Spring 2021

Untouchable Money and Impossible Clones: Applications of Quantum Picturalism and ZX-Calculus

Shea A. Roccaforte
Bard College

Untouchable Money and Impossible Clones: Applications of Quantum Picturalism and ZX-Calculus

A Senior Project submitted to
The Division of Science, Mathematics, and Computing
of
Bard College

by
Shea Ana Nye Roccaforte

Annandale-on-Hudson, New York
May, 2021

Abstract

Quantum Picturalism allows a new technique for researchers and students alike in the areas of quantum computation and quantum information. This picturalistic method represents fundamental math concepts and quantum theory in a diagrammatic manner. This method is a high-level language that allows for the exploitation of quantum weirdness. Using these techniques, quantum processes and the composition of those processes are highlighted as a structure referred to as process theory. Viewing these processes in a purely diagrammatic language allows for an unambiguous universal language for qubits, and the manipulation of these diagrams is referred to as ZX-calculus. These concepts allow for a hybrid diagram-symbol formalism that allows for research in finite spaces. In this thesis, we focus on the connectivity between processes and study quantum algorithms such as the no-cloning theorem and quantum money to see what can be explained and explored diagrammatically.

Contents

Dedication

To all the young girls with a curiosity of mathematics—may your wonder ignite you.

Acknowledgments

I want to thank all of my mathematics professors that I have had the pleasure of working with over the years: Ethan Bloch, Steve Simon, Stefan Méndez-Diez, John Cullinan, and Lauren Rose. I want to thank Sven Anderson and Kerri-Ann Norton in the computer science department for their support. I especially want to thank my SPROJ advisors—Stefan Méndez-Diez and Paul Cadden-Zimansky—for their immense amount of guidance in completion of this thesis during a global pandemic. Thanks to Paul and Henry Chang for helping me dip my toes into the physics realm through our weekly quantum computation and information theory book club and our BSRI qubit visualization research project. I want to thank Jack Calman, for the endless zooms on physics research, and thanks to Henning Fischel for the accountability zooms for working on our projects. Thanks to all my friends and family for their encouragement to complete this project, but especially thanks to (in no particular order): Sarah Shaffer, Makenna Hulett, Kate Geores, Rosie Levi, Philip Barnet, Mark Roccaforte, Wray Roccaforte, Judy Calman, Lexy Hernandez, Riti Bahl, Autumn Perez, Liam Bach, and Josie Weck. I am so thankful for all of you. Lastly, thank you Daniel for believing in me.

1 Introduction

Coined by Bob Coecke and Aleks Kissinger [1], Quantum Picturalism is a way to use diagrammatic reasoning, graph theory, set theory, and category theory to define a new formalism for quantum processes. Focusing on the connectivity of processes, this abstract tensor notation can be used to exploit quantum algorithms. As they put it in their own words: "By using diagrams we eliminate a huge amount of redundant syntactic garbage in representing mathematical objects, freeing us to concentrate on the important features of the mathematical objects themselves". [1, Ch.1].

This project involves a closer look at this diagrammatic notation to see these algorithms differently. Using their own notation, Coecke and Kissinger created ZX-Calculus as a way to work through these diagrams through deformation and substitution. The classifications of these diagrams each have their own sets of rules, and could be viewed as a strict symmetric monoidal category, or process theory. This allows for a closer look at quantum computation and quantum information as the formalism can be applied to qubits, or quantum bits. Since this formalism allows the representation of these atomic building blocks of quantum computation, we can work to exploit this quantum weirdness through a new, intuitive mathematical representation.

Using this formalism to capture the essential features of a quantum process, this diagrammatic intuition can be used as a high-level language for both mathematicians and physicists. This logic focuses on the interactions of these processes to truly think about theory diagrammatically.

This language allows for many mathematical formalisms—from algebra to quantum mechanics—to be expressed diagrammatically. While focusing on the essential components of the connectivity of these processes, the foundation of the mathematics adjusts to the new formalism.

With the clarity of this language, qubits can be manipulated unambiguously through the process of ZX-Calculus. An important caveat to this, however, is that this picturalism does not work with infinite spaces. This is a result of the operators that Coecke and Kissinger formalised not being able to be bounded between infinite-dimensional Hilbert spaces [1, Ch. 1]. Instead, this model focuses on the change between two times t_1 and t_2 without considering the changes that occur between those times.

This thesis will approach quantum picturalism from a mathematical perspective with quantum references and examples throughout the project. In short, we will be focusing on the interpretation and manipulation of these diagrams. After an understanding of these concepts is built in chapters 3-5, we will focus on the no-cloning theorem (chapter 6), and use that structure to explore the representation of a quantum money application (chapter 7) in this formalism. [1]

[1] Diagrams in this thesis were made using Inkscape 1.0.2, an open source scalable vector graphics editor.

2
Preliminaries

Before understanding the diagrammatic notation, there are some linear algebra and quantum mechanics definitions that should be reviewed in a traditional formalism. Note that the following definitions will be re-represented in the diagrammatic formalism in later sections. This section will also go over some important terminology that will be used heavily throughout this thesis. These definitions, theorems, and proofs follow from [3, Ch. 1] and [2, Ch. 1.2.1, 1.3.3, 2.1.6, and 2.2.8].

2.1 Linear Algebra and Quantum Mechanics

The following quantum mechanics concepts that we will review here in traditional notation will give an overview of qubits and their properties and structure in our quantum money application.

2.1.1 Definitions

Definition 2.1.1. An orthonormal basis is a set of basis vectors that are pairwise orthogonal and normalized.

Definition 2.1.2. A "ket" is a special way of writing a column vector in Dirac Notation:

$$|\psi\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \tag{2.1.1}$$

Where a and b are complex numbers.

Definition 2.1.3. A “bra” is a special way of writing a row vector in Dirac Notation:

$$\langle\psi| = \begin{pmatrix} c & d \end{pmatrix} \tag{2.1.2}$$

Where c and d are the complex conjugates of the elements from $|\psi\rangle$ in definition 2.1.1.

Definition 2.1.4. The inner product is a generalization of the dot product, and results in a scalar. In Dirac Notation, this is represented as:

$$\langle\phi|\psi\rangle \tag{2.1.3}$$

Definition 2.1.5. We define the tensor product of $\boldsymbol{x}$ and $\boldsymbol{y}$ to be the outer product—or $|x\rangle\langle y|$ — defined as:

$$\boldsymbol{x} \otimes \boldsymbol{y} = xy^T \tag{2.1.4}$$

Noting that x is a column vector and y is a row vector from definitions 2.1.1 and 2.1.2.

Example 2.1.6. Let $\boldsymbol{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, and let $\boldsymbol{y} = \begin{pmatrix} 3 & 4 \end{pmatrix}$. Then:

$$\boldsymbol{x} \otimes \boldsymbol{y} = \begin{pmatrix} 1\cdot 3 & 1\cdot 4 \\ 2\cdot 3 & 2\cdot 4 \end{pmatrix} = \begin{pmatrix} 3 & 4 \\ 6 & 8 \end{pmatrix}. \tag{2.1.5}$$

Definition 2.1.7. An operator Ω is Hermitian if $\Omega^\dagger = \Omega$.

Note that $\dagger$ refers to taking the transposition and complex conjugate of a vector.

Definition 2.1.8. An operator U is unitary if:

$$UU^\dagger = I \tag{2.1.6}$$

Note that this means U and $U^\dagger$ are inverses of each other. Thus, it follows that

$$U^\dagger U = I \tag{2.1.7}$$

Theorem 2.1.9. *Unitary operators preserve the inner product between the vectors they act on.*

Proof. Let $|V_1'\rangle = U|V_1\rangle$ and let $|V_2'\rangle = U|V_2\rangle$. Then, by taking the inner product of these vectors, we obtain:

$$\langle V_2'|V_1'\rangle = \langle UV_2|UV_1\rangle \tag{2.1.8}$$

By definition 2.1.7 and our assumptions, this can be simplified to:

$$\langle V_2| U^\dagger U |V_1\rangle = \langle V_2|V_1\rangle \tag{2.1.9}$$

□

2.1.2 Qubits, States, and Probabilities

We can think of a qubit, or a quantum bit, in terms of a classical bit. In short. a classical bit is a two-state system that has a state definitively of 0 or 1. A qubit can also have those states, represented as $|0\rangle$ and $|1\rangle$. However, a qubit has a special property: it can have many, many more states than a classical bit. This is done by taking linear combinations of states, and these are called superpositions:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.1.10}$$

Although it is not important for our purposes, it is worth noting that α and β are complex numbers, leading us to the following:

Definition 2.1.10. The state of a qubit is a unit vector in a two-dimensional complex vector space.

As we will see in chapter 6, we should mention that $|0\rangle$ and $|1\rangle$ are known as the computational basis states.

