

UNIVERSITY OF CALGARY

Classifying reversible logic gates with ancillary bits

by

Cole Robert Comfort

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE

CALGARY, ALBERTA

JULY, 2019

© Cole Robert Comfort 2019

Abstract

In this thesis, two models of reversible computing are classified, and the relation of reversible computing to quantum computing is explored.

First, a finite, complete set of identities is given for the symmetric monoidal category generated by the computational ancillary bits along with the controlled-not gate. In doing so, it is proven that this category is equivalent to the category of partial isomorphisms between non-empty finitely-generated commutative torsors of characteristic 2.

Next, a finite, complete set of identities is given for the symmetric monoidal category generated by the computational ancillary bits along with the Toffoli gate. In doing so, it is proven that this category is equivalent to the category of partial isomorphisms between finite powers of the two element set.

The relation between reversible and quantum computing is also explored. In particular, the category with the controlled-not gate as a generator is extended to be complete for the real stabilizer fragment of quantum mechanics. This is performed by translating the identities to and from the angle-free fragment of the ZX-calculus, and showing that these translations are inverse to each other.

Preface

This thesis includes results which are proven in “The Category CNOT” [14] and “The Category TOF” [13] which I presented, respectively, at the 14th and 15th International Conferences on Quantum Physics and Logic. Parts of the “The Category CNOT” were in my undergraduate thesis. This also contains material in the preprint, “Circuit Relations for Real Stabilizers: Towards TOF+H” which I presented at the 4th Symposium on Compositional Structures [20].

Acknowledgements

First, I would like to thank my supervisor Dr. Robin Cockett for teaching me category theory and inspiring me to do math!

I would also like to thank my fellow graduate students and postdocs (in alphabetical order) Matthew Burke, Joseph Collins, Antonin Delpuch, Richard East, Robert Furber, Lukas Heidemann, Martti Karvonen, Prashant Kumar, Jean-Simon Lemay, Chad Nester, Paolo Perrone, Priyaa Srinivasan, Daniel Satanove, Sander Uijlen and John van de Wetering (and anyone else I have forgotten to mention) for their company, enlightening conversations, passionate political debates and merriment. A *particularly grateful* thanks is extended to Benjamin MacAdam and Jonathan Gallagher who provided me with whiteboard markers; without which, this thesis wouldn't be possible.

I would also like to thank the academic faculty at the University of Calgary for teaching and encouragement, especially Dr. Peter Høyer for inspiring me to study quantum computing and Dr. Kristine Bauer for giving me advice.

I am particularly grateful for the comments received by my examiners Dr. Bob Coecke, Dr. Gilad Gour and Dr. Renate Scheidler.

I would also like to thank Dr. Bob Coecke and Dr. Chris Heunen for hosting me for a semester at the University of Oxford and the University of Edinburgh (and an extra thanks to Bob for the beer).

I would also like to thank Brett Giles, whose PhD thesis is vital to this work.

I would also like to thank countless graduate students, postdocs and professors from

FMCS and QPL for accepting me into their company.

Finally, I would like to thank my family for their emotional support.

To the homies...

Table of Contents

Abstract	ii
Preface	iii
Acknowledgements	iv
Dedication	vi
Table of Contents	vii
List of Figures and Illustrations	ix
List of Symbols, Abbreviations and Nomenclature	x
Epigraph	xiv
1 Introduction	1
1.1 Reversible computing	2
1.2 Quantum computing as reversible computing	4
1.3 Chapter outline	5
2 Category Theory	6
2.1 Basic category theory	6
2.2 Monoidal categories	15
2.2.1 Strict monoidal categories and PROPs	17
2.2.2 Monoidal functors	19
3 Quantum Computing	22
3.1 Categorical quantum computing	22
3.1.1 †-categories and compact closed categories	22
3.1.2 Hilbert spaces	25
3.1.3 Frobenius algebras and complementarity	29
3.2 Stabilizer quantum mechanics	35
4 Restriction and Inverse Categories	39
4.1 Restriction and inverse categories	39
4.1.1 Discrete restriction categories and inverse products	41

4.2	Restriction and discrete inverse functors	46
4.3	Barr's ℓ^2 functor	52
5	The Controlled-not Gate	54
5.1	The category CNOT	55
5.2	Preliminary results for CNOT	57
5.3	CNOT is a discrete inverse category	58
5.3.1	Inverse products in CNOT	58
5.3.2	CNOT is an inverse category	63
5.4	Torsors	68
5.5	The equivalence between CNOT and $\text{Parlso}(\text{CTor}_2)^*$	74
5.5.1	Defining the functor $\tilde{H}_0 : \text{CNOT} \rightarrow \text{Parlso}(\text{CTor}_2)^*$	74
5.5.2	The points of CNOT	76
5.5.3	Internal torsor structures in CNOT	78
5.5.4	Normal form for the idempotents of CNOT	82
5.5.5	$\tilde{H}_0 : \text{CNOT} \rightarrow \text{Parlso}(\text{CTor}_2)^*$ is a discrete inverse equivalence	87
6	CNOT and ZX_π	92
6.1	ZX_π	92
6.2	Embedding CNOT into ZX_π	95
6.3	Extending CNOT to ZX_π	102
6.3.1	The completeness of $\text{CNOT} + H$	105
7	The Toffoli Gate	117
7.1	The category TOF	118
7.2	Controlled-not gates and the Iwama identities	124
7.3	TOF is a discrete inverse category	133
7.4	The points of TOF	136
7.5	Partial injective functions and TOF	140
7.6	A normal form for the idempotents of TOF	140
7.6.1	$\tilde{H}_0 : \text{TOF} \rightarrow \text{FPinj}_2$ is a discrete inverse equivalence	146
8	Conclusions and future work	148
8.1	Conclusions	148
8.2	Future work	149
	Bibliography	151
	A Matrices and Constants	157
	B Basic Calculations for CNOT	161

List of Figures and Illustrations

5.1	The identities of CNOT	56
6.1	The identities of ZX_α (where $\alpha \in \{0, \pi\}$)	93
6.2	The identities of CNOT + H (in addition to the identities of CNOT)	103
7.1	The identities of TOF	120

List of Symbols, Abbreviations and Nomenclature

<i>Symbol or abbreviation</i>	<i>Definition</i>
$g \circ f$:	The composite fg .
$x \in X$:	x is an element of the set X .
\forall :	For all.
\exists :	There exists.
\geq :	Greater than or equal to.
$>$:	Greater than.
\leq :	Less than or equal to.
$<$:	Less than.
∞ :	Infinity.
$\mathbb{X}, \mathbb{Y}, \mathbb{Z}, \dots$:	Arbitrary categories.
A, B, C, X, Y, Z, \dots :	Arbitrary objects.
$\lim_{\leftarrow} F$:	The limit of a functor F .
\top :	Initial object.
\mathbb{X}_0 :	The objects of \mathbb{X} .
\mathbb{X}_1 :	The maps of \mathbb{X} .
$\partial_0(f)$:	Domain object of map f .
$\partial_1(f)$:	Codomain object of map f .
$X \xrightarrow{f} Y$:	A map f from object X to Y .
$f : X \rightarrow Y$:	A map f from object X to Y .
\hookrightarrow :	A monic map, or faithful functor.
\twoheadrightarrow :	An epic map, or full functor.
\hookrightarrow :	The inclusion functor.
$\mathbb{X}(-, -)$:	The hom functor of \mathbb{X} .
$[\mathbb{X}, \mathbb{Y}]$:	Functor category from \mathbb{X} to \mathbb{Y} .
$X \multimap Y$:	The internal hom from X to Y .
1_X :	The identity map on X .
\mathbb{X}^{op} :	The opposite category of \mathbb{X} .
$\alpha : F \Rightarrow G$:	A natural transformation α from functors F to G .
$(-)^{\dagger}$:	A dagger functor.
$(-)^*$:	The dualizing functor.

\mathbb{Z} :	The integers.
\mathbb{Z}_n :	The integers modulo n .
CNOT :	The symmetric monoidal category generated by computational ancillary bits and the controlled-not gate.
ZX_π :	The angle-free fragment of the ZX-calculus.
ΔZX :	The angle-free fragment of the ZX-calculus with the triangle generator.
ZH_π :	The phase-free fragment of the ZH-calculus.
Mat_k :	The category matrices over k .
Vect_k :	The category of vector spaces over k .
FdVect_k :	The category of finite dimensional vector spaces over k .
$\llbracket - \rrbracket_{\text{CNOT}}$:	The canonical interpretation $\llbracket - \rrbracket_{\text{CNOT}} : \text{CNOT} \rightarrow \text{Mat}_{\mathbb{C}}$.
CNOT + H :	The symmetric monoidal category generated by computational ancillary bits, controlled-not gate, the Hadamard gate and the scalar $\sqrt{2}$.
cnot :	The controlled-not gate.
not :	The not gate.
FHilb :	The category of finite dimensional Hilbert spaces.
Hilb :	The category of Hilbert spaces.
\oplus :	The operating bit of a controlled-not gate. Also used for addition in \mathbb{Z}_2 .
\bar{f} :	The restriction of a map f to its domain of definition.
$\text{dom } f$:	The domain of definition of a map f .
$\text{cod } f$:	The codomain of definition of a map f .
$\text{Span}(\mathbb{X})$:	The category of spans in \mathbb{X} .
$\text{Total}(\mathbb{X})$:	The subcategory of total maps of a restriction category \mathbb{X} .
$\text{Par}(\mathbb{X})$:	The partial map category of a category \mathbb{X} with all monics.
$\text{ParIso}(\mathbb{X})$:	The partial isomorphism category of a category \mathbb{X} with all monics.
$(-)^{\circ}$:	The partial inverse functor.
$\langle f, g \rangle$:	The pairing $\Delta(f \otimes g)$ of maps f and g .
$\langle f g \rangle$:	The inner product of f and g .
$!$:	The unique map to the restriction final object.
Δ_A :	The diagonal map on A .
∇_A :	The partial inverse of Δ_A .
ℓ^2 :	Barr's $\ell^2 : \text{Pinj} \rightarrow \text{Hilb}$ functor.
δ_x :	The Kronecker delta function for x .
$f \cap g$:	The meet of f and g .

$a \times_b c$:	The torsor paramultiplication of a, b, c .
$_ \times _$:	The product functor.
π_i :	The i th projection.
\sum :	The sum.
\prod :	The product.
$_ \otimes _$:	A tensor product.
$(_)^{\otimes n}$:	The n -fold iterated tensor product.
u_X^L :	The left unitor for X .
u_X^R :	The right unitor for X .
$a_{A,B,C}$:	The associator for A, B, C .
$c_{A,B}$:	The symmetry for A, B .
m_I :	The coherence map for a monoidal functor preserving the tensor unit.
m_{\otimes} :	The coherent natural transformation for a monoidal functor preserving the tensor product.
I :	The tensor unit.
$X_f \times_g Y$:	The pullback of $X \xrightarrow{f} Z \xleftarrow{g} Y$.
\lrcorner :	Denotes that an object is a pullback.
\mathbf{CTor}_2 :	The category of finitely-generated commutative torsors of characteristic 2.
$X \cong Y$:	X and Y are isomorphic.
$\mathbf{ParIso}(\mathbf{CTor}_2)^*$:	The category of partial isomorphisms of nonempty finitely-generated commutative torsors of characteristic 2.
\emptyset :	The zero map from 0 to 0.
$\emptyset_{n,m}$:	The zero map from n to m .
$\mathbf{swap}_{(i,n)}$:	The gate swapping the 1st and i th wires in an n wire array.
\mathbf{TOF} :	The symmetric monoidal category generated by the computational ancillary bits and Toffoli not gate.
$\llbracket _ \rrbracket_{\mathbf{TOF}}$:	The canonical interpretation $\llbracket _ \rrbracket_{\mathbf{TOF}} : \mathbf{TOF} \rightarrow \mathbf{Mat}_{\mathbb{C}}$.
\mathbf{Set} :	The category of sets and functions.
\mathbf{Rel} :	The category of sets and relations.
\mathbf{Pinj} :	The category of sets and partial isomorphisms.
\mathbf{FPinj} :	The category of finite sets and partial isomorphisms.
\mathbf{FPinj}_2 :	The category of finite powers of the two element set as objects and partial isomorphisms as maps.
\mathbf{tof} :	The Toffoli gate.
\mathbf{cnot}_n :	The generalized controlled-not gate with n control bits.

$[x, X]:$	Generalized controlled-not gate operating on wires indexed by X , targeting $x \notin X$.
$\triangleright_n:$	An input 0 ancillary bit on wire n .
$\triangleleft_n:$	An output 0 ancillary bit on wire n .
$X \cup Y:$	The union of sets X and Y .
$X \sqcup Y:$	The disjoint union of sets X and Y .

Epigraph

Un symbole dépasse toujours celui qui en use et lui fait dire en réalité plus qu'il n'a conscience d'exprimer.

- Albert Camus, *Le Mythe de Sisyphe: essai sur L'absurde* [12]

Chapter 1

Introduction

In this thesis, complete sets of identities are given for two fragments of reversible computing. The first fragment, the category **CNOT**, is generated by the controlled-not gate and ancillary bits. This is the affine fragment of reversible computing, and in particular, it is shown that this is equivalent to the inverse category of partial isomorphisms between non-empty finitely-generated commutative torsors of characteristic 2. The second fragment, the category **TOF**, is generated by the Toffoli gate and ancillary bits. This is shown to be equivalent to the inverse category of partial isomorphisms between finite powers of the two element set. Thus it is a full fragment of reversible computing. In Section 1.1 of this chapter, we give an overview of the field of reversible computing; and in particular, we motivate modeling reversible computing with inverse categories.

We also explore the relation of reversible and quantum computing. In particular, we extend **CNOT** to the angle-free fragment of the **ZX-calculus**, \mathbf{ZX}_π , by adding the Hadamard gate as a generator. In Section 1.2 of this chapter, an overview of the relation of quantum and reversible computing is given. In particular, the relation between the doctrine of inverse categories and that of categorical quantum mechanics is sketched.

Finally, at the end of this chapter, in Section 1.3, the general structure, and an overview of the content of this thesis is given.

1.1 Reversible computing

Landauer's principle, originally posed in [39], posits that the information lost about a state, before and after a computation step corresponds to heat being produced. This connects the Shannon entropy of the process to the thermodynamic entropy of physically computing a process. In particular, the Shannon entropy of a system is a constant multiple of the thermodynamic entropy: 1 bit of information corresponds to $k_B \ln 2$ Joules per Kelvin, where k_B is Boltzmann's constant. Landauer's principle presents a problem for classical computing: because classical computers are built with irreversible logic gates, they are prone to overheating.

The limits that Landauer's principle imposes on irreversible computing has motivated finding models of reversible computing. Category theory is a particularly useful framework through which this objective can be explored, because it reveals the *compositional structure* of computation. Category theory allows one to structurally examine how the constituent components connect together to form coherent computations. Specifically, monoidal categories naturally model circuits using the tensor product [46]. Monoidal categories have been applied in the field of quantum computing, and, in particular to quantum circuit optimization [36], to which reversible computing is intimately connected.

Naïvely, one might hope to model reversible computation with groupoids, that is, categories where all maps are isomorphisms. However, this approach has its drawbacks. Reversible circuits, and in particular quantum circuits, often make use of auxiliary space. That is, memory that is prepared in a certain configuration and post-selected to remain in the same configuration. In quantum computing, this trick allows circuits to be implemented with limited gate sets. For example, in Grover's algorithm, auxiliary bits allow one to kickback the output of the oracle into the phase [53].

However, if we are going to take this compositional approach to reversible computing seriously, preparing and post-selecting the state must be regarded as maps in the category. Preparing and post-selecting the state are not isomorphisms, rather, they are only one-sided

inverses to each other in Hilbert spaces. Therefore, state preparation and post-selection cannot be regarded as generators of a groupoid.

A similar problem arises when one considers reversible computing in general, and not just reversible circuits. A Turing machine with a reversible transition function (an operationally reversible Turing machine) can produce a program that never halts; thus having a denotational interpretation which is not an isomorphism, but rather a partial isomorphism [35, §1.1.1].

Since groupoids fail to capture many features of reversible computing, one could try to model reversible computing in \dagger -categories (pronounced dagger-categories): that is, categories \mathbb{X} for which there is a contravariant, identity on objects involution $(-)^{\dagger} : \mathbb{X}^{\text{op}} \rightarrow \mathbb{X}$. That is, for all maps f , $(f^{\dagger})^{\dagger} = f$ and all objects X , $X^{\dagger} = X$. This approach is used in quantum computing, where the dagger is the Hermetian adjoint. In this setting, instead of every map having a proper inverse, every map f would only have an adjoint f^{\dagger} . However merely having a distinguished \dagger -functor is not necessarily useful for modeling computation.

Many of these aforementioned problems, insofar as classical reversible computing is concerned, can be resolved with inverse categories. Inverse categories generalize groupoids by stipulating that every map need only have a *partial* inverse with respect to some chosen restriction structure. The restriction structure should be chosen to distinguish when information is lost. Furthermore, this restriction structure determines a unique \dagger -functor. Therefore, if the input is known a priori to be in the domain of definition, then the input can be computed from the output by composition with the adjoint: the partial inverse. When a map is composed with its partial inverse, then a weaker form of identity called a restriction idempotent is produced. Restriction idempotents commute with each other and are closed under composition and taking partial inverses, therefore they generalize much of the structure of identity maps. This solves the problem of modeling reversible circuits; namely, preparing a state and post-selecting the same state are not inverse to each other, but only partial inverses. In the case of reversible Turing machines, one can take the partial inverse

by inverting the transition function.

Inverse categories can support different structure than groupoids. For example, consider a monoidal category with copying. Although copying is not an isomorphism, in many cases, it has a partial inverse given by cocopying. Copying and cocopying structures that form a (nonunital) special commutative \dagger -Frobenius algebra are called inverse products. Furthermore, categories with inverse products compatible with the tensor product are called discrete inverse categories. Discrete inverse categories, as we will see, are useful tools for modeling reversible computing.

1.2 Quantum computing as reversible computing

The study of discrete inverse categories, and thus (partial) reversible computing, is closely connected to categorical quantum computing. Both reversible computing in this sense and categorical quantum mechanics model computation in categories which have structure generalizing the linear nature of Hilbert spaces. First, \dagger -functors generalize the structure of the Hermitian adjoint in Hilbert spaces (\mathbf{Hilb}). Moreover, unital special commutative \dagger -Frobenius algebras correspond to bases in finite dimensional Hilbert spaces (\mathbf{FHilb}) [19]. It is not unsurprising that these structures are central in both theories. Bases, and thus Frobenius algebras, allow one to define a diagonal map; that is, a map which copies the basis elements. Miraculously, this map is a partial isomorphism, and thus, can be seen as a reversible gate. However, the unit of the Frobenius algebra cannot be copied. This is because composing the unit with the comultiplication corresponds to a maximally mixed state in \mathbf{FHilb} , which cannot be copied because of the no-cloning theorem.

This draws a close connection between these two models of computing, and reveals the mid-point between classical computing and quantum (reversible) computing: namely, the unitality of the chosen Frobenius algebra.

1.3 Chapter outline

In Chapter 2, some aspects of category theory are summarized. In Chapter 3, some aspects of quantum computing and categorical quantum mechanics are reviewed. In Chapter 4, restriction and inverse categories are reviewed and some new lemmas are proved, which are crucial to Chapters 5 and 7. The remainder of the thesis is original work, excluding Section 6.1 on the ZX-calculus. In Chapter 5, a complete set of identities is given for the symmetric monoidal category, CNOT, generated by the state preparation and post-selected measurement in the computational basis and the controlled-not gate. This shows that CNOT is equivalent to the category of affine partial isomorphisms between finite dimensional \mathbb{Z}_2 vector spaces. In Chapter 6, this category is completed to the angle-free fragment of the ZX-calculus, ZX_π , by adding the Hadamard gate as a generator. In particular, this completes the inverse products of CNOT to a *unital* Frobenius algebra structure. In Chapter 7, the identities of Chapter 5 are extended to give a complete set of identities for the symmetric monoidal category, TOF, generated by state preparation and post-selected measurement in the computational basis and the *Toffoli gate*. It is shown that TOF is equivalent to the category of partial isomorphisms between sets with cardinalities of powers of 2; a full subcategory of sets and partial injections. Finally, in Chapter 8, we discuss the ramifications of this research, and potential future work.

Chapter 2

Category Theory

In this chapter, we review some aspects of category theory and, in particular, the basic theory of monoidal categories. All definitions and results in this chapter are well-known; I refer the reader to [40] for further reading.

2.1 Basic category theory

Definition 2.1.1. A **category** \mathbb{X} consists of a class \mathbb{X}_0 of **objects** and a class \mathbb{X}_1 of **maps** such that each map $f \in \mathbb{X}_1$ has an associated **domain** $\partial_0(f) \in \mathbb{X}_0$ and **codomain** $\partial_1(f) \in \mathbb{X}_0$. A map f with domain X and codomain Y , is denoted by $X \xrightarrow{f} Y$. The class of maps from an object X to Y is denoted by $\mathbb{X}(X, Y)$.

Given maps $X \xrightarrow{f} Y$ and $Y \xrightarrow{g} Z$, we can form the **composite** $X \xrightarrow{fg} Z$. Every object X has a canonical, associated map $X \xrightarrow{1_X} X$ called **the identity** on X . The maps of \mathbb{X} are subject to the following two axioms:

Identity law: For all maps $X \xrightarrow{f} Y$, $1_X f = f = f 1_Y$.

Associative law: For all maps $X \xrightarrow{f} Y$, $Y \xrightarrow{g} Z$, $W \xrightarrow{h} V$; $(fg)h = f(gh)$.

A **locally small category** is a category, where for all objects X and Y , $\mathbb{X}(X, Y)$ is a set—and not for example a proper class.

Maps $X \xrightarrow{f} Y$ are sometimes denoted by $f : X \rightarrow Y$. The identity map on X is often denoted by $X \xrightarrow{=} X$. The composite of maps $X \xrightarrow{f} Y \xrightarrow{g} Z$ is sometimes denoted by $g \circ f$, instead of fg . This order of composition using \circ is called the **applicative order of composition**, as opposed to the default order of composition in this document which is called **diagrammatic order of composition**. The applicative order of composition is used in the bra-ket notation in quantum computing, as discussed in Chapter 3.

Example 2.1.2. The **category of sets**, Set , is a category where the objects are sets, maps are functions and composition is function composition.

Two different categories can have the same class of objects:

Example 2.1.3. The **category of relations**, Rel , is a category with:

Objects: Sets.

Maps: Maps $R : X \rightarrow Y$ are subsets of $X \times Y$.

Composition: The composite of maps $R : X \rightarrow Y$ and $S : Y \rightarrow Z$ is given by:

$$RS := \{(x, z) \in X \times Z \mid \exists y \in Y : (x, y) \in R, (y, z) \in S\}$$

Identity: $1_X : X \rightarrow X$ is given by the map $\{(x, x) \mid \forall x \in X\}$.

A relation $R : X \rightarrow X$ is **reflexive** if for all x in X , $(x, x) \in R$. A relation is **symmetric** if $(x, y) \in R$ implies $(y, x) \in R$. A relation is **transitive** if $(x, y), (y, z) \in R$ implies $(x, z) \in R$. A relation is an **equivalence relation** in case it is reflexive, symmetric and transitive. If X is endowed with extra structure (for example, being a group), then an equivalence relation on X preserving this structure is called a **congruence**.

Definition 2.1.4. A category \mathbb{X} is a **subcategory** of another category \mathbb{Y} , when \mathbb{X} can be obtained by forgetting some of the objects and maps of \mathbb{Y} .

A category \mathbb{X} is a **full subcategory** of another category \mathbb{Y} , when \mathbb{X} can be obtained by forgetting some of the objects of \mathbb{Y} .

Example 2.1.5. The **category of vector spaces** over a field k , \mathbf{Vect}_k , is a category where the objects are k -vector spaces and the maps are linear transformations between k -vector spaces. The composition and identities are given by the underlying composition and identities in \mathbf{Set} . Denote the full subcategory of \mathbf{Vect}_k , where the objects are finite dimensional k -vector spaces, by \mathbf{FdVect}_k .

The following category is much like \mathbf{FdVect}_k , except it doesn't have all of the objects:

Example 2.1.6. The **category of matrices** over a field k , \mathbf{Mat}_k , is a category where the objects are natural numbers and the maps from n to m are k -valued $n \times m$ matrices. Composition is given by matrix multiplication and the identity is the identity matrix.

A new category can be obtained by reversing the order of composition:

Definition 2.1.7. Given a category \mathbb{X} , the **opposite category** of \mathbb{X} , \mathbb{X}^{op} is defined with the same objects as \mathbb{X} and maps given by:

$$\frac{X \xrightarrow{f} Y \text{ in } \mathbb{X}}{Y \xrightarrow{f^{\text{op}}} X \text{ in } \mathbb{X}^{\text{op}}}$$

The composite of maps and identities are defined in the obvious way. Every true proposition about a category \mathbb{X} entails a **dual** proposition which holds in \mathbb{X}^{op} . The prefix **co** is used to describe dual properties.

Many familiar algebraic structures are categories. Recall that an **isomorphism** is a map f with an inverse f^{-1} so that $ff^{-1} = f^{-1}f = 1$. A **monoid** is a one object category. A **groupoid** is a category in which every map is an isomorphism. A **group** is a category which is simultaneously a monoid and a groupoid: that is, a one object groupoid.

A **commutative diagram** in a category \mathbb{X} is a directed graph with a start point and end point where the nodes are objects, and arrows are maps of \mathbb{X} . Moreover, the composites

of all paths from the start point to the end point are required to be equal; where all maps must be covered by such a path. For example to say that the following is a commutative diagram in \mathbb{Z} ,

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

is to say that $fg = hk$. In order to assert that a diagram is commutative, one often says that *the diagram commutes*.

Definition 2.1.8. A map $X \xrightarrow{f} Y$ in a category \mathbb{X} is **epic** (an epimorphism) in case for all objects Z and parallel arrows $g, h : Y \rightarrow Z$: whenever $fg = fh$, then $g = h$. The dual notion of an epic map is a **monic** map (a monomorphism).

Epic maps are drawn as \twoheadrightarrow and monics as \rightarrowtail .

Just as categories are composed of objects with maps between them, one can consider categories as objects themselves with maps between them:

Definition 2.1.9. Given categories \mathbb{X} and \mathbb{Y} , a **functor** $F : \mathbb{X} \rightarrow \mathbb{Y}$ associates to each object X in \mathbb{X} , an object $F(X)$ in \mathbb{Y} and to each map $X \xrightarrow{f} Y$ in \mathbb{X} , a map $F(X) \xrightarrow{F(f)} F(Y)$ in \mathbb{Y} . A functor must satisfy the following two axioms:

Preservation of Identity: For all objects X of \mathbb{X} , $F(1_X) = 1_{F(X)}$.

Preservation of Composition: For all maps $X \xrightarrow{f} Y, Y \xrightarrow{g} Z$, $F(fg) = F(f)F(g)$.

For example, given a k -vector space V , taking the the dual of V , V^* is a functor from $\mathbf{Vect}_k^{\text{op}} \rightarrow \mathbf{Vect}_k$. Similarly, taking the matrix transpose is a functor from $\mathbf{Mat}_k^{\text{op}} \rightarrow \mathbf{Mat}_k$.

Every category \mathbb{X} is endowed with an identity functor denoted by \mathbb{X} acting as the identity on maps, moreover, functors compose. Therefore, locally small categories and functors themselves form a category, \mathbf{Cat} , with locally small categories as objects and functors as maps. One must be careful of size issues: \mathbf{Cat} is not itself a locally small category, although its objects are.

The canonical inclusion functor induced by a full subcategory is drawn as \hookrightarrow .

Every functor $F : \mathbb{X} \rightarrow \mathbb{Y}$ between locally small categories induces a function:

$$F_{X,Y} : \mathbb{X}(X, Y) \rightarrow \mathbb{Y}(F(X), F(Y))$$

Taking maps $X \xrightarrow{f} Y$ to the map $F(X) \xrightarrow{F(f)} F(Y)$ for all objects X and Y in \mathbb{X} . This leads to three important notions:

Definition 2.1.10. A functor $F : \mathbb{X} \rightarrow \mathbb{Y}$ is **full** if for all objects X and Y in \mathbb{X} , the induced map $F_{X,Y} : \mathbb{X}(X, Y) \rightarrow \mathbb{Y}(F(X), F(Y))$ is surjective, and the functor is **faithful** if this induced map is injective.

A functor $F : \mathbb{X} \rightarrow \mathbb{Y}$ is **essentially surjective** if, for every object Y of \mathbb{Y} , there is an isomorphism $f : Y \rightarrow F(X)$ in \mathbb{Y} for some object X in \mathbb{X} .

Full functors are drawn as \twoheadrightarrow and faithful functors as \rightarrowtail . This notation is in reference to the induced map $F_{X,Y}$ being epic or monic.

The functor $\mathbf{Vect}_k \rightarrow \mathbf{Set}$, forgetting the linear structure is faithful. Given a full subcategory \mathbb{X} of \mathbb{Y} , the inclusion functor $\mathbb{X} \hookrightarrow \mathbb{Y}$ is also faithful. Since monoids are one object categories, a surjective monoid homomorphism is an example of a full functor. Similarly, an injective monoid homomorphism is faithful.

Just as functors are maps between categories, there is also a notion of a map between functors.

Definition 2.1.11. Given parallel functors $F, G : \mathbb{X} \rightarrow \mathbb{Y}$, a **natural transformation** $\alpha : F \Rightarrow G$ is an indexed family of maps $\{\alpha_X : F(X) \rightarrow G(X)\}_{X \in \mathbb{X}_0}$ such that for all maps $X \xrightarrow{f} Y$ in \mathbb{X} , the following diagram commutes:

$$\begin{array}{ccc} F(X) & \xrightarrow{F(f)} & F(Y) \\ \alpha_X \downarrow & & \downarrow \alpha_Y \\ G(X) & \xrightarrow{G(f)} & G(Y) \end{array}$$

The map $\alpha_X : F(X) \rightarrow G(X)$ is called the **component** of α at X . The subscripts of the components are often suppressed for ease of notation, as they can be inferred by context.

Categories \mathbb{X} and \mathbb{Y} induce a category $[\mathbb{X}, \mathbb{Y}]$, the **functor category** from \mathbb{X} to \mathbb{Y} , with functors from $\mathbb{X} \rightarrow \mathbb{Y}$ as objects and natural transformations as maps. An isomorphism $\alpha : F \Rightarrow G$ in this category is called a **natural isomorphism**. This is made explicit by the notation $F \cong G$.

For example, in \mathbf{Vect}_k , there is a natural injection $\iota : (-) \Rightarrow (-)^{**}$ with components $(\iota_V(v))(f) := f(v)$. The injection of a vector space into a double dual is an isomorphism if and only if the vector space is finite dimensional. The map from V to V^* is basis dependent, and therefore fails to be natural. For every choice of basis, there is a different isomorphism.

Definition 2.1.12. Let $F : \mathbb{D} \rightarrow \mathbb{X}$ be a functor. A **cone** to F is a pair (X, ψ) where X is an object of \mathbb{X} , and $\psi_D : X \rightarrow F(D)$ is a family of maps indexed by the objects D of \mathbb{D} , so that for all maps f in $\mathbb{D}(C, D)$, $\psi_C F(f) = \psi_D$.

The **limit** of $F : \mathbb{D} \rightarrow \mathbb{X}$, if it exists, is a cone $(\lim_{\leftarrow} F, \varphi)$, so that for any cone (X, ψ) and any map f in $\mathbb{D}(C, D)$, the following diagram commutes:

$$\begin{array}{ccc}
 & X & \\
 \psi_C \swarrow & \vdots & \searrow \psi_D \\
 & \lim_{\leftarrow} F & \\
 \varphi_C \swarrow & \leftarrow & \searrow \varphi_D \\
 F(C) & \xrightarrow{F(f)} & F(D)
 \end{array}$$

The dotted line is notation asking for the unique existence of a map $X \rightarrow \lim_{\leftarrow} F$ making the diagram commute.

The dual notion of a limit is a **colimit**. A category with all (finite) limits is called (finitely) **complete**.

Definition 2.1.13. An object \top in \mathbb{X} is **initial** when for every object X of \mathbb{X} , there is a

unique map $\top \rightarrow X$. The dual notion of an initial object is a **final** object.

Definition 2.1.14. The **equalizer** of two parallel arrows $f, g : X \rightarrow Y$ in a category \mathbb{X} (if it exists) is the pair $(E, e : E \rightarrow X)$ such that given any pair $(Z, h : Z \rightarrow X)$ so that $hg = hf$, the following diagram commutes:

$$\begin{array}{ccc} E & \xrightarrow{e} & X & \begin{array}{c} \xrightarrow{g} \\ \xrightarrow{f} \end{array} & Y \\ \downarrow m & \nearrow h & & & \\ Z & & & & \end{array}$$

Lemma 2.1.15. The equalizer map $e : E \rightarrow X$ is monic.

Proof. Suppose that $e : E \rightarrow X$ is the equalizer of $X \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} Y$. Consider parallel $Z \begin{array}{c} \xrightarrow{h} \\ \xrightarrow{k} \end{array} E$, so that $he = ke$. Then $hef = heg$, so that, because e is an equalizer, there exists a unique map $m : Z \rightarrow E$ so that $me = he$. Therefore, by the uniqueness of m , $m = k$; so that, by symmetry, $h = m = k$.

□

Monics that arise as equalizers are called **regular monics**. The dual notion to the equalizer is the **coequalizer**. Moreover, a **regular epic** is an epic that is a coequalizer. All monics and epics in **Set** are regular.

Definition 2.1.16. The **product** of objects X and Y , if it exists, is an object $X \times Y$ and maps $X \xleftarrow{\pi_1} X \times Y \xrightarrow{\pi_2} Y$, such that for any object Z and maps $X \leftarrow Z \rightarrow Y$, the following diagram commutes:

$$\begin{array}{ccccc} & & Z & & \\ & \swarrow & \downarrow & \searrow & \\ X & \xleftarrow{\pi_1} & X \times Y & \xrightarrow{\pi_2} & Y \end{array}$$

The dual notion to the product is called the **coproduct**.

