$ such that for any orthonormal basis $\{x_1, ..., x_n\}$ of H, as*

$$Tr(T) = \sum_{i=1}^{n} \langle x_i|Tx_i\rangle \tag{A.3}$$

Definition A.21. *A self-adjoint operator T is **positive** if $\langle x|Tx\rangle \geq 0$ for all $x \in H$.*

Definition A.22. *A **density operator** on a Hilbert space is a bounded linear map $\rho in \mathbb{L}(H)$ that is*

- *positive and*

- $Tr(\rho) = 1$.

Definition A.23. *The set D_n is the set of all density matrices of dimension n, $D_n = \{A \in \mathcal{C}^{v\times n} \mid A$ positive hermitian and tr $A \leq 1\}$*

Definition A.24. *(The **Löwner order**) For matrices $A, B \in \mathcal{C}^{n\times n}$, then $A \leq B$ if $B - A$ is positive.*

Remark A.1. *The poset $(D_n, \leq)$ is a complete partial order.*

Remark A.2. *Let H_1 and H_2 are Hilbert spaces then $H_1 \oplus H_2$ is also a Hilbert space with the inner product*

$$\langle (h_1, k_1), |(h_2, k_2)\rangle = \langle h_1, h_2\rangle + \langle k_1, k_2\rangle$$

Definition A.25. *Let H_1 and H_2 be Hilbert spaces. The tensor product $H_1 \otimes H_2$ is a Hilbert space, such that*

$$x \in H_1 \otimes H_2 \text{ iff } x = \sum_{i,j}^{m,n} c_{ij} e_i \otimes e_j \tag{A.4}$$

where $\{e_i\}^m$ and $\{e_j\}^n$ are basis are of H_1 and H_2, respectively.The inner product of $H_1 \otimes H_2$ is defined as

$$\langle \phi \otimes \phi_2, \psi_1 \otimes \psi_2\rangle_{H_0\otimes H_1} = \langle \phi_1, \psi_1\rangle_{H_0} \langle \phi_2, \psi_2\rangle_{H_1}$$

Additionally, given two linear maps $S : V \to X$ and $T : W \to Y$, the tensor product of the two linear maps is a linear map

$$S \otimes T : V \otimes W \to X \otimes Y$$

defined by

$$(S \otimes T)(v \otimes w) = S(v) \otimes T(w)$$

Definition A.26. *The **partial trace** over a state space B Tr_B is a mapping $L(H_A \otimes H_B) \to L(H_A)$ defined as follows. Given an operator $O = M_A \otimes N_B \in L(H_A \otimes H_B)$, then the partial trace is given by the linear extension of the mapping*

$$Tr_B(M_A \otimes N_B) = M_A Tr(N_B) = M_A \sum_u \langle u|N_B|u\rangle$$

where $\{|u\rangle\}$ is a basis of H_B.

A.2 CATEGORY THEORY

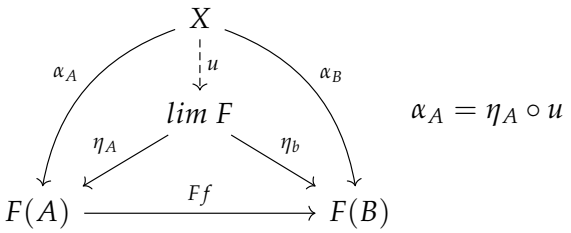
Definition A.27. *A diagram of shape $\mathcal{I}$ in $\mathcal{C}$ is a functor $\mathcal{I} \to \mathcal{C}$.*

Let Δ_X be the constant diagram mapping every object in $\mathcal{I}$ to $X \in \mathcal{C}$ and every morphism in $\mathcal{C}$ to the identity on X. Then

Definition A.28. *The **limit** of a diagram $F : \mathcal{I} \to \mathcal{C}$ is an object $lim\ F$ together with the isomorphism*

$$u : \mathcal{C}(X, lim\ F) \cong [\mathcal{I}, \mathcal{C}](\Delta_X, F)$$

natural in X. The correspondence can be seen diagrammatically as



This notion can be dualised, giving the notion colimit, i.e., the natural isomorphism

$$u : \mathcal{C}(colim\ F, X) \cong [\mathcal{I}, \mathcal{C}](F, \Delta_X)$$

It can be easily shown that the the universal constructions above, can be given by limits a a simple diagrams. For instance the limit/colimit of the diagram $F := \bullet \quad \bullet \mapsto A \quad B$ is a product/copruduct of A and B.