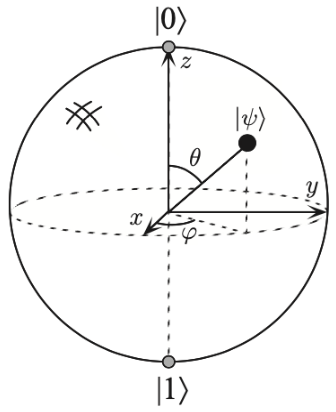


Figure 2.1 "Bloch Sphere representation of a qubit."[2, Ch. 1.2]

Note that Figure 2.2.1 is the standard way of visualizing a single qubit. For our purposes, it is only important to note that a superposed state, such as $|\psi\rangle$ could be any point on the sphere. In addition, the computational basis chosen for this measurement is $|0\rangle$ and $|1\rangle$.

When we take a measurement of a qubit, we are extracting quantum information in regards to what state it is in from α and β, as shown later in equations 2.1.14 and 2.1.15. The measurement will only yield a value of $|0\rangle$ or $|1\rangle$, and so we must recognize that to "measure" a qubit, we must collapse it from its superposition, which we can recall from earlier is a linear combination of these states. In other words, the measurements corresponds to the projections of these vectors.

Now, consider two qubits. This implies that we have four computational basis states, which we denote as:

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle \tag{2.1.11}$$

Definition 2.1.11. The amplitude of a quantum state of two qubits is represented as α, a linear combination of basis states. This yields a state vector of the two qubits in the following form:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \tag{2.1.12}$$

Definition 2.1.12. Consider the two qubit state:

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{2.1.13}$$

Since $|\psi\rangle \neq |a\rangle |b\rangle$ for any single qubit states $|a\rangle$ and $|b\rangle$, then $|\psi\rangle$ is defined to be an entangled state.

More generally, given any basis states $|a\rangle$ and $|b\rangle$ for a qubit, we can represent an arbitrary state, say $|\psi\rangle$, as a linear combination of these states in the now generalized form of equation 2.1.10:

$$|\psi\rangle = \alpha |a\rangle + \beta |b\rangle \tag{2.1.14}$$

We will see in chapter 6 that there are other orthonormal states besides $|0\rangle$ and $|1\rangle$ that can be chosen as the basis states for a qubit, but it is important to understand how a measurement on any basis, say $|a\rangle$ and $|b\rangle$, will yield a probability of quantum information about the qubit itself.

Definition 2.1.13. A quantum system is normalized when every state sums to one.

Given that the qubit's states are normalized—

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2.1.15}$$

where α and β follow from equation 2.2.11, a measurement is possible with respect to the $|a\rangle, |b\rangle$ basis. This measurement yields the following results: the result is a with a probability of $|\alpha|^2$, and the result is b with a probability of $|\beta|^2$.

3
String Diagrams

We will now introduce the new formalism by defining the diagrams that we use for these algorithms. The first subset of this process theory that we will consider is a group referred to as *string diagrams*. In short, string diagrams highlight non-separability of qubits, the no-cloning theorem, and unitarity. This chapter will go over the fundamental building blocks and the properties of these diagrams given our knowledge of these topics in a traditional formalism. We will also use a bit of hybrid formalism for clarity, which will be expanded upon in later chapters. This chapter will give us a structure that we will be able to use for later applications.

3.1 Diagrams, System-Types, and Processes

These categories are created to break down the most important elements of the string diagrams. String diagrams are the subset of diagrams that we will use for the no-cloning theorem and for our quantum money application.

Definition 3.1.1. A diagram is processes that are wired together.

This definition leaves us with two essential building blocks—the wires and the boxes.

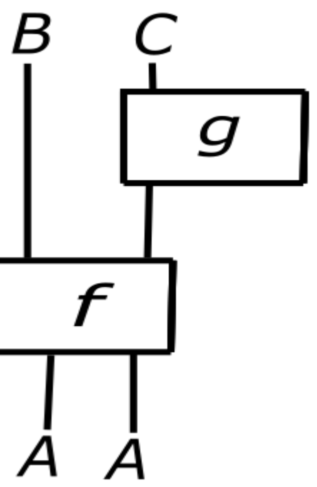


Figure 3.1.1

Definition 3.1.2. System-Types represent the labels on the wires themselves in a diagram.

In Figure 3.1.1, the system-types would be A, B, and C. The wires are connecting the processes by demonstrating the connectivity between inputs and outputs. There is a restriction, however, in that these types have to match each other as they can be thought of as data-types. The boxes in Figure 3.1.1 are f and g respectively. These boxes represent the processes of the diagram, or the fundamental interpretation of the system.

It is also important to define how these diagrams are read. In this diagrammatic notation, time flows from bottom to top. In the example of Figure 3.1.1, process f would occur before process g. In other words, this diagram could be thought of as a digraph, as it is only read in one direction (see figure 3.1.2).

Now, we have defined two fundamental components of the string diagram—the system-types and the processes. The understanding of wiring these processes together is the fundamental process theory for string diagrams, which can be simplified into two simple operations, which will be discussed later in section 3.2.

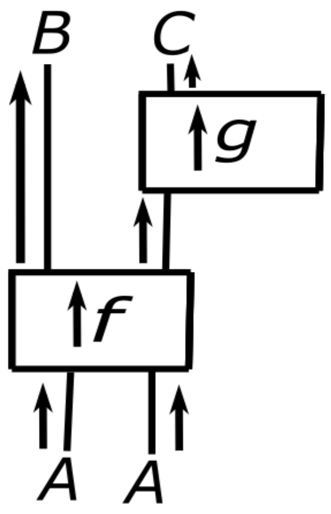


Figure 3.1.2

3.1.1 Spiders

Spiders are a way to allow for the generalization of wires. [1, Ch. 2.2] In short, they allow a way to interpret the uniqueness of data and information between inputs and outputs. These spiders themselves are the representation of interaction between objects in these diagrams. When considering String Diagrams, we use the formalism to have spiders represent Hilbert Spaces. We also state that different spiders are a way of representing quantum entanglement. We will see these spiders come into play later in chapter 6.

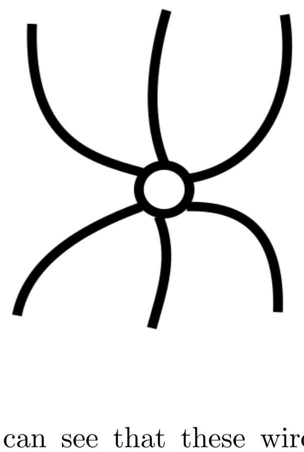
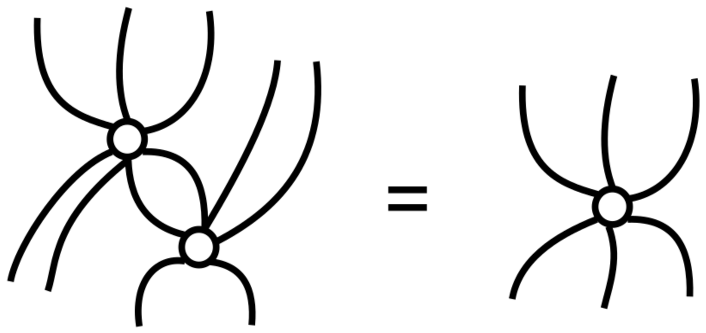


Figure 3.1.3 [1, Ch. 8.1]

In figure 3.1.3, we can see that these wires represent classical information, and they are compounded by this spider. [1, Ch. 8.1]

A spider can be thought of as a bounded operator, and represents the merging and splitting of classical data when information is copied or deleted.

Definition 3.1.3. The behavior of spiders follow from the "fusion" rule—



Example 3.1.4. Figure 3.1.4

[1, Ch. 8.1]

Note that spiders contain the yanking equations (which will be defined in chapter 5). Spiders embody the idea that only connectivity matters, as we focus on the process itself.

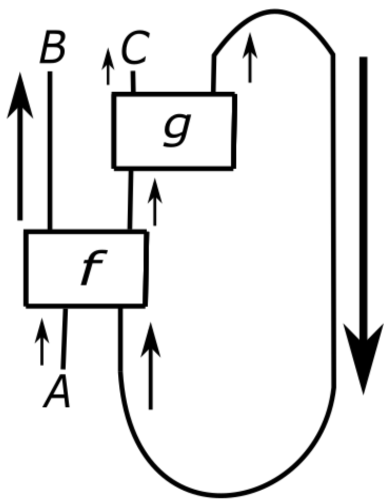
3.1.2 Circuit Diagrams

Yet another category of string diagrams that is worth mentioning are circuit diagrams.

Definition 3.1.5. A circuit diagram is any diagram that can be represented non-uniquely in terms of $\otimes$ and $\circ$ if it does not contain directed cycles.

Thus, a circuit diagram is equivalent to a directed acyclic graph. These operations will be explained in the next section. Circuit diagrams will be our main focus of string diagrams.

Revisiting Figure 3.1.1, this is an example of a circuit diagram. This diagram would not be a circuit diagram if there was a wire connecting the output of g back to f, as demonstrated in the following example:



Example 3.1.6. Figure 3.1.6

Theorem 3.1.7. *The following two statements are equivalent: A diagram is a circuit, and a diagram contains no directed cycles.*

The proof of theorem 3.1.6 follows from [1, Ch. 3.2]. Note that some terminology, including operators, is yet to be defined (see section 3.2 below).