For example, the product in **Set** is the Cartesian product. **Cat** has products: explicitly, the product $\mathbb{X} \times \mathbb{Y}$ just has objects $\mathbb{X}_0 \times \mathbb{Y}_0$ and maps $\mathbb{X}_1 \times \mathbb{Y}_1$. A **bifunctor** $F : \mathbb{X} \times \mathbb{Y} \rightarrow \mathbb{Z}$ is a functor from the product category of \mathbb{X} and \mathbb{Y} to \mathbb{Z} .

Definition 2.1.17. Similarly, the **pullback** of a pair of maps $X \xrightarrow{f} Z \xleftarrow{g} Y$, if it exists, is the limit of the span diagram $X \xrightarrow{f} Z \xleftarrow{g} Y$. The pullback object of such a diagram is denoted by $X_f \times_g Y$. To make it clear that a diagram is a pullback, a corner is drawn in the diagram beside the pullback object:

$$\begin{array}{ccc} X_f \times_g Y & \xrightarrow{\pi_1} & Y \\ \pi_0 \downarrow & \lrcorner & \downarrow g \\ X & \xrightarrow{f} & Z \end{array}$$

The dual notion to the pullback is called the **pushout**.

For example, given two maps $X \xrightarrow{f} Z \xleftarrow{g} Y$ in **Set**, The pullback object $X_f \times_g Y$ is

$$\{(x, y) \in X \times Y \mid f(x) = g(y)\}$$

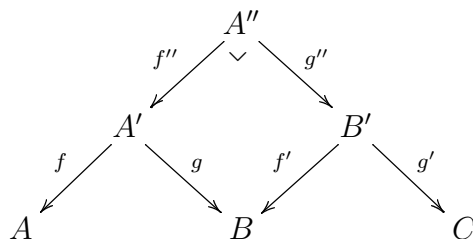
Definition 2.1.18. Given a category with pullbacks \mathbb{X} , define the **span category**, $\text{Span}(\mathbb{X})$:

Objects: Objects in \mathbb{X} .

Maps: A map from A to B is a span $A \xleftarrow{f} A' \xrightarrow{g} B$, for arbitrary A' , which we denote as (f, g) modulo an equivalence relation \sim ; $(f, g) \sim (f', g')$ if and only if there exists an isomorphism α such that $\alpha f' = f$ and $\alpha g' = g$.

Identities: The identity on A is the pair $(1_A, 1_A)$.

Composition: For maps $(f, g) : A \rightarrow B$ and $(f', g') : B \rightarrow C$, $(f, g)(f', g') := (f''f, g''g')$ where f'' and g'' are determined by the following pullback:



Composition is well-defined even though pullbacks are determined only up to isomorphism as the maps are taken modulo the equivalence relation.

An isomorphism of categories is a very restrictive notion. In general, one usually only asks for their identity functors to be naturally isomorphic:

Definition 2.1.19. An **equivalence of categories** between \mathbb{X} and \mathbb{Y} is a pair of functors $F : \mathbb{X} \rightarrow \mathbb{Y}$ and $G : \mathbb{Y} \rightarrow \mathbb{X}$ such that there are natural isomorphisms $FG \cong \mathbb{X}$ and $GF \cong \mathbb{Y}$. Two categories are equivalent in case there exists an equivalence of categories between them.

For example, the categories \mathbf{FdVect}_k and \mathbf{Mat}_k are equivalent, although they are not isomorphic. This is because the objects are different: in \mathbf{Mat}_k there is one object for each vector space of dimension n ; whereas, in \mathbf{Vect}_k there are many isomorphic copies.

There is an alternate characterization of an equivalence of categories:

Theorem 2.1.20. Categories \mathbb{X} and \mathbb{Y} are equivalent if and only if there exists a full, faithful and essentially surjective functor $F : \mathbb{X} \rightarrow \mathbb{Y}$.

The proof invokes the axiom of choice, as one must choose the isomorphisms $X \rightarrow G(Y)$, for every $X \in \mathbb{X}$, for some $G : \mathbb{Y} \rightarrow \mathbb{X}$.

Recall that the set of maps from object X to object Y in a category \mathbb{X} is denoted by $\mathbb{X}(X, Y)$. $\mathbb{X}(-, -)$ is also a functor:

Definition 2.1.21. Given a category \mathbb{X} , the **hom-functor** $\mathbb{X}(-, -) : \mathbb{X}^{\text{op}} \times \mathbb{X} \rightarrow \mathbf{Set}$ takes:

Objects: $(X, Y) \mapsto \mathbb{X}(X, Y)$

Maps: $(X \xrightarrow{f^{\text{op}}} Y, Z \xrightarrow{g} W)$ to the function $\mathbb{X}(X, Y) \rightarrow \mathbb{X}(Z, W)$ taking:

$$(h : Y \rightarrow Z) \mapsto fhg$$

A functor $F : \mathbb{X} \rightarrow \mathbf{Set}$ is **representable** when there exists an object X of \mathbb{X} such that there is a natural isomorphism $F \cong \mathbb{X}(X, -)$.

Definition 2.1.22. Two functors $L : \mathbb{X} \rightarrow \mathbb{Y}$ and $R : \mathbb{Y} \rightarrow \mathbb{X}$ form an **adjoint pair**, $L \dashv R$, when there is a bijection between $\mathbb{Y}(L(X), Y)$ and $\mathbb{X}(X, R(Y))$ natural in X and Y . L is the **left adjoint** of the adjunction and R is the **right adjoint** of the adjunction.

Equivalently, the functor $L : \mathbb{X} \rightarrow \mathbb{Y}$ is left adjoint to $R : \mathbb{Y} \rightarrow \mathbb{X}$ when there are two natural transformations $\eta : \mathbb{X} \rightarrow LR$ and $\varepsilon : RL \rightarrow \mathbb{Y}$ so that the triangle identities hold:

$$\begin{array}{ccc}
 L & & \\
 L\eta \downarrow & \searrow & \\
 LRL & \xrightarrow{\varepsilon} & L
 \end{array}
 \qquad
 \begin{array}{ccc}
 R & \xrightarrow{\eta} & RLR \\
 & \searrow & \downarrow R\varepsilon \\
 & & R
 \end{array}$$

2.2 Monoidal categories

There is an important class of categories in which circuits can be expressed naturally:

Definition 2.2.1. A **monoidal category** \mathbb{X} is a category equipped with:

- A bifunctor $_ \otimes _ : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{X}$ called the **tensor product**.
- A distinguished object I of \mathbb{X} called the **tensor unit**.
- A natural isomorphism a with components $a_{X,Y,Z} : (X \otimes Y) \otimes Z \rightarrow X \otimes (Y \otimes Z)$ called the **associator**.
- A natural isomorphism u^L with components $u_X^L : I \otimes X \rightarrow X$ called the **left unitor**.
- A natural isomorphism u^R with components $u_X^R : X \otimes I \rightarrow X$ called the **right unitor**.

Such that the following diagrams commute:

The Mac Lane pentagon diagram:

$$\begin{array}{ccc}
 ((A \otimes B) \otimes C) \otimes D & \xrightarrow{a_{A,B,C} \otimes 1_D} & (A \otimes (B \otimes C)) \otimes D & \xrightarrow{a_{A,B \otimes C,D}} & A \otimes ((B \otimes C) \otimes D) \\
 a_{A \otimes B,C,D} \downarrow & & & & \downarrow a_{1 \otimes a_{B,C,D}} \\
 (A \otimes B) \otimes (C \otimes D) & \xrightarrow{a_{A,B,C \otimes D}} & & & A \otimes (B \otimes (C \otimes D))
 \end{array}$$

Interaction of unitors and associator:

$$\begin{array}{ccc}
 (A \otimes I) \otimes B & \xrightarrow{a_{A,I,B}} & A \otimes (I \otimes B) \\
 \searrow u_A^R \otimes 1_B & & \swarrow 1_A \otimes u_B^L \\
 & A \otimes B &
 \end{array}$$

A monoidal category is said to be **symmetric** if additionally, it is equipped a natural isomorphism c with components $c_{X,Y} : X \otimes Y \rightarrow Y \otimes X$ called the **symmetry** such that the following diagrams commute:

Inverse law:

$$\begin{array}{ccc}
 A \otimes B & & \\
 c_{A,B} \downarrow & \searrow & \\
 B \otimes A & \xrightarrow{c_{B,A}} & A \otimes B
 \end{array}$$

Interaction of symmetry with unitors:

$$\begin{array}{ccc}
 A \otimes I & \xrightarrow{c_{A,I}} & I \otimes A \\
 \searrow u_A^R & & \swarrow u_A^L \\
 & A &
 \end{array}$$

Interaction of symmetry with associator:

$$\begin{array}{ccc}
 (A \otimes B) \otimes C & \xrightarrow{c_{A,B} \otimes 1_C} & (B \otimes A) \otimes C \\
 a_{A,B,C} \downarrow & & \downarrow a_{B,A,C} \\
 A \otimes (B \otimes C) & & B \otimes (A \otimes C) \\
 c_{A,B} \otimes 1_C \downarrow & & \downarrow 1_B \otimes c_{A,C} \\
 (C \otimes B) \otimes A & \xrightarrow{a_{A,B,C}} & B \otimes (C \otimes A)
 \end{array}$$

Maps from the tensor unit are called **points**.

Definition 2.2.2. A symmetric monoidal \mathbb{X} category is **symmetric monoidal closed** when there is a functor $-\circ - : \mathbb{X}^{\text{op}} \times \mathbb{X} \rightarrow \mathbb{X}$, such that for every object X of \mathbb{X} , there is an

adjunction $_ \otimes X \dashv X \multimap _$. The functor $_ \multimap _ : \mathbb{X}^{\text{op}} \times \mathbb{X} \rightarrow \mathbb{X}$ is called the **internal hom**, because it generalizes the hom functor $\mathbb{X}(-, -) : \mathbb{X} \times \mathbb{X} \rightarrow \mathbf{Set}$.

For example, \mathbf{Vect}_k is closed. Therefore, taking the dual vector space is a functor from $\mathbf{Vect}_k^{\text{op}}$ to \mathbf{Vect}_k .

2.2.1 Strict monoidal categories and PROPs

Definition 2.2.3. A **strict monoidal category** is a monoidal category in which all of the coherent natural isomorphisms are identity maps. Moreover, a **strict symmetric monoidal category** is a strict monoidal category with a symmetry. The symmetry is not required to be the identity, because this would force the category to be a monoid.

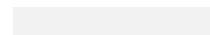
There is a succinct graphical representation of maps in strict (symmetric) monoidal categories. A map $X \xrightarrow{f} Y$ is graphically depicted as the following circuit:



The identity arrow on X is graphically depicted as a straight line:



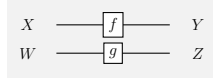
The identity arrow on I is depicted as a blank space:



The composite $X \xrightarrow{f} Y \xrightarrow{g} Z$ is graphically depicted by horizontal pasting:



The tensor $X \otimes W \xrightarrow{f \otimes g} Y \otimes Z$ is graphically depicted by vertical pasting:



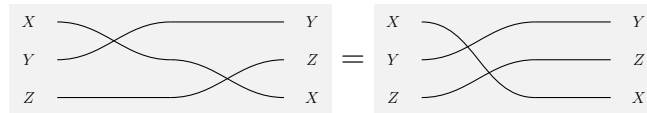
The object labels on the wires are often omitted. Notice how the grey background distinguishes tensored maps with maps merely pasted vertically on the same page:



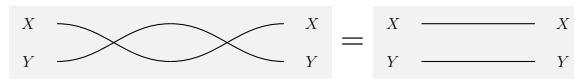
The symmetry $c_{X,Y}$ is graphically depicted as the following circuit:



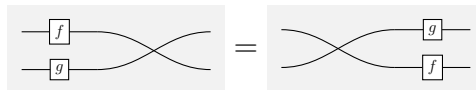
The coherences of a strict monoidal category are simple to draw. Because unitors and associators are identities, we only have to draw a few diagrams. First, the interaction of the symmetry with the associator:



As well as the inverse law:



The naturality of c can also be graphically depicted as follows:



Strict monoidal categories are called graphical calculi when the maps are semantically interpreted as computations. The ZX-calculus [17], the ZH-calculus [8], the ZW-calculus [29],

as well as CNOT (Chapter 5), CNOT + H (Chapter 6), TOF (Chapter 7) are all strict symmetric monoidal categories which are used to model computation in terms of generators and relations. There are also many variations of the ZX/W/H-calculi, [9, 51, 34, 22, 43, 6, 52, 50]. Lafont also gave complete sets of circuit identities for various fragments of classical computing; notably, he gave a complete set of identities for circuits generated by the controlled-not gate without ancillary bits [38, §3.1].

Definition 2.2.4. A **PROP** (short for **product** and **permutation** category) is a strict symmetric monoidal category where there is a fixed object X such that all the objects are of the form $X^{\otimes n}$ (where $X^{\otimes n}$ is the n -fold tensor product of X). CNOT, CNOT + H , TOF and the ZX/W/H-calculi are all PROPs.

2.2.2 Monoidal functors

This aforementioned assignment of a matrix to a circuit motivates the discussion of monoidal functors; translations between monoidal categories:

Definition 2.2.5. Given two monoidal categories \mathbb{X} and \mathbb{Y} , a **monoidal functor** from \mathbb{X} to \mathbb{Y} consists of functor $F : \mathbb{X} \rightarrow \mathbb{Y}$, a map $m_I : I \rightarrow F(I)$, and a natural transformation $m_{\otimes} : F(X) \otimes F(Y) \rightarrow F(X \otimes Y)$ satisfying the following conditions:

Associativity:

$$\begin{array}{ccc}
 (F(X) \otimes F(Y)) \otimes F(Z) & \xrightarrow{a} & F(X) \otimes (F(Y) \otimes F(Z)) \\
 m_{\otimes} \otimes 1 \downarrow & & 1 \otimes m_{\otimes} \downarrow \\
 F(X \otimes Y) \otimes F(Z) & & F(X) \otimes F(Y \otimes Z) \\
 m_{\otimes} \downarrow & & m_{\otimes} \downarrow \\
 F((X \otimes Y) \otimes Z) & \xrightarrow{F(a)} & F(X \otimes (Y \otimes Z))
 \end{array}$$

Unitality:

$$\begin{array}{ccc}
I \otimes F(X) & \xrightarrow{m_I \otimes 1} & F(I) \otimes F(X) & & F(X) \otimes I & \xrightarrow{1 \otimes m_I} & F(X) \otimes F(I) \\
u^L \downarrow & & \downarrow m_\otimes & & u^R \downarrow & & \downarrow m_\otimes \\
F(X) & \xleftarrow{F(u^L)} & F(I \otimes X) & & F(X) & \xleftarrow{F(u^R)} & F(X \otimes I)
\end{array}$$

A **symmetric monoidal functor** is a monoidal functor, so that the following diagram commutes:

$$\begin{array}{ccc}
F(X) \otimes F(Y) & \xrightarrow{c} & F(X) \otimes F(Y) \\
m_\otimes \downarrow & & \downarrow m_\otimes \\
F(X \otimes Y) & \xrightarrow{F(c)} & F(Y \otimes X)
\end{array}$$

A **strong monoidal functor** is a monoidal functor where m_I is an isomorphism and m_\otimes is a natural isomorphism. A **strict monoidal functor** is a monoidal functor where m_I and m_\otimes are identities. Strong symmetric monoidal functors and strict symmetric monoidal functors are defined in the obvious way.

We can view completeness, soundness and universality of a graphical calculus with respect to different properties of a map $\llbracket - \rrbracket : \mathbb{X} \rightarrow \mathbb{Y}$ interpreting \mathbb{X} in \mathbb{Y} . This observation was previously made for the strict symmetric monoidal case $\mathbb{X} \rightarrow \mathbf{Mat}_{\mathbb{C}}$ [7, §2.4]. An interpretation is **sound** if it preserves structure and, and thus, is a strong or strict symmetric monoidal functor. An interpretation is **complete** if it is faithful and surjective on objects. An interpretation is **universal** if it is full.

When \mathbb{X} is presented in terms of generators and relations, as in the ZX/W/H-calculus, CNOT or TOF etc., the soundness and fullness are mechanical to verify. Proving the completeness, on the other hand, is almost always nontrivial; for CNOT and TOF, this involves constructing normal forms. A normal form for circuits presented by generators and relations is a form from which the equality of any two circuits can be easily determined; along with an algorithm applying the equational identities to transform arbitrary circuits into the normal form. In CNOT, the normal form is constructed by performing Gaussian elimination. In TOF, the normal form is constructed by computing the expansions of multivariate

polynomials.

Chapter 3

Quantum Computing

The analysis of finite-dimensional quantum physics with category theory is known as *categorical quantum mechanics*. We will first discuss quantum computing abstractly in these terms and then restrict our attention to the stabilizer fragment of quantum mechanics which is needed for Chapter 6.

3.1 Categorical quantum computing

I refer the reader to [18] for a survey of categorical quantum mechanics, in which the definitions and results in this section are discussed in greater depth.

3.1.1 †-categories and compact closed categories

Quantum circuits can naturally be expressed in terms of monoidal categories. However, the broad structure of monoidal categories must be further refined to model interesting quantum phenomena:

Definition 3.1.1. A **†-category** (dagger-category) \mathbb{X} is a category equipped with a functor $(_)^\dagger : \mathbb{X}^{\text{op}} \rightarrow \mathbb{X}$ that is an identity on objects involution. That is, for all objects X and maps f , $X^\dagger = X$ and $((f)^\dagger)^\dagger = f$.

A map f in a dagger category is an **isometry** if $ff^\dagger = 1$ and **unitary** if $f^\dagger = f^{-1}$. On the other hand, f is **Hermetian** when $f^\dagger = f$.

Example 3.1.2. The category $\text{Mat}_{\mathbb{C}}$ of matrices over \mathbb{C} from Example 2.1.6 is a \dagger -category, where the dagger is given by the complex conjugate transpose.

A **\dagger -monoidal category** \mathbb{X} is simultaneously a \dagger -category and a monoidal category where the dagger distributes over the tensor (ie. $(X \otimes Y)^\dagger = X^\dagger \otimes Y^\dagger$) and all of the components of the given natural isomorphisms are unitary, i.e. $a_{A,B,C}^\dagger = a_{A,B,C}^{-1}$, $(u_A^R)^\dagger = (u_A^R)^{-1}$, $(u_A^L)^\dagger = (u_A^L)^{-1}$ and $c_{A,B}^\dagger = c_{A,B}^{-1}$.

A **\dagger -symmetric monoidal category** \mathbb{X} is simultaneously a \dagger -monoidal category and a symmetric monoidal category where the symmetry is also unitary.

A **\dagger -monoidal functor** between \dagger -monoidal categories is a monoidal functor that also preserves the \dagger -functor.

Example 3.1.3. $\text{Mat}_{\mathbb{C}}$ is not only a \dagger -category, it is a strict \dagger -symmetric monoidal category. The tensor product is given by the Kronecker product, and the tensor unit is the identity matrix on 1. Recall that the Kronecker product is defined as follows:

$$X \otimes Y = \begin{bmatrix} x_{1,1}Y & \cdots & x_{1,n}Y \\ \vdots & \ddots & \vdots \\ x_{m,1}Y & \cdots & x_{m,n}Y \end{bmatrix}$$

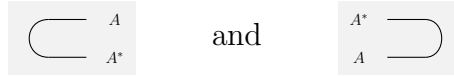
Monoidal categories can be refined so that wires can, not only cross over each other, but also be bent into cups and caps:

Definition 3.1.4. A symmetric monoidal closed category \mathbb{X} is **compact closed** when the canonical map $B \otimes A^* \rightarrow A \multimap B$ obtained by currying:

$$\begin{array}{c}
\frac{A \multimap I = A \multimap I}{(A \multimap I) \otimes A \rightarrow I} \multimap A \dashv A \multimap \\
\frac{B = B \quad \frac{A^* \otimes A \rightarrow I}{B \otimes (A^* \otimes A) \rightarrow B \otimes I} \otimes}{B \otimes (A^* \otimes A) \rightarrow B} u_B^R \\
\frac{B \otimes (A^* \otimes A) \rightarrow B}{(B \otimes A^*) \otimes A \rightarrow B} a_{B, A^*, A}^{-1} \\
\frac{(B \otimes A^*) \otimes A \rightarrow B}{B \otimes A^* \rightarrow A \multimap B} \multimap A \dashv A \multimap
\end{array}$$

is an isomorphism.

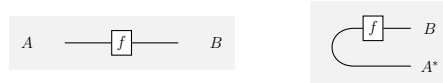
In the strict case, the evaluation and coevaluation maps for the components of the adjunction $\eta : I \rightarrow A \otimes A^*$ and $\varepsilon : A^* \otimes A \rightarrow I$ are graphically depicted by:



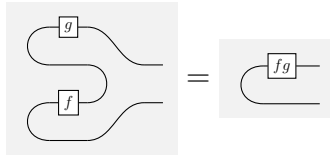
The triangle identities are thus graphically depicted as follows:



This means that processes $f : A \rightarrow B$ can be curried into states $I \rightarrow B \otimes A^*$:



Where states can be composed as follows:



In a compact closed category, the component at A of the natural isomorphism $\iota_A : A \rightarrow A^{**}$ is given by uncurrying 1_{A^*} ; graphically:



A **†-compact closed category** is a compact closed, †-symmetric-monoidal category for which $\eta_A^\dagger c_{A,A^*} = \varepsilon_A$ for all objects A that is:

$$\boxed{\text{Cup}}^\dagger = \boxed{\text{Cap}} \quad \text{so that, as a consequence} \quad \boxed{\text{Cap}}^\dagger = \boxed{\text{Cup}}$$

3.1.2 Hilbert spaces

Definition 3.1.5. A **Hilbert space** H is a vector space over \mathbb{C} equipped with an inner product $\langle \cdot | \cdot \rangle : H \times H \rightarrow \mathbb{C}$ such that H is also a complete metric space with respect to the distance induced by the inner product.

The **category of Hilbert spaces** \mathbf{Hilb} has:

Objects: Hilbert spaces.

Maps: Bounded linear maps.

Tensor: The tensor of products Hilbert spaces is metric space completion of the usual bilinear tensor product of vector spaces. The tensor unit is \mathbb{C} .

Dagger: The dagger $(\cdot)^\dagger : \mathbf{Hilb}^{\text{op}} \rightarrow \mathbf{Hilb}$ is the Hermitian adjoint; that is for all maps $H_1 \xrightarrow{f} H_2$, and all vectors h_1 in H_1 and h_2 in H_2 :

$$\langle f \circ h_1 | h_2 \rangle = \langle h_1 | f^\dagger \circ h_2 \rangle$$

Let \mathbf{FHilb} denote the full subcategory of \mathbf{Hilb} where the objects are finite dimensional Hilbert spaces.

The vectors (points) of a Hilbert space are denoted by $|\varphi\rangle$, and the linear functionals are denoted by $\langle\psi|$. This allows the inner product to be split into two parts, bras and kets, so that $\langle\varphi|\psi\rangle = \langle\varphi| \circ |\psi\rangle$. Likewise the outer product is thus written as $|\varphi\rangle \circ \langle\psi|$. *We must use the application order of composition when working in \mathbf{Hilb} , so that the inner product and outer product are not confused.*

The vectors of 2^n -dimensional Hilbert spaces are called **qubits**.

Given a Hilbert space H_1 with basis $\{b_i\}_{i \in I}$ and another Hilbert space H_2 with basis $\{e_j\}_{j \in J}$, any linear map from H_1 to H_2 can be expressed as the following sum, for some $a_{i,j} \in \mathbb{C}$:

$$\sum_{i \in I, j \in J} a_{i,j} |e_j\rangle \langle b_i|$$

This notation for matrices often makes calculations much easier. As a matter of convention, the computational basis (standard basis) of dimension n is denoted by the set $\{|i\rangle\}_{i \in [0, n-1]}$.

In general, when one speaks of a graphical calculus that is sound for a fragment of quantum computing, this posits the existence of a strict symmetric monoidal functor $\mathbb{X} \rightarrow \mathbf{Mat}_{\mathbb{C}}$. This is because $\mathbf{Mat}_{\mathbb{C}}$ is equivalent to \mathbf{FHilb} via the functor $\mathbf{Mat}_{\mathbb{C}} \rightarrow \mathbf{FHilb}$ which chooses a basis [5, Example 2.3.13].

\mathbf{FHilb} is a \dagger -compact closed category. For every finite dimensional Hilbert space H and basis $\{|b_i\rangle\}_{i \in I}$, the coevaluation is given by:

$$\varepsilon_H := \sum_{i \in I} |b_i\rangle \otimes \overline{|b_i\rangle}$$

It is important to note that the evaluation and coevaluation maps for the \dagger -compact closed structure of \mathbf{FHilb} are basis independent.

Lemma 3.1.6. The dimension of a Hilbert space is given by the circuit:



Proof. Given some (finite) N -dimensional Hilbert space H :

$$\begin{aligned}
\eta_H^\dagger \eta_H &= \left(\sum_{i=0}^{N-1} \langle i | \overline{\langle i |} \right) \circ \left(\sum_{i=0}^{N-1} |i\rangle \overline{|i\rangle} \right) \\
&= \sum_{i=0}^{N-1} \sum_{j=0}^i (\langle j | \overline{\langle j |} \circ (|i-j\rangle \overline{|i-j\rangle})) \\
&= \sum_{i=0}^{N-1} \sum_{j=0}^i \langle j | i-j \rangle \overline{\langle j | i-j \rangle} \\
&= \sum_{i=0}^{N-1} \sum_{j=0}^i \delta_{i,i-j} \\
&= \sum_{i=0}^{N-1} 1 \\
&= N
\end{aligned}$$

□

Definition 3.1.7. A **quantum observable** is a Hermetian operator. Given an eigenvector $h|v\rangle = \lambda|v\rangle$ of a Hermetian h , measuring a state $\langle w|$ in the normalized eigenbasis of h produces $|v\rangle$ with probability $|\langle w|v\rangle|^2$.

A vector $|\varphi\rangle$ in a Hilbert space is a **quantum state** if it has unit norm, that is, $|\langle\varphi|\varphi\rangle|^2 = 1$. The coevaluation is not a quantum state as it is unnormalized. However, the normalized state is the **maximally mixed state**.

Definition 3.1.8. Two observables on the same space with respective eigenbases $|a_i\rangle$ and $|b_i\rangle$ are said to be **mutually unbiased** if for all indices i, j and k :

$$|\langle a_i | b_k \rangle|^2 = |\langle a_j | b_k \rangle|^2$$

and

$$|\langle b_i | a_k \rangle|^2 = |\langle b_j | a_k \rangle|^2$$

That is to say, that the measurement of one observable reveals no information about what would have happened if the other observable were measured, and vice versa. In a Hilbert space with finite dimension N , because the probability is uniformly random, we have:

$$|\langle a_i | b_k \rangle|^2 = |\langle b_i | a_k \rangle|^2 = 1/N$$

Example 3.1.9. Consider the Pauli X and Z matrices:

$$X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

These matrices are Hermitian, where the X observable has normalized eigenbasis:

$$|x_+\rangle := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad |x_-\rangle := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

and the Z eigenbasis is the computational basis:

$$|z_+\rangle := \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |z_-\rangle := \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

It is mechanical to check that the Pauli X and Z bases are mutually unbiased. It is also worth noting that the Z and X observables are related via the change of basis matrix called the Hadamard gate:

$$H|x_\pm\rangle = |z_\pm\rangle \quad H|z_\pm\rangle = |x_\pm\rangle$$

where:

$$H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The Hadamard gate is the 1-qubit quantum Fourier transform:

Definition 3.1.10. The n -qubit **Fourier transform** is given by the matrix:

$$\mathcal{F}_{2^n} := \sum_{j,k=0}^{2^n-1} \omega_k^j |j\rangle\langle k|$$

Where ω_k is the 2^n th root of unity $e^{-2\pi ik/2^n}$.

The Hadamard gate is the 1-qubit quantum Fourier transform.

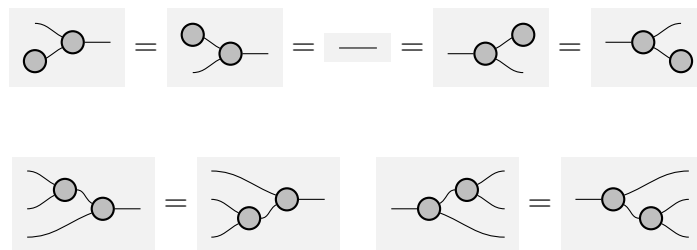
3.1.3 Frobenius algebras and complementarity

Structures such as Frobenius algebras, bialgebras and Hopf algebras generalize much of the structure of \mathbf{FHilb} to arbitrary \dagger -compact closed categories. We draw the components of Frobenius algebras by shaded circles:

Definition 3.1.11. A **Frobenius algebra** in a monoidal category is a 5-tuple:

$$(A, \begin{array}{|c|} \hline \text{---} \circ \text{---} \\ \hline \end{array}, \begin{array}{|c|} \hline \circ \text{---} \\ \hline \end{array}, \begin{array}{|c|} \hline \text{---} \circ \text{---} \\ \hline \end{array}, \begin{array}{|c|} \hline \text{---} \circ \text{---} \\ \hline \end{array})$$

such that $(A, \begin{array}{|c|} \hline \text{---} \circ \text{---} \\ \hline \end{array}, \begin{array}{|c|} \hline \circ \text{---} \\ \hline \end{array})$ is a monoid and $(A, \begin{array}{|c|} \hline \text{---} \circ \text{---} \\ \hline \end{array}, \begin{array}{|c|} \hline \text{---} \circ \text{---} \\ \hline \end{array})$ is a comonoid:

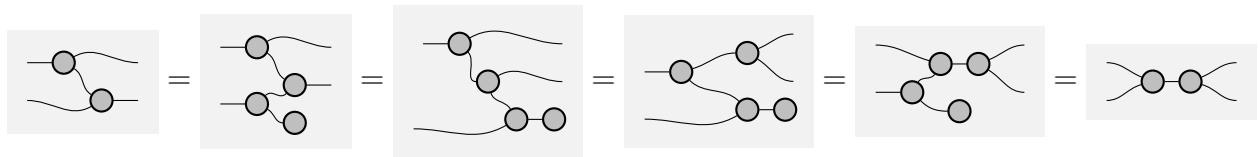


And the Frobenius law holds:

[F]

The middle term in the equality is redundant, although we consider non-unital Frobenius algebras in Chapter 4 as it follows from the unitality/counitality of the monoid and comonoid

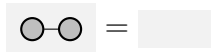
and the equality of the first and second terms:



A Frobenius algebra is **special** if



extra if



and **commutative** if the underlying monoid and comonoids are commutative and cocommutative:



A **†-Frobenius algebra** $(A, \triangleright, \circlearrowleft)$ is a Frobenius algebra of the form

$$(A, \triangleright, \circlearrowleft, \triangleright^\dagger, \circlearrowleft^\dagger)$$

That is to say, the monoid and comonoid are daggers of each other.

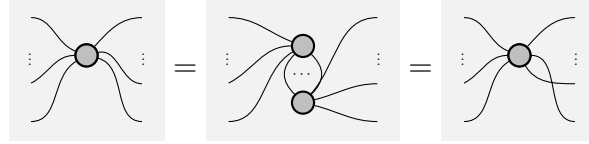
A non-(co)unital special commutative \dagger -Frobenius algebra is called a **semi-Frobenius algebra**. Semi-Frobenius algebras are used to construct a weak product structure for inverse categories which we will explore in Chapter 4. Special, commutative, \dagger -Frobenius algebras are called **classical structures**.

The following theorem is useful for manipulation of Frobenius algebras:

Theorem 3.1.12. [2] [**Commutative spider Theorem**] All connected sub-graphs consisting of the components of a classical structure with the same type (same domain and

codomain) have a unique normal form.

In graphical notation, that means that we can represent the connected components of a Frobenius algebra by a spider. The multiplication is the spider of type $X \times X \otimes X$, the unit is the spider of type $I \rightarrow X$ and so on. When any two spiders are connected, they merge into a larger spider. Graphically:



If there is a Frobenius algebra on an object X , an isomorphism $X \rightarrow Y$ induces a Frobenius algebra on Y :

Lemma 3.1.13. If $h : X \rightarrow Y$ is an isomorphism and

$$(X, \text{multiplication}, \text{unit}, \text{comultiplication}, \text{counit})$$

is a Frobenius algebra, then so is

$$\left(Y, \text{multiplication} \circ h, \text{unit} \circ h, \text{comultiplication} \circ h^{-1}, \text{counit} \circ h^{-1} \right)$$

And in particular, if h is unitary and the Frobenius algebra is a \dagger -Frobenius algebra, the induced Frobenius algebra is a \dagger -Frobenius algebra.

Proof. Observe that $h^{-1}h$ and $h^{-1}h$ cancel; where the identities of a Frobenius algebra can be applied. □

Moreover,

Lemma 3.1.14. [31, Theorem 5.15] If there is a Frobenius algebra on X , then X is self dual.

Proof. Consider a Frobenius algebra $(X, \nabla, e, \Delta, m)$

The evaluation map is given by $X \otimes X \xrightarrow{\nabla} X \xrightarrow{m} I$ and the coevaluation map is given by $I \xrightarrow{e} X \xrightarrow{\Delta} X \otimes X$. □

We can say more about Frobenius algebras in **FHilb**:

Theorem 3.1.15. [19, Theorem 5.1] Orthogonal bases $\{|b_i\rangle\}_{i \in I}$ in **FHilb** are in bijective correspondence with commutative \dagger -Frobenius algebras:

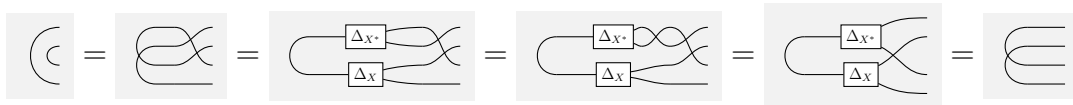
$$\left(\Delta := \sum_{i \in I} |b_i\rangle \langle b_i|, e := \sum_{i \in I} |b_i\rangle \right)$$

Moreover, this correspondence can be refined to that of *special* commutative \dagger -Frobenius algebras (classical structures) and *orthonormal* bases.