Definition A.29. *An adjunction between categories $\mathcal{C}$ and $\mathcal{D}$ is a pair of functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ together with natural transformations $\eta : id_{\mathcal{C}} \to GF$, called the unit, and $\epsilon : FG \to id_{\mathcal{D}}$, called the counit, so that for all objects $X \in \mathcal{C}$ and $Y \in \mathcal{D}$ the diagrams below commute*

Theorem A.4. *Let $\mathcal{C}$ and $\mathcal{D}$ be categories and*

$$\mathcal{C} \underset{G}{\overset{F}{\rightleftarrows}} \mathcal{D}$$

a pair of adjoint functors between them. Then If $X : \mathcal{I} \to \mathcal{C}$ is a diagram whose limit $lim\ X$ exists in $\mathcal{C}$, then this limit is preserved by the right adjoint G in that there is a natural isomorphism

$$G(lim\ X) \cong lim\ (G(X)),$$

where on the right we have the limit in $\mathcal{D}$ over the diagram $R \circ X : \mathcal{I} \to D$. Dually it can be said that left adjoints preserves colimits.

One very useful corollary of the theorem is the fact that right adjoints preserve products and left adjoints preserve coproducts.

Definition A.30. *Let $(\mathcal{C}, \otimes, I)$ be a symmetric monoidal category. The internal hom in $\mathcal{C}$ is a functor*

$$[-,-] : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{C}$$

such that for every $X \in Ob(\mathcal{C})$ there is a pair of adjoint functors

$$(- \otimes X \dashv [X,-]) : \mathcal{C} \to \mathcal{C}$$

If a symmetric monoidal category has such an internal hom then it is called a closed monoidal category.

A.3 KLEENE'S FIXED-POINT THEOREM

Definition A.31. *A partial order, or a poset $(D, \sqsubseteq)$ consists of a set D and a binary relation $\sqsubseteq$ on D satisfying the properties*

1. ***reflexive**, i.e., $x \sqsubseteq x$ for all $x \in D$.*
2. ***transitive**, i.e., for all $x, y, z \in D$, $x \sqsubseteq y$, and $y \sqsubseteq y$ imply $x \sqsubseteq z$.*
3. ***anti-symmetric**, i.e., if $x \sqsubseteq y$ and $y \sqsubseteq x$ for some $x, y \in D$ then $x = y$.*

A poset (D,$\sqsubseteq$) forms a category. The objects are the elements of D, and if $A \sqsubseteq B$ for $A,\ B \in D$, then there is a morphism $f : A \to B$. Reflexivity gives the identity, and transitivity gives composition. The hom-sets of (D,$\sqsubseteq$) contain either one or zero elements.

Definition A.32. *- Suppose D is a poset and $S \subseteq D$. An element $d \in S$ is the least element of S if $\forall x \in S. d \sqsubseteq x$. Moreover, if it exists the least element of the whole poset is denoted $\bot_D$ and is uniquely determined by $\forall_{d \in D}.\ \bot \sqsubseteq d$.*

The object $\bot_D$ is the initial object in (D,$\sqsubseteq$).

Definition A.33. *- A countable increasing chain ($\omega-$chain) in a poset is a sequence of elements of D such that $d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq$ .... An upper bound for the chain is any $d \in D$ such that $\forall_{n \in \mathbb{N}}. d_n \sqsubseteq d$. If it exists, the least upper bound of the chain is denoted as $\bigcup_{n \geq 0} d_n$. Thus by definition:*

$$\forall_{m \in \mathbb{N}}. d_m \sqsubseteq \bigcup_{n \geq 0} d_n.$$
$$\forall_{d \in D} \forall_{m \geq 0}.\ d_m \sqsubseteq d \implies \bigcup_{n \geq 0} d_n \sqsubseteq d$$

In (D,$\sqsubseteq$) the product of two objects A and B is given by the least upper bound of A and B.

Definition A.34. *A poset $(D, \sqsubseteq)$ is called a **complete partial order** iff all of its chains have a least upper bound.*

Definition A.35. *If $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ are two posets and $F : D \to D'$ is a function, then F is called **monotone** iff $\forall_{x,y \in D}.\ x \sqsubseteq y \implies F(x) \sqsubseteq' F(y)$.*

A monotone function between posets - considered as categories - are functors.