Proof. A diagram is a circuit if and only if it contains no directed cycles. First, consider the case when a diagram contains no directed cycles. Since the diagram is directed acyclic, we can break down the diagram into layers, as demonstrated in figure 3.1.6, where l_1 and l_2 are defined. For an arbitrary diagram, consider layers $l_1, \cdots, l_n$. Note that these layers contain all wires as outputs or inputs respectively, and include any identities or swaps (see 3.2.3). So, if we consider layer l_i, everything contained in that layer with only connect to boxes in layer l_{i+1}. By definition of our operators, $\otimes$ will combine all boxes together, and $\circ$ will collapse all layers together. Thus, by definition, we have a circuit diagram. On the other hand, consider an arbitrary circuit diagram. This diagram can then be represented non-uniquely in terms of $\otimes$ and $\circ$. We know that any diagram of this form can be decomposed into these layers, which are separated by time. Thus, there cannot exist any directed cycles. $\square$

In short, the important takeaway is that the only difference between string and circuit diagrams is the existence of directed cycles.

3.1.3 Reading a Diagram

This provides a nice segue into how these diagrams are read, particularly circuit diagrams. Since there are no directed cycles, this allows for a "layering" of time, which we will see in 3.2.2 is represented as sequential compositions. We can also begin experimenting with some hybrid-symbol diagrams, which will show up again in chapter 4.

Consider figure 3.1.1. We can break this down into further "layers" to understand the operations and processes that are going on. This process is called **foliation**. We can redraw our diagram to obtain:

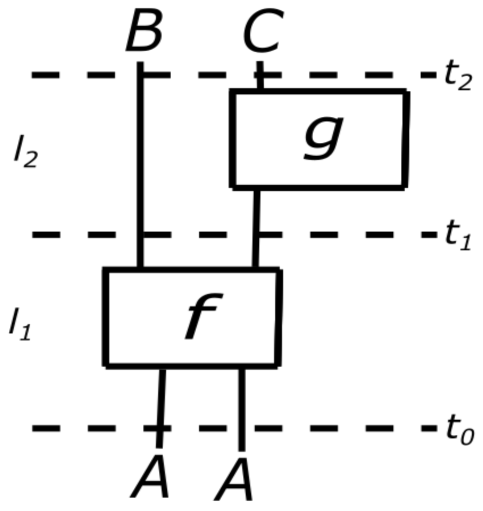


Figure 3.1.7

We will return to further interpretations of this diagram in chapter 4.

3.2 Compositions

String diagrams are defined in such a way that the entire structure can be created solely on two operations: the parallel composition and the sequential composition. We will define these operations and then compute a few small examples.

3.2.1 Parallel Compositions

Parallel Compositions [1, Ch. 3.2.1] are represented by the tensor operator, $\otimes$. If we had two processes, f and g, we would say $f \otimes g$ if f and g were happening simultaneously—or in parallel. This is represented by the two processes being side by side, as shown below.

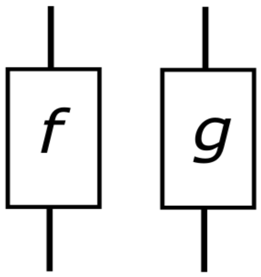


Figure 3.2.1

In other words, this implies that these operations, for instance f and g in figure 3.2.1, are happening independently from each other, as there is no connectivity.

Example 3.2.1.

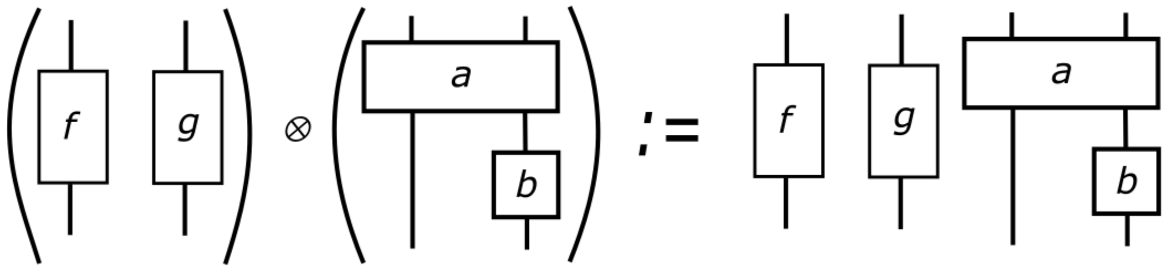


Figure 3.2.2

The parallel composition holds special properties as follows:

The operation is associative:

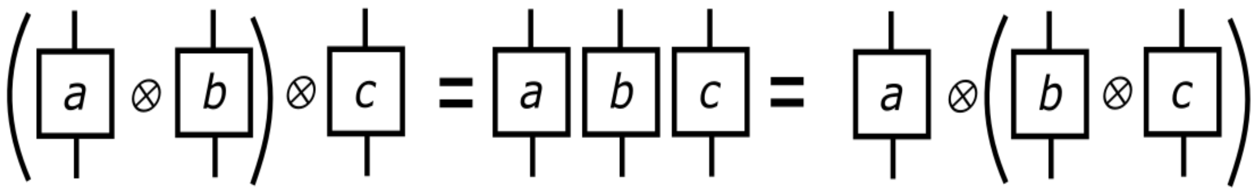


Figure 3.2.3

The operation contains a unit, or an empty diagram.

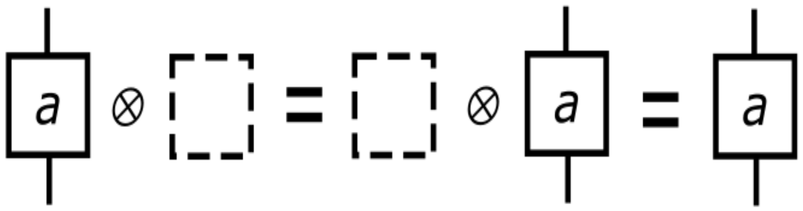


Figure 3.2.4

In other words, if we considered $f \otimes g$, we know that this implies associativity and composition with the empty diagram returning the same input.

In addition to our boxes/processes, parallel composition also applies to the system-types, or the wires of the diagrams.

Definition 3.2.2. A joint-system type is formed from two system-types, say A and B, such that:

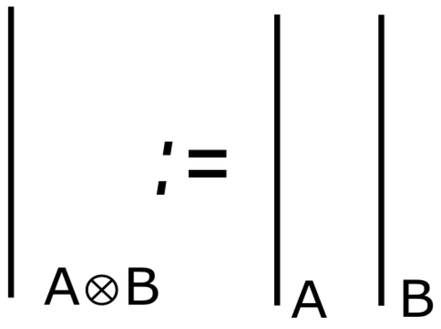
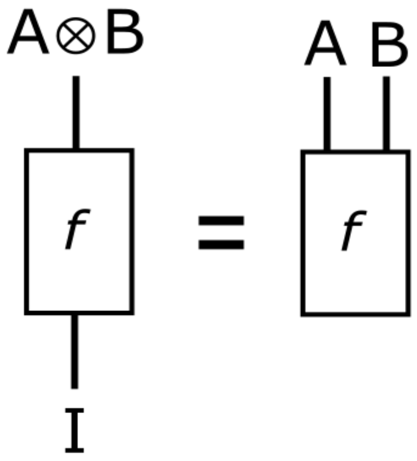


Figure 3.2.5

Lastly, there exists an empty system type denoted with I; this represents no inputs, outputs or both. An example of this follows below:



Example 3.2.3. Figure 3.2.6

3.2.2 Sequential Compositions

Sequential Compositions are represented by a composition of functions, or $\circ$. If we look at two other processes, say g and h, we would say $g \circ h$ if g and h were happening sequentially, or one after another. This is represented by the two processes being stacked on each other, as time flows from the bottom to the top (recall figure 3.1.2).

Example 3.2.4.

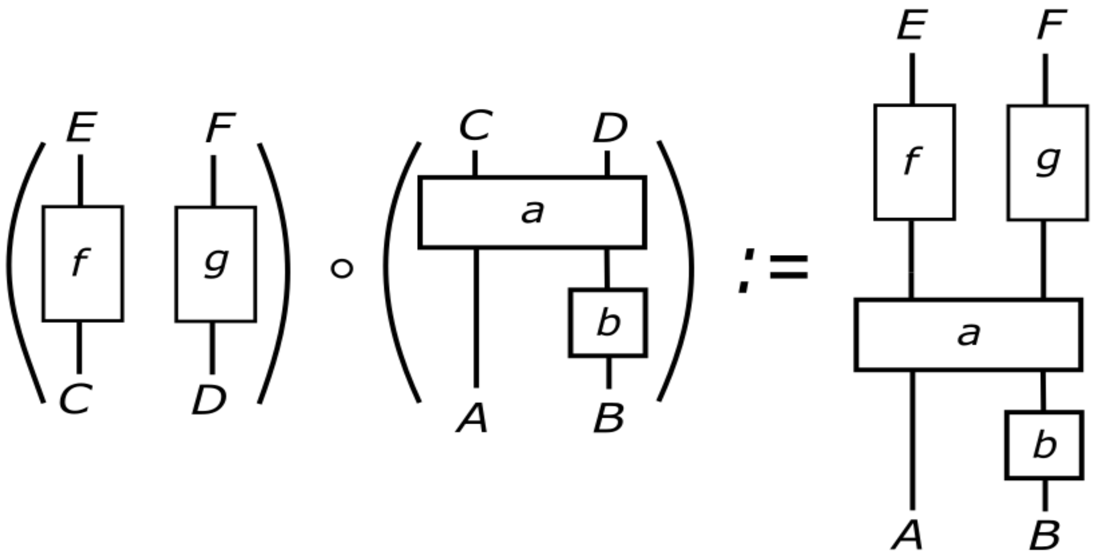


Figure 3.2.7

The sequential composition holds special properties as follows:

The operation is associative.