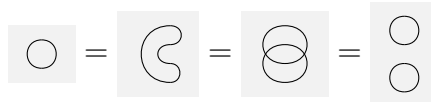
The elements of a basis are copied by the induced special commutative \dagger -Frobenius algebra. However, not all points are copied in any category with a chosen classical structure, the coevaluation cannot be copied. A graphical proof of this well known result is given in [18], which we give here:

Theorem 3.1.16. [No cloning] Quantum channels cannot be copied.

Proof. Suppose for sake of contradiction that quantum channels can be copied. Then there is a commutative natural transformation Δ so that:



This implies that



□


Since every orthonormal basis in **FHilb** induces a classical structure and every vector space has a basis, **FHilb** is \dagger -compact closed.

Mutually unbiased bases can also be generalized beyond \mathbf{FHilb} . We use different colours for the monoid and comonoid to indicate that they do not necessarily form a Frobenius algebra.

Definition 3.1.17. A **bialgebra** is a monoid, comonoid pair

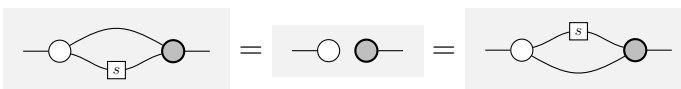
$$(X, \text{---} \circlearrowleft \text{---}, \text{---} \circlearrowright \text{---}, \text{---} \circlearrowright \text{---}, \text{---} \circlearrowleft \text{---})$$

so that:

[B.U] 

[B.M] 

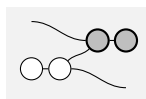
A bialgebra is a **Hopf-algebra** when there is a map $X \xrightarrow{s} X$ called the **antipode**, such that:

[B.H] 

Definition 3.1.18. Two classical structures on the same object

$$(X, \text{---} \circlearrowleft \text{---}, \text{---} \circlearrowright \text{---}, \text{---} \circlearrowright \text{---}, \text{---} \circlearrowleft \text{---}) \quad \text{and} \quad (X, \text{---} \circlearrowleft \text{---}, \text{---} \circlearrowleft \text{---}, \text{---} \circlearrowright \text{---}, \text{---} \circlearrowleft \text{---})$$

are **complementary**, when they interact satisfying the Hopf law up to an invertible scalar. Complementary Frobenius algebras are **strongly complementary** when they also satisfy the bialgebra laws up to an invertible scalar, so that they form a Hopf algebra up to an invertible scalar with the following antipode [17]:



Theorem 3.1.19. [31, Proposition 5.12] In \mathbf{FHilb} mutually unbiased bases are complementary classical structures.

The Pauli Z and X observables are strongly complementary. It was this observation that led to the development of the ZX-calculus [17], which we discuss in Chapter 6.

The Hopf law should be interpreted as the result of preparing a state in the white basis and then measuring in the black basis; where the distribution of measurement outcomes is uniform. The reason that the bialgebra/Hopf rule only holds up to a scalar, is because in \mathbf{FHilb} , mutually unbiased bases correspond to Frobenius algebras interacting to satisfy the Hopf law up to a nonzero scalar. In particular, they are scaled with respect to the dimension of the underlying vector space.

Even more of the structure of finite-dimensional quantum mechanics can be abstractly characterized:

Definition 3.1.20. [21, Definition 4.3] A **prephase-shift** for a \dagger -Frobenius algebra:

$$(X, \text{---}\bigcirc\text{---}, \bigcirc\text{---}, \text{---}\bigcirc\text{---}, \bigcirc\text{---})$$

is an endomorphism $\alpha : X \rightarrow X$ so that:

$$\text{---}\bigcirc\text{---} \stackrel{\alpha}{=} \text{---}\bigcirc\text{---} \stackrel{\alpha}{=} \text{---}\bigcirc\text{---}$$

Unitary prephase-shifts are called **phase-shifts**

A (pre)phase-shift $\alpha : X \rightarrow X$ induces a (pre)phase:

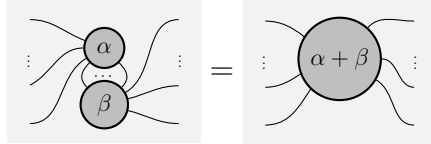
$$\text{---}\square\bigcirc\text{---}$$

The phase-shifts of a \dagger -Frobenius algebra form a group called **the phase group** with group multiplication given by the Frobenius multiplication, the unit given by the identity and the inverse given taking the dagger (as phases are unitary).

The commutative spider theorem can be further refined to account for phases:

Theorem 3.1.21. [18, Theorem 9.15] [**Phased commutative spider Theorem**] All connected sub-graphs consisting of the components of a classical structure with the same type (same domain and codomain) have a unique normal form.

That is to say, we can augment spiders to be labeled by phases so that the following spider fusion law holds:



The label $\alpha + \beta$ denotes the phase group product of phases α and β . Note that the original spider fusion rule is just the case when α and β are the identity for the phase group¹.

3.2 Stabilizer quantum mechanics

In this section, I refer the reader to [7, §3.3] or [44, §10.5] for further details.

The following sets of gates are known to be relatively inexpensive to implement and are of particular interest in quantum error correction [25, 44]:

Definition 3.2.1. The **Pauli matrices** are the complex matrices:

$$X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y := \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

The **Pauli group** on n is the closure of the set:

$$P_n := \{\lambda a_1 \otimes \cdots \otimes a_n \mid \lambda \in \{\pm 1, \pm i\}, a_i \in \{I_2, X, Y, Z\}\}$$

under matrix multiplication.

¹This can cause confusion: in \mathbf{FHilb} , the identity for the phase group is denoted by 0 , corresponding to the phase $e^{i0} = 1$, which is the identity for composition, hence additive notation is used.

Definition 3.2.2. A unitary U **stabilizes** a quantum state $|\varphi\rangle$ when $|\varphi\rangle$ is a +1 eigenvector of U . That is to say $U|\varphi\rangle = |\varphi\rangle$. The unitaries stabilizing a quantum state $|\varphi\rangle$ form a group called the **stabilizer group of** $|\varphi\rangle$ denoted by $S_{|\varphi\rangle}$.

Because of the importance of Pauli products in quantum error correction, it is important to identify what states are stabilized by them:

Definition 3.2.3. An n qubit quantum state $|\varphi\rangle$ is called a **stabilizer state** in case $S_{|\varphi\rangle}$ is a subgroup of P_n .

This gives an alternative way to think about stabilizer states:

Lemma 3.2.4. [44, §10.5.1] Stabilizer states are uniquely determined by their stabilizer subgroups (up to an invertible scalar).

The following class of maps preserve the property of being a stabilizer state:

Definition 3.2.5. The **Clifford group** on n is the group of operators which acts on the Pauli group on n by conjugation (where $U(2^n)$ is the group of 2^n dimensional unitaries in $\text{Mat}_{\mathbb{C}}$):

$$C_n := \{U \in U(2^n) | \forall p \in P_n, UpU^{-1} \in P_n\}$$

There is an algebraic description of stabilizer states:

Lemma 3.2.6. [7, Theorem 3.3.6] All n qubit stabilizer states have the form $C|0\rangle^{\otimes n}$, for some member C of the Clifford group on n qubits.

This characterization of stabilizer states in terms of Clifford operators leads to the following notion:

Definition 3.2.7. A **stabilizer circuit** is a circuit composed of Clifford operators, state preparation and postselected measurement in the computational basis.

We also consider a subgroup of C_n :

Definition 3.2.8. [22] The **real Clifford group** on n qubits, is the subgroup of the Clifford group with real elements, ie:

$$C_n^{re} := \{U \in C_n | \bar{U} = U\}$$

An n -qubit **real stabilizer state** is a state of the form $C|0\rangle^{\otimes n}$ for some $C \in C_n^{re}$.

Similarly, a **real stabilizer circuit** is a circuit composed of real Clifford operators, state preparation and postselected measurement in the computational basis.

There is a generating set for the Clifford group on n qubits.

Theorem 3.2.9. [25] For $n > 1$, the Clifford group on n is generated by ω (global $\pi/4$ phase), S ($\pi/2$ phase-shift gate), H (Hadamard gate) and **cnot** (controlled-not gate). Explicitly, these matrices are:

$$\omega := e^{i\pi/4} \quad S := \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{cnot} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

It is worth mentioning that Selinger gave a complete set of circuit relations for Cliffords [47].

Corollary 3.2.10. [22] For $n > 1$ the real Clifford group is generated by the Hadamard gate, the controlled-not gate and the not gate.

In [22], this group is presented in terms of the Hadamard gate, the controlled- Z gate and the Z gate, but these two presentations are equivalent.

Although these circuits exhibit quantum entanglement; they are not computationally powerful:

Theorem 3.2.11. [26] [**Gottesmann-Knill**] Stabilizer circuits can be simulated on probabilistic a classical computer in polynomial time.

Of course, quantum circuits cannot, in general, be simulated on classical computers in polynomial time. This means that stabilizer circuits are a very restrictive class of quantum circuits.

Chapter 4

Restriction and Inverse Categories

In this chapter, we shall turn our attention from quantum computing to partial, and reversible computing.

4.1 Restriction and inverse categories

In this section, we recall the basic theory and terminology of restriction and inverse categories.

Definition 4.1.1. [16, Definition 2.1.1] A **restriction structure** on a category \mathbb{X} is an assignment $\bar{f} : A \rightarrow A$ for each map $f : A \rightarrow B$ in \mathbb{X} satisfying the following four axioms:

$$\text{[R.1]} \quad \bar{f}f = f \qquad \text{[R.2]} \quad \bar{f}\bar{g} = \bar{g}\bar{f} \qquad \text{[R.3]} \quad \overline{\bar{f}} = \bar{f} \qquad \text{[R.4]} \quad f\bar{g} = \bar{f}gf$$

A **restriction category** is a category equipped with a restriction structure. An endomorphism $e : A \rightarrow A$ is called a **restriction idempotent** if $e = \bar{e}$. In particular, for any map f , $\bar{f}\bar{f} = \overline{\bar{f}f} = \bar{f}$, so that \bar{f} is idempotent. Restriction categories have a partial order on homsets with $f \leq g$ if and only if $\bar{f}g = f$. This partial order is called the **restriction order**.

In a restriction category, a **total map** is a map f such that $\bar{f} = 1$. The total maps of a restriction category \mathbb{X} form a subcategory $\text{Total}(\mathbb{X})$ of \mathbb{X} .

Example 4.1.2. The canonical example of a restriction category is the category of sets and partial functions, Par . The restriction of a partial function $f : A \rightarrow B$ is given by:

$$\overline{f}(x) := \begin{cases} x & \text{If } f(x) \downarrow \\ \uparrow & \text{Otherwise} \end{cases}$$

The notation $f(x) = \uparrow$ means that f is undefined at x ; similarly, $f(x) \downarrow$ means that f is undefined at x .

Since the assignment of restriction is structure, one needs a notion of a functor between restriction categories that preserves this structure:

Definition 4.1.3. A **restriction functor** $F : \mathbb{X} \rightarrow \mathbb{Y}$ between restriction categories is a functor so that for any map $f \in \mathbb{X}$, $F(\overline{f}) = \overline{F(f)}$.

Definition 4.1.4. [16, Sec. 3] A **stable system of monics** \mathcal{M} for a category \mathbb{X} is a class of monics of \mathbb{X} closed to composition, containing all isomorphisms, so that given any cospan $X \xrightarrow{f} Z \xleftarrow{m} Y$, where m is in \mathcal{M} , the following pullback exists:

$$\begin{array}{ccc} X_f \times_m Y & \xrightarrow{\pi_1} & Y \\ \pi_0 \downarrow & \lrcorner & \downarrow m \\ X & \xrightarrow{f} & Z \end{array}$$

and π_0 is in \mathcal{M} .

Definition 4.1.5. [16, Sec. 3] Given a stable system of monics \mathcal{M} of a category \mathbb{X} , **the category of partial maps**, $\text{Par}(\mathbb{X}, \mathcal{M})$ is the subcategory of $\text{Span}(\mathbb{X})$ where the left leg is in \mathcal{M} . $\text{Par}(\mathbb{X}, \mathcal{M})$ is endowed with a restriction structure given by $\overline{(m, f)} := (m, m)$.

When all of the monics of \mathbb{X} form a stable system of monics, denote the category of partial maps with all monics by $\text{Par}(\mathbb{X})$.

For example, $\text{Par}(\text{Set})$ is equivalent to Par .

Definition 4.1.6. [16, Sec. 2.3]. A map f in a restriction category is a **partial isomorphism** when there exists another map g , called the **partial inverse** of f , such that $\bar{f} = fg$ and $\bar{g} = gf$. A restriction category \mathbb{X} is an **inverse category** when all its maps are partial isomorphisms.

In an inverse category, every idempotent is a restriction idempotent. Partial isomorphisms generalize the notion of isomorphisms to restriction categories; thus, the composition of partial isomorphisms is a partial isomorphism and partial inverses are unique. Furthermore, every restriction category \mathbb{X} has a subcategory of partial isomorphisms $\text{ParIso}(\mathbb{X})$ which is an inverse category. The category $\text{ParIso}(\text{Set})$ is denoted by Pinj . The full subcategory of Pinj with objects given by finite sets is denoted by FPinj .

Inverse categories can be equivalently characterized by having a special type of dagger structure:

Theorem 4.1.7. [16, Theorem 2.20] A category \mathbb{X} is an inverse category if and only if there exists an functor $(-)^{\circ} : \mathbb{X}^{\text{op}} \rightarrow \mathbb{X}$ which is the identity on objects, satisfying the following three axioms:

$$[\text{INV.1}] (f^{\circ})^{\circ} = f \qquad [\text{INV.2}] ff^{\circ}f = f \qquad [\text{INV.3}] ff^{\circ}gg^{\circ} = gg^{\circ}ff^{\circ}$$

The restriction structure is given by $\bar{c} := cc^{\circ}$.

4.1.1 Discrete restriction categories and inverse products

If a category \mathbb{X} has products then $\text{Par}(\mathbb{X})$ has restriction products: these are “lax” products for which the pairing operation satisfies $\langle f, g \rangle \pi_0 = \bar{g}f$ (and $\langle f, g \rangle \pi_1 = \bar{f}g$). If the category \mathbb{X} has a final object then $\text{Par}(\mathbb{X})$ has a restriction final object. That is an object 1 for which, for each object A , there is a unique total map $! : A \rightarrow 1$ so that for any map $k : A \rightarrow B$, $k!_B = \bar{k}!_A$. A restriction category with restriction products and a restriction terminal object is called a **Cartesian restriction category**.

A Cartesian restriction category in which the diagonal map $\Delta_A : A \rightarrow A \times A$ is a partial isomorphism is called a **discrete Cartesian restriction category**. Given a category with products and pullbacks, the category of partial maps, $\text{Par}(\mathbb{X})$, is always a discrete Cartesian restriction category. This is because the diagonal map is monic, and thus a partial isomorphism. Discrete Cartesian restriction categories can be equivalently characterized as Cartesian restriction categories which have meets with respect to the restriction ordering:

Definition 4.1.8. [15, Definition 2.9] A restriction category \mathbb{X} has **restriction meets** when it has a combinator $_ \cap _ : \mathbb{X}(A, B) \times \mathbb{X}(A, B) \rightarrow \mathbb{X}(A, B)$ for all objects A and B in \mathbb{X} such that for all $f, g : A \rightarrow B$ and $h : B \rightarrow C$ in \mathbb{X} the following axioms hold:

$$[\mathbf{M.1}] \quad f \cap f = f$$

$$[\mathbf{M.2}] \quad f \cap g \leq f \text{ and } f \cap g \leq g \quad (\text{with respect to the restriction ordering})$$

$$[\mathbf{M.3}] \quad (f \cap g)h = (fh) \cap (gh)$$

In a discrete Cartesian restriction category the meet is $f \cap g := \Delta(f \times g)\Delta^\circ$. Conversely if one has meets in a Cartesian restriction category, one can define the partial inverse of the diagonal map as $\Delta^\circ := \pi_0 \cap \pi_1$. The subcategory of partial isomorphism of a discrete Cartesian category is an inverse category which has a residual product structure [24, Proposition 4.3.7]. As the projections are not partial isomorphisms this structure hinges on the behaviour of Δ and Δ° :

Definition 4.1.9. [24, Definition 4.3.1] Consider a symmetric monoidal inverse category \mathbb{X} where the tensor preserves restriction in both components. Such a category \mathbb{X} has **inverse products** if there exists a natural diagonal transformation Δ whose components induce semi-Frobenius algebras with respect to the dagger functor $(_)^\circ : \mathbb{X}^{\text{op}} \rightarrow \mathbb{X}$, satisfying uniform copying. Δ is **uniform copying**¹ if:

¹In [24, Definition 4.3.1], there is no requirement that $\Delta_I = (u_I^R)^{-1} = (u_I^L)^{-1}$.

$$\Delta_I^\circ = u_I^R = u_I^L \qquad \Delta_I = (u_I^R)^{-1} = (u_I^L)^{-1}$$

Recall from Definition 3.1.11, that a semi-Frobenius algebra is a nonunital, special, commutative, \dagger -Frobenius algebra. Therefore, the speciality makes the map Δ total on all components.

Denote the partial inverse of Δ by $\nabla := \Delta^\circ$. We depict Δ and ∇ graphically as follows:



A **discrete inverse category** is a category with inverse products. Discrete inverse categories have meets given by $f \cap g := \Delta(f \otimes g)\nabla$. Inverse products are extra structure, so an inverse category can be a discrete inverse category in different ways (see [24, Example 4.3.4]). Therefore, we need functors between discrete inverse categories to preserve more than just restriction.

Definition 4.1.10. A **discrete inverse functor** from a discrete inverse category \mathbb{X} to \mathbb{Y} is a strong monoidal restriction functor $F : \mathbb{X} \rightarrow \mathbb{Y}$ so that the following diagram commutes:

$$\begin{array}{ccc} F(X) & \xrightarrow{\Delta} & F(X) \otimes F(X) \\ & \searrow F(\Delta) & \downarrow m_\otimes \\ & & F(X \otimes X) \end{array}$$

Example 4.1.11. [24, Example 4.3.2] Pinj is a discrete inverse category with Cartesian monoidal structure. The dagger is given by the relational converse and the inverse products are given by components:

$$\Delta_X = \lambda x.(x, x) : X \rightarrow X \times X$$

We identify a class of maps with restriction idempotents using inverse products:

Definition 4.1.12. Consider a \dagger -monoidal category \mathbb{X} with a natural semi-Frobenius structure, without necessarily being an inverse category. A map $e : X \rightarrow X$ in \mathbb{X} is **latchable** when

$$e = \Delta_X(f \otimes 1_X)\nabla_X$$

That is, when the following holds:

The following Lemma was proven, in slightly different terms, by Giles in [24, Lemma 4.3.5 (i)]:

Lemma 4.1.13. If \mathbb{X} is a \dagger -monoidal category with a natural semi-Frobenius algebra structure satisfying uniform copying (without necessarily being an inverse category), then the latchable maps commute and are idempotent.

Moreover, if \mathbb{X} is a discrete inverse category, the latchable maps are precisely the restriction idempotents.

Proof.

- Consider a \dagger -monoidal category \mathbb{X} with a natural semi-Frobenius structure satisfying uniform copying. First we observe that latchable maps e in \mathbb{X} are idempotent.

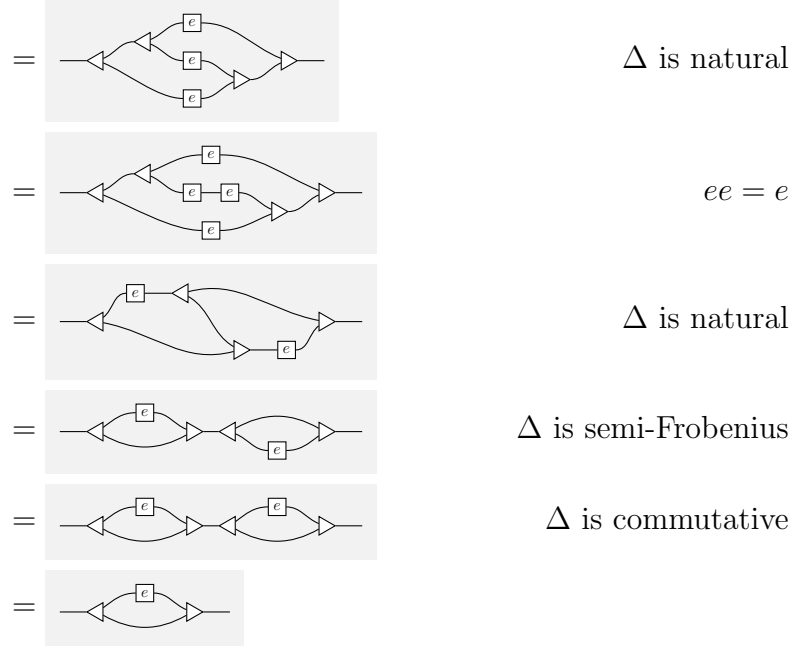
$$\begin{aligned}
&= \text{[Diagram: A box labeled } e \text{ with two parallel lines entering from the left and exiting to the right, with a loop above and below the box.]} && e \text{ is latching} \\
&= \text{[Diagram: A box labeled } e \text{ with a single line entering from the left, a loop above and below, and a single line exiting to the right.]} && \Delta \text{ is natural} \\
&= \text{[Diagram: A box labeled } e \text{ with a single line entering from the left and exiting to the right.]} && \Delta \text{ is special}
\end{aligned}$$

Next, observe that latching maps e and m and \mathbb{X} commute:

$$\begin{aligned}
\text{[Diagram: A box labeled } e \text{ followed by a box labeled } m \text{ on a single line.]} &= \text{[Diagram: A box labeled } e \text{ with a loop above and below, followed by a box labeled } m \text{ with a loop above and below.]} && e \text{ is latching} \\
&= \text{[Diagram: A box labeled } e \text{ with a loop above and below, followed by a box labeled } m \text{ with a loop below and above.]} && \Delta \text{ is commutative} \\
&= \text{[Diagram: A box labeled } e \text{ with a loop above and below, followed by a box labeled } m \text{ with a loop below and above, with lines crossing.]} && \Delta \text{ is semi-Frobenius} \\
&= \text{[Diagram: A box labeled } e \text{ with a loop above and below, followed by a box labeled } m \text{ with a loop above and below, with lines crossing.]} && \Delta \text{ is associative} \\
&= \text{[Diagram: A box labeled } e \text{ with a loop above and below, followed by a box labeled } m \text{ with a loop below and above.]} && e \text{ is latching} \\
&= \text{[Diagram: A box labeled } m \text{ with a loop above and below, followed by a box labeled } e \text{ with a loop below and above.]} && \Delta \text{ is commutative} \\
&= \text{[Diagram: A box labeled } m \text{ followed by a box labeled } e \text{ on a single line.]} && \text{By symmetry}
\end{aligned}$$

- For the second claim, consider a discrete inverse category \mathbb{X} . By the previous bullet we already know that latching maps are idempotent. We show, conversely, that idempotents e of \mathbb{X} are latching:

$$\begin{aligned}
\text{[Diagram: A box labeled } e \text{ with a single line entering from the left and exiting to the right.]} &= \text{[Diagram: A box labeled } e \text{ with a loop above and below, followed by a box labeled } e \text{ with a loop above and below.]} && \Delta \text{ is special} \\
&= \text{[Diagram: A box labeled } e \text{ with a loop above and below, followed by a box labeled } e \text{ with a loop below and above.]} && \Delta \text{ is semi-Frobenius}
\end{aligned}$$



Where the last line follows as latching maps are idempotent.

□

4.2 Restriction and discrete inverse functors

In Chapters 5 and 7, we demonstrate equivalences between discrete inverse categories using restriction functors which are representable. The pursuit of proving that these functors are equivalences resulted in the realization that the preservation of discrete inverse structure is very powerful property for a functor to have: in particular, the proof of the fullness and faithfulness of these functors is reduced to much easier cases (Lemmas 4.2.6 and 4.2.7).

Definition 4.2.1. Given a restriction category \mathbb{X} and any object X in \mathbb{X} , define the functor $h_X := \text{Total}(\mathbb{X})(X, -) : \mathbb{X} \rightarrow \text{Par}$ as follows:

On objects: For each object $Y \in \mathbb{X}$, $h_X(Y) := \{f \in \mathbb{X}(X, Y) \mid \bar{f} = 1_x\}$;

On maps: For each map $Y \xrightarrow{f} Z$ in \mathbb{X} , for all $g \in h_X(Y)$,

$$(h_X(f))(g) := \begin{cases} gf & \text{if } \overline{gf} = 1_X \\ \uparrow & \text{otherwise} \end{cases}$$

The notation $(h_X(f))(g) = \uparrow$ means $h_X(f)$ is undefined on g .

Lemma 4.2.2. $h_X : \mathbb{X} \rightarrow \text{Par}$ is a restriction functor.

Proof. To prove h_X is a restriction functor is to prove h_X preserves identities, composition and restriction structure.

- First, we prove that h_X preserves identities. Take any object $Y \in \mathbb{X}$ and any map $f \in h_X(Y)$. Then, $(h_X(1_Y))(f) = f1_Y = f$ as $\overline{f1_Y} = \overline{f} = 1_X$.
- Next we prove that h_X preserves composition. Consider arbitrary maps $Y \xrightarrow{f} Z \xrightarrow{g} W$ and an $h \in h_X(Y)$.

Suppose that $\overline{hfg} = 1_X$, then $(h_X(fg))(h) = hfg$ and $\overline{hf} = 1_X$, then

$$h_X(g)((h_X(f))(h)) = (h_X(g))(hf) = hfg.$$

On the other hand, suppose that $\overline{hfg} \neq 1_X$ then

$$h_X(g)((h_X(f))(h)) = h_X(g)(\uparrow) = \uparrow$$

If $\overline{hf} = 1_X$, then

$$h_X(g)(h_X(f)(h)) = h_X(g)(hf) = \uparrow.$$

Therefore, h_X preserves composition.

- Finally, we prove that h_X preserves restriction.

For any $f : Y \rightarrow Z$ and any $g \in h_X(Y)$:

$$(h_X(\bar{f}))(g) = \begin{cases} g\bar{f} & \text{if } \overline{g\bar{f}} = \overline{g\bar{f}} = 1_X \\ \uparrow & \text{otherwise} \end{cases}$$

However,

$$\overline{h_X(f)}(g) = \begin{cases} g & \text{if } (h_X(f))(g) \downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

But, $(h_X(f))(g)$ is defined if and only if $\overline{g\bar{f}} = 1_X$; moreover if $\overline{g\bar{f}} = 1_X$, then $g\bar{f} = \overline{g\bar{f}}g = 1_Xg = g$. Therefore, h_X preserves restriction.

□

Because $h_X : \mathbb{X} \rightarrow \mathbf{Par}$ is a restriction functor, and restriction functors preserve partial isomorphisms, it follows that:

Lemma 4.2.3. Given an object X of an inverse category \mathbb{X} , $h_X : \mathbb{X} \rightarrow \mathbf{Par}$ restricts to a \dagger -functor $\tilde{h}_X : \mathbb{X} \rightarrow \mathbf{Pinj}$.

Note that \tilde{h}_X is a \dagger -functor because the \dagger -structures of both categories are determined by their restriction structures, where restriction is preserved by \tilde{h}_X .

This functor sometimes preserves inverse products:

Lemma 4.2.4. Given a discrete inverse category \mathbb{X} , if $\tilde{h}_I : \mathbb{X} \rightarrow \mathbf{Pinj}$ is a strong monoidal functor, it is also a discrete inverse functor (where I is the tensor unit).

Proof. Recall that \mathbf{Pinj} has a canonical inverse product structure given by:

$$\Delta_X := \lambda x.(x, x) : X \rightarrow X \times X$$

so that for any $f : X \rightarrow Y$ in Pinj , $f\Delta_X = \Delta_Y(f \times f)$.

First, recall from the definition of a semi-Frobenius structure, that $\Delta_I = (u_I^R)^{-1} = (u_I^L)^{-1}$. Moreover, as \tilde{h}_I is a strong monoidal functor, $\text{Total}(\mathbb{X})(I, I) = \tilde{h}_I(I) \cong I$, so there is only one total endomorphism on the tensor unit in \mathbb{X} ; namely, 1_I .

Consider some f in $\tilde{h}_I(X)$. Then:

$$\begin{array}{ll}
f\tilde{h}_I(\Delta_X) = \tilde{h}_I(f)\tilde{h}_I(\Delta_X) & \text{As } \tilde{h}_I(I) = \{1_I\} \\
= \tilde{h}_I(f\Delta_X) & \tilde{h}_I \text{ is a functor} \\
= \tilde{h}_I(\Delta_I(f \otimes f)) & \Delta \text{ is natural} \\
= \tilde{h}_I(\Delta_I)\tilde{h}_I(f \otimes f) & \tilde{h}_I \text{ is a functor} \\
= \tilde{h}_I(u_I^R)\tilde{h}_I(f \otimes f) & \Delta_I = u_I^R \\
\cong u_I^R(f \times f) & \tilde{h}_I \text{ is a strong monoidal functor} \\
= \Delta_I(f \times f) & \Delta_I = u_I^R
\end{array}$$

Therefore $\tilde{h}_I(\Delta_X) = \Delta_{\tilde{h}_I(X)}$ for all objects X in \mathbb{X} . □

We develop a criterion for when an inverse functor is faithful:

Lemma 4.2.5. A restriction functor $F : \mathbb{X} \rightarrow \mathbb{Y}$ between inverse categories is faithful if and only if it reflects and is faithful on restriction idempotents.

F reflects restriction idempotents when for all endomorphisms $h : A \rightarrow A$ in \mathbb{X} , $\overline{F(h)} = F(h)$ implies $\bar{h} = h$.

Proof.

- If F is faithful then it is faithful on restriction idempotents. To show it reflects idempotents, if $F(g) = \overline{F(g)}$, then $F(g)F(g) = F(gg) = F(g)$. So g is an idempotent

and thus a restriction idempotent, as all idempotents are restriction idempotents in an inverse category.

- Conversely, suppose F reflects and is faithful on restriction idempotents and that $F(f) = F(g)$ for f, g which are parallel maps in \mathbb{X} . This means that

$$\overline{F(f)} = F(f)F(f)^\circ = F(f)F(g)^\circ = F(fg^\circ)$$

So fg° is a restriction idempotent as is $g^\circ f$. But

$$F(fg^\circ) = F(f)F(g)^\circ = F(f)F(f)^\circ = F(ff^\circ)$$

So $fg^\circ = ff^\circ$ and $g^\circ f = f^\circ f$. Thus g° is the partial inverse of f , and hence $g^\circ = f^\circ$.

□

If F is a *discrete* inverse functor—then we can do even better:

Lemma 4.2.6. A discrete inverse functor $F : \mathbb{X} \rightarrow \mathbb{Y}$ between discrete inverse categories is faithful if and only if it is faithful on restriction idempotents.

Proof.

- If F is faithful it is certainly faithful on restriction idempotents.
- Conversely, by Lemma 4.2.5, it suffices to prove that F reflects restriction idempotents.

Suppose $F(f) = F(\bar{f})$, then

$$F(\bar{f}) = F(f) \cap \overline{F(f)} = F(f) \cap F(\bar{f}) = F(f \cap \bar{f})$$

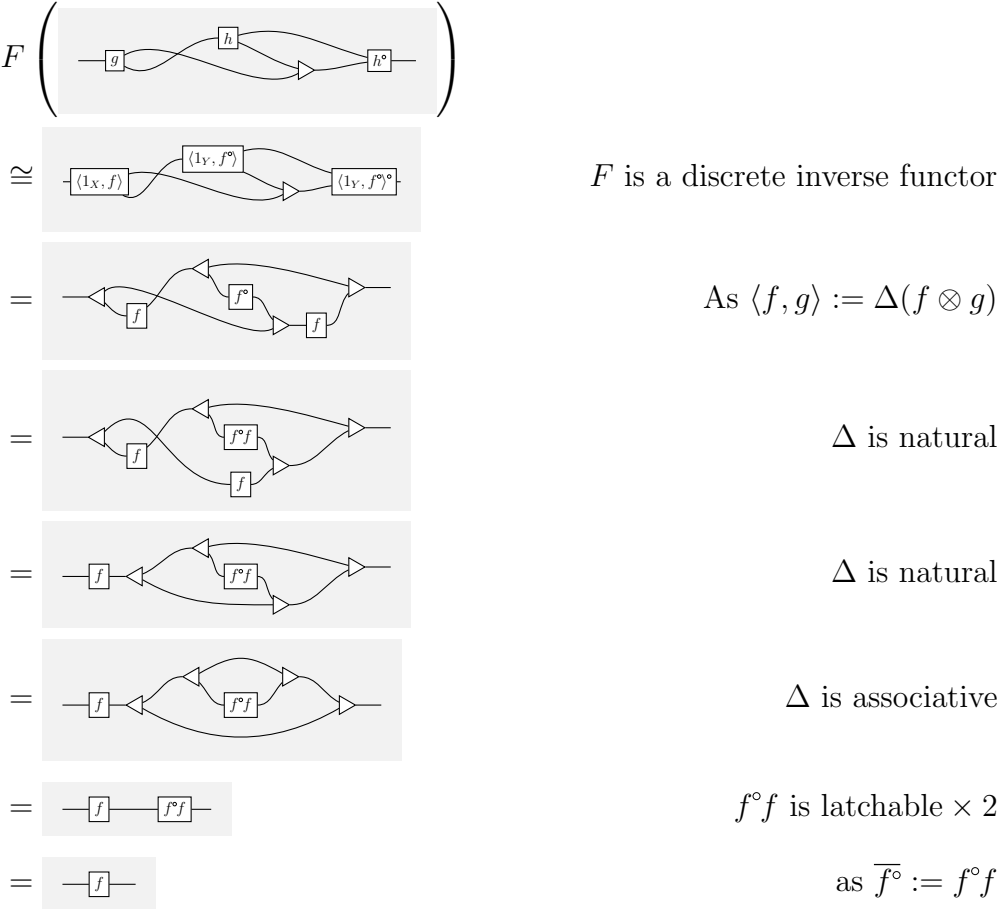
Since \bar{f} and $f \cap \bar{f}$ are restriction idempotents and F is faithful on restriction idempotents, then $\bar{f} = f \cap \bar{f} \leq f$. But then $\bar{f} \leq f$ iff $\bar{f}f = \bar{f}$. So $f = \bar{f}$ as $\bar{f}f = f$.