Definition A.36. *A monotone function $F : (D, \sqsubseteq) \to (D', \sqsubseteq')$ is **Scott continuous** iff $F(\bigcup_{n \geq 0} d_n) \sqsubseteq' \bigcup_{n \geq 0} F(d_n)$, for any chain $\{d_n | n \in \mathbb{N}\}$ in $(D, \sqsubseteq)$.*

Theorem A.5. *(Kleene) Let $(D, \leq)$ be a complete partial order with least element $\bot$. Then every Scott-continuous function $f : D \to D$ has a least fixed point given by the supremum of its w-chain, i.e*

$$lfp(f) = sup(\{f^n(\bot) | n \in \mathbb{N}\}$$

Proof. First, by induction: $\bot \leq f(\bot)$ (definition of least element) and $f^{n-1}(\bot) \leq f^n(\bot) \implies f^n(\bot) \leq f^{n+1}(\bot)$ (by monotonicity of f). So $M = \{f^n(\bot) | n \in \mathbb{N}\}$ is a w-chain in D and (by the definition of CPO) it has a supremum $sup(M) = \bigcup_n f^n(\bot)$. By continuity $f \bigcup_n f^n(\bot) = \bigcup_n f(f^n(\bot))$ and immediately follows (by the definition of supremum) $\bigcup_n f^n(\bot) = \bigcup_n f(f^n(\bot))$, thus $sup(M)$ is a fixed point f. It remains to show that $sup(M)$ is the least fixed point. Again by induction, let k be another fixed point of f, then $\bot \leq k$ and $f^n(\bot) \leq k \implies f^{n+1} \leq f(k)$ (by monotonicity of f). Since k is a fixed point then $f^{n+1} \leq k$ and it immediately follows $\bigcup_n f^n(\bot) \leq k$. □

A.4 QUANTUM CONDITIONAL WITH TWO 'PREDICATES'

Definition A.37. *Let $(\mathcal{C}, \otimes, I, +, dist)$ be a monoidal distributive category and $C \cong I + I$ be its control unit object. Then a generalized conditional on $A \in \mathcal{C}$ with respect to $C \otimes C$ is a morphism $cond_{q\otimes q}(g,h,f,l) : (C \otimes C) \otimes A \to (C \otimes C) \otimes B$ given by the following diagram*

$$
\begin{array}{c}
(C \otimes C) \otimes A \\
\downarrow{\scriptstyle (q\otimes id)\otimes id} \\
((I \oplus I) \otimes C) \otimes A \\
\downarrow{\scriptstyle distl\otimes id} \\
((I \otimes C) + (I \otimes C)) \otimes A \\
\downarrow{\scriptstyle (\lambda_C^{-1}+\lambda_C^{-1})\otimes id} \\
(C + C) \otimes A \\
\downarrow{\scriptstyle (q+q)\otimes id} \\
((I + I) + (I + I)) \otimes A \\
\downarrow{\scriptstyle distl} \\
((I + I) \otimes A) + ((I + I) \otimes A) \\
\downarrow{\scriptstyle distl+distl} \\
((I \otimes A) + (I \otimes A)) + ((I \otimes A) + (I \otimes A)) \\
\downarrow{\scriptstyle ((id\otimes g)+(id\otimes h))+((id\otimes f)+(id\otimes l))} \\
((I \otimes B) + (I \otimes B)) + ((I \otimes B) + (I \otimes B)) \\
\downarrow{\scriptstyle distl^{-1}+distl^{-1}} \\
((I + I) \otimes B) + ((I + I) \otimes B) \\
\downarrow{\scriptstyle distl^{-1}} \\
((I + I) + (I + I)) \otimes B \\
\downarrow{\scriptstyle (q^{-1}+q^{-1})\otimes id} \\
(C + C) \otimes B \\
\downarrow{\scriptstyle (\lambda_C+\lambda_C)\otimes id} \\
((I \otimes C) + (I \otimes C)) \otimes B \\
\downarrow{\scriptstyle distl^{-1}\otimes id} \\
((I \oplus I) \otimes C) \otimes B \\
\downarrow{\scriptstyle (q^{-1}\otimes id)\otimes id} \\
(C \otimes C) \otimes B
\end{array}
$$

for $g,h,f,l : A \to B$.