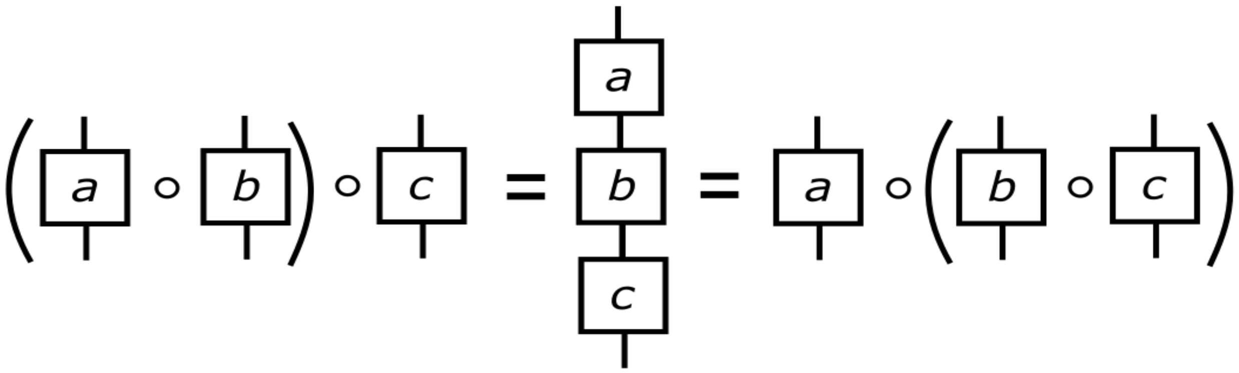


Figure 3.2.8

The operation contains a unit, or an empty diagram.

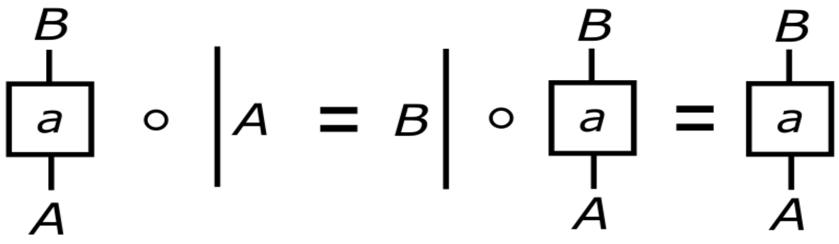


Figure 3.2.9

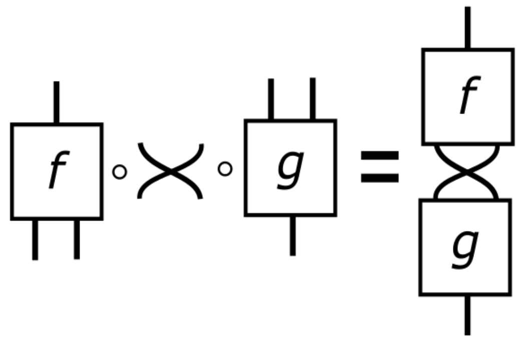
In other words, if we considered $g \circ h$, we know that this implies associativity and composition with the empty diagram returning the same input.

3.2.3 Temporal Flow: Identities and Swaps

Definition 3.2.5. An identity is represented as a wire.

Example 3.2.6. Considering l_2 in Figure 3.1.6, we can note that the wire on the left has no process acting upon it, thus resulting in the identity. We will see a more symbolic representation of this example in section 4.2.

Definition 3.2.7. Since sequential compositions connect outputs to inputs in order, we define a swap to vary that order:



Example 3.2.8. To visualize definition 3.2.7, consider a hybrid diagram with additional information for understanding. If we think of a simple algebraic equation, such as:

$$2x - 2y = (x - y) \cdot 2 \tag{3.2.1}$$

We can represent this with labelled processes for a clear understanding as:

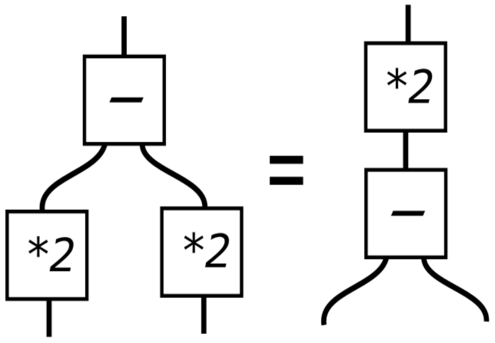


Figure 3.2.10

Note that when we add a swap, we have the following:

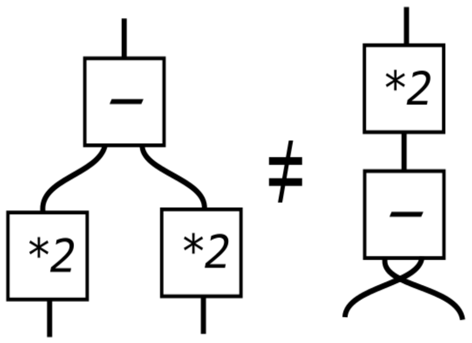


Figure 3.2.11

Because we note that this is:

$$2x - 2y \neq (y - x) \cdot 2 \tag{3.2.2}$$

Showcasing the importance of input and output order.

3.2.4 Combinations of Compositions

You might find yourself asking, how are these diagrams different than traditional algebraic notation? Following from [1, Ch. 3.2.4], we will see how these diagrams use the combination of these operators to do some useful tricks.

First, it is important to note that diagrammatically, associativity and unitality of parallel composition are handled automatically by the nature of the diagram itself. By Figure 3.2.3, we see that parallel associativity is implied by processes being drawn side by side, and unitality follows because the composition of a process with empty space does nothing.

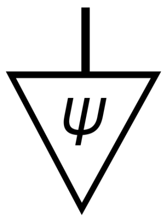
Similarly, these two properties follow for sequential composition as well, as identities are simply wires with no processes and associativity follows from removing brackets.

4
Introduction to Diagrammatic Quantum Mechanics

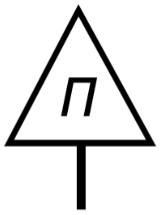
4.1 States, Effects, and Numbers

Coecke and Kissinger created formalities for major concepts that will later allow for a look at quantum algorithms and quantum information, such as special processes called states, effects, and numbers.

Definition 4.1.1. States are processes that do not have any inputs or preparation. We don't care about what happened before, we only care about what state the system is currently in. We represent this as a triangle, which has no inputs.



Definition 4.1.2. Effects are processes that do not have any outputs. We don't care about what happens afterwards, we only care about what effect the system ends up in. This is helpful for verification and checks to see if a state ends up in a certain effect. We also represent this as a triangle, but with no outputs.



We can also compose a state and effect to get a new type of process altogether, which is simply a number.

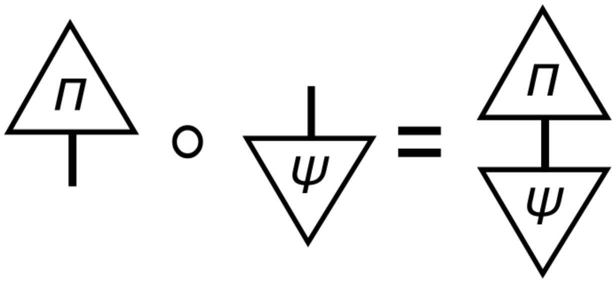
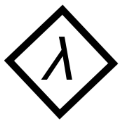


Figure 4.1.1

Definition 4.1.3. Numbers are processes that do not have any inputs or outputs. This is typically noted as λ, (with or without a process box) and is thought of as an eigenvalue:



4.1.1 A Note on Eigenvalues

We must note that the numbers shown in definition 4.1.3 are a very special type of process, as they have both no inputs and outputs. We can further note that these numbers are simply a set of objects that can be multiplied by the $\otimes$ operation of the $\circ$ operation. Given two numbers λ and μ, it follows that:

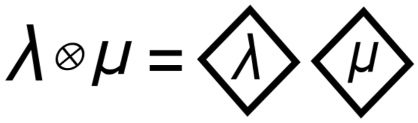
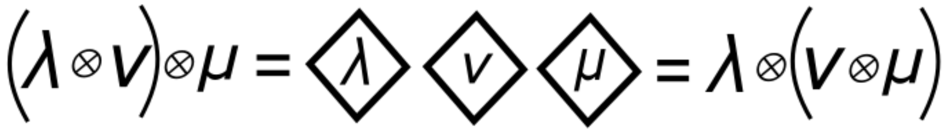


Figure 4.1.2

We also know that our diagrams are associative, so it follows that our numerical multiplication is as well:



$(\lambda \otimes \nu) \otimes \mu =$
λ
ν
μ
$= \lambda \otimes (\nu \otimes \mu)$

Figure 4.1.3

In addition, these processes have a unit, which is given by the empty diagram:

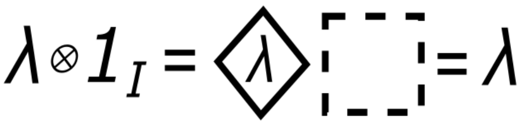


Figure 4.1.4

Lastly, due to the lack of inputs and outputs of these processes, they can jump around to any part of a given diagram, thus making them commutative.