□

The following criterion is used in Chapters 5 and 7 for proving the fullness of a restricted class of discrete inverse functors:

Lemma 4.2.7. Let $F : \mathbb{X} \rightarrow \mathbb{Y}$ be a discrete inverse functor between discrete inverse categories and let $f : X \rightarrow Y$ be a partial isomorphism in \mathbb{Y} . If $\langle 1_Y, f^\circ \rangle := \Delta(1_Y \otimes f^\circ)$ and $\langle 1_X, f \rangle := \Delta(1_X \otimes f)$ are in the image of F , then so is f .

Proof. For all maps $f : X \rightarrow Y$ in \mathbb{Y} there are maps: g and h in \mathbb{X} such that :



Therefore, f is in the image of F and by symmetry f° is as well.

□

If F is seen as realising a restricted class of circuits, this has an interpretation in terms

of circuit synthesis. Given a partial isomorphism $f : X \rightarrow Y$, algorithms for the synthesis of $\langle 1_X, f \rangle$, $\langle 1_Y, f^\circ \rangle$ and $\langle 1_Y, f^\circ \rangle^\circ$, an algorithm for synthesizing f is obtained by Lemma 4.2.7.

4.3 Barr's ℓ^2 functor

In [10], Barr describes a contravariant functor from sets and partial injections to Hilbert spaces. We will use the covariant version of the functor described in [32] (although he considers the restriction to finite sets).

The functor $\ell^2 : \text{Pinj} \rightarrow \text{Hilb}$ takes:

Objects: For any object X :

$$\ell^2(X) := \left\{ \varphi : X \rightarrow \mathbb{C} \mid \sum_{x \in X} |\varphi(x)|^2 < \infty \right\}$$

It is easily verified that $\ell^2(X)$ is a vector space with basis $\{\delta_x\}_{x \in X}$, where $\delta_x : X \rightarrow \mathbb{C}$ is the Kronecker delta function taking $x \neq y \mapsto 0$ and $x \mapsto 1$. Moreover, $\ell^2(X)$ is a Hilbert space as there is an induced inner product:

$$\langle \psi | \varphi \rangle := \sum_{x \in X} \overline{\psi(x)} \varphi(x)$$

Where the $\overline{\psi(x)}$ is the complex conjugate of $\psi(x)$.

Maps: For any map $f : X \rightarrow Y$, $\ell^2(f)$ is defined on basis elements, so that for any $x \in X$:

$$(\ell^2(f))(\delta_x) := \begin{cases} \delta_{f(x)} & \text{if } \overline{f(x)} = 1_X \\ 0 & \text{otherwise} \end{cases}$$

Theorem 4.3.1. [30, §4.2, §4.3, §4.9] $\ell^2 : \text{Pinj} \rightarrow \text{Hilb}$ is a strong \dagger -symmetric monoidal, essentially surjective, faithful functor.

In the case where \mathbb{X} has a terminal object I , note the similarity of $\tilde{h}_I : \mathbb{X} \rightarrow \text{Pinj}$ and $\ell^2 : \text{Pinj} \rightarrow \text{Hilb}$. Whereas $\tilde{h}_I : \mathbb{X} \rightarrow \text{Pinj}$ evaluates maps on points, sending undefined evaluations to \uparrow ; $\ell^2 : \text{Pinj} \rightarrow \text{Hilb}$ evaluates maps on basis elements sending undefined evaluations to the zero matrix.

Chapter 5

The Controlled-not Gate

In this chapter we model the behaviour of circuits comprised of controlled-not gates and computational ancillary bits (corresponding to state preparation and post-selected measurement in the standard basis). The controlled-not gate is the unitary matrix taking $|b_1, b_2\rangle$ to $|b_1, b_1 \oplus b_2\rangle$. We model these circuits as maps in the symmetric monoidal category **CNOT** given by finite generators and relations. This follows the work of Lafont, wherein a finite complete set of identities is given for circuits generated by the controlled-not gate *without ancillary bits* [38, §3.1].

We prove that **CNOT** is discrete inverse equivalent to a concrete category of torsors and partial maps – in other words the category of affine partial isomorphisms between finite dimensional \mathbb{Z}_2 vector spaces. This allows us to use the faithful strong \dagger -symmetric monoidal functor $\ell^2 : \mathbf{Pinj} \rightarrow \mathbf{Hilb}$ to prove the completeness of these identities with respect to the canonical interpretation into $\mathbf{Mat}_{\mathbb{C}}$.

5.1 The category CNOT

Define the category **CNOT** to be the PROP generated by the 1 ancillary bits $|1\rangle$ and $\langle 1|$ as well as the controlled-not gate:

$$|1\rangle := \text{---}\blacktriangleright \quad \langle 1| := \blacktriangleleft\text{---} \quad \text{cnot} := \begin{array}{c} \text{---} \\ | \\ \oplus \\ \text{---} \end{array}$$

These generators have a canonical interpretation $\llbracket - \rrbracket_{\text{CNOT}} : \text{CNOT} \rightarrow \text{Mat}_{\mathbb{C}}$:

$$\llbracket |1\rangle \rrbracket_{\text{CNOT}} := \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \llbracket \langle 1| \rrbracket_{\text{CNOT}} := \begin{bmatrix} 0 & 1 \end{bmatrix} \quad \llbracket \text{cnot} \rrbracket_{\text{CNOT}} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The controlled-not gate and the 1-ancillary bits allow the **not** gate, $|0\rangle$ gate, $\langle 0|$ gate and flipped **cnot** gate to be defined:

$$\text{---}\oplus := \text{---}\oplus\blacktriangleright\blacktriangleleft, \quad \blacktriangleright\text{---} := \blacktriangleright\oplus, \quad \blacktriangleleft\text{---} := \oplus\blacktriangleleft, \quad \text{---}\oplus := \text{---}\oplus\text{---}$$

We also allow for “gaps” **cnot** gates to suppress symmetry maps:

$$\text{---}\oplus := \text{---}\oplus\text{---}$$

These gates must satisfy the identities given in Figure 5.1.

[**CNOT.1**] can be stated algebraically as $\text{cnot } c \text{ cnot } c \text{ cnot} = c$. [**CNOT.2**] shows that **cnot** is self-inverse. [**CNOT.3**] shows that control bits can be slid past each other; likewise, [**CNOT.5**] says that target bits can be slid past each other. [**CNOT.4**] shows how controlling from $|1\rangle$, produces a **not** gate. [**CNOT.6**] shows that preparing the state $|1\rangle$ entails that $|1\rangle$ will be measured. [**CNOT.7**] shows that controlling from $|0\rangle$ makes the **cnot** gate disappear. [**CNOT.9**] shows that multiplication by the zero matrix collapses the

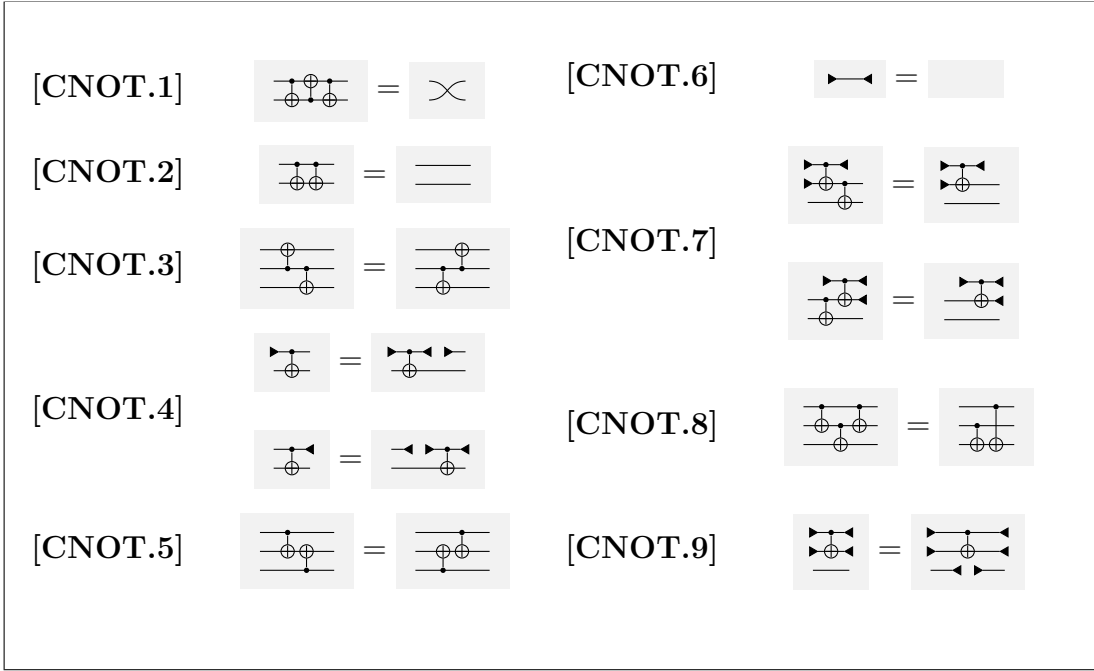


Figure 5.1: The identities of CNOT

connectivity of circuits.

[CNOT.8] (in conjunction with [CNOT.2]) allows certain configurations of `cnot` gates to be pushed “past each other” with another trailing `cnot` gate as a side effect. This is the `cnot` case of a class of identities given by Iwama et al. [33], which we discuss in great detail in Chapter 7.

$$\begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \ominus \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \\ \ominus \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array}$$

Up to sliding control wires and target wires, all identities are graphically horizontally symmetric. Therefore there is an obvious \dagger -functor, $(-)^{\circ} : \text{CNOT}^{\text{op}} \rightarrow \text{CNOT}$ given by `cnot` \mapsto `cnot`, $|1\rangle \mapsto \langle 1|$, $\langle 1| \mapsto |1\rangle$.

It is mechanical to verify that the interpretation $\llbracket - \rrbracket_{\text{CNOT}} : \text{CNOT} \rightarrow \text{Mat}_{\mathbb{C}}$ is a strict \dagger -symmetric monoidal functor which reflects the dagger, so this interpretation is sound. However, proving completeness is highly nontrivial.

5.2 Preliminary results for CNOT

Because of Axiom [CNOT.9], it follows that CNOT has zero-maps (although there is no zero-object). This warrants discussion as zero maps play an important role later.

Definition 5.2.1. The **zero circuit** in CNOT is:

$$\emptyset := \begin{array}{|c|} \hline \text{---} \oplus \text{---} \\ \hline \end{array} : 0 \rightarrow 0$$

In order to show this is a zero map, first:

Lemma 5.2.2. $\emptyset \otimes \emptyset = \emptyset$

Proof.

$$\begin{aligned} \begin{array}{|c|} \hline \text{---} \oplus \text{---} \\ \oplus \\ \text{---} \oplus \text{---} \\ \oplus \\ \text{---} \oplus \text{---} \\ \hline \end{array} &= \begin{array}{|c|} \hline \text{---} \oplus \text{---} \\ \oplus \\ \text{---} \oplus \text{---} \\ \hline \end{array} \\ &= \begin{array}{|c|} \hline \text{---} \oplus \text{---} \\ \oplus \\ \text{---} \oplus \text{---} \\ \hline \end{array} && \text{[CNOT.4]} \times 2 \\ &= \begin{array}{|c|} \hline \text{---} \oplus \text{---} \\ \oplus \\ \text{---} \oplus \text{---} \\ \oplus \\ \text{---} \oplus \text{---} \\ \hline \end{array} && \text{[CNOT.2]} \\ &= \begin{array}{|c|} \hline \text{---} \oplus \text{---} \\ \oplus \\ \text{---} \oplus \text{---} \\ \hline \end{array} && \text{[CNOT.1]} \\ &= \begin{array}{|c|} \hline \text{---} \oplus \text{---} \\ \hline \end{array} \end{aligned}$$

□

We generalize \emptyset to an arbitrary domain and codomain:

Definition 5.2.3. For any n and m in \mathbb{N} , define the zero map from n to m , $\emptyset_{n,m} : n \rightarrow m$ by the circuit $|1\rangle^{\otimes m} \emptyset \langle 1|^{\otimes n}$.

These are zero maps:

Lemma 5.2.4.

- (i) If $f = f \otimes \emptyset$, for some $f : n \rightarrow m$ then $f = \emptyset_{n,m}$

(ii) If $g : m \rightarrow p$, then $\emptyset_{n,m}g = \emptyset_{n,p}$

(iii) If $h : p \rightarrow n$, then $h\emptyset_{n,m} = \emptyset_{p,n}$

Proof.

(i) Consider an arbitrary circuit $f : n \rightarrow m$ such that $f = f \otimes \emptyset$. Use [CNOT.9] to cut the wires around every gate in f . Then every cut gate must either be \emptyset or $\langle 1|_0|1 \rangle$. In the first case, use Lemma 5.2.2 to consume the \emptyset obtained by cutting. In the second case, by [CNOT.6], allow one to remove the circuit. Thus $f = f \otimes \emptyset = \emptyset_{n,m}$.

(ii) Clearly $\emptyset_{n,m}\emptyset = \emptyset_{n,m} \otimes \emptyset = \emptyset_{n,m}$ but then $h\emptyset_{n,m} = (h\emptyset_{n,m})\emptyset$ so $h\emptyset_{n,m} = \emptyset_{p,m}$.

(iii) Dual to (ii).

□

5.3 CNOT is a discrete inverse category

In this section, we prove that CNOT is a discrete inverse category. We show “discreteness” before establishing the inverse category properties.

5.3.1 Inverse products in CNOT

We begin by defining two families of maps Δ_n and ∇_n for all $n \in \mathbb{N}$. We then show that these maps are semi-Frobenius algebras.

Definition 5.3.1. Define two families of maps $\{\Delta_n : n \rightarrow 2n\}_{n \in \mathbb{N}}$ and $\{\nabla_n : 2n \rightarrow n\}_{n \in \mathbb{N}}$ as follows.

On zero wires, define $\Delta_0 := 1_0$.

On one wire, define Δ_1, ∇_1 , respectively by:

$$\begin{array}{c} \text{---} \diagup \quad \diagdown \text{---} \\ \text{---} \end{array} := \begin{array}{c} \triangleright \oplus \\ \text{---} \\ \text{---} \end{array} \quad \text{and dually} \quad \begin{array}{c} \text{---} \diagdown \quad \diagup \text{---} \\ \text{---} \end{array} := \begin{array}{c} \oplus \triangleleft \\ \text{---} \\ \text{---} \end{array}$$

On n wires define Δ, ∇ inductively as follows:

$$\begin{array}{c} \text{---} \\ \diagup \\ \text{---} \end{array}^{n+1} := \begin{array}{c} \text{---} \\ \diagup \\ \text{---} \\ \diagdown \\ \text{---} \end{array}^n \quad \text{and dually} \quad \begin{array}{c} \text{---} \\ \diagdown \\ \text{---} \end{array}^{n+1} := \begin{array}{c} \text{---} \\ \diagdown \\ \text{---} \\ \diagup \\ \text{---} \end{array}^n$$

As we have defined CNOT by its generators, most of the proofs which follow involve structural induction; that is, by induction on the generators of CNOT. Moreover, since CNOT is a PROP, often we can just prove the inductive step.

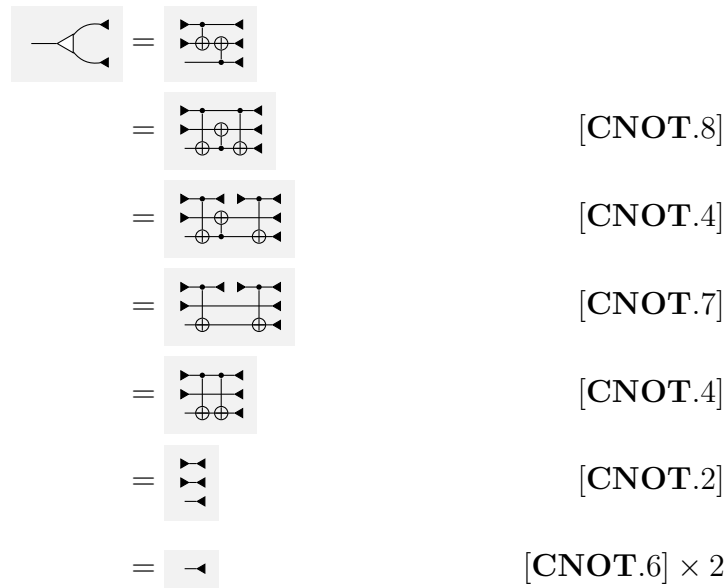
Lemma 5.3.2. Δ is a natural transformation.

Proof. We prove Δ is natural by structural induction on circuits.

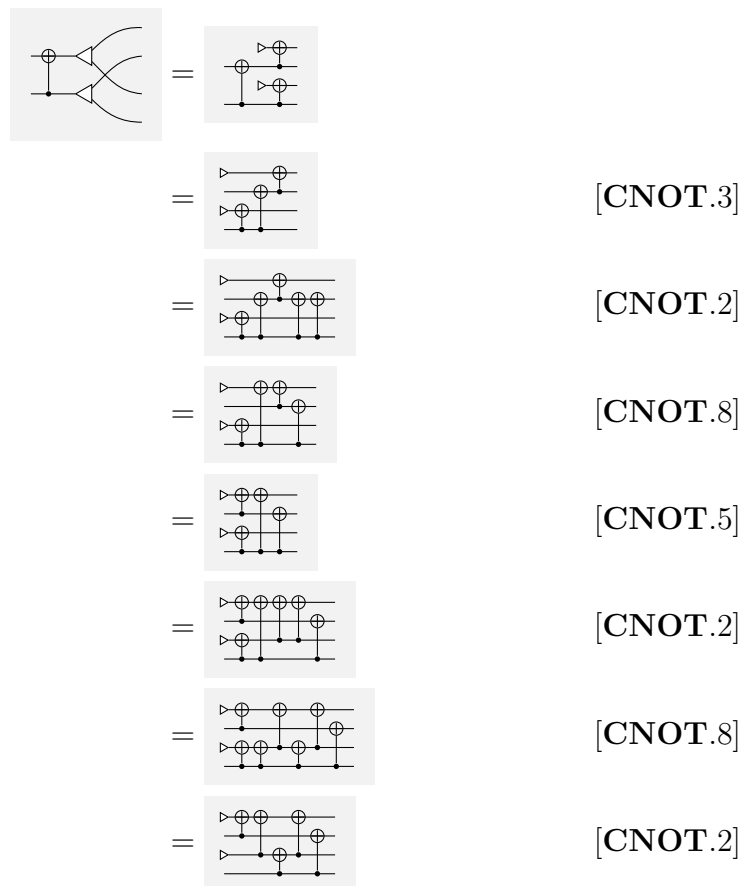
For |1):

$$\begin{array}{l}
 \begin{array}{c} \text{---} \\ \diagup \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} \\
 = \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} \quad \text{[CNOT.8]} \\
 = \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} \quad \text{[CNOT.4]} \\
 = \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} \quad \text{[CNOT.7]} \\
 = \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} \quad \text{[CNOT.4]} \\
 = \begin{array}{c} \text{---} \\ \diagdown \\ \text{---} \end{array} \quad \text{[CNOT.2]} \\
 = \begin{array}{c} \text{---} \\ \diagdown \\ \text{---} \end{array} \quad \text{[CNOT.6]}
 \end{array}$$

For $\langle 1|$:



For cnot:



$$\begin{aligned}
&= \text{[CNOT.8]} \\
&=
\end{aligned}$$

□

Lemma 5.3.3. Δ is cocommutative.

Proof. We prove that Δ is cocommutative by induction on the number of wires:

- On no wires the result is immediate.
- It suffices to observe, on one wire:

$$\begin{aligned}
&= \text{[CNOT.1]} \\
&= \text{[CNOT.2]} \\
&= \text{[CNOT.7]} \\
&=
\end{aligned}$$

□

Lemma 5.3.4. Δ is special.

Proof. We prove that Δ is special by induction on the number of wires.

- On zero wires it is the identity.
- It suffices to observe, on one wire:

$$\begin{aligned}
\text{[Diagram: } \Delta \text{]} &= \text{[Diagram: } \Delta \text{ with } \oplus \text{ gates]} \\
&= \text{[Diagram: } \Delta \text{ with } \ominus \text{ gate]} && \text{[CNOT.2]} \\
&= \text{[Diagram: } \Delta \text{]} && \text{Lemma B.0.1 (ii)}
\end{aligned}$$

□

Lemma 5.3.5. Δ is coassociative.

Proof. We prove that Δ is coassociative by induction on the number of wires.

- On zero wires it is immediate.
- It suffices to observe, on one wire:

$$\begin{aligned}
\text{[Diagram: } \Delta \text{]} &= \text{[Diagram: } \Delta \text{ with } \oplus \text{ gates]} \\
&= \text{[Diagram: } \Delta \text{ with } \oplus \text{ gates]} && \text{[CNOT.2]} \\
&= \text{[Diagram: } \Delta \text{ with } \oplus \text{ gates]} && \text{Lemma B.0.1 (i)} \\
&= \text{[Diagram: } \Delta \text{ with } \oplus \text{ gates]} && \text{[CNOT.7]} \times 2 \\
&= \text{[Diagram: } \Delta \text{]}
\end{aligned}$$

□

The dual propositions for ∇ hold since, $\Delta^\circ = \nabla$.

Lemma 5.3.6. (n, Δ_n, ∇_n) is a semi-Frobenius algebra for all $n \in \mathbb{N}$.

Proof. We prove that (Δ, ∇) is a semi-Frobenius algebra by induction on the number of wires.

- On zero wires it is immediate.
- It suffices to observe, on one wire:

$$\begin{aligned}
 & \text{Diagram of } \Delta \text{ and } \nabla \text{ on one wire} = \text{Diagram 1} \\
 & = \text{Diagram 2} \quad \text{[CNOT.3]} \\
 & = \text{Diagram 3} \quad \text{[CNOT.2]} \\
 & = \text{Diagram 4} \quad \text{[CNOT.8]} \\
 & = \text{Diagram 5} \quad \text{[CNOT.2]} \\
 & = \text{Diagram 6} \quad \text{[CNOT.8]} \\
 & = \text{Diagram of } \Delta \text{ and } \nabla \text{ on one wire}
 \end{aligned}$$

□

Note that Δ satisfies the uniform copying law by construction.

5.3.2 CNOT is an inverse category

To prove that CNOT is an inverse category, we need to prove that the functor $(-)^{\circ} : \text{CNOT}^{\text{op}} \rightarrow \text{CNOT}$ which horizontally flips circuits satisfies [INV.1], [INV.2] and [INV.3]. It is immediate that [INV.1] holds, as $(-)^{\circ} : \text{CNOT}^{\text{op}} \rightarrow \text{CNOT}$ is a dagger functor. It remains to show that [INV.2] and [INV.3] hold.

To prove that [INV.3] holds, we show that circuits of the form ff° are latchable, and thus commute.

Proposition 5.3.7. All circuits of the form ff° are latchable.

Proof. We prove that all circuits of the form ff° are latchable by structural induction on f .

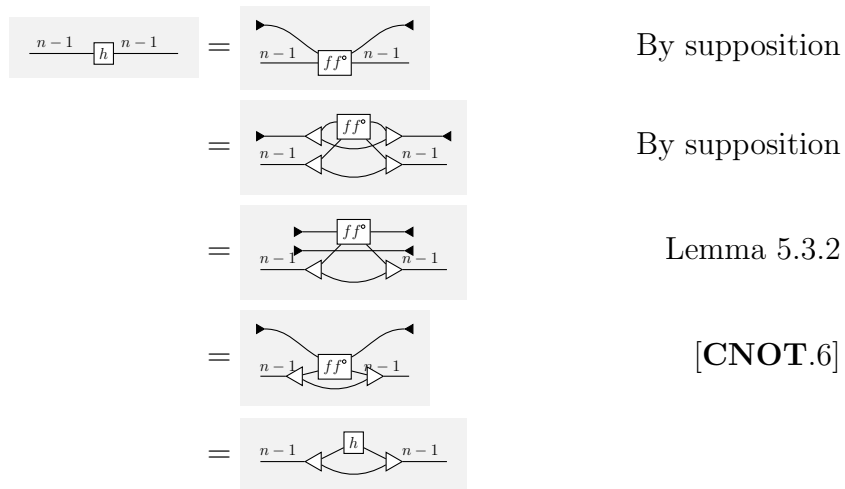
Any circuit p with no components is a permutation and, thus, pp° is the identity. So being latchable in this case comes from the fact that the semi-Frobenius algebras are special. Furthermore, adding a permutation, p , in front of any latchable circuit, h , gives a latchable circuit as:

$$(ph)(ph)^\circ = phh^\circ p^\circ = p\Delta(hh^\circ \otimes 1)\nabla p^\circ = \Delta(phh^\circ p^\circ \otimes pp^\circ)\nabla = \Delta(((ph)(ph)^\circ) \otimes 1)\nabla$$

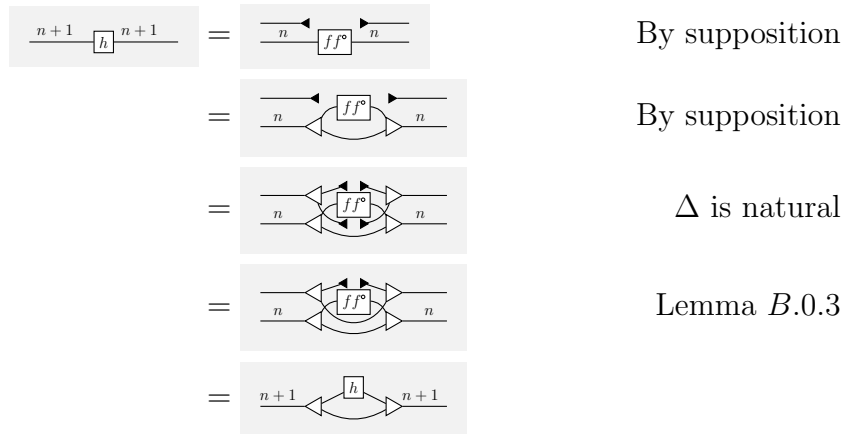
Thus we need only consider adding gates to the top left of circuits: adding a gate anywhere else can be simulated by precomposing with a permutation to move the gates wires to the top, then adding the gate at the top left, and then precomposing with the inverse of the permutation.

Thus, it suffices to show inductively that when a circuit of the form ff° is latchable, adding a generator to the top left of f results in a circuit which is still latchable.

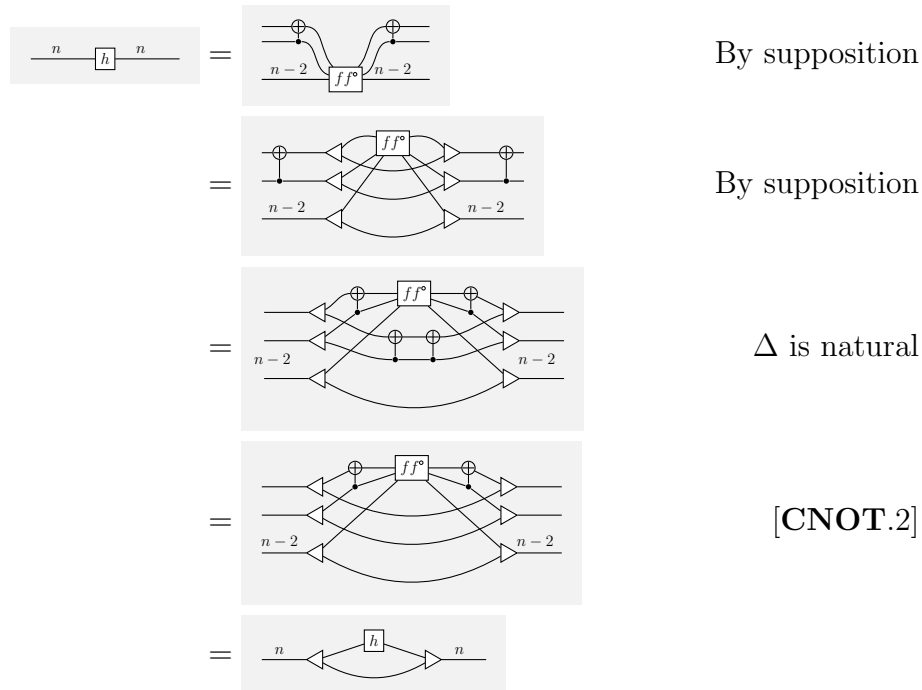
For |1):



For $\langle 1|$:



For cnot:



□

Therefore, as latchable circuits commute, by Lemma 4.1.13:

Proposition 5.3.8. Circuits of the form ff° commute.

Proof. Circuits of the form ff° are latchable, by Proposition 5.3.7. By Lemma 4.1.13, latchable circuits are idempotent. \square

It remains to prove that [INV.2] holds which we prove by induction.

Lemma 5.3.9. [INV.2] holds in CNOT with respect to the functor $(-)^{\circ} : \text{CNOT}^{\text{op}} \rightarrow \text{CNOT}$.

Proof. We will prove [INV.2] holds under the functor $(-)^{\circ}$ by structural induction.

- For the base case, for a permutation f , $ff^\circ f = f$.
- For the inductive hypothesis, suppose that $gg^\circ g = g$.

We proceed by cases for each generator:

For $|1\rangle$:

$$\begin{aligned}
 & \frac{n}{f} \frac{m}{f^\circ} \frac{n}{f} \frac{m}{f} \\
 = & \frac{n-1}{gg^\circ} \frac{n-1}{g} \frac{m}{g} && \text{By supposition} \\
 = & \frac{n-1}{gg^\circ g} \frac{m}{g} && \text{[INV.3]} \\
 = & \frac{n-1}{gg^\circ g} \frac{m}{g} && \text{[CNOT.6]} \\
 = & \frac{n-1}{g} \frac{m}{g} && \text{By the inductive hypothesis} \\
 = & \frac{n}{f} \frac{m}{f} && \text{By supposition}
 \end{aligned}$$

For $\langle 1 \rangle$:

$$\overline{n} \boxed{f}^m \boxed{f^s}^n \boxed{f}^m$$

$$= \overline{n-1} \boxed{gg^s}^{n-1} \boxed{g}^m$$

By supposition

$$= \overline{n-1} \boxed{gg^s}^{n-1} \boxed{g}^m$$

[CNOT.6]

$$= \overline{n-1} \boxed{g}^m$$

By the inductive hypothesis

$$= \overline{n} \boxed{f}^m$$

By supposition

For cnot:

$$\overline{n} \boxed{f}^m \boxed{f^s}^n \boxed{f}^m$$

$$= \overline{n-2} \boxed{gg^s}^{n-2} \boxed{g}^m$$

By supposition

$$= \overline{n-2} \boxed{gg^s}^{n-2} \boxed{g}^m$$

[CNOT.2]

$$= \overline{n-2} \boxed{g}^m$$

By the inductive hypothesis

$$= \overline{n} \boxed{f}^m$$

By supposition

□

Therefore:

Theorem 5.3.10. CNOT is a discrete inverse category.

5.4 Torsors

Our aim is to prove that CNOT is a discrete inverse category, equivalent to the category of partial isomorphism between finitely generated non-empty commutative torsors of characteristic 2, $\text{ParIso}(\text{CTor}_2)^*$. Torsors are essentially groups without a fixed multiplicative identity: the category $\text{ParIso}(\text{CTor}_2)^*$ may be viewed, perhaps more familiarly, as the partial isomorphism category of finite-dimensional \mathbb{Z}_2 vector spaces with affine maps.

Definition 5.4.1. A **torsor** is a set X along with a ternary operation $(-) \times_{(\cdot)} (-) : X \times X \times X \rightarrow X$ called para-multiplication, such that for any $a, b, c, d, e \in X$, the following laws hold [37]:

Para-associativity:
$$(a \times_b c) \times_d e = a \times_{d \times_c b} e = a \times_b (c \times_d e)$$

Para-identity:
$$a \times_b b = b \times_b a = a$$

A torsor is said to be **commutative**, when $a \times_b c = c \times_b a$ and a torsor is said to have **characteristic 2**, when $a \times_b a = b$.

The category of torsors Tor has objects torsors and maps homomorphisms of torsors. A homomorphism of torsors, $f : (X, \times) \rightarrow (Y, \times)$, is a function $X \rightarrow Y$ which preserves para-multiplication.

As this is a category of algebras we know that it is a finitely complete category. This allows us to form the discrete restriction category $\text{Par}(\text{Tor})$, and the discrete inverse category $\text{ParIso}(\text{Tor})$ immediately.

If (X, \times) is a non-empty torsor, then X has, for each element z of X , a group structure with multiplication given by:

$$\cdot : X^2 \rightarrow X; (x, y) \mapsto x \times_z y.$$

Conversely, if (X, \cdot) is a group, then X has a non-empty torsor structure $\times : X^3 \rightarrow X$ such that $(x, z, y) \mapsto x \cdot z^{-1} \cdot y$ [11, Sec. 0.2]. This correspondence, however, does not imply

that the category of nonempty torsors and groups are equivalent since their homomorphisms are different. Although it does give a functor from the category of Torsors to groups.