Definition 4.1.4. Since these processes are a set of objects that can be multiplied and are associative, commutative, and has a unit (namely 1), they are a commutative monoid. See [1, Ch. 3.6.1, 3.6.2] for more on Monoidal Category theory and Abstract Tensor Notation.

4.1.2 Dirac Notation Application

These states and effects can also be thought of as Kets and Bras, and the composition of both (Figure 4.1.1) as Bra-Kets. Dirac notation itself actually represents a 1-D subset of diagrammatic notation.

Example 4.1.5. The figure in definition 4.1.1 could be written as $|\psi\rangle$ when rotated 90 degrees counterclockwise. Similarly, the figure in definition 4.1.2 could be written as $\langle\pi|$ when rotated 90 degrees. Lastly, the composed process in Figure 4.1.1 represents a Bra-Ket and can be written as $\langle\pi|\psi\rangle$ when rotated 90 degrees counterclockwise.

Putting this together with our number definitions from section 4.1.1, we can form expressions like:

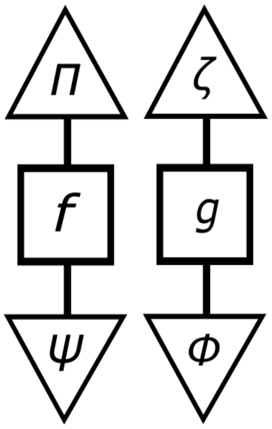


Figure 4.1.5

which is simply:

$$\langle\pi|\, f\, |\psi\rangle\, \langle\zeta|\, g\, |\phi\rangle \tag{4.1.1}$$

and:

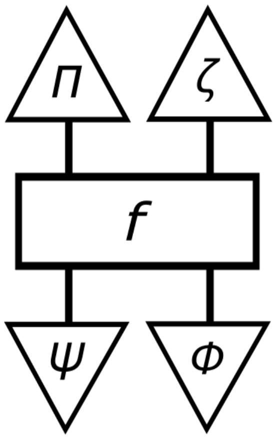


Figure 4.1.6

becomes:

$$\langle\pi\zeta|\, f\, |\psi\phi\rangle \tag{4.1.2}$$

because writing multiple states inside of a ket or bra indicates parallel composition.

4.1.3 Generalized Born Rule

In chapter 7, we will see the importance of measurements and probabilities, and so it is important to see how we find this probability diagrammatically.

Definition 4.1.6. When a state meets and effect, we obtain a number, and that number yields the probability that the effect happens, given the system is in a particular state [1, Ch. 3.4]:

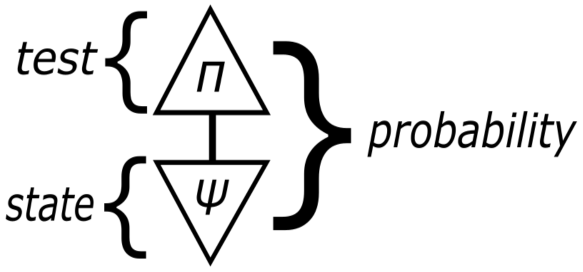


Figure 4.1.7

We can also note that this probability will be a real number, as the quantum map ensures that the number—which is a complex number (or amplitude)—is multiplied by its conjugate to obtain the probability.

4.2 Separability and Non-Separability

So far, most of the diagrammatic language that we have encountered applies to classical systems, and we have yet to introduce the quantum aspects of these systems. So far, these diagrams are separable, meaning they can be broken down into disconnected segments. Now, we will be looking at non-separable features of diagrams that allow for states and effects to not be disconnected as easily. We will introduce new features such as cups, caps and adjoints (vertical reflection of a diagram). These tools will allow for many of the tools mentioned in chapter 2, such as unitary processes and probabilities from measurements (inner products).

Due to these additions, these sets of processes will be string diagrams. These differences will become more evident in chapter 6, when we see something that is impossible in process theory with string diagrams, although it would be expected in a classical system.

4.2.1 Cups and Caps

In short, cups and caps are special states and effects that are used to convert processes to states and back with certain operations. These two processes will be defined to be inverses of each other, or yield **process-state duality** if and only if: [1, Ch. 4.1.2]

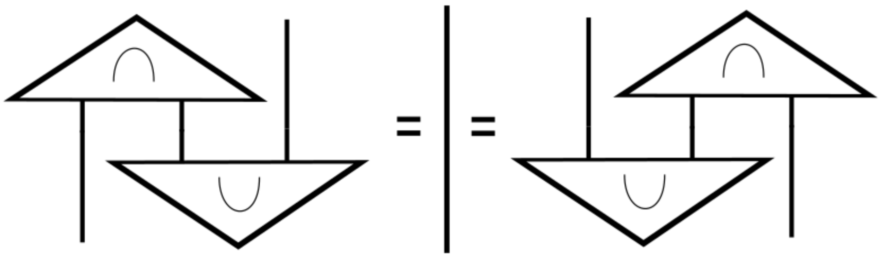


Figure 4.2.1

Now, the following notation will later help us intuitively understand section 4.2.2. We can define these special states and effects to be bent wires, or cups and caps:

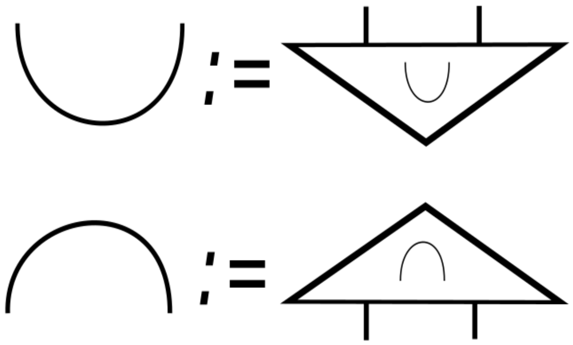


Figure 4.2.2

and Figure 4.2.1 then becomes:

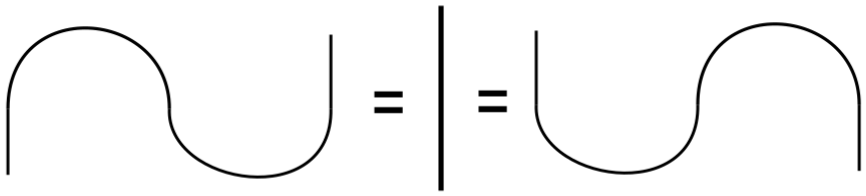


Figure 4.2.3

Definition 4.2.1. Figure 4.2.3 visually represents what are known as the yanking equations, which states that these bent wires can be "yanked" into a straight wire.

This also follows for crossings—or swaps—and loops. Thus, we have these three equations:

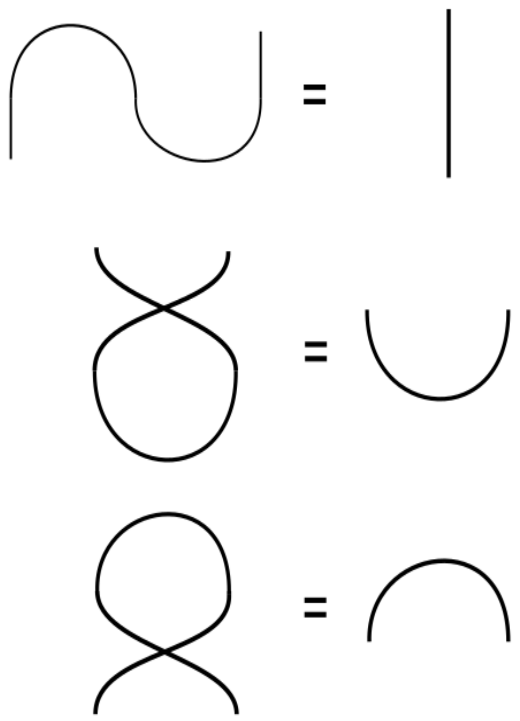


Figure 4.2.4

We will see how these cups and caps are read and used in the next section.

5
Process Theory

Here we will begin unpacking some of the tools of this diagrammatic process theory, and we will define the operations that will help us better understand quantum concepts in this formalism.

5.1 ZX-Calculus Introduction

As introduced earlier, we will now take a deeper look into the manipulation of these diagrams, or the act of ZX-Calculus itself. As we will see, for our diagrams and purposes, there are two methods of manipulation: diagram manipulation and deformation. After those are defined and demonstrated, we will see more on how these diagrams are read, and we will be using more foliation methods throughout the formulas and diagrams.

5.1.1 Diagram Substitution and Deformation

Of the two calculations, substitution is the easier one to understand. The idea is straightforward; a sub-piece of a diagram can replace another piece if those processes are equal. This is very similar to traditional notation, but with one important reminder:

Since connectivity in diagrams is critically important, inputs and outputs must be tracked when manipulating the diagrams. In short, when performing a diagram substitution, every wire must still connect to the same processes outside of the manipulation.