Some authors, including their originator [37, Definition 18], require the underlying set of a torsor to be nonempty so torsors always arise as groups. A torsor is also known variously as a “heap, group, flock, herd, principal homogeneous space, abstract coset [or] pregroup,” [11, Sec. 0.2] with the non-emptiness condition appearing in some cases. However, following [11, Sec. 0.2], we will not impose this condition as we need a category closed to pullbacks, and the empty torsor can arise as a pullback of non-empty torsors. In particular, we will see how the empty torsor arises as a subobject of the pullback of non-empty finitely generated commutative torsors of characteristic 2 .

Definition 5.4.2. Define \mathbf{CTor}_2 to be the full subcategory of torsors whose objects are finitely generated commutative torsors of characteristic 2 (including the empty torsor).

There is an equivalent characterization of the objects of \mathbf{CTor}_2 .

Proposition 5.4.3. Every object in \mathbf{CTor}_2 is either empty or isomorphic to $(\mathbb{Z}_2^n, _ \oplus _ \oplus _)$, where $_ \oplus _$ is componentwise addition mod 2. Furthermore, torsor homomorphisms between non-empty torsors are precisely the affine maps.

An affine map between vector spaces is a linear map that doesn’t necessarily preserve 0.

Proof. For the first claim, consider an object $(X, _ \times _ : X^3 \rightarrow X)$ in \mathbf{CTor}_2 , where X is nonempty. As X is inhabited, choose some element z in X . Then X is also an Abelian group with multiplication $_ + _ := _ \times_z _ : X^2 \rightarrow X$. As the corresponding torsor is finitely generated, the group is finitely generated as well. Moreover, as the corresponding torsor has characteristic 2, all generators have order 2. Furthermore, as the torsor is commutative, the group is Abelian. As $(X, +)$ is finitely generated, any element of X is the (possibly infinite) product of finitely many generators. However, as the group is Abelian, we can regroup the generators together. However, as every generator has finite order, this element is a finite

product of finitely many generators. Therefore, $(X, +)$ has finite order as a group, so the set X is finite.

By the fundamental theorem of finitely generated Abelian groups, as $(X, +)$ has finite order, it is isomorphic as a group to \mathbb{Z}_2^n (where the group multiplication of \mathbb{Z}_2 is given by componentwise addition). Let $f : \mathbb{Z}_2^n \rightarrow X$ be the underlying bijection induced by this isomorphism of groups. As \mathbb{Z}_2^n is nonempty, it is endowed with a para-multiplication $-\oplus - \oplus - : X^3 \rightarrow X$. This isomorphism of groups extends to an isomorphism of torsors, as, for any $x, y, w \in \mathbb{X}$:

$$\begin{aligned}
f(x \oplus y \oplus w) &= f(x \oplus (y \oplus w)) \\
&= f(x) +_z f(y \oplus w) && f \text{ is a group homomorphism} \\
&= f(x) +_z (f(y) +_z f(w)) && f \text{ is a group homomorphism} \\
&= f(x) +_{z+f(y)z} f(w) && \text{Para-associativity of } + \\
&= f(x) +_{f(y)} f(w) && (X, +) \text{ has characteristic } 2
\end{aligned}$$

so the para-multiplication is preserved: thus $(X, -+_ -)$ and $(\mathbb{Z}_2^n, -\oplus - \oplus -)$ are isomorphic *as torsors*.

For the second claim, consider morphism of non-empty torsors, $f : (\mathbb{Z}_2^n, -\oplus - \oplus -) \rightarrow (\mathbb{Z}_2^m, -\oplus - \oplus -)$. Then, for any x and y in \mathbb{Z}_2^n :

$$f(x \oplus y) = f(x \oplus 0 \oplus y) = f(x) \oplus f(y) \oplus f(0)$$

Therefore, f is an affine transformation when \mathbb{Z}_2^n and \mathbb{Z}_2^m are seen as vector spaces over \mathbb{Z}_2 .

Conversely, consider an affine transformation of vector spaces $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$. Then f can

be regarded as morphism of torsors as for any $x, y, z \in \mathbb{Z}_2^n$,

$$\begin{aligned}
f(x \oplus y \oplus z) &= f(x \oplus (y \oplus z)) \\
&= f(x) \oplus f(y \oplus z) \oplus f(0) \\
&= f(x) \oplus f(y) \oplus f(z) \oplus f(0) \oplus f(0) \\
&= f(x) \oplus f(y) \oplus f(z)
\end{aligned}$$

□

As \mathbf{CTor}_2 is a category of algebras and so is finitely complete, we may construct $\mathbf{Par}(\mathbf{CTor}_2)$ and $\mathbf{Parlso}(\mathbf{CTor}_2)$. Let $\mathbf{Parlso}(\mathbf{CTor}_2)^*$ denote $\mathbf{Parlso}(\mathbf{CTor}_2)$ without the empty torsor.

Proposition 5.4.4. $\mathbf{Parlso}(\mathbf{CTor}_2)^*$ is a discrete inverse category.

Proof. $\mathbf{Parlso}(\mathbf{CTor}_2)^*$ is an inverse category by construction. Because $\mathbf{Par}(\mathbf{CTor}_2)^*$ is a discrete Cartesian restriction category with inverse products and $\mathbf{Parlso}(\mathbf{CTor}_2)^*$ is the category of partial isomorphisms, it is a discrete inverse category. □

One further property of \mathbf{CTor}_2 is worth mentioning, all its monic maps are *regular* monics. This means, more concretely, every subobject of a torsor is determined by some set of equations. Therefore in $\mathbf{Par}(\mathbf{CTor}_2)^*$, the restriction idempotents correspond to equations:

Lemma 5.4.5. The monics of \mathbf{CTor}_2 are regular.

Proof. Consider a monic $m : X \rightarrow Y$ in \mathbf{CTor}_2 .

Suppose that X is empty. Then we have the following equalizer diagram, where 1 and 2 are the 1 and 2 object commutative torsors of characteristic 2, respectively:

$$\emptyset \xrightarrow{m} Y \xrightarrow{!} 1 \begin{array}{c} \xrightarrow{* \mapsto 0} \\ \xrightarrow{* \mapsto 1} \end{array} 2$$

Suppose, otherwise, that $X \ni x_0$ is inhabited. Consider the relation \sim on Y given by:

$$y \sim w \iff \exists x \in X : y \cdot_{m(x_0)} m(x) = w$$

First, we show that it is an equivalence relation.

By the para-identity law, we have $y \cdot_{m(x_0)} m(x_0) = y$, so that $y \sim y$. So the relation is reflexive. Suppose $y \sim y'$ for some $y, y' \in Y$, so that there exists an element x of X such that $y \cdot_{m(x_0)} m(x) = y'$. Then:

$$y' \cdot_{m(x_0)} m(x) = y \cdot_{m(x_0)} m(x) \cdot_{m(x_0)} m(x) = y \cdot_{m(x_0)} m(x_0) = y$$

So that $y' \sim y$, and thus the relation is symmetric.

Furthermore, suppose that $y \sim y'$ and $y' \sim y''$, where $y, y', y'' \in Y$, so that there exists $x, x' \in X$ such that $y \cdot_{m(x_0)} m(x) = y'$ and $y' \cdot_{m(x_0)} m(x') = y''$.

Then

$$y'' = y' \cdot_{m(x_0)} m(x') = y \cdot_{m(x_0)} m(x) \cdot_{m(x_0)} m(x')$$

Since $m(x) \cdot_{m(x_0)} m(x')$ is an element of $m(X)$ the relation is also transitive.

We now show that this relation is a congruence with respect to the para-multiplication. Take some $y, y', z, w \in Y$. Suppose $y \sim y'$, so that there exists an element x of X such that $y \cdot_{m(x_0)} m(x) = y'$.

Then:

$$\begin{aligned} y' \cdot_z w &= y \cdot_{m(x_0)} x \cdot_z w && \text{As } y \cdot_{m(x_0)} m(x) = y' \\ &= w \cdot_z y \cdot_{m(x_0)} m(x) && \text{Commutativity} \\ &= y \cdot_z w' \cdot_{m(x_0)} m(x) && \text{Commutativity} \end{aligned}$$

So $y' \cdot_z w \sim y \cdot_z w$.

By symmetry, we also have $w \cdot_z y' \sim w \cdot_z y$.

Moreover

$$\begin{array}{ll}
z \cdot_{y'} w = z \cdot_{y \cdot_{m(x_0)} m(x)} w & \text{As } y \cdot_{m(x_0)} m(x) = y' \\
= z \cdot_{m(x)} m(x_0) \cdot_y w & \text{Para-associativity} \\
= w \cdot_y z \cdot_{m(x)} m(x_0) & \text{Commutativity} \\
= w \cdot_y z \cdot_{m(x)} m(x_0) \cdot_{m(x_0)} m(x_0) & \text{Para-identity} \\
= w \cdot_y m(x_0) \cdot_{m(x)} z \cdot_{m(x_0)} m(x_0) & \text{Commutativity} \\
= w \cdot_y m(x_0) \cdot_{m(x)} m(x_0) \cdot_{m(x_0)} z & \text{Commutativity} \\
= w \cdot_y m(x) \cdot_{m(x_0)} z & \text{Characteristic 2} \\
= w \cdot_y z \cdot_{m(x_0)} m(x) & \text{Commutativity}
\end{array}$$

So $z \cdot_{y'} w \sim w \cdot_y z$.

Define the Torsor X/Y to be the subtorsor of X modulo the congruence \sim . We show that the following diagram is an equalizer:

$$\begin{array}{ccccccc}
X & \xrightarrow{m} & Y & \xrightarrow{!} & 1 & \xrightarrow{*\mapsto [m(x_0)]\sim} & Y/X \\
& & & \searrow & & \nearrow & \\
& & & & & & y \mapsto [y]\sim
\end{array}$$

If $x \in X$, then

$$m(x) = m(x) \cdot_{m(x_0)} m(x_0) = m(x_0) \cdot_{m(x_0)} m(x)$$

Therefore, as $x_0 \in X$, $m(x) \sim m(x_0)$.

On the other hand, consider some $y \notin X$. Suppose for the sake of contradiction that

$y \sim m(x_0)$. Then there exists some $x \in X$ so that $y \cdot_{m(x_0)} m(x) = m(x_0)$. Therefore,

$$\begin{aligned}
m(x) &= m(x) \cdot_{m(x_0)} m(x_0) && \text{Para-identity} \\
&= m(x) \cdot_{y \cdot_{m(x_0)} m(x)} m(x_0) && \text{As } y \cdot_{m(x_0)} m(x) = m(x_0) \\
&= m(x) \cdot_{m(x)} m(x_0) \cdot_y m(x_0) && \text{Para-associativity} \\
&= m(x_0) \cdot_y m(x_0) && \text{Characteristic 2} \\
&= y && \text{Para-identity}
\end{aligned}$$

This is a contradiction. □

5.5 The equivalence between CNOT and $\text{Parlso}(\text{CTor}_2)^*$

The objective of this section is to prove that CNOT and $\text{Parlso}(\text{CTor}_2)^*$ are (structure-preserving) equivalent. The proof involves several steps:

- (1) Defining a discrete inverse functor $\tilde{H}_0 : \text{CNOT} \rightarrow \text{Parlso}(\text{CTor}_2)^*$.
- (2) Showing that \tilde{H}_0 is full and faithful on restriction idempotents.
- (3) Showing that \tilde{H}_0 is essentially surjective.
- (4) Showing that \tilde{H}_0 is full and faithful.

A key technical step is to reduce the fullness and faithfulness of \tilde{H}_0 to its fullness and faithfulness on restriction idempotents (step (2) above). This latter result is based on the normal form for restriction idempotents in CNOT, which is developed in Section 5.5.4.

5.5.1 Defining the functor $\tilde{H}_0 : \text{CNOT} \rightarrow \text{Parlso}(\text{CTor}_2)^*$

To construct a functor $\tilde{H}_0 : \text{CNOT} \rightarrow \text{Parlso}(\text{CTor}_2)^*$ we consider the following pullback, where $U : \text{CTor}_2 \rightarrow \text{Set}$ is the forgetful functor:

$$\begin{array}{ccc}
\text{ParIso}(\text{CTor}_2)^* & \xrightarrow{\text{ParIso}(U)} & \text{Pinj} \\
\downarrow \lrcorner & & \downarrow \lrcorner \\
\text{Par}(\text{CTor}_2)^* & \xrightarrow{\text{Par}(U)} & \text{Par}
\end{array}$$

To prove that CNOT is equivalent to $\text{ParIso}(\text{CTor}_2)^*$, the category of partial isomorphisms of finitely generated non-empty commutative torsors of characteristic 2, we start by considering a functor $h_0 : \text{CNOT} \rightarrow \text{Par}$. On the one hand, we lift it to a functor $H_0 : \text{CNOT} \rightarrow \text{Par}(\text{CTor}_2)^*$; and on the other hand, we lift it to a functor $\tilde{h}_0 : \text{CNOT} \rightarrow \text{Pinj}$. Then by the pullback of the diagram $\text{Par}(\text{CTor}_2)^* \xrightarrow{\text{Par}(U)} \text{Par} \leftarrow \text{Pinj}$ we are given a unique functor $\tilde{H}_0 : \text{CNOT} \rightarrow \text{ParIso}(\text{CTor}_2)^*$:

$$\begin{array}{ccccc}
& & & & \tilde{h}_0 \\
& & & & \curvearrowright \\
\text{CNOT} & & & & \text{Pinj} \\
& \searrow^{h_0} & & \searrow^{\text{ParIso}(U)} & \\
& & \text{ParIso}(\text{CTor}_2)^* & \xrightarrow{\text{ParIso}(U)} & \text{Pinj} \\
& \searrow^{\tilde{H}_0} & \downarrow \lrcorner & & \downarrow \lrcorner \\
& & \text{Par}(\text{CTor}_2)^* & \xrightarrow{\text{Par}(U)} & \text{Par} \\
& \searrow^{H_0} & & & \\
& & & &
\end{array}$$

Recall the representable restriction functor $h_X : \mathbb{X} \rightarrow \text{Par}$ from Section 4.2. Fixing $\mathbb{X} = \text{CNOT}$ and $X = 0$, we obtain a functor $h_0 := \text{Total}(\text{CNOT})(0, -) : \text{CNOT} \rightarrow \text{Par}$. As CNOT is an inverse category, it follows by Lemma 4.2.2 that every map in $h_0(\text{CNOT})$ is a partial isomorphism. Therefore $h_0 : \text{CNOT} \rightarrow \text{Par}$ factors through Pinj as $\tilde{h}_0 : \text{CNOT} \rightarrow \text{Pinj}$.

Now that we have the candidate functor $\tilde{h}_0 : \text{CNOT} \rightarrow \text{Pinj}$ for the pullback, we must also show that we can factor h_0 through $\text{Par}(U) : \text{Par}(\text{CTor}_2)^* \rightarrow \text{Par}$. To do so, we exhibit internal torsor structures in CNOT and show that h_0 preserves this structure: thus showing it can be factored through $\text{Par}(\text{CTor}_2)^*$. This will allow us to construct the functor $H_0 : \text{CNOT} \rightarrow \text{Par}(\text{CTor}_2)^*$ and whence $\tilde{H}_0 : \text{CNOT} \rightarrow \text{ParIso}(\text{CTor}_2)^*$, by pullback.

Proving that \tilde{H}_0 is a functor from CNOT to $\text{Par}(\text{CTor}_2)^*$ is not a trivial task.

5.5.2 The points of CNOT

Because \tilde{h}_0 evaluates maps in CNOT on points; their classification is needed. To express total maps in $\text{CNOT}(0, n)$ for any $n \in \mathbb{N}$, we define the following family of functions:

Definition 5.5.1. Define a family of functions $|_ \rangle : \mathbb{Z}_2^n \rightarrow \text{Total}(\text{CNOT})(0, n)$ indexed by n for all $n \in \mathbb{N}$ as follows.

Take $| \rangle := 1_0$. For any $b \in \mathbb{Z}_2^1$ define:

$$|b\rangle := \begin{cases} |1\rangle & \text{if } n = 1 \\ |0\rangle & \text{otherwise} \end{cases}$$

Moreover, for all $n \in \mathbb{N}$ such that $n > 1$, define:

$$|b_1, \dots, b_n\rangle := |b_1\rangle \otimes \dots \otimes |b_n\rangle$$

Lemma 5.5.2. Consider a circuit $f : n \rightarrow m$ with no output ancillary bits. Then, for any $x \in \mathbb{Z}_2^n$, there is some $y \in \mathbb{Z}_2^m$ such that $f \circ |x\rangle = |y\rangle$

Proof. It will suffice to prove our claim on a single controlled-not gate, then by induction, the more general claim follows immediately.

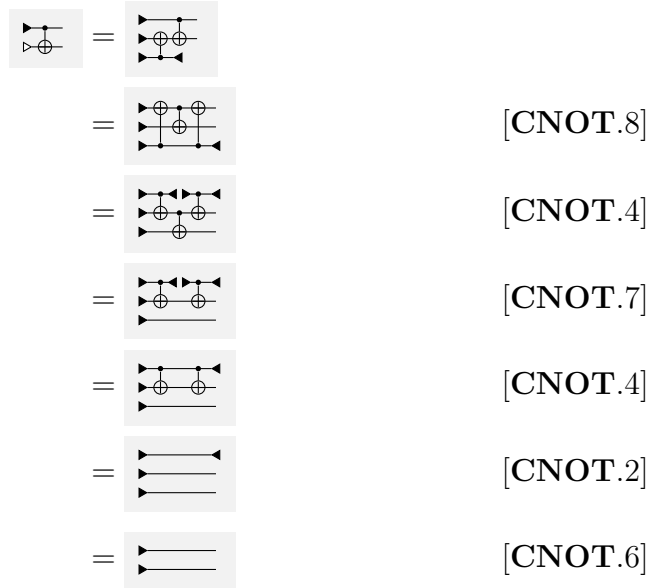
$\text{cnot} \circ |0, 0\rangle = |0, 0\rangle$:

$$\begin{array}{|c|} \hline \text{CNOT} \\ \hline \end{array} = \begin{array}{|c|} \hline \text{CNOT} \\ \hline \end{array} \quad [\text{CNOT.7}]$$

$\text{cnot} \circ |0, 1\rangle = |0, 1\rangle$:

$$\begin{array}{|c|} \hline \text{CNOT} \\ \hline \end{array} = \begin{array}{|c|} \hline \text{CNOT} \\ \hline \end{array} \quad [\text{CNOT.7}]$$

$\text{cnot}_\circ |1, 0\rangle = |1, 1\rangle$:



$\text{cnot}_\circ |1, 1\rangle = |1, 0\rangle$:



□

Using this, we observe that the points of CNOT satisfy:

Lemma 5.5.3. For every $f \in \text{CNOT}(0, n)$, f is either total or \emptyset .

Proof. Consider any circuit $f : 0 \rightarrow n$ for any $n \in \mathbb{N}$. We prove that f is either total or \emptyset by structural induction.

- For the base case, remark that permutations are total.
- For the inductive case, suppose that f is either total or zero. Consider a $g : n \rightarrow m$ with one component. If f is zero, then fg is zero as well by Lemma 5.2.4. Otherwise, suppose that f is total. There are three cases:

- If $|1\rangle \in g$, then $\overline{fg} = \overline{f \otimes g} = \overline{f} \otimes \overline{g} = 1 \otimes 1 = 1$.
- If $\langle 1| \in g$, then the gate to the left of $\langle 1|$ is either $|1\rangle$ or $|0\rangle$. If it is $|1\rangle$, then as $\langle 1| \circ |1\rangle = 1$, it is total. Otherwise, if it is $\langle 1|$, then $\langle 1| \circ |1\rangle = \emptyset$, so fg is zero by Lemma 5.2.4.
- If $\text{cnot} \in g$, then fg is total by Lemma 5.5.2.

□

This allows us to show:

Lemma 5.5.4. $\tilde{h}_0 : \text{CNOT} \rightarrow \text{Pinj}$ is a strong symmetric monoidal functor.

Proof. By Lemma 5.5.3, both

$$\tilde{h}_0(n \otimes m) = \text{Total}(\text{CNOT})(0, nm)$$

and

$$\tilde{h}_0(n) \times \tilde{h}_0(m) = \text{Total}(\text{CNOT})(0, m) \times \text{Total}(\text{CNOT})(0, m)$$

have 2^{nm} elements, so they are isomorphic in Pinj .

Lemma 5.5.3 also implies that $\tilde{h}_0(0) = \{1_0\}$, so that $\tilde{h}_0(0)$ is isomorphic to the tensor unit in Pinj . □

Therefore, by Lemma 4.2.4:

Lemma 5.5.5. $\tilde{h}_0 : \text{CNOT} \rightarrow \text{Pinj}$ is a discrete inverse functor.

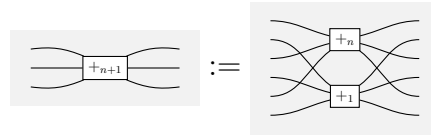
5.5.3 Internal torsor structures in CNOT

To prove that h_0 takes maps in CNOT to well-defined maps in $\text{Par}(\text{CTor}_2)^*$, we construct a torsor-like operation in CNOT which gives the internal torsor structure which was mentioned previously. This will act as the para-multiplication when we project onto the last 1/3 wires.

Definition 5.5.6. Define a family of maps $+_n : 3n \rightarrow 3n$ in CNOT inductively such that on no wires $+_0 := 1_0$, and on one wire:



Furthermore, for any n :



Now we show that $\text{Total}(\text{CNOT})(0, _)$ really does produce maps which preserve torsor structure.

Lemma 5.5.7. For any map $f : n \rightarrow m$ in CNOT, $f^{\otimes 3} +_m = +_n f^{\otimes 3}$.

Proof. For any map $f : n \rightarrow m$ in CNOT, we prove $f^{\otimes 3} +_m = +_n f^{\otimes 3}$ by induction on the size of f .

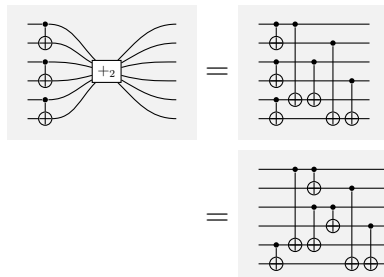
- For the base case, f is a permutation, so it is immediate that $f^{\otimes 3} +_m = +_n f^{\otimes 3}$.
- Suppose that $f^{\otimes 3} + = + f^{\otimes 3}$.

We proceed by structural induction:

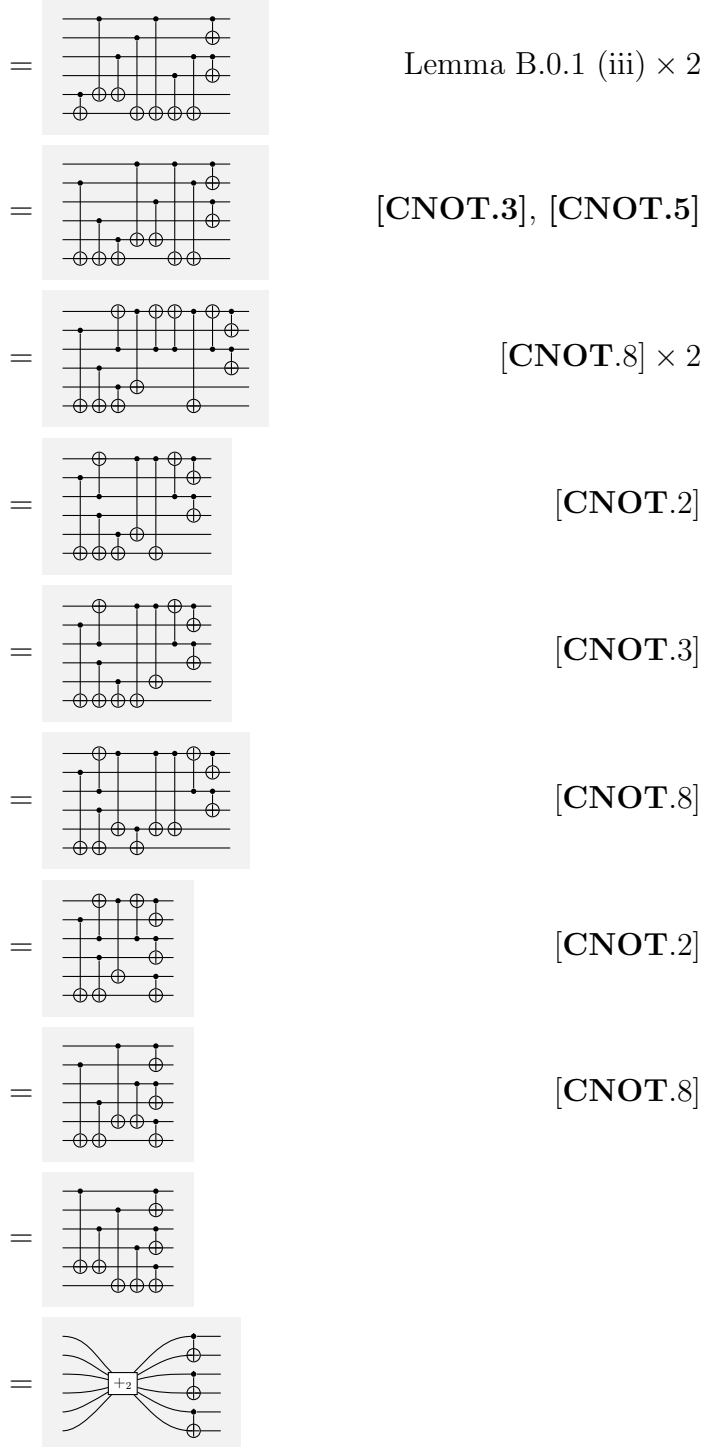
For $|1\rangle$: By Lemma 5.5.2, $f^{\otimes 3} + = g^{\otimes 3} + h^{\otimes 3} = +(gh)^{\otimes 3} = + f^{\otimes 3}$.

For $\langle 1|$: Dually to the previous case, $f^{\otimes 3} + = g^{\otimes 3} + h^{\otimes 3} = +(gh)^{\otimes 3} = + f^{\otimes 3}$.

For cnot:



[CNOT.3]



Therefore, $f^{\otimes 3} + = g^{\otimes 3} + h^{\otimes 3} = +(gh)^{\otimes 3} = +f^{\otimes 3}$.

□

We also show that h_0 produces well defined maps in $\text{Par}(\text{CTor}_2)$.

Lemma 5.5.8. Consider any $n, m \in \mathbb{N}$ and map $f \in \text{CNOT}(n, m)$. For any $x, y, z \in \text{CNOT}(0, n)$ such that $\overline{xf} = \overline{yf} = \overline{zf} = 1_0$, it follows that $\overline{(x \otimes y \otimes z) +_n f^{\otimes 3}} = 1_0$.

Proof. Consider an arbitrary map $f \in \text{CNOT}(n, m)$ and any $x, y, z \in \text{CNOT}(0, n)$ such that $\overline{xf} = \overline{yf} = \overline{zf} = 1_0$. By Lemma 5.5.7:

$$\begin{aligned} \overline{(x \otimes y \otimes z) +_n f^{\otimes 3}} &= \overline{(x \otimes y \otimes z)f^{\otimes 3} +_m} = \overline{(xf \otimes yf \otimes zf) +_m} \\ &= \overline{(xf \otimes yf \otimes zf) +_m} = \overline{(xf \otimes yf \otimes zf)1_{3m}} \\ &= \overline{xf \otimes yf \otimes zf} = \overline{xf} \otimes \overline{yf} \otimes \overline{zf} = 1_0 \otimes 1_0 \otimes 1_0 = 1_0 \end{aligned}$$

□

We now prove that $H_0 : \text{CNOT} \rightarrow \text{Par}(\text{Tor}_2)^*$ is a functor.

Lemma 5.5.9. h_0 can be factored as $H_0 \text{Par}(U)$ and h_0 preserves torsor structure. Thus, CNOT has internal torsor structure which is preserved by h_0 .

Proof. First, we prove for every $f : n \rightarrow m$ in CNOT , $h_0(f)$ can be regarded as a map in $\text{Par}(\text{CTor}_2)^*$. Consider an arbitrary map $f : n \rightarrow m$ in CNOT . If f is a zero map, then $h_0(f)$ vacuously preserves torsor structure. Suppose otherwise that there exists some $x, y, z \in \mathbb{Z}_2^n$ and $x', y', z' \in \mathbb{Z}_2^m$ such that $|x'\rangle = f \circ |x\rangle$, $|y'\rangle = f \circ |y\rangle$ and $|z'\rangle = f \circ |z\rangle$. Projecting the first 2 out of 3 wires with the Cartesian restriction structure, by Lemma 5.5.8, $h_0(f)(x \oplus y \oplus z)$ is defined. Moreover, by Lemma 5.5.7, $h_0(f)(x \oplus y \oplus z) = h_0(f)(x) \oplus h_0(f)(y) \oplus h_0(f)(z)$, so $h_0(f)$ preserves the para-multiplication. □

Corollary 5.5.10. The functor h_0 can be lifted to a functor $\tilde{H}_0 : \text{CNOT} \rightarrow \text{ParIso}(\text{CTor}_2)^*$.

Proof. As h_0 can be factored through Pinj by Lemma 4.2.2 and $\text{Par}(\text{CTor}_2)^*$ by Lemma 5.5.9, it is also a functor to $\tilde{H}_0 : \text{CNOT} \rightarrow \text{ParIso}(\text{CTor}_2)^*$ by pullback. □

5.5.4 Normal form for the idempotents of CNOT

As we will reduce the fullness and faithfulness of \tilde{H}_0 to its fullness and faithfulness on restriction idempotents, this section is dedicated to describing a normal form for restriction idempotents in CNOT. Such a normal form helps establish the fullness and faithfulness of \tilde{H}_0 on restriction idempotents.

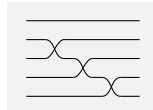
Recall that restriction idempotents of $\text{Par}(\text{CTor})^*$ are determined by finite sets of torsor equations. The restriction idempotents of CNOT also have a normal form, as a conjunction of clauses: we call this the clausal form for restriction idempotents in CNOT. As torsor equations can be translated into clauses it follows that \tilde{H}_0 is full on restriction idempotents. Furthermore, we also show that we perform Gaussian elimination on these clauses for the faithfulness on idempotents.

Definition 5.5.11. For any $n \in \mathbb{N}$, and $1 \leq i \leq n$, define the map $\text{swap}_{(i,n)} : n \rightarrow n$ on m wires inductively as follows:

$$\text{swap}_{(i,n)} := \begin{cases} 1_n & \text{If } i = 0 \\ (1_{i-1} \otimes c \otimes 1_{n-(i+1)})\text{swap}_{(i+1,n)} & \text{Otherwise} \end{cases}$$

That is, it pulls the i th wire from the top to the bottom.

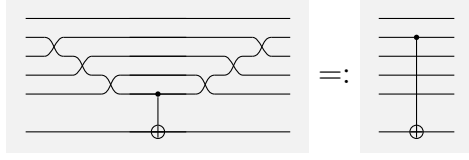
For example, consider the circuit $\text{swap}_{(2,5)}$:



Definition 5.5.12. Given any $n \in \mathbb{N}$ and $1 \leq i \leq n$, define the **literal** $l_{i,n} : n \rightarrow n$:

$$l_{i,n} := \begin{cases} (\text{swap}_{(i,n-1)} \otimes 1_1)(1_{n-2} \otimes \text{cnot})(\text{swap}_{(i,n-1)} \otimes 1_1)^\circ & \text{If } i < n \\ 1_{n-1} \otimes \text{not} & \text{Otherwise} \end{cases}$$

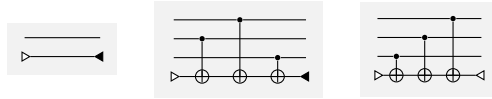
For example, consider the literal $l_{2,6} : 6 \rightarrow 6$:



Definition 5.5.13. A **clause** $e : n \rightarrow n$ is a map in CNOT which is the composition of literals in the following form:

$$e = (1_n \otimes \langle 0|) l_{i_1, n} l_{i_2, n} \cdots l_{i_m, n} (1_n \otimes |0\rangle)$$

In a clause, the wire which begins with an input ancillary bit and ends with an output ancillary bit, on which literals act is called the **clause wire**. The following are examples of clauses in which the bottom wire is the clause wire:



Note that the parity of the output bit can be flipped as a consequence of having a clause containing a literal on the diagonal, ie one of the form $l_{n, n}$.

Definition 5.5.14. A map in CNOT is said to be in **clausal form** if it is a sequence of clauses.

We wish now to show that every restriction idempotent in CNOT can be written in clausal form:

Lemma 5.5.15. Clauses are idempotent.

Proof. To show idempotence of a clause, we describe how to duplicate it. First, using the naturality of Δ , split the input ancilla bit on the clause wire $|0\rangle = \nabla(|0\rangle \otimes |0\rangle)$. Then by repeatedly using Lemma B.0.2 (iii), copy all of the literals onto both of the new clause wires. Then use the naturality of ∇ on the output ancillary bit $\langle b|\nabla := \langle 0| \otimes \langle 0|$ to split both clauses apart.

□

Proposition 5.5.16. In CNOT

- (i) Every restriction idempotent is equivalent to a circuit in clausal form.
- (ii) Every circuit in clausal form is a restriction idempotent.

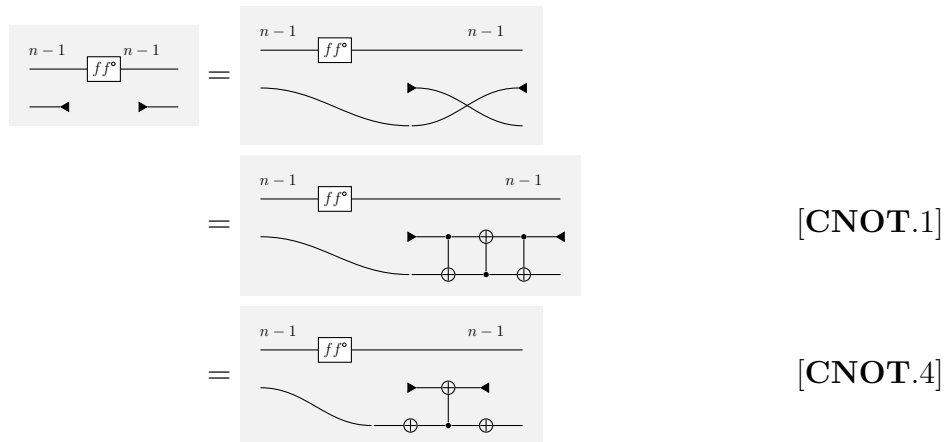
Proof.

- (i) Given a restriction idempotent $e : n \rightarrow n$ for some $n \in \mathbb{N}$, we prove e is in clausal form by structural induction.

- 1_n is in clausal form as $1_n = 1_n \otimes 1_0 = 1_n \otimes \langle 1| \circ |1\rangle$.
- Suppose that $ff^\circ : n \rightarrow n$ is in clausal form.
- We add components to the left of f :

For $|1\rangle$: Push the $|1\rangle$ past the clauses until it cancels with $\langle 1|$. For every literal which is on the same wire i ; the literal will be replaced with the literal $l_{n,n}$ (the not gate). Therefore, the clauses which the $|1\rangle$ gate passes by remain clauses.

For $\langle 1|$: If the $\langle 1|$ is on wire j , a clause with precisely the literals $l_{n,n}$ and $l_{j,n}$ is added:



$$= \begin{array}{c} \begin{array}{c} \text{\scriptsize } n-1 \qquad \qquad \text{\scriptsize } n-1 \\ \hline \text{\scriptsize } ff^{\circ} \\ \hline \end{array} \\ \text{\scriptsize } \oplus \\ \text{\scriptsize } \leftarrow \qquad \qquad \rightarrow \end{array} \quad [\mathbf{CNOT.2}], [\mathbf{CNOT.8}]$$

For cnot: Push the cnot gate past all the literals with **[CNOT.8]**, until by **[CNOT.1]**, cnot annihilates with $\text{cnot}^{-1} = \text{cnot}$.

Note, that given any clause, for every literal controlled from the target of cnot a literal controlled on the control bit of cnot will be added to said clause. Therefore, as the cnot gates are being pushed inwards, the clauses which they are pushed past remain clauses.

(ii) Given a circuit $t = d_1 d_2 \cdots d_m$ in clausal form where d_1, \dots, d_m are clauses,

$$tt = d_1 \cdots d_m d_1 \cdots d_m = d_1^2 d_2^2 \cdots d_m^2 = d_1 d_2 \cdots d_m$$

as clauses commute by **[CNOT.5]** and are idempotent by Lemma 5.5.15. Therefore, as CNOT is an inverse category t is a restriction idempotent.

□

Theorem 5.5.17. $\tilde{H}_0 : \mathbf{CNOT} \rightarrow \mathbf{ParIso}(\mathbf{CTor}_2)^*$ is full and faithful on restriction idempotents.

Proof. A restriction in $\mathbf{Par}(\mathbf{Tor}_2)$ is given by a span in which both legs are equal and, thus, monic. Thus, restrictions correspond precisely to subobjects in \mathbf{Tor}_2 . However, these are determined by sets of torsor equations of the form:

$$\left\{ \sum_j b_{i,j} = a_i \right\}_i$$

Each equation, $\sum_j b_{i,j} = a_i$, corresponds in turn to a clause which picks out the wires $b_{i,j}$ and has output ancillary bit $\langle a_i |$. This immediately means that \tilde{H}_0 is full on restriction idempotents.

Remark that an arbitrary restriction idempotent expressed as a circuit in clausal form, under \widetilde{H}_0 , corresponds to a set of equations. We must show that two restriction idempotents in CNOT, whose corresponding sets of equations are equivalent in CTor_2 , must be equal in CNOT. This amounts to showing that we can perform Gaussian elimination on clauses in CNOT, as two sets of equations are equivalent in CTor_2 if and only if they can be shown so by Gaussian elimination steps.

Given two clauses c and c' we show that we can perform the Gaussian elimination step

$$\{c, c'\} \mapsto \{c, c + c'\}$$

and maintain equality.

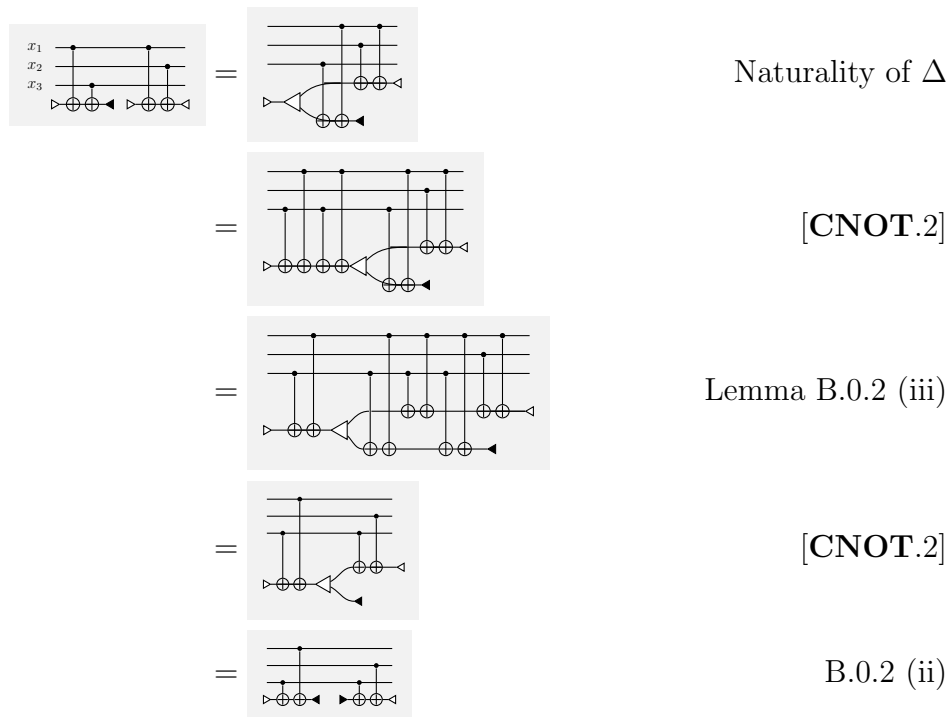
We first join the input ancillary bits of both clause wires into $|0\rangle \otimes |0\rangle := \Delta \circ |0\rangle$ using naturality of Δ .

By [CNOT.2], we copy two copies of each literal in c to the right of the input ancilla. By Lemma B.0.2 (iii) push one copy of each new literal through Δ . On one wire all of the literals will annihilate, and on the other only the common literals between c and c' will annihilate. Use Lemma B.0.2 (i) and B.0.2 (ii) to split the literals to the left and right of Δ . This may have shifted the input ancilla of the second clause to be $|1\rangle$. In this case, use [CNOT.2] to push a **not** gate from the left to right of the clause wire and negate the output ancillary bit of the second clause. The result is two clauses corresponding to the Gaussian elimination step. Therefore, we can perform Gaussian elimination on clauses in CNOT.

For example, suppose we are given a circuit determined by the equations:

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

We can perform Gaussian elimination as follows



Which represents the reduced system of linear equations:

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Hence, if the image of two circuits are equal under the functor \tilde{H}_0 , then they are the same. □

5.5.5 $\tilde{H}_0 : \text{CNOT} \rightarrow \text{Parlso}(\text{CTor}_2)^*$ is a discrete inverse equivalence

We already know that $\tilde{H}_0 : \text{CNOT} \rightarrow \text{Parlso}(\text{CTor}_2)^*$ is discrete inverse functor. It remains to show that this functor is essentially surjective, full and faithful.

In order to prove that \tilde{H}_0 is essentially surjective, we invoke the alternative characterization of CTor_2 given by Proposition 5.4.3.

Proposition 5.5.18. \tilde{H}_0 is essentially surjective.

Proof. Consider any torsor (X, \times) in $\mathbf{ParIso}(\mathbf{CTor}_2)^*$. There is some $n \in \mathbb{N}$ such that $(X, \times) \cong (\mathbb{Z}_2^n, - \oplus - \oplus -)$ by Proposition 5.4.3. However, by Lemma 5.5.3, $\mathbf{Total}(\mathbf{CNOT})(0, n) \cong \mathbb{Z}_2^n$. \square

We can simulate total maps in \mathbf{CTor}_2 :

Lemma 5.5.19. If $f \in \mathbf{CTor}_2(\mathbb{Z}_2^n, \mathbb{Z}_2^m)$, then there is a map $g \in \mathbf{CNOT}(n, m)$ with $\tilde{H}_0(g) = \langle 1, f \rangle$.

Proof. Consider $f \in \mathbf{CTor}_2(\mathbb{Z}_2^n, \mathbb{Z}_2^m)$. Recall that f may be regarded as a linear map $t : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$ with a shift $b : m \rightarrow m$. Consider the standard bases $\{e_i\}$ and $\{m_j\}$ of \mathbb{Z}_2^n and \mathbb{Z}_2^m , respectively. As t is a linear map, for any $1 \leq i \leq n$ there are unique coefficients $a_{i,j} \in \mathbb{Z}_2$ for all $1 \leq j \leq m$ such that:

$$f(e_i) = \sum_{j=1}^m a_{i,j} m_j$$

However, as \mathbb{Z}_2^n is a vector space over \mathbb{Z}_2 , the coefficients $a_{i,j}$ are either 0 or 1 so they determine for each i a subset of the m_i .

Consider the circuit

$$g := g_1 \circ g_2 \circ (1_n \otimes |0, \dots, 0\rangle)$$

Where g_1 is the composite of literals $l_{i,n+j} \otimes 1_{m-j}$ for which $a_{i,j} = 1$. And g_2 is the composite of literals $l_{n+j} \otimes 1_{m-j}$ for which $b_j = 1$.

Given any $(c_1, \dots, c_n) \in \mathbb{Z}_2^n$, by Lemma 5.5.2:

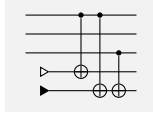
$$g \circ |c_1, \dots, c_n\rangle = |c_1, \dots, c_n\rangle \otimes \left(\bigotimes_{j=1}^m \left| b_j + \sum_{i=1}^n a_{i,j} c_i \right\rangle \right)$$

Therefore, $\tilde{H}_0(g)(c_1, \dots, c_n) = f(c_1, \dots, c_n)$ and thus $\tilde{H}_0(g) = f$.

For example, consider the map $f \in \mathbf{CTor}_2(\mathbb{Z}_2^3, \mathbb{Z}_2^2)$ given by the affine transformation with a linear component T and shift S such that:

$$T = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad S = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Then the corresponding circuit g such that $\tilde{H}_0(g) = \langle 1, f \rangle$ is:

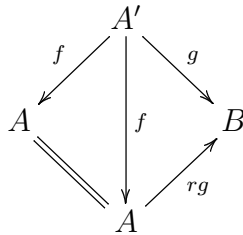


□

We have almost proven fullness; however, we must combine the fullness results for total maps and restriction idempotents:

Proposition 5.5.20. $\tilde{H}_0 : \text{CNOT} \rightarrow \text{ParIso}(\text{CTor}_2)^*$ is full.

Proof. Suppose $A \xleftarrow{f} A' \xrightarrow{g} B$ is a partial isomorphism in $\text{Par}(\text{CTor}_2)^*$. Thus f and g are monics. If A' is empty we can simulate the map as $\tilde{H}_0(\emptyset_{n,m})$ for some $n, m \in \mathbb{N}$. On the other hand, if A' is non-empty then there is a total map r with $fr = 1_{A'}$ as the object A' is injective (as it is injective as a \mathbb{Z}_2 -vector space). This means the total map $A \xlongequal{rg} A \xrightarrow{rg} B$ extends (f, g) (so $(f, g) \leq (1_A, rg)$) as



But by Lemma 5.5.19 there is a map $k \in \text{CNOT}$ with $\tilde{H}_0(k) = \langle 1, rg \rangle$. By the fullness of \tilde{H}_0 on restriction idempotents there is an e with $\tilde{H}_0(e) = (f, f)$ but then

$$\tilde{H}_0(ek) = \tilde{H}_0(e)\tilde{H}_0(k) = (f, f)\langle 1_a, rg \rangle = \langle \overline{(f, g)}, (f, g) \rangle.$$

Similarly we can implement $\langle \overline{(g, f)}, (g, f) \rangle$ and therefore by Lemma 4.2.7 we can implement (f, g) . \square

Note that since \tilde{h}_0 is a strong \dagger -monoidal functor, by Lemma 4.2.4 it is also a discrete inverse functor. Therefore, by Lemma 4.2.6, it suffices to show that \tilde{H}_0 is faithful on restriction idempotents to prove that it is faithful. However, we already have proven that \tilde{H}_0 is faithful on idempotents in Theorem 5.5.17 so we have:

Proposition 5.5.21. $\tilde{H}_0 : \text{CNOT} \rightarrow \text{Parlso}(\text{CTor}_2)^*$ is faithful.

This implies:

Theorem 5.5.22. The identities of CNOT are complete.

Proof. Because $\tilde{H}_0 : \text{CNOT} \rightarrow \text{Parlso}(\text{CTor}_2)^*$ and the forgetful functor $\text{Parlso}(U) : \text{Parlso}(\text{CTor}_2)^* \rightarrow \text{FPinj}$ are strong \dagger -symmetric monoidal and faithful, so is their composite $\tilde{H}_0 \text{Parlso}(U) : \text{CNOT} \rightarrow \text{FPinj}$. Post-composing $\tilde{H}_0 \text{Parlso}(U) : \text{CNOT} \rightarrow \text{FPinj}$ with the strong \dagger -symmetric monoidal, faithful functor $\ell^2 : \text{FPinj} \rightarrow \text{FHilb}$, we obtain a strong \dagger -symmetric monoidal, faithful functor $\tilde{H}_0 \ell^2 : \text{CNOT} \rightarrow \text{Hilb}$.

Recall the canonical interpretation $\llbracket - \rrbracket_{\text{CNOT}} : \text{CNOT} \rightarrow \text{Mat}_{\mathbb{C}}$ is a strict \dagger -symmetric monoidal, \dagger -reflecting functor.

Consider, moreover, the faithful strong \dagger -symmetric monoidal equivalence $F : \text{Mat}_{\mathbb{C}} \rightarrow \text{FHilb}$ taking objects n to the Hilbert space $\tilde{H}_0 \ell^2(n)$.

Remark that the following diagram of strong \dagger -symmetric monoidal functors commutes:

$$\begin{array}{ccc}
 \text{CNOT} & \xrightarrow{\tilde{H}_0} & \text{FPinj} \\
 \llbracket - \rrbracket_{\text{CNOT}} \downarrow & & \downarrow \ell^2 \\
 \text{Mat}_{\mathbb{C}} & \xrightarrow[\sim]{F} & \text{FHilb}
 \end{array}$$

Therefore, $\llbracket - \rrbracket_{\text{CNOT}}$ is faithful, so the identities are complete (as well as sound). \square

Because $\tilde{H}_0 : \text{TOF} \rightarrow \text{ParIso}(\text{CTor}_2)^*$ is a discrete inverse functor, which is full, faithful and essentially surjective:

Theorem 5.5.23. There is an equivalence of categories preserving discrete inverse category structure, between CNOT and $\text{ParIso}(\text{CTor})^*$.

Chapter 6

CNOT and ZX_π

In this chapter, we relate the category CNOT to a category which is complete for real stabilizers, namely the angle-free fragment of the ZX-calculus: ZX_π .

6.1 ZX_π

The **ZX-calculus** is a collection of calculi describing the interaction of the complementary Frobenius algebras corresponding to the Pauli Z and X observables and their phases. The first iteration of the ZX-calculus was described in [17].

However, we are interested in a simple fragment of the ZX-calculus, namely the angle-free calculus for real stabilizer circuits, ZX_π , described in [22] (slightly modified to account for scalars):

Definition 6.1.1. Let ZX_π denote the \dagger -compact closed PROP with generators:



such that

$$(\text{circle with dot on left}, \text{circle with dot on right}, \text{circle with dot on top}, \text{circle with dot on bottom})$$

is a classical structure, corresponding to the Z basis, and the following identities also hold

up to swapping colours:

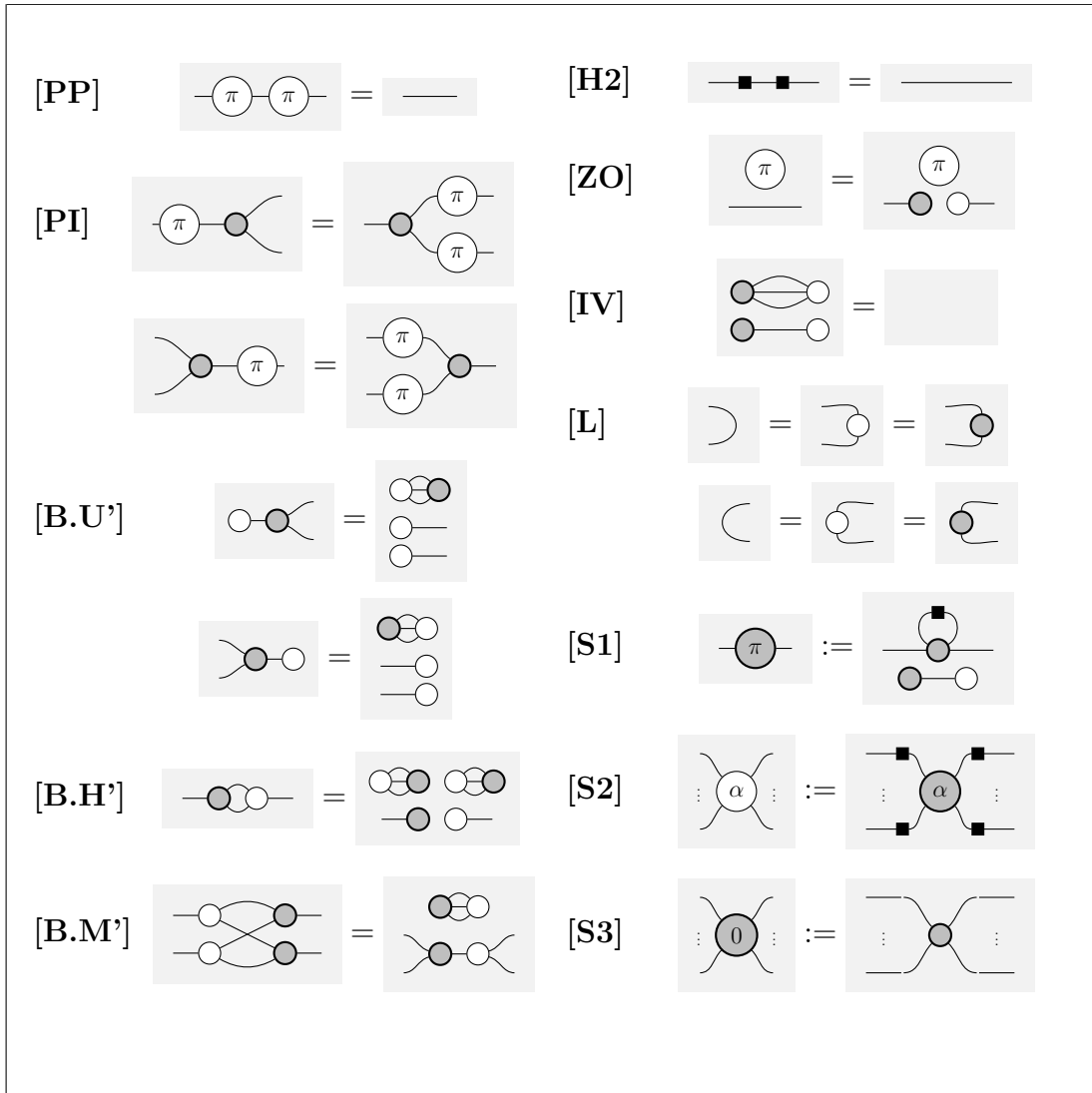


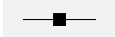
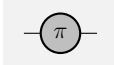
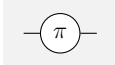
Figure 6.1: The identities of ZX_{π} (where $\alpha \in \{0, \pi\}$)

The last 3 Axioms are actually definitions, which simplify the presentation of ZX_{π} . Note that the axioms of a classical structure are omitted from this box to save space.

These axioms imply that the black and white Frobenius algebras are complementary where the antipode is the identity.

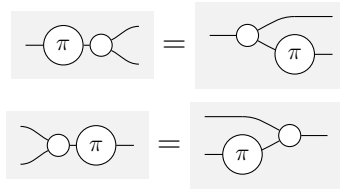
This category has a canonical \dagger -functor, as all of the stated axioms are horizontally

symmetric. It is also \dagger -compact closed.

This category embeds \mathbf{FHilb} ; the black Frobenius algebra corresponds to the Pauli Z basis; the white Frobenius algebra corresponds to the Pauli X basis; the gate  corresponds to the Hadamard gate and  and  correspond to Z and X π -phase-shifts respectively. In particular, the X π -phase-shift is the not gate (see Appendix A for the aforementioned matrices).

Because the π -phases are given by **[S3]**, by the commutative spider theorem, it is immediate that they are phase shifts:

[PH]



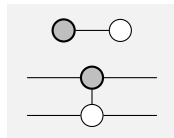
In bra-ket notation, a black spider from n to m with angle θ is interpreted as follows in \mathbf{FHilb} :

$$|0\rangle^{\otimes n} \circ \langle 0|^{\otimes m} + e^{i\theta} |1\rangle^{\otimes n} \circ \langle 1|^{\otimes m}$$

and a white spider from n to m with angle θ is interpreted as follows in \mathbf{FHilb} :

$$|+\rangle^{\otimes n} \circ \langle +|^{\otimes m} + e^{i\theta} |-\rangle^{\otimes n} \circ \langle -|^{\otimes m}$$

Note that the controlled-not gate has a succinct representation in \mathbf{ZX}_π (this can be verified by calculation):



This means that \mathbf{ZX}_π contains all of the generators of the real Clifford group. Furthermore, the following is known:

Theorem 6.1.2. [22]

\mathbf{ZX}_π is complete for real stabilizer states.

The original presentation of ZX_π in [22] did not account for scalars; instead, it imposed the equivalence relation on circuits up to an invertible scalar and ignored the zero scalar entirely. Therefore, the original completeness result described in [22] is not actually as strong as Theorem 6.1.2. This means, of course, that this original calculus does not embed in $\text{Mat}_{\mathbb{C}}$ as the relations are not sound. For example, the following map is interpreted as $\sqrt{2}$, not 1, in $\text{Mat}_{\mathbb{C}}$:



Later on, [6, 9] showed that by scaling certain axioms to make them sound, and by adding Axioms [IV] and [ZO] this fragment of the ZX-calculus is also complete for scalars. The properly scaled axioms have all been collected in Figure 6.1.

6.2 Embedding CNOT into ZX_π

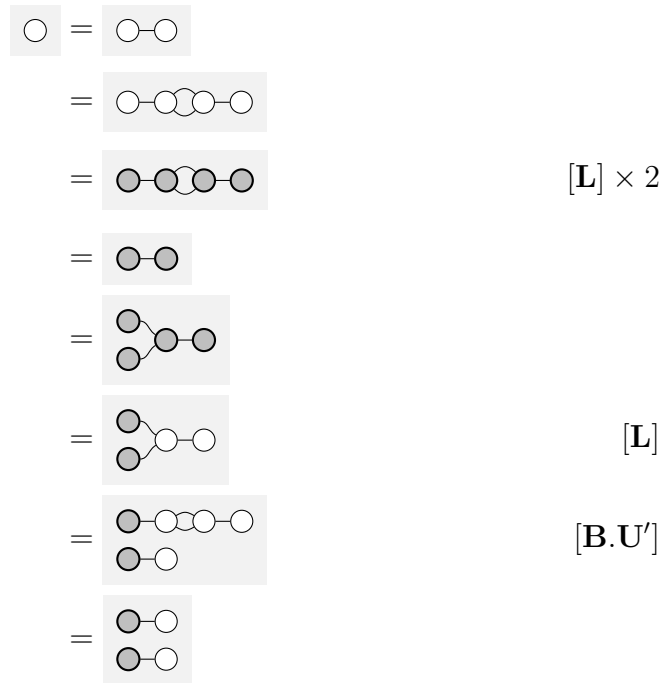
Consider the interpretation of CNOT into ZX_π , sending:



First we observe:

Lemma 6.2.1. [51, Lemma 19] $\bigcirc = \begin{array}{c} \bullet \circ \\ \bullet \circ \end{array}$

Proof.



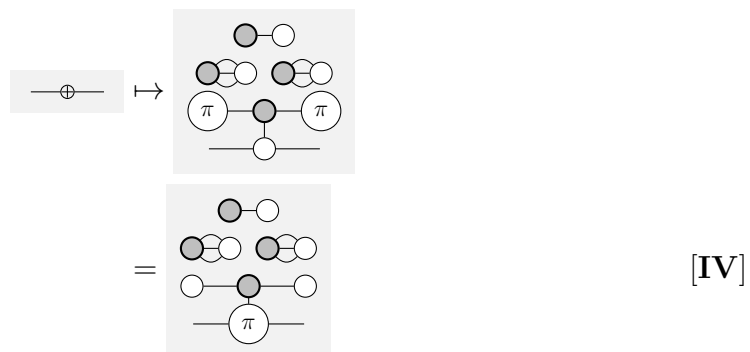
□

We also need:

Lemma 6.2.2.



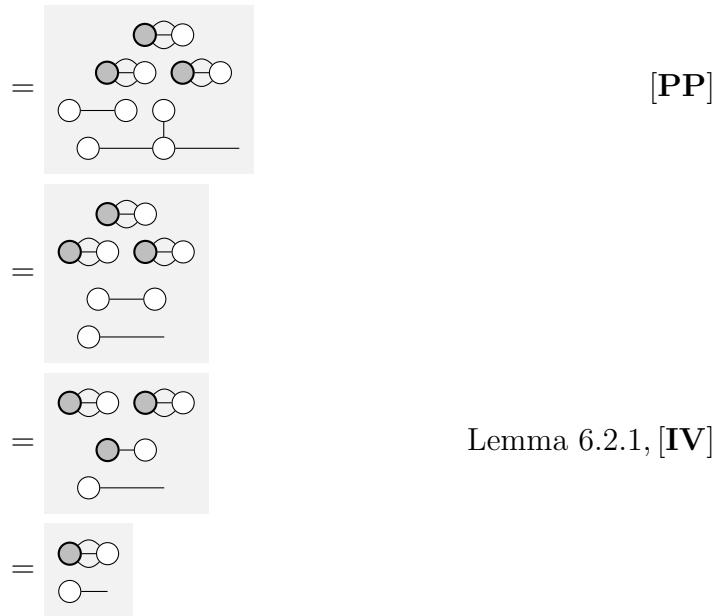
Proof. (i)



$$\begin{aligned}
&= \begin{array}{c} \text{[Diagram: 2 rows of 4 circles, top row has 2 shaded circles]} \\ \text{[Diagram: 1 row of 3 circles]} \\ \text{[Diagram: circle } \pi \text{ with line below]} \end{array} & \text{[B.U']} \\
&= \begin{array}{c} \text{[Diagram: 2 rows of 4 circles, top row has 2 shaded circles]} \\ \text{[Diagram: 1 circle]} \\ \text{[Diagram: circle } \pi \text{ with line below]} \end{array} \\
&= \begin{array}{c} \text{[Diagram: 2 rows of 4 circles, top row has 2 shaded circles]} \\ \text{[Diagram: 1 row of 4 circles, top 2 shaded]} \\ \text{[Diagram: circle } \pi \text{ with line below]} \end{array} & \text{Lemma 6.2.1} \\
&= \begin{array}{c} \text{[Diagram: circle } \pi \text{ with line below]} \end{array} & \text{[IV]}
\end{aligned}$$

(ii)

$$\begin{aligned}
\text{[Diagram: triangle symbol]} &:= \text{[Diagram: triangle symbol with line and circle below]} \\
&\mapsto \begin{array}{c} \text{[Diagram: 2 rows of 4 circles, top row has 2 shaded circles]} \\ \text{[Diagram: 1 row of 4 circles, top 2 shaded]} \\ \text{[Diagram: circle } \pi \text{ with line below]} \\ \text{[Diagram: circle } \pi \text{ with line below]} \end{array} \\
&= \begin{array}{c} \text{[Diagram: 2 rows of 4 circles, top row has 2 shaded circles]} \\ \text{[Diagram: 1 row of 4 circles, top 2 shaded]} \\ \text{[Diagram: circle } \pi \text{ with line below]} \\ \text{[Diagram: circle } \pi \text{ with line below]} \end{array} & \text{[PI]} \\
&= \begin{array}{c} \text{[Diagram: 2 rows of 4 circles, top row has 2 shaded circles]} \\ \text{[Diagram: 1 row of 4 circles, top 2 shaded]} \\ \text{[Diagram: circle } \pi \text{ with line below]} \\ \text{[Diagram: circle } \pi \text{ with line below]} \end{array} & \text{[PH]} \\
&= \begin{array}{c} \text{[Diagram: 1 row of 3 circles]} \\ \text{[Diagram: 2 rows of 4 circles, top row has 2 shaded circles]} \\ \text{[Diagram: circle } \pi \text{ with line below]} \\ \text{[Diagram: circle } \pi \text{ with line below]} \end{array} & \text{[B.U']}
\end{aligned}$$



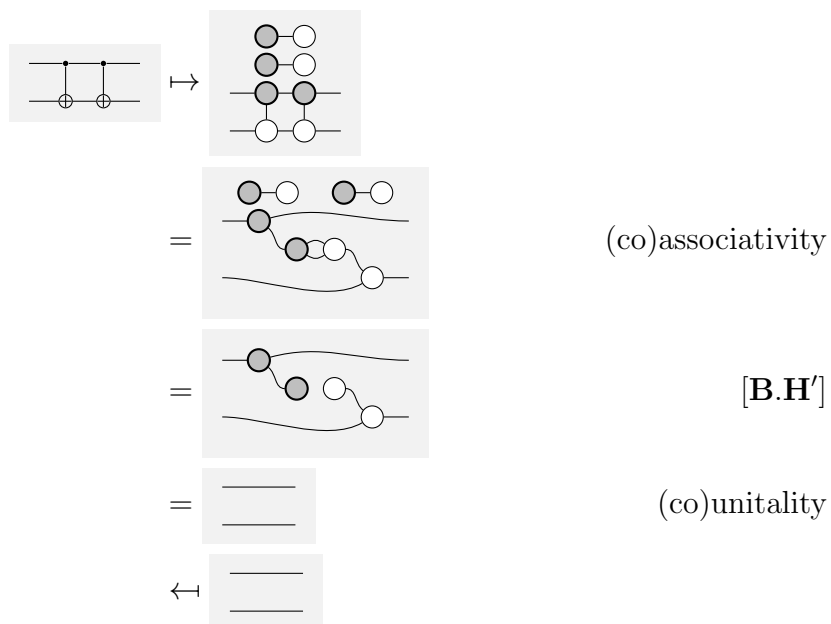
□

We explicitly prove that this interpretation is functorial.

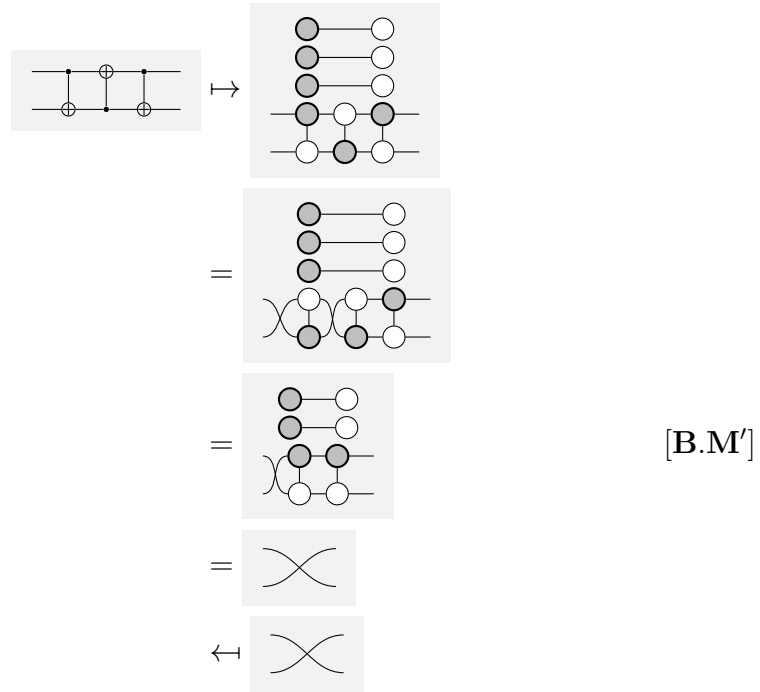
Lemma 6.2.3. The interpretation of CNOT into ZX_π is functorial.

Proof. We prove that each axiom holds

[CNOT.2]

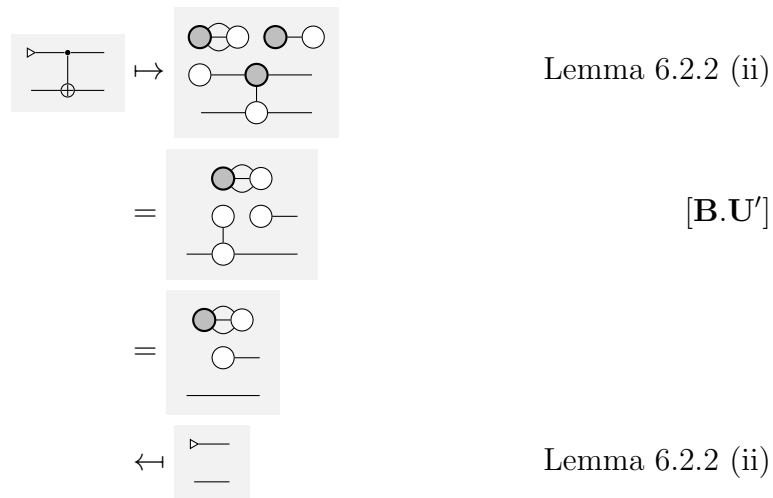


[CNOT.1]



[CNOT.3] Immediate from commutative spider theorem.

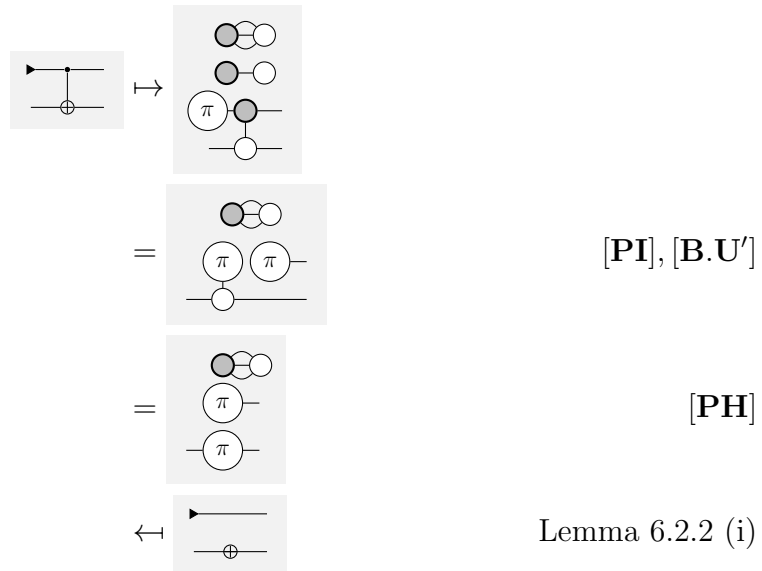
[CNOT.4]



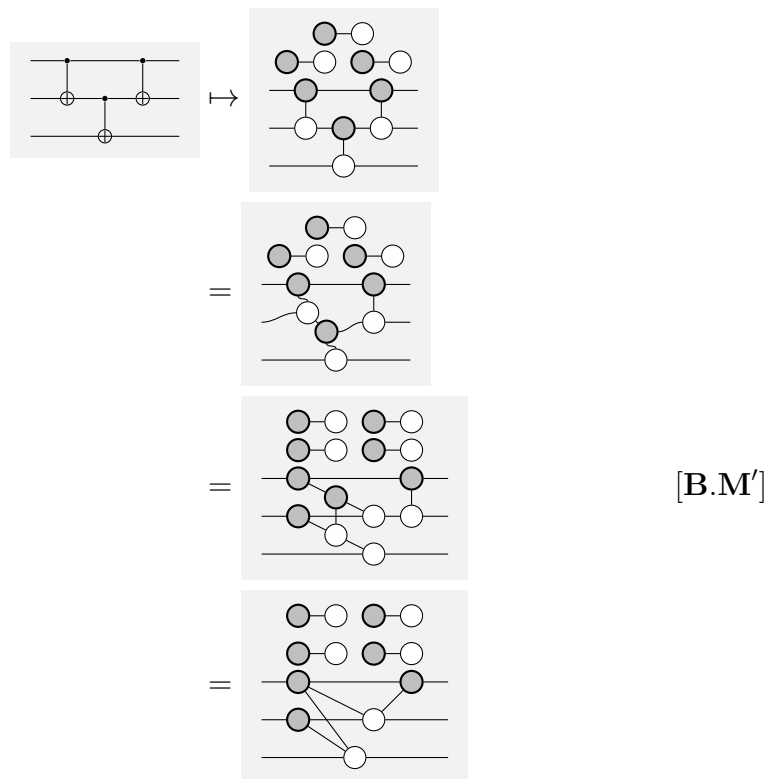
[CNOT.5] Immediate from commutative spider theorem.

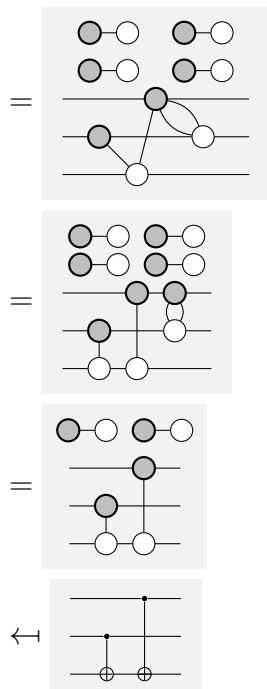
[CNOT.6] Frobenius algebra is special.

[CNOT.7]



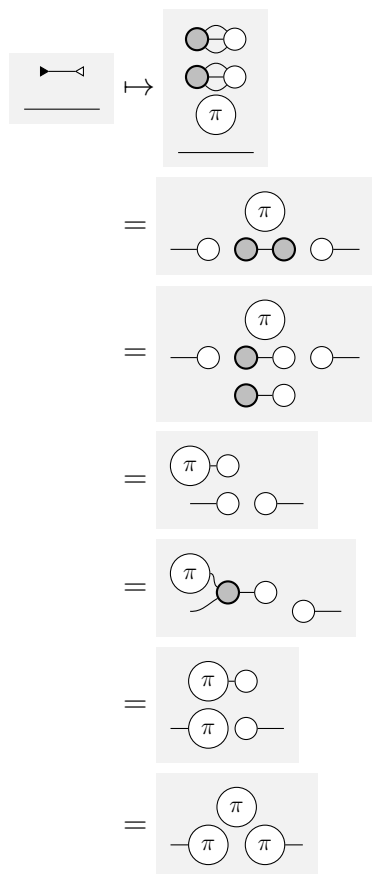
[CNOT.8]





[B.H']

[CNOT.9]



Lemma 6.2.2 (ii)

[ZO]

Lemma 6.2.1

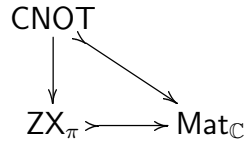
[B.U']

By symmetry



□

Because of the various completeness results discussed so far, the following diagram of strict \dagger -symmetric monoidal functors commutes, making this functor $\text{CNOT} \rightarrow \text{ZX}_\pi$ faithful:



6.3 Extending CNOT to ZX_π

As opposed to the ZX-calculus, the identities of CNOT are given in terms of *circuit relations*. When applying rules of the ZX calculus, circuits can be transformed into intermediary representations so that the flow of information is lost. Various authors have found complete circuit relations for various fragments of quantum computing. Notably, Selinger found a complete set of identities for Clifford circuits (stabilizer circuits without ancillary bits) [47]. Similarly, Amy et al. found a complete set of identities for cnot-dihedral circuits (without ancillary bits) [4].

In this section, we provide a complete set of circuit relations for *real* stabilizer circuits (although circuits can have norms greater than 1). We show that CNOT is embedded in ZX_π and we complete CNOT to ZX_π by adding the Hadamard gate and the scalar $\sqrt{2}$ as generators along with 5 relations.

Definition 6.3.1. Let $\text{CNOT} + H$ denote the PROP freely generated by the axioms of CNOT with additional generators the Hadamard gate and $\sqrt{2}$:



satisfying the following identities:

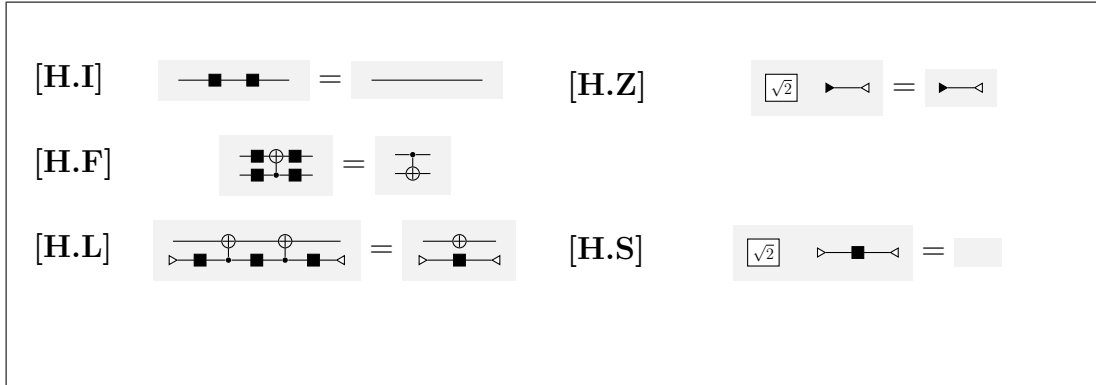
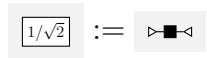


Figure 6.2: The identities of CNOT + H (in addition to the identities of CNOT)

The inverse of $\sqrt{2}$ is given the alias:



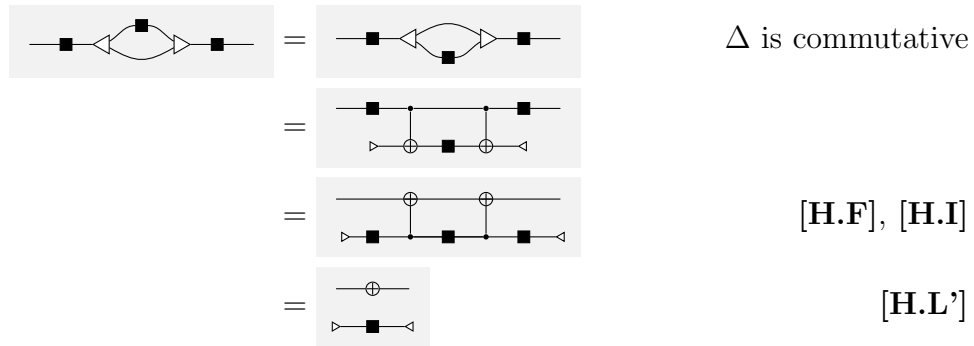
[H.I] is stating that the Hadamard gate is self-inverse. [H.F] reflects the fact that composing the controlled-not gate with Hadamards reverses the control and operating bits. [H.Z] is stating that $\sqrt{2}$ composed with the zero matrix is again the zero matrix. [H.S] makes $\sqrt{2}$ and $1/\sqrt{2}$ inverses to each other.

[H.L] can be restated to resemble [S1]:

Lemma 6.3.2.



Proof.



□

[H.Z] can be restated in slightly different terms:

Lemma 6.3.3.

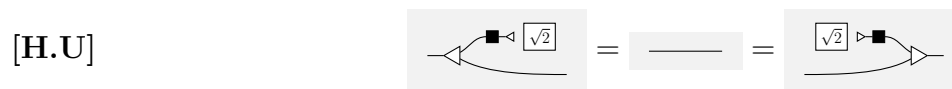


Proof. Immediate by [H.S] and [H.Z].

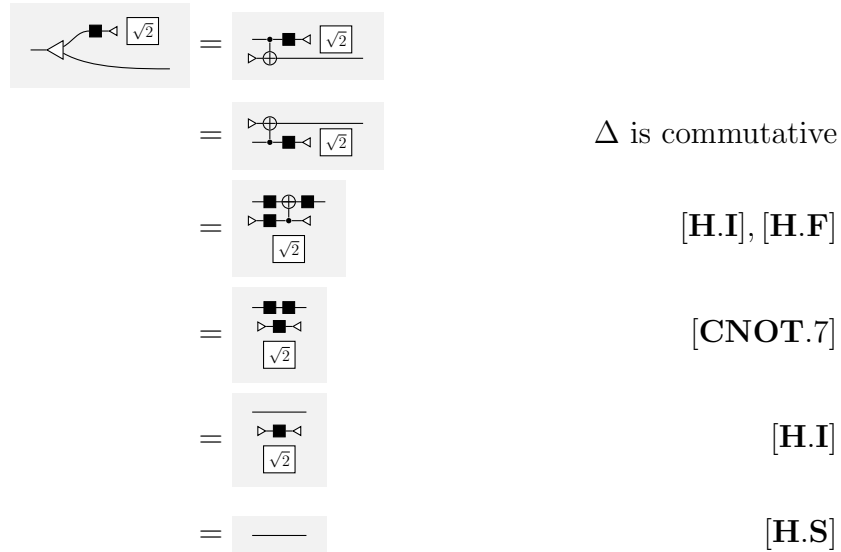
□

There is a derived identity, showing that the Frobenius structure identified with the inverse products of CNOT is unital:

Lemma 6.3.4.



Proof.



□

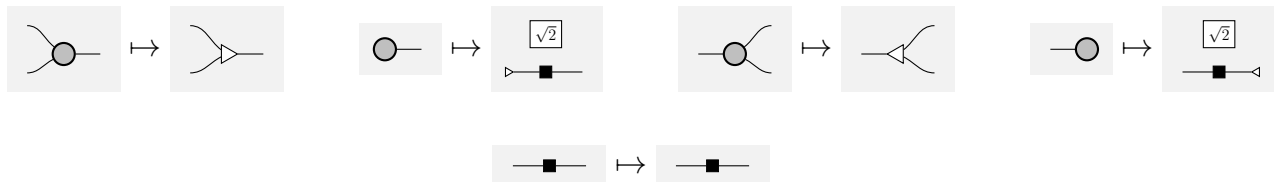
6.3.1 The completeness of CNOT + H

We construct two functors between CNOT + H and ZX_π and show that they are pairwise inverses.

Definition 6.3.5. Let $F : \text{CNOT} + H \rightarrow \text{ZX}_\pi$, be the extension of the interpretation CNOT → ZX_π which takes:



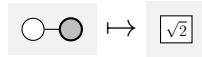
Let $G : \text{ZX}_\pi \rightarrow \text{CNOT} + H$ be the interpretation sending:



We first observe:

Lemma 6.3.6.

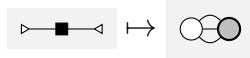
(i)



(ii)

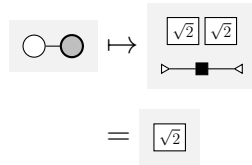


(iii)



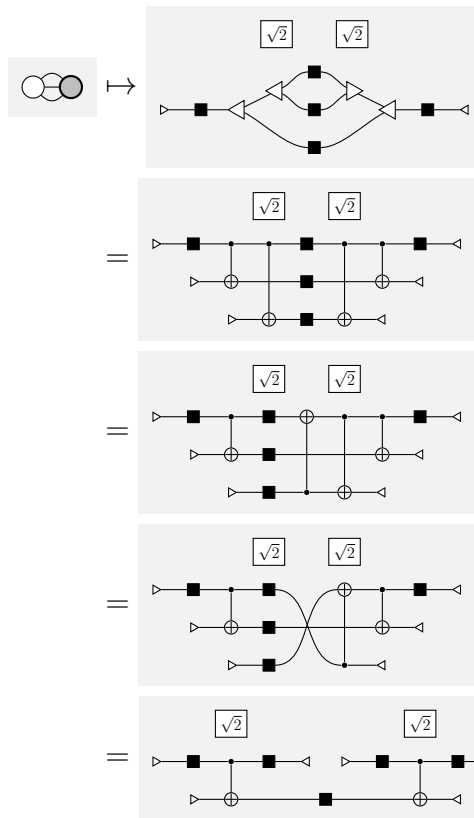
Proof.

(i)



[H.S]

(ii)



[H.F]

[CNOT.1], [CNOT.2]

$$\begin{aligned}
&= \text{[H.F]} \\
&= \text{[H.F]} \\
&= \text{Lemma B.0.2 (i)} \\
&= \\
&= \text{[H.S]} \\
&=
\end{aligned}$$

(iii)

$$\begin{aligned}
&\mapsto \\
&= \text{[H.S]}
\end{aligned}$$

□

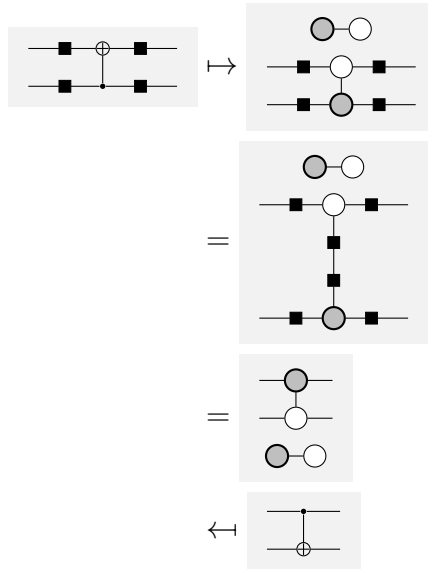
We show that these interpretations are functors:

Lemma 6.3.7. $F : \text{CNOT} + H \rightarrow \text{ZX}_\pi$ is a strict \dagger -symmetric monoidal functor.

Proof. The preservation of the \dagger -symmetric monoidal structure is immediate. As the restriction of F to CNOT is a functor, it suffices to show that [H.I], [H.F], [H.U], [H.L'], [H.S] and [H.Z'] hold.

[H.I] Immediate.

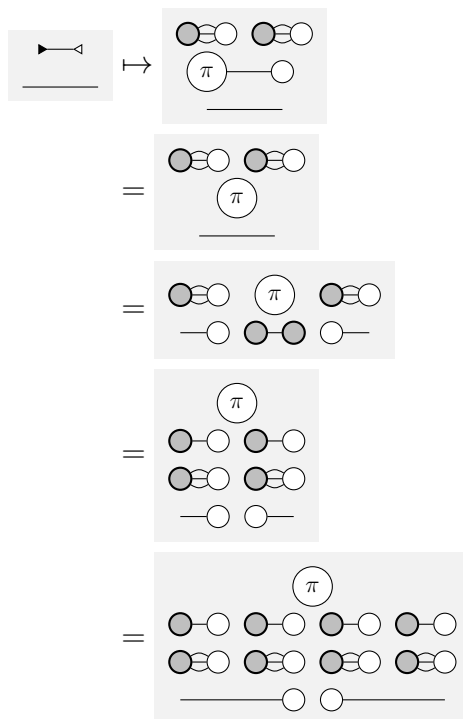
[H.F]



[H.L'] Immediate.

[H.S] Immediate.

[H.Z']

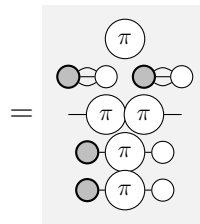
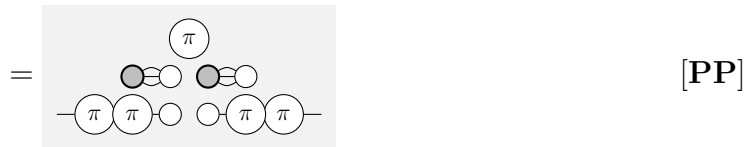
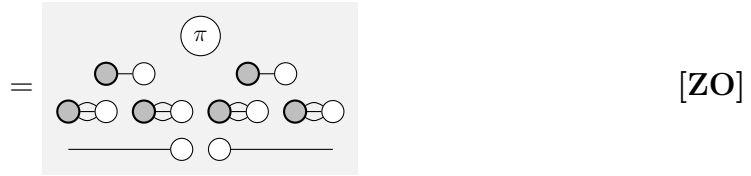


Lemma 6.2.2 (ii)

[ZO]

Lemma 6.2.1

[IV]



$$\begin{array}{c}
\begin{array}{c} \pi \\ \bullet \text{---} \bullet \text{---} \bullet \\ \pi \text{---} \pi \\ \bullet \text{---} \bullet \\ \bullet \text{---} \bullet \end{array} \\
= \\
\begin{array}{c} \pi \\ \bullet \text{---} \bullet \text{---} \bullet \text{---} \bullet \\ \pi \text{---} \pi \\ \bullet \text{---} \bullet \\ \bullet \text{---} \bullet \end{array} \\
\leftarrow \\
\begin{array}{c} \blacktriangleleft \text{---} \blacktriangleright \\ \blackrightarrow \text{---} \blackleftarrow \end{array}
\end{array}
\quad \begin{array}{l} \text{[ZO]} \\ \\ \text{Lemma 6.2.2 (ii)} \end{array}$$

□

For the other way around:

Lemma 6.3.8. $G : ZX_\pi \rightarrow \text{CNOT} + H$ is a strict \dagger -symmetric monoidal functor.

Proof. We prove that each axiom holds:

[PI] This follows by naturality of Δ in CNOT.

[B.U'] This follows by naturality of Δ in CNOT and [H.S].

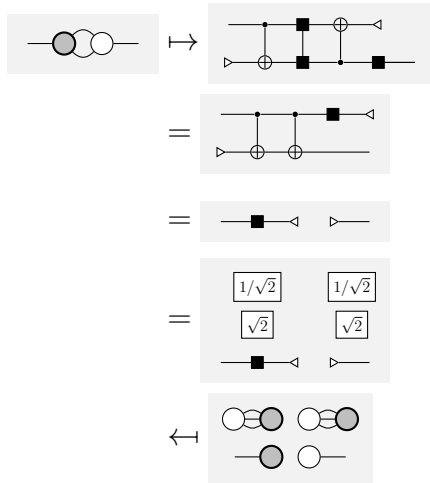
[H2] This follows immediately from [H.I].

[H2] This follows immediately from [H.S].

[PP]

$$\begin{array}{c}
\begin{array}{c} \pi \text{---} \pi \end{array} \mapsto \begin{array}{c} \blacksquare \oplus \blacksquare \blacksquare \oplus \blacksquare \end{array} \\
= \begin{array}{c} \blacksquare \oplus \oplus \blacksquare \end{array} \\
= \begin{array}{c} \blacksquare \blacksquare \end{array} \\
= \begin{array}{c} \text{---} \end{array} \\
\leftarrow \begin{array}{c} \text{---} \end{array}
\end{array}
\quad \begin{array}{l} \text{[H.I]} \\ \\ \text{[H.I]} \end{array}$$

[B.H']



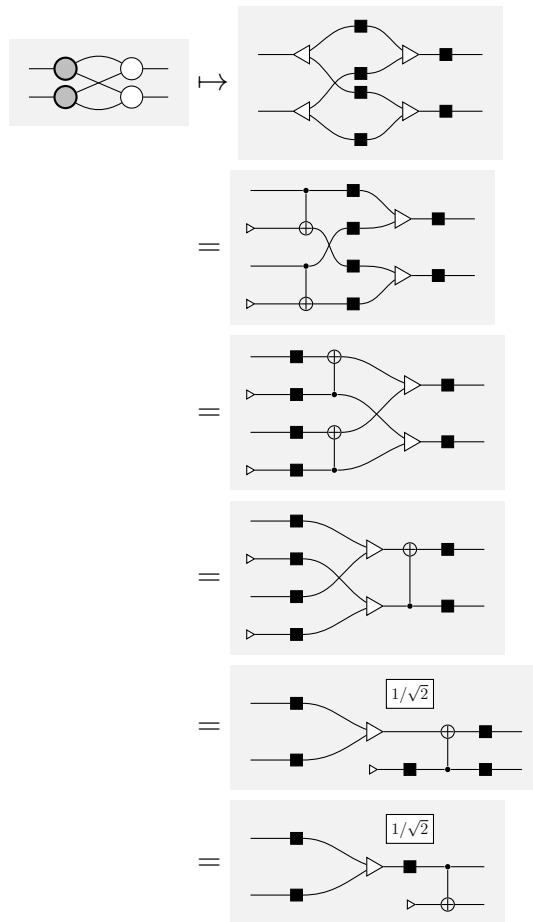
[H.F]

[CNOT.2]

[H.Z']

Lemma 6.3.6 (ii) $\times 2$

[B.M']

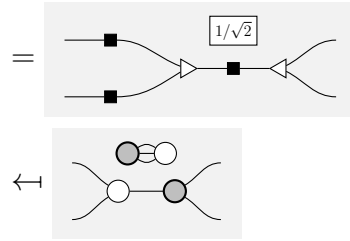


[H.F]

Δ natural in CNOT

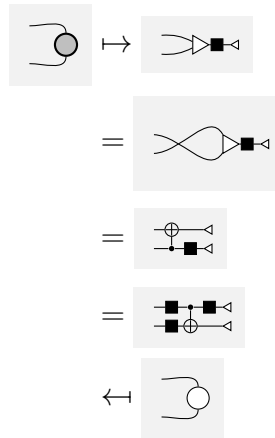
[H.U]

[H.F]



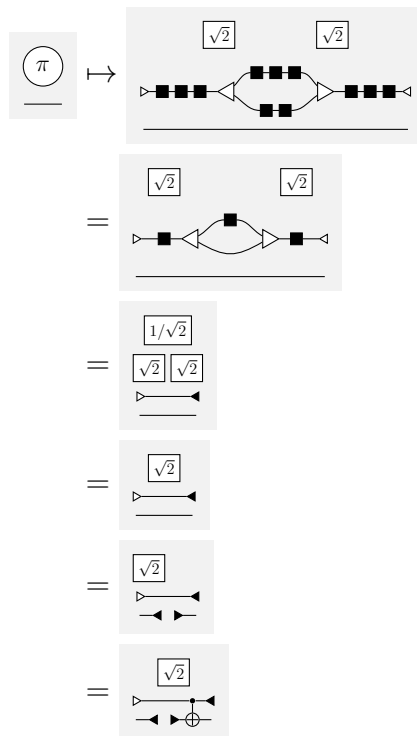
Lemma 6.3.6 (ii)

[L]



[H.F]

[ZO]

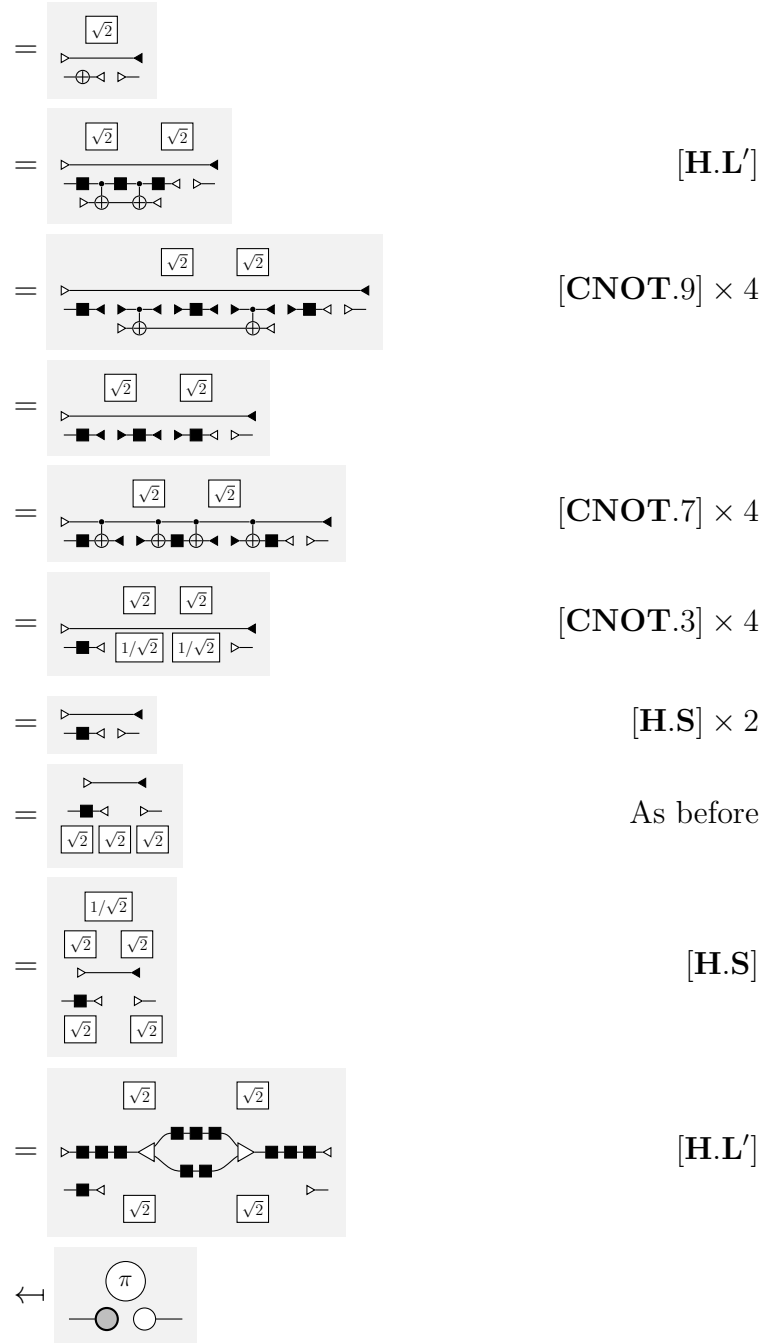


[H.I]

[H.S]

[CNOT.9]

[CNOT.7]



Classical structure: Remark that rules [H.U] and [H.S] complete the semi-Frobenius structure to the appropriate classical structure.

□

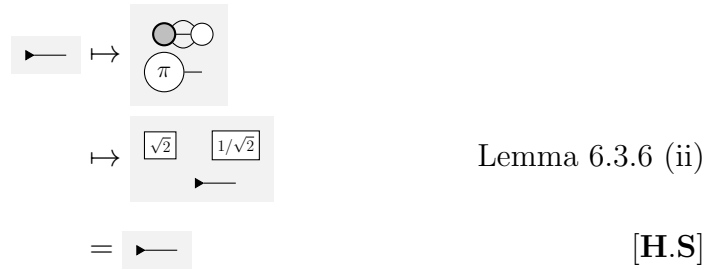
Proposition 6.3.9. $\text{CNOT} + H \xrightarrow{F} \text{ZX}_\pi$ and $\text{ZX}_\pi \xrightarrow{G} \text{CNOT} + H$ are inverses.

Proof.

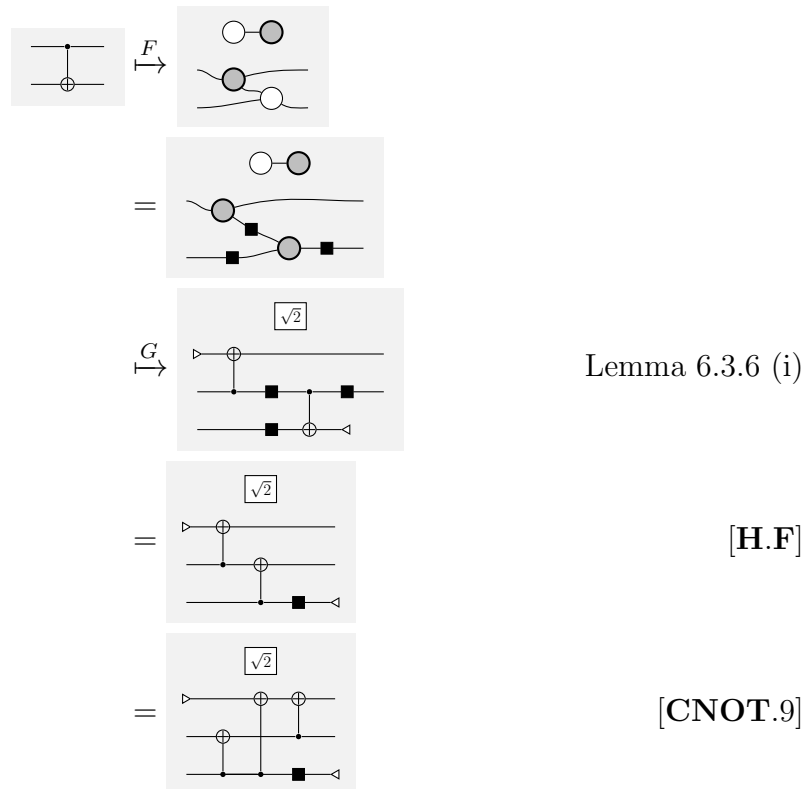
1. First, we show that $GF = 1$.

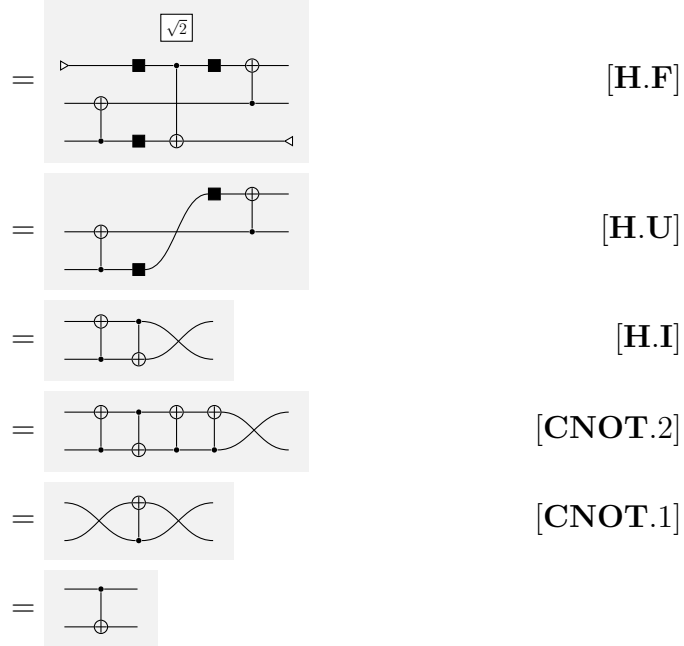
We only prove the cases for the generators **cnot** and $|1\rangle$ as the claim follows trivially for the Hadamard gate and by symmetry for $\langle 1|$:

For $|1\rangle$:



For cnot:





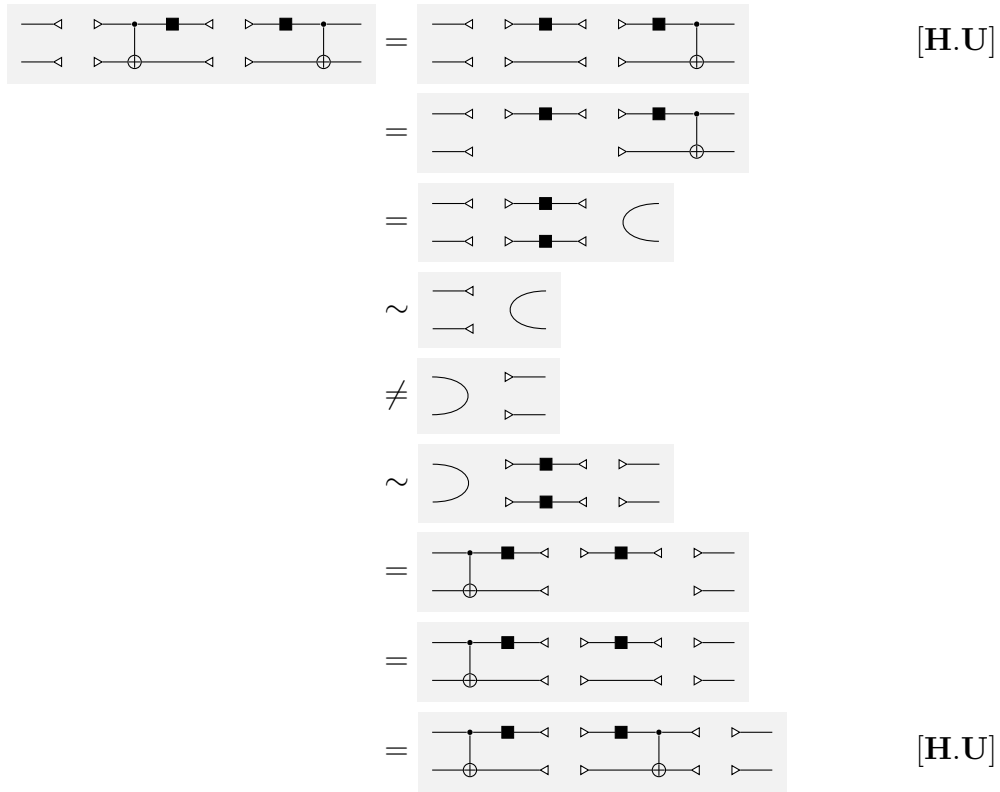
It is trivial to observe that $GF = 1$.

□

Because all of the axioms of $\text{CNOT} + H$ and ZX_π satisfy the same “horizontal symmetry”; we can not only conclude that they are isomorphic, but rather:

Theorem 6.3.10. $\text{CNOT} + H$ and ZX_π are strictly \dagger -symmetric monoidally isomorphic.

As CNOT is a discrete inverse category, and the dagger of $\text{CNOT} + H$ extends the dagger of CNOT , it is natural to ask if the inverse category structure, or inverse products extend to $\text{CNOT} + H$. It clearly isn’t a discrete inverse category, because the no-cloning-theorem forces Δ to not be natural. Even if we don’t ask for inverse products, one will find that the non-zero scalars which aren’t units cause **[INV.2]** to fail. Even if we quotient by scalars, we still fail to obtain a discrete inverse category. Fix $f := \langle 0, 0 |$ and $g := \langle 0, 0 | \circ (H \otimes 1) \circ \text{cnot}$. As g is the subnormalised counit for the compact closed structure, it is trivial to observe that $ff^\dagger gg^\dagger \neq gg^\dagger ff^\dagger$, by examining the connectivity of both (non-zero) circuits:



We see that compact closed inverse categories are pathological; delineating classical from quantum computing. In the classical setting, we have copying; on the other hand, in the quantum setting, we instead have entanglement.

Chapter 7

The Toffoli Gate

The Toffoli gate generalizes the controlled-not gate. It is the 3-qubit unitary matrix taking $|b_1, b_2, b_3\rangle$ to $|b_1, b_2, b_1 \cdot b_2 \oplus b_3\rangle$. One can see that `cnot` can be simulated with a Toffoli gate by restricting $|b_1\rangle := |1\rangle$, so that $|1, b_2, b_3\rangle$ gets sent to $|1, b_2, b_2 \oplus b_3\rangle$.

The Toffoli gate is a cornerstone for reversible computing. It was among the first gates proven to be universal for quantum computing [23]. The Toffoli gate is also used frequently in quantum computation (especially in quantum error correction [49, 25]); and has even been physically realized [42, 45].

This chapter builds on Chapter 5 and provides a finite, complete set of identities for the fragment of quantum computing generated by the Toffoli gate with *ancillary bits*. Recall that ancillary bits are a more general notion than auxiliary bits. We call the PROP generated by the Toffoli gate and ancillary bits with these identities, TOF. In fact, generalized controlled-not gates are derivable in TOF. Generalized controlled-not gates are not gates with arbitrarily many control bits; the not gate, controlled-not gate and Toffoli gate being examples thereof.

There is already an *infinite*, complete set of identities for circuits comprised of generalized controlled-not gates and *auxiliary bits* [33]. However, we provide a *finite*, complete set of identities for circuits comprised of Toffoli gates and *ancillary bits*, generalizing this result. Without ancillary bits, by [38, Lemma 14], it is not possible to give a *finite* complete set of

identities for circuits with generalized controlled-not gates.

In addition to providing a complete set of identities for these circuits, we also prove a concrete equivalence into the full subcategory of Pinj where the objects are finite powers of the 2 element set. The key step of this proof is to prove that the functor \tilde{H}_0 , which takes an object to its (total) points, is full and faithful: this, in turn, relies on providing a normal form for the restriction idempotents of TOF . This is the same approach that was taken in Chapter 5.

7.1 The category TOF

Define the category TOF to be the PROP, generated by the 1 ancillary bits $|1\rangle$ and $\langle 1|$ (depicted graphically as in CNOT) as well as the Toffoli gate:

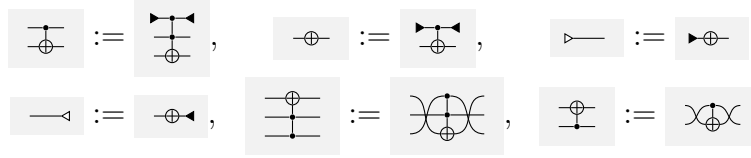
$$|1\rangle := \text{---}\blacktriangleright \qquad \langle 1| := \text{---}\blacktriangleleft \qquad \text{tof} := \begin{array}{c} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ \oplus \end{array}$$

These generators have an interpretation $\llbracket - \rrbracket_{\text{TOF}} : \text{TOF} \rightarrow \text{Mat}_{\mathbb{C}}$:

$$\llbracket |1\rangle \rrbracket_{\text{TOF}} := \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \llbracket \langle 1| \rrbracket_{\text{TOF}} := \begin{bmatrix} 0 & 1 \end{bmatrix} \qquad \llbracket \text{tof} \rrbracket_{\text{TOF}} := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The Toffoli gate and the 1-ancillary bits allow cnot , not , $|0\rangle$, $\langle 0|$, and flipped tof gate and

flipped **cnot** gate to be defined by:



We also allow for “gaps” in the **tof** and **cnot** gates, as in **CNOT**.

These gates must satisfy the identities given in Figure 7.1 ¹.

Axioms **[TOF.1]**-**[TOF.6]**, **[TOF.8]**-**[TOF.9]** are relatively intuitive. **[TOF.7]** corresponds to tensoring a matrix with the 1×1 zero matrix: this is useful for establishing the restriction structure of **TOF**. **[TOF.7]** is used for establishing a normal form for the restriction idempotents. Axioms **[TOF.10]**-**[TOF.13]** combined with **[TOF.9]** can be used to push **cnot/tof** gates past each other thereby generating another trailing **cnot/tof** gate: as show in Lemma 7.2.6 (v). **[TOF.14]** is inherited from **CNOT** and is used to establish the inverse products of **TOF**. **[TOF.15]** expresses that the order of the control bits does not matter for the Toffoli gate. **[TOF.16]** is similar to **[TOF.15]**, except for the 3-bit-controlled-not gate.

Up to sliding control wires and target wires, all identities are graphically horizontally symmetric. The flipped equivalentents of Axioms **[TOF.7]**, **[TOF.10]**, **[TOF.11]**, **[TOF.12]**, **[TOF.13]** follow from applying one of the rules **[TOF.3]**, **[TOF.4]**, **[TOF.5]** or **[TOF.6]**. Therefore there is an obvious \dagger -functor, $(-)^{\circ} : \mathbf{TOF}^{\text{op}} \rightarrow \mathbf{TOF}$ given by $\text{tof} \mapsto \text{tof}$, $|1\rangle \mapsto \langle 1|$, $\langle 1| \mapsto |1\rangle$.

It is mechanical to verify that, $\llbracket - \rrbracket_{\mathbf{CNOT}} : \mathbf{CNOT} \rightarrow \mathbf{Mat}_{\mathbb{C}}$ is a strict \dagger -symmetric monoidal functor that reflects the dagger, so this interpretation is sound.

As an exercise, we first show that $\langle 0| \circ |0\rangle = 1_0$ (just as $\langle 1| \circ |1\rangle = 1_0$ in **[TOF.8]**):

¹The identities given here are not numbered the same as in the original paper [13], as one identity was found to be redundant (**[CNOT.9]**).

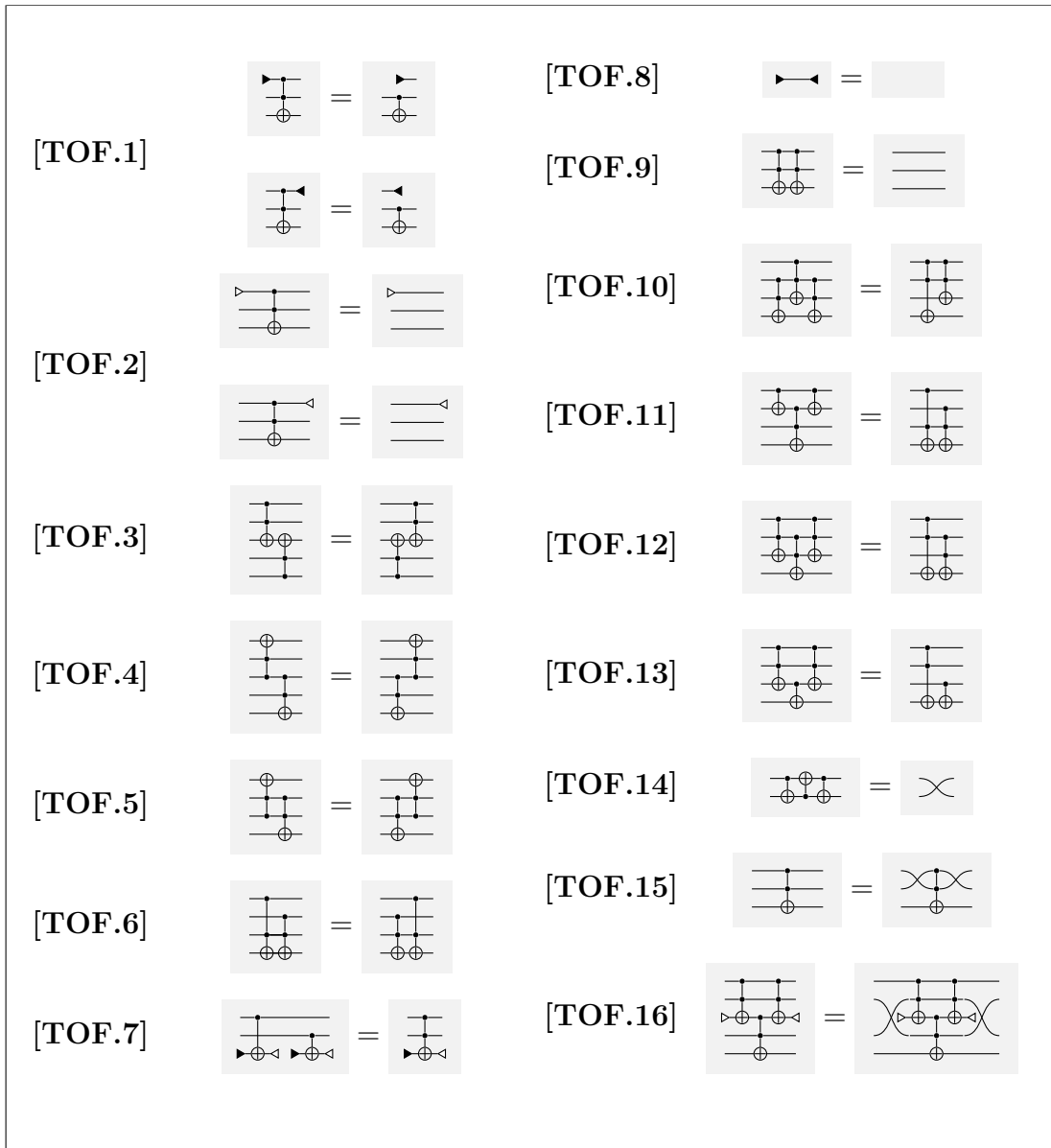


Figure 7.1: The identities of TOF

$$\begin{aligned}
\text{[Diagram]} &= \text{[Diagram]} && \text{[TOF.1]} \times 2 \\
&= \text{[Diagram]} && \text{[TOF.15]} \times 2 \\
&= \text{[Diagram]} \\
&= \text{[Diagram]} && \text{[TOF.1]} \\
&= \text{[Diagram]} && \text{[TOF.9]} \\
&= \text{[Diagram]} && \text{[TOF.8]} \times 3
\end{aligned}$$

Since we build on the work done in Chapter 5, we start by establishing that the interpretation of CNOT into TOF, where $|1\rangle \mapsto |1\rangle$, $\langle 1| \mapsto \langle 1|$, $\text{cnot} \mapsto \text{cnot}$ is sound:

Lemma 7.1.1. The interpretation of CNOT in TOF, using the derived generators in TOF, is a strict \dagger -symmetric monoidal dagger reflecting functor.

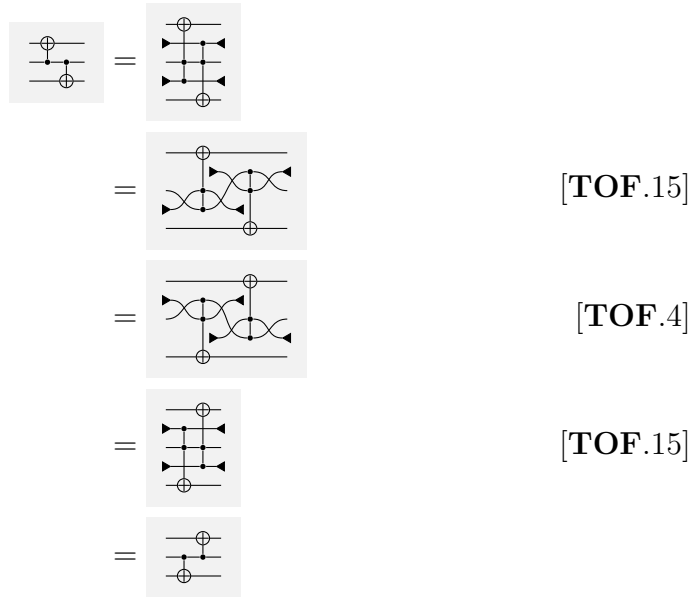
Proof. The functoriality is the only nontrivial property which must be proven. We verify that all of the identities of CNOT hold. To this end:

[CNOT.1]: This is [TOF.14].

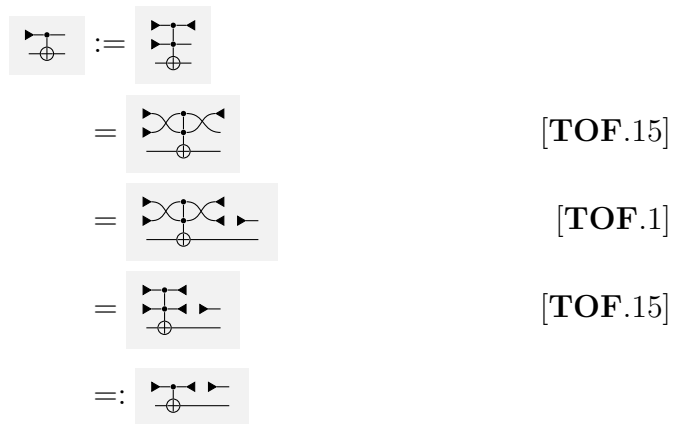
[CNOT.2]:

$$\begin{aligned}
\text{[Diagram]} &:= \text{[Diagram]} \\
&= \text{[Diagram]} && \text{[TOF.1]} \\
&= \text{[Diagram]} && \text{[TOF.9]} \\
&= \text{[Diagram]} && \text{[TOF.8]}
\end{aligned}$$

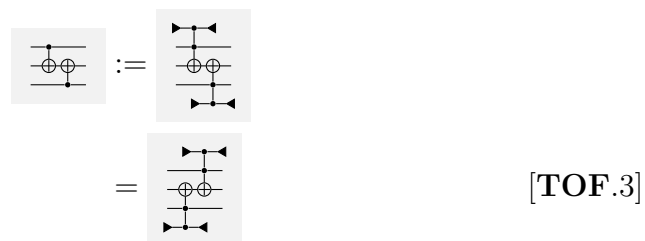
[CNOT.3]:




[CNOT.4]:



[CNOT.5]:



$$=:$$


[CNOT.6]: This is [TOF.8].

[CNOT.7]:

$$\begin{aligned}
 & \text{[CNOT.7]} := \text{[TOF.15]} \\
 & = \text{[TOF.15]} \\
 & = \text{[TOF.1]} \\
 & = \text{[TOF.1]} \\
 & = \text{[TOF.8]}
 \end{aligned}$$

[CNOT.8]:

$$\begin{aligned}
 & \text{[CNOT.8]} := \text{[TOF.15], [TOF.13], [TOF.9]} \\
 & = \text{[TOF.15], [TOF.13], [TOF.9]} \\
 & = \text{[TOF.1]} \\
 & = \text{[TOF.9]} \\
 & =:
 \end{aligned}$$

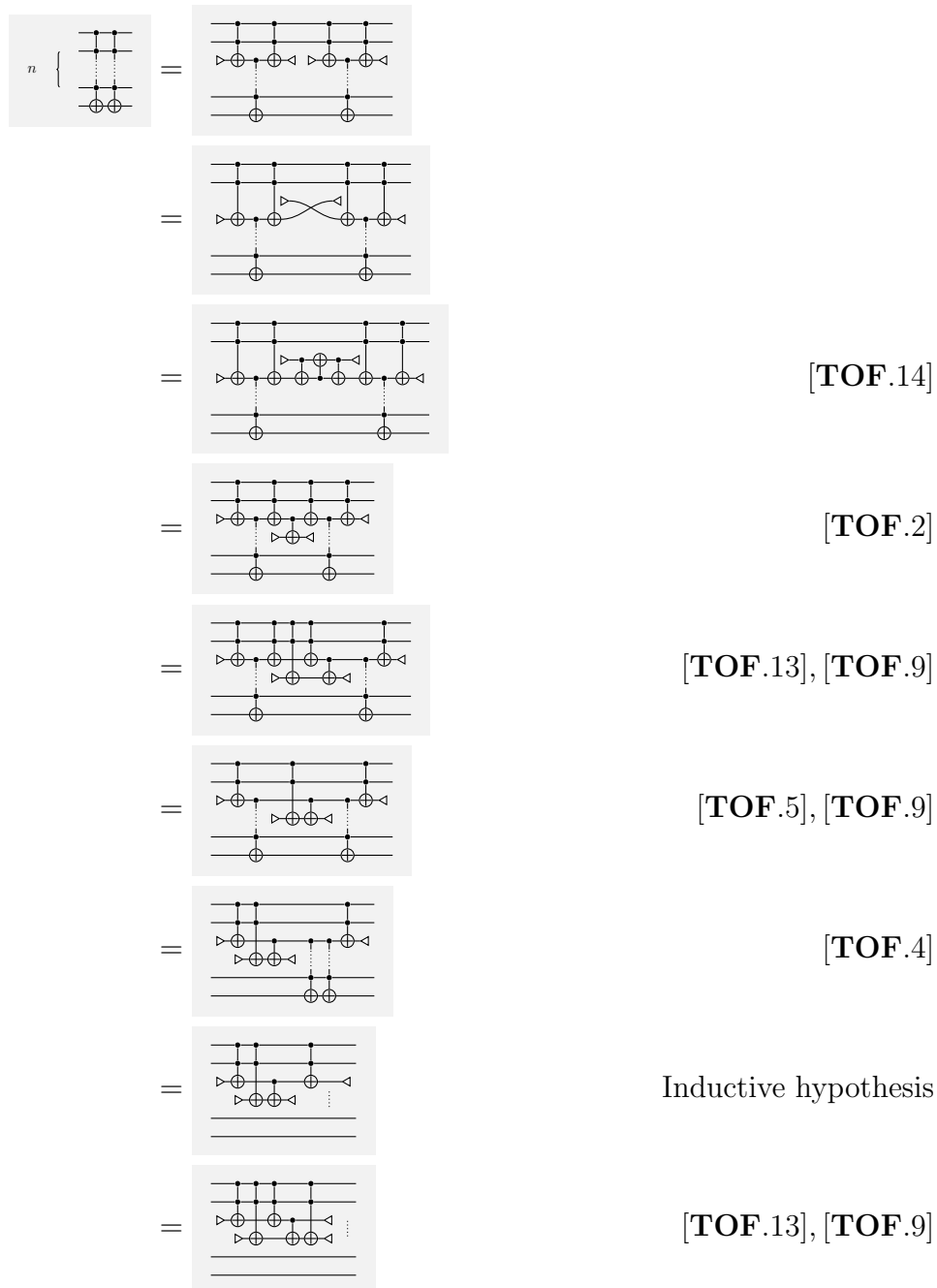
[CNOT.9]:

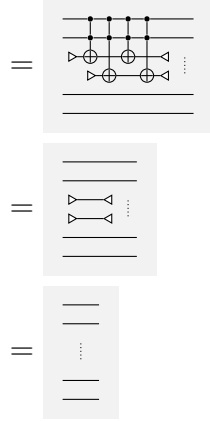
$$=$$


wires (and the target wire).

Lemma 7.2.2. Every cnot_n gate is self-inverse.

Proof. The base cases are trivial. For the inductive step, suppose that cnot_n gates are self-inverse, then:



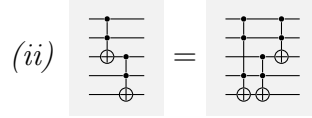
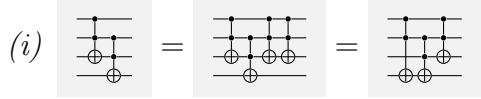


[TOF.5] \times 2, [TOF.9] \times 2

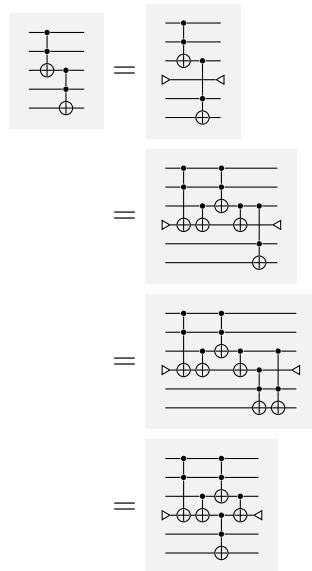
□

Next, we show how **tof** gates can be “pushed past” each other with a “trailing” cnot_n gate:

Lemma 7.2.3.



Proof. Of part (ii):



[TOF.9], [TOF.10]

[TOF.2]

[TOF.9], [TOF.11]

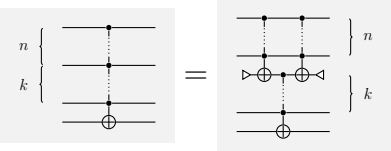
$$\begin{aligned}
&= \text{[TOF.9]} \\
&= \text{[TOF.9], [TOF.12]} \\
&= \text{[TOF.11]} \\
&= \text{[CNOT.2]} \\
&=:
\end{aligned}$$

□

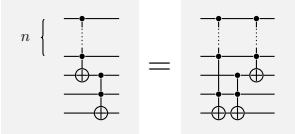
Next we show how we can “unzip” cnot_n gates; and simultaneously, we generalize Lemma 7.2.3 to show how a cnot_n gate can be “pushed” past a Toffoli gate:

Proposition 7.2.4. Given some $n \geq 1$ and $k \geq 1$:

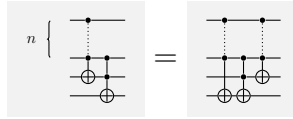
(i) cnot_{n+k} gates can be zipped and unzipped:



(ii) cnot_n gates can be pushed past Toffoli gates in the following sense:



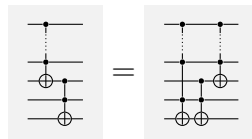
(iii) And likewise:



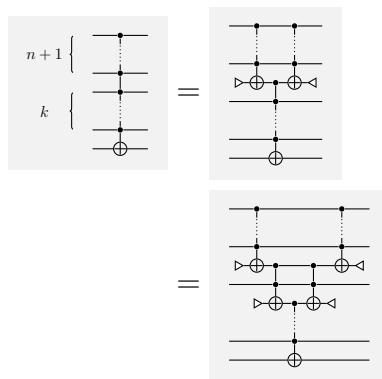
Proof. The proof is by a simultaneous induction for claims (i) and (ii) for all natural numbers k , by induction on the number of control wires, n , to unzip and the number of control wires being pushed past a Toffoli gate. Claim (iii) follows as a consequence of (ii) and Lemma 7.2.3.

For the induction, suppose that $n, k \geq 2$. The cases when $n = 1$ or $n = 2$ follow as a consequence.

- The base cases of claim (i) follows by the definition of the cnot_n gate. The base cases of claim (ii) Lemma is precisely Lemma 7.2.3(ii).
- Suppose that the inductive claim holds for all k and for some $n \geq 2$, graphically:



Consider the two identities for $n + 1$; first for the zipper:



Inductive hypothesis

$$= \text{[Diagram]} \tag{ii}$$

$$= \text{[Diagram]} \tag{[TOF.2]}$$

$$= \text{[Diagram]} \tag{[TOF.5], Lemma 7.2.2}$$

For the second inductive step:

$$n \left\{ \text{[Diagram]} \right. = \text{[Diagram]} \tag{Inductive step of (i)}$$

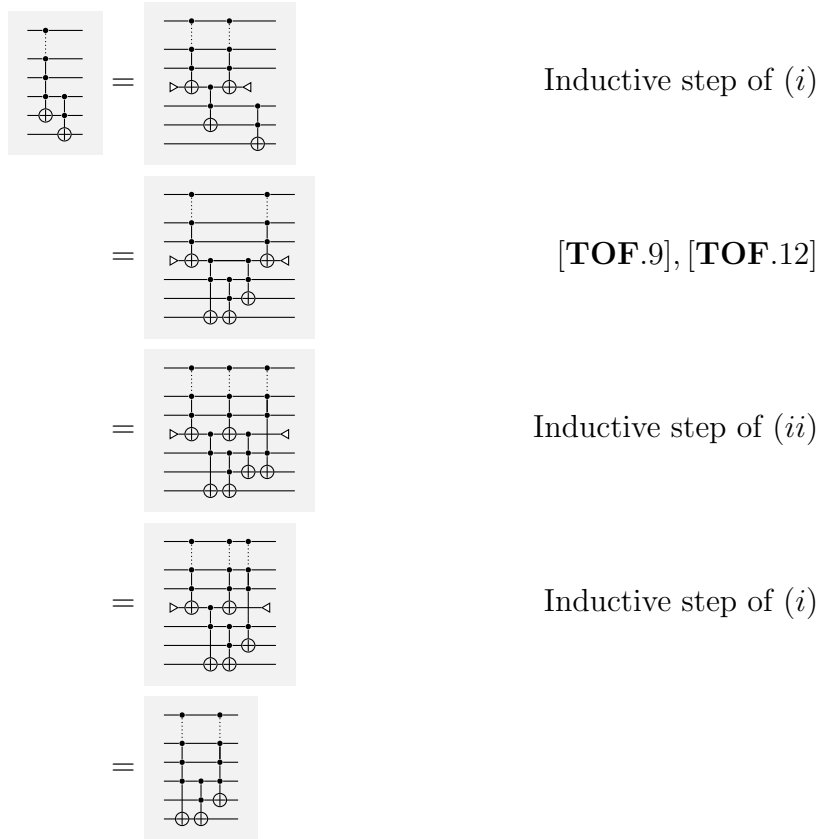
$$= \text{[Diagram]} \tag{[TOF.9], Lemma 7.2.3(ii)}$$

$$= \text{[Diagram]} \tag{Inductive hypothesis}$$

$$= \text{[Diagram]} \tag{[TOF.2]}$$

$$= \text{[Diagram]} \tag{Inductive step of (i)}$$

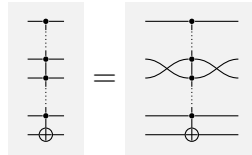
For claim (iii), we use claims (i) and (ii):



□

The ability to zip and unzip controlled-not gates allows us to transpose control wires:

Corollary 7.2.5.



Proof. The claim follows vacuously for $n < 2$. The base case when $n = 2$ follows immediately from [TOF.15]. Suppose that the claim holds for some cnot_n . Consider a cnot_{n+1} gate and unzip this gate one level down. We can transpose the top two control wires by the base case for $n = 2$ and we can transpose the bottom $n - 1$ control wires by the inductive hypothesis. Finally, to transpose the second and third wire from the top, try to unzip the cnot_{n+1} gate

down by one more step. If this cannot be done – because there is nothing more to unzip – it means one can directly apply [TOF.16]. \square

Since transpositions generate the symmetric group, the control wires of a cnot_n gate can, therefore, be freely permuted.

Let $[x, X]$ denote a $\text{cnot}_{|X|}$ gate with control wires in the set X targeting the wire $x \notin X$. Similarly let \triangleright_x and \triangleleft_x be, respectively, the $|0\rangle$ and $\langle 0|$ gates on the wire x .

The following identities are due to Iwama et al. [33]. They show how to push generalized control not gates past each other in certain key situations:

Lemma 7.2.6 (Iwama’s Identities).

(i) $[x, X][x, X] = 1$

(ii) When $x \in X$ then $\triangleright_x[y, X] = \triangleright_x$

(iii) When the target wire are the same $[x, X][x, Y] = [x, Y][x, X]$

(iv) When $x \notin Y$ and $y \notin X$ then $[x, X][y, Y] = [y, Y][x, X]$

(v) $[x, X][y, \{x\} \sqcup Y] = [y, X \cup Y][y, \{x\} \sqcup Y][x, X]$

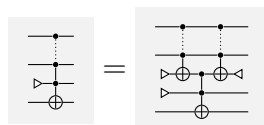
(vi) $\triangleright_z[z, \{x\}][y, \{x\} \sqcup X] = \triangleright_z[z, \{x\}][y, \{z\} \sqcup X]$

Proof.

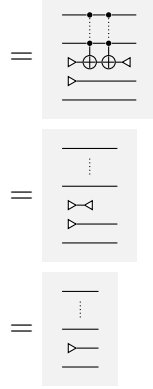
(i) This is precisely Lemma 7.2.2.

(ii) Using Corollary 7.2.5 it suffices to consider the case where x is the bottom control wire.

Using Proposition 7.2.4 (i) and Lemma 7.2.2:



Lemma 7.2.4(i)



Lemma 7.2.2

(iii) The proof follows easily from Axioms [TOF.3], [TOF.4], [TOF.5] and [TOF.7].

(iv) The proof follows easily from Axioms [TOF.4] and [TOF.5].

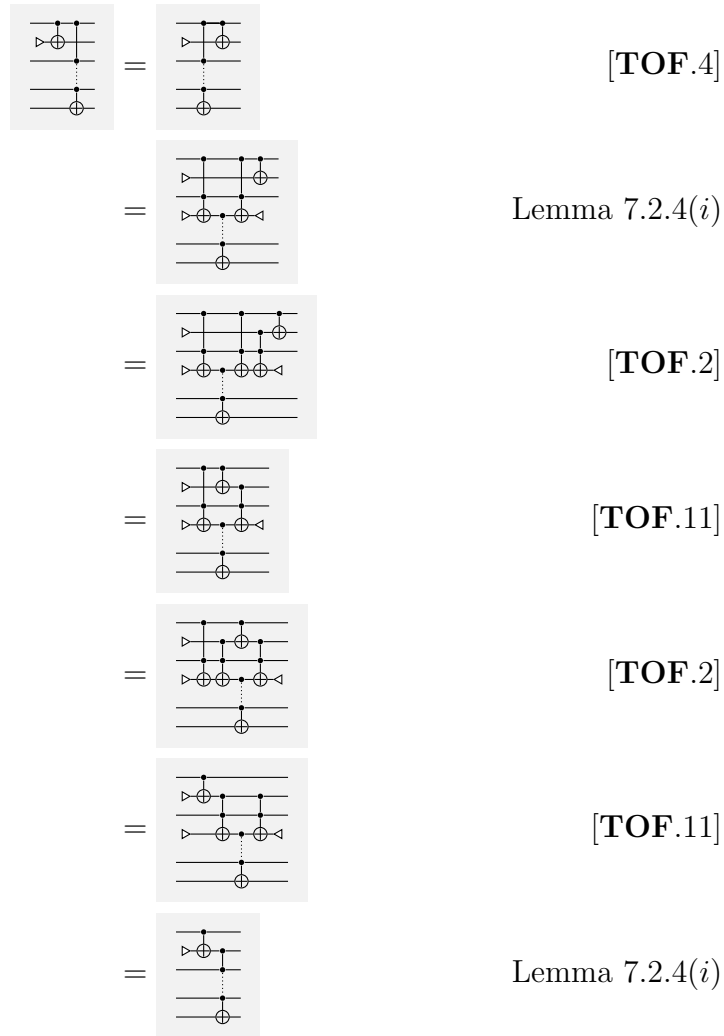
(v) The proof is by induction on the number of control wires of the second gate.

- The base cases are provided by Proposition 7.2.4 (ii) and (iii).
- Suppose now that the claim holds for all cases in which the second gate has no more than n control wires. Consider the case when the second gate has $n + 1$ control wires. Using Corollary 7.2.5 it suffices to consider the case where y is the bottom wire and x is the second bottom wire:

$$\begin{aligned}
& [x, X][y, Y \sqcup \{x\}] \\
&= [x, X] \triangleright_z [z, Y][y, \{z, x\}][z, Y] \triangleleft_z && \text{Use Prop. 7.2.4 (i) to unzip } [y, Y \sqcup \{x\}] \\
&= \triangleright_z [z, Y][x, X][y, \{z, x\}][z, Y] \triangleleft_z \\
&= \triangleright_z [z, Y][y, X \sqcup \{z\}][y, \{