The situation is similar with decomposition, or deformation, as everything must still match up. This idea is all about simplification, and will be clear in the proceeding two examples.

5.2 Hybrid Diagram and Symbol Formalism

To gain an understanding on how these diagrams are read, we will do a hybrid-diagram and symbol formalism to better understand the parallel and sequential compositions from the previous chapter.

Example 5.2.1. Recall figure 3.1.6. Using foliation, we divided this figure into layers to see what was happening in each segment. To further clarify what actions are happening in parallel, we will add another segmented dashed line in l_2:

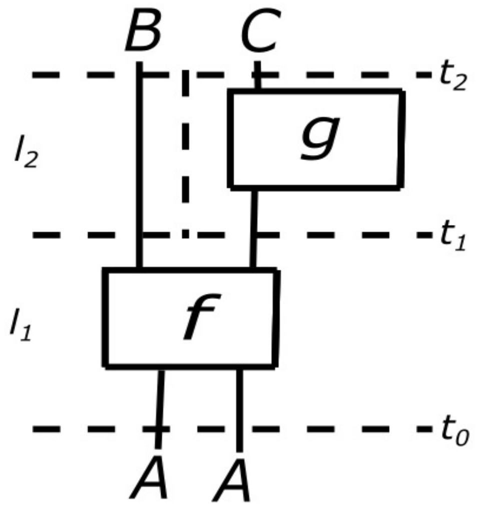


Figure 5.2.1

Working our way from the bottom to the top—following time—we can translate this to a hybrid-symbol formalism as follows:

$$(1_B \otimes g) \circ (f) \tag{5.2.1}$$

Using the same logic as before, we can do another translation of a diagram, which also happens to be the definition of a diagrammatic transposition (definition 5.2.2):

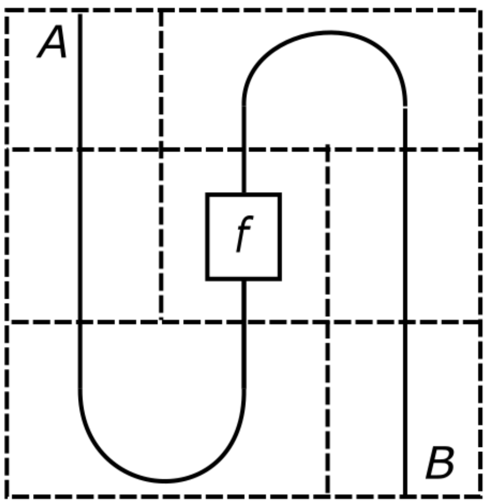


Figure 5.2.2

becomes:

$$(1_A \otimes \cap_B) \circ (1_A \otimes f \otimes 1_B) \circ (\cup_A \otimes 1_B) \tag{5.2.2}$$

Definition 5.2.2. A transposition on a process is shown as:

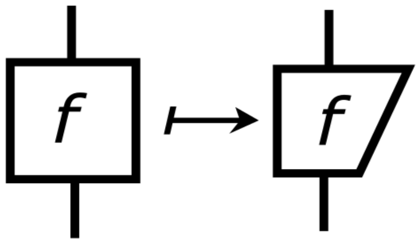


Figure 5.2.3

This definition makes the process asymmetrical, and thus will be rotated when following the yanking equations from the last chapter:

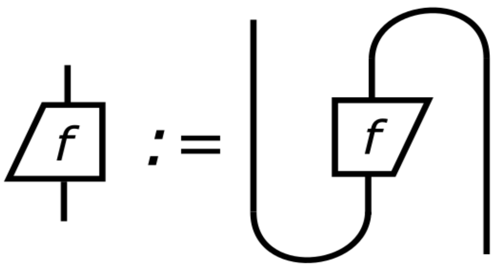


Figure 5.2.4

The same process also works for states and effects:

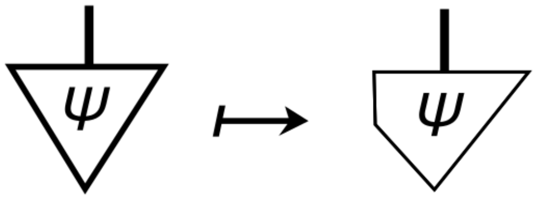


Figure 5.2.5

We will see how these processes and diagrammatic readings will work their way into our algorithms in chapters 6 and 7. Similar to other branches and formalisms of mathematics, everything builds to make a nice diagrammatic proof.

6
No-Cloning Theorem

The no-cloning theorem was discovered by W.K. Wootters and W.H. Zurek [5] in the early 1980's and has helped researchers understand more details about how a quantum cloning device might operate [2].

6.1 Background

The no-cloning theorem is used as a tool in several quantum algorithms, and we will see a direct application to our quantum money example in chapter 6. Using our definitions and review from chapter 2, we will first go over a tradition approach to the no-cloning theorem before completely restructuring it in the diagrammatic formalism. We will see that everything that was defined in chapter 2 will be approached diagrammatically, as shown in section 6.5. We will then apply this result to our quantum money application.

6.2 Proof

Theorem 6.2.1. *The no cloning theorem states that there is no single unitary operation U such that a state $|\psi\rangle$ (pure and unknown) in system A can be "copied" into system B while maintaining $|\psi\rangle$ in A.*

Before jumping into the diagrammatic reasoning behind this theorem, we'll start with the traditional mathematical proof, which is as follows:

Proof. Consider a unitary operator U that can copy state $|\psi\rangle$. Thus, acting U on these states would result in $U(|\psi\rangle \otimes |x\rangle) = |\psi\rangle \otimes |\psi\rangle$, where $|x\rangle$ is the initial state in system B. Taking the inner product, we obtain $\langle x| \langle\psi_1| U^\dagger U |\psi_2\rangle |x\rangle = \langle\psi_1| \langle\psi_1|\psi_2\rangle |\psi_2\rangle$. By definition, we know that $U^\dagger U = 1$. We also know that $\langle x|x\rangle = 1$, so this equation becomes $\langle\psi_1|\psi_2\rangle = (\langle\psi_1|\psi_2\rangle)^2$. Algebraically, these states are either 0 or 1. This implies that either $|\psi_1\rangle = |\psi_2\rangle$ or that $|\psi_1\rangle$ and $|\psi_2\rangle$ are orthogonal. Thus, a general cloning operator does not exist, as U can only be used to copy states that are orthogonal. $\square$

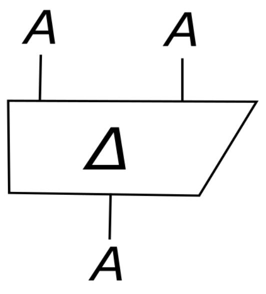
6.3 Algebraic vs Diagrammatic Reasoning

The section above demonstrated the no cloning theorem in a mathematically traditional way. We will see in the conversion process to the diagrammatic language how the no cloning theorem arises in a simpler context that some may find easier to understand.

6.4 Cloning Processes

To understand the no cloning theorem diagrammatically, there are a few attributes of the proof that need to be defined. The fundamental component of this proof relies on a cloning process. We will be following this proof from Coecke and Kissinger (section/chapter)

Definition 6.4.1. A cloning process takes any state as an input, and returns two copies.



In figure 5.3, note that system-type A could be anything, but for this process, A is actually referencing a state ψ.

We can also note that since these states are identical by definition, swaps do not effect the process.

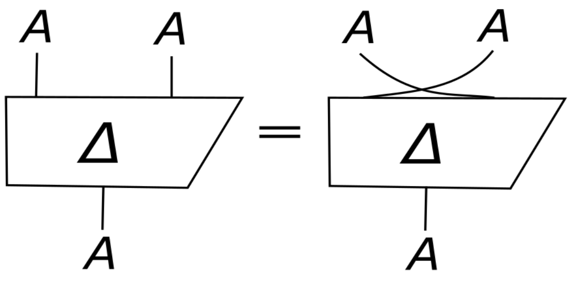
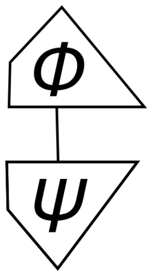


Figure 6.4.1

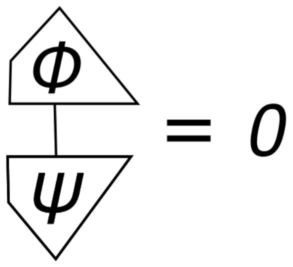
Throughout the no-cloning theorem, we will show that any process theory that involves string diagrams yields it impossible to have a cloning process.

6.5 Definitions

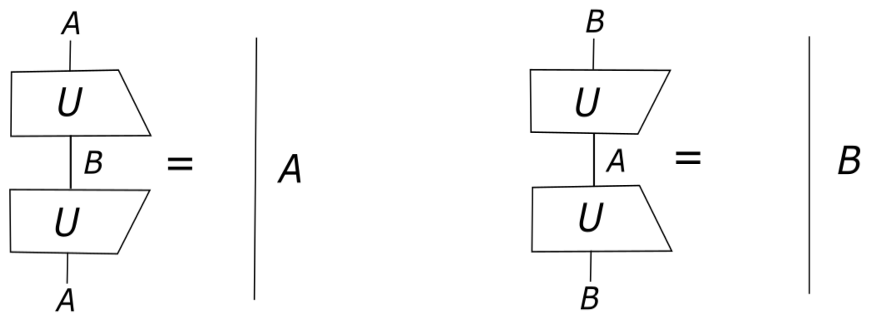
Definition 6.5.1. If we have two states of the same system type, the inner product is defined to be:



Definition 6.5.2. These states of the same type are orthogonal when:

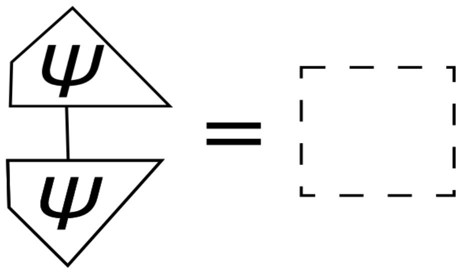


Definition 6.5.3. We say a process U is Unitary if:

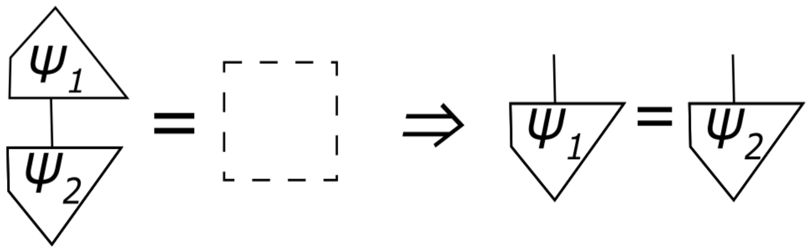


Note that a Unitary process preserves the inner product.

Definition 6.5.4. We say that a state ψ is normalised when:



Definition 6.5.5. When we have two normalized states, ψ_1 and ψ_2 it follows:



6.6 Diagrammatic Proof

We now have the definitions to complete the diagrammatic proof.

Proof. Consider a cloning operator with two normalised states, ψ and ϕ. We will show that these states must either be equal or orthogonal. We will define this cloning operator to be a one input, two output, as follows:

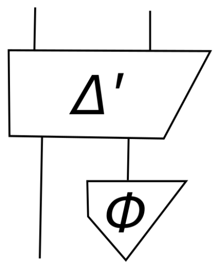


Figure 6.6.1

Then, we use this operator as follows:

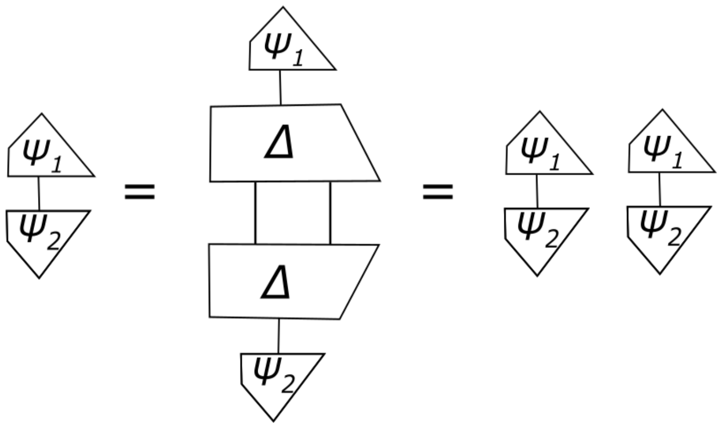


Figure 6.6.2

Simplifying, we have:

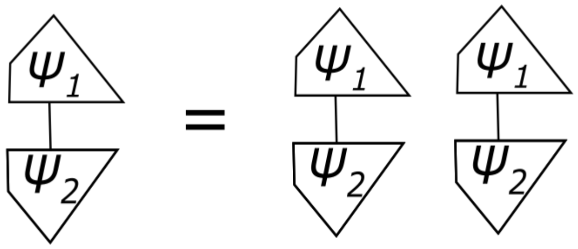


Figure 6.6.3

Then, we consider two cases. In the first, assume:

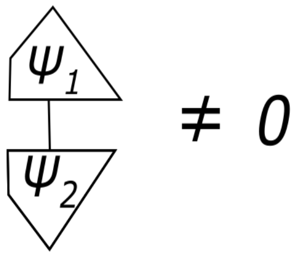


Figure 6.6.4

Then, this number can be cancelled on both sides to obtain:

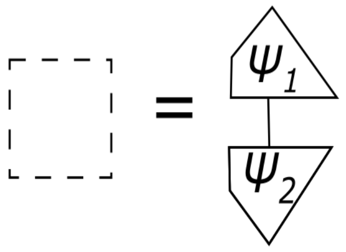


Figure 6.6.5

In the second case, assume:

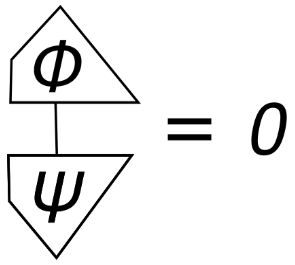


Figure 6.6.6

This means that in this case, the states are orthogonal.

Thus, the states must be 0 or 1: in other words, orthogonal or equal. □

7
Quantum Money

7.1 Introduction

The general idea of the traditional mathematical approach to the quantum money algorithm is to show the indistinguishability of non-orthogonal states. If there was distiguishability between arbitrary states, that would imply communication that would be faster than light [2]. We will find that these states contain hidden information that is not accessible to measurement. In other words, we will show that it is possible to make money that is impossible to counterfeit using quantum systems.

7.2 Algebraic Understanding and Motivation

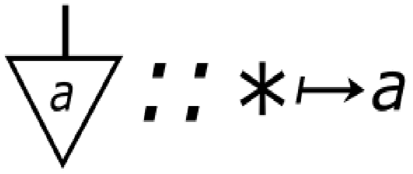
The motivation for this section is simple; we will show that quantum systems allow for the creation of money that is impossible to counterfeit. We will show this based on what we have learned with the no-cloning theorem and what we know about indistinguishability, following [2, Ch. 1] and [4]. Two systems, say Alice and Bob, will be trying to transmit information. Since our systems will share an entangled pair of qubits, there will be 2 bases with a total of four states. We will conclude that if Bob had a device to distinguish these states, say $|0\rangle, |1\rangle, |+\rangle, |-\rangle$, then Bob would be able to tell which basis Alice was using, which would then imply that information was received instantaneously, as soon as Alice made the measurement. Thus, we can conclude from the no-cloning theorem that this would be impossible, due to the indistinguishability of non-orthogonal quantum states.

7.3 Definitions

Definition 7.3.1. [1, Ch. 3.3.1] Bit strings can be represented as an n-fold Cartesian product.

$$\mathbb{B} = \{0,1\} \implies \mathbb{B} \times \mathbb{B} \times \cdots \mathbb{B} \tag{7.3.1}$$

Definition 7.3.2. [1, Ch. 3.4] States correspond to elements of a set. We can think of a state as a function from * (no wire, or single element set) into another set A. This function will yield the image of *, or a single element $a \in A$:



Note that for all $a \in A$, there exists a unique function mapping * to a, thus elements of A and these functions from $\{*\}$ to A are the same. Thus, this function is simply a.

For this application, we will let A from definition 7.3.2 be defined as $\mathbb{B}$, yielding exactly two states from definition 7.3.1:

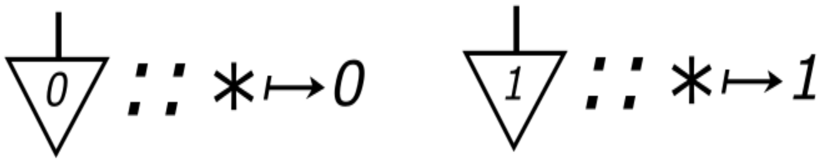
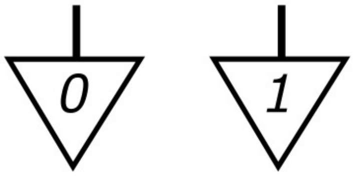


Figure 7.3.1

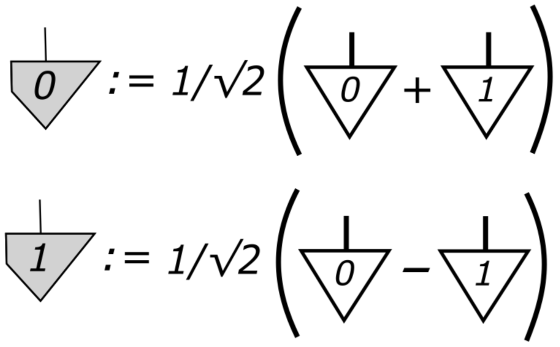
Example 7.3.3. Given a state ψ, a $|0\rangle$ measurement yields 0 with a probability of 1. However, a $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ measurement yields 0 with a probability of $\frac{1}{2}$ and 1 with a probability of $\frac{1}{2}$.

We will have two pairs of orthonormal basis states, which we will label as follows:

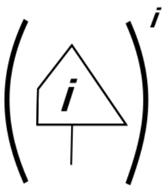
Definition 7.3.4. The computational basis, or the Z-basis is defined as follows: $\{|0\rangle, |1\rangle\}$



Definition 7.3.5. The X-basis is defined as follows: $\{|+\rangle, |-\rangle\}$, or as $\{\frac{1}{\sqrt{2}(a+b)}, \frac{1}{\sqrt{2}(a-b)}\}$, where a and b refer to the computational basis states $|0\rangle$ and $|1\rangle$ respectively. Note the different shading of these states in the diagram below, as it will be important for distinguishability.



Definition 7.3.6. We show a measurement as:



Example 7.3.7. A bit string is represented as follows:

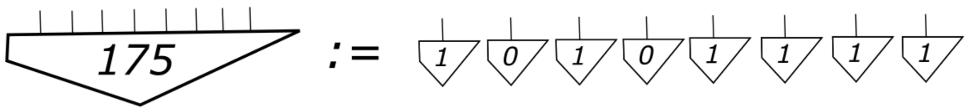


Figure 7.3.2

7.4 Diagrammatic Understanding

Consider a bank note that has a sequence of 10 bits, or isolated systems, which we will call S_i, where $i = 1, 2, \cdots 10$.

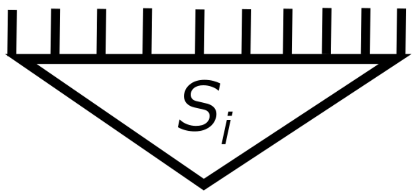


Figure 7.4.1

Next, we will create two random binary sequences of length 10, called m and n.

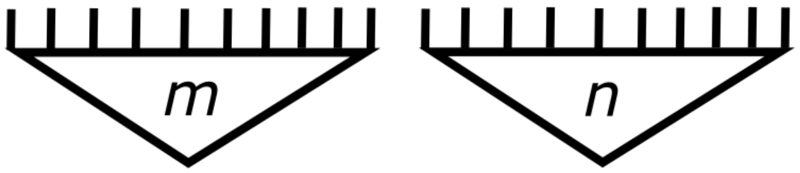


Figure 7.4.2

By definition 6.5.3, each 2-state output represents a 0 or a 1 in these sequences.

Next, we will place our 2 state system in one of the four states:

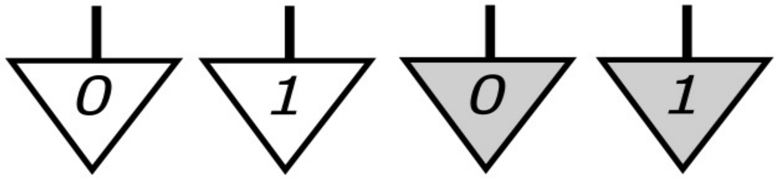


Figure 7.4.3

The check arises when we see if each isolated system is still in the initial state or if it has switch to it's orthogonal state. Since n cannot be recovered, because m is unknown, we cannot conclude what measurements to make on S_i. We know that whatever measurement we make on S_i, we will never know information about both bases. For example, if a measurement is made in the $\{|+\rangle, |-\rangle\}$ basis to distinguish the two states, that measurement will not yield any information about the distinguishability of the $\{|0\rangle, |1\rangle\}$ states in the computational basis. Thus, if a measurement is made on S_i anyway, for every system, there is a 50% chance of the wrong measurement, and thus a 50% chance that the system is in the wrong state. Thus, there is a $\frac{1}{4}$

chance that each digit is incorrect in the sequence. Following from definition 7.3.6, we see that for both basis states, we will get the form:

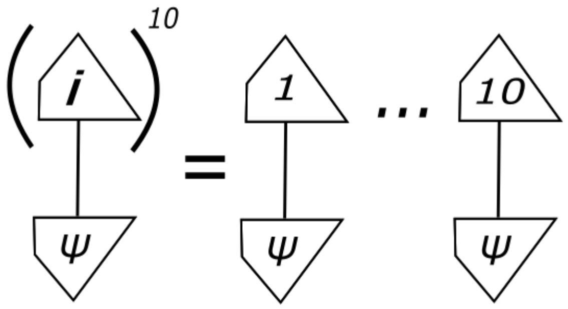


Figure 7.4.4

This yields a probability of $(\frac{3}{4})^{10} = 0.056$, a very small chance of getting a correct duplicate. Now, imagine if that sequence of bits was a length of 20, or 200. This probability would yield an impossible counterfeit, as we reach a sequence of n bits:

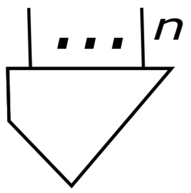


Figure 7.4.5

Similarly, we obtain:

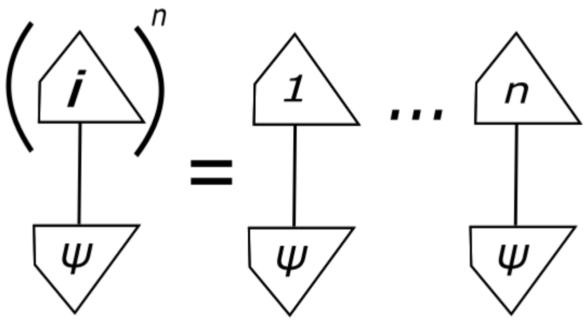


Figure 7.4.6

thus yielding in a probability of $\frac{3}{4}^n$.

This idea of money that is impossible to counterfeit is extremely useful; we can guarantee that a transaction is cryptographically secure by using these systems. However, [6] notes that the no-cloning theorem does not guarantee that secure quantum money is possible, but it instead leads us to conclude that there cannot be a machine that can copy all quantum states. It is a current area of research to see how these systems could be implemented to digital currencies.

8 Conclusion

8.1 Process Theory as a Language

In short, process theory is a structure containing all of the allowed processes of a diagram, and how they interact with each other. This is the fundamental understanding of quantum picturalism: the connectivity. Process theory involves looking at the mathematical definition of these structures, and how these structures interact with each other. The method really did become a language itself, as it brought new meaning to how the mathematical objects themselves fit into the larger process or goal of the theorem, concept, or algorithm.

8.1.1 Discussion

This project involved learning a completely new formalism of mathematics. Everything from basic algebra to quantum entanglement was interpreted diagrammatically. I sometimes found myself wondering, “is that it?” This picturalistic method seemed to ease my understanding of these concepts.

This was a brand new approach to mathematics for me, and it was difficult to switch. Bridging the gaps in my understanding between two rigorous formalism’s allowed me to obtain a firmer grasp of both languages.

Focusing on connectivity allowed me to understand quantum information theory on a deeper level, as I was paying attention to the process itself.

This work goes beyond modelling and re-representations, however. In addition to figuring out quantum algorithms and applications, there are many areas of current research with these techniques, and there are many listed in section 8.2.

There are many quantum algorithms and applications that would be interesting to explore with ZX-Calculus. One application that has not been expanded upon with quantum picturalism is Shor's Algorithm, which computes prime number factorization in polynomial time. This is a very important algorithm because many cryptography systems rely on the difficulty to factor large numbers. Having an algorithm like Shor's would deem all these encryption systems worthless.

8.2 Further Reading

In addition to everything listed in the biblography, here are some addition papers with research being done using this formalism:

1. Provides a new application of quantum computing to the field of natural language processing. A model is introduced based on tensor product composition.

William Zeng and Bob Coecke, *Quantum Algorithms for Compositional Natural Language Processing*, Electronic Proceedings in Theoretical Computer Science **221** (2016), 67–75.

2. In this thesis, Hadzihasanovic continues workig with string diagrams, and presents ZW-calculus, the first complete diagrammatic axiomatisation of the theory of qubits.

Amar Hadzihasanovic, *The algebra of entanglement and the geometry of composition* (2017).

3. Added two new axioms to the formalism to make the diagrammatic ZX-Calculus language complete for Clifford+T quantum mechanics. It is called the /4-fragment of the ZX-Calculus, proved with ZW-Calculus (integer matrices).

Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart, *A Complete Axiomatisation of the ZX-Calculus for Clifford+T Quantum Mechanics*, The 33rd Annual Symposium on Logic in Computer Science **221** (2018), 559–568.

4. Has ZX-Calculus demonstrations of tensors, and applies ZX-Calculus to surface code lattice surgery.

Dominic Horsman and Niel de Beaudrap, *The ZX calculus is a language for surface code lattice surgery*, Quantum **4** (2020), 218.

Bibliography

[1] Bob Coecke and Aleks Kissinger, *Picturing Quantum Processes*, Cambridge University Press, Cambridge, UK, 2017.

[2] Michael A. Nielsen and Isaac L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK, 2010.

[3] Ramamurti Shankar, *Principles of Quantum Mechanics, 2ed.*, Kluwer Academic/Plenum Publishers, New York, NY, 1994.

[4] Stephen Wiesner, *Conjugate Coding*, ACM SIGACT News **15** (1983), 83–85.

[5] W. K. Wootters and W. H. Zurek, *A single quantum cannot be cloned.*, Nature **299** (1982), 802–803.

[6] Scott Aaronson, Edward Farhi, David Gosset, Avinatan Hassidim, Jonathan Kelner, and Andrew Lutomirski, *Quantum Money*, Communications of the ACM **55** (2012), 84–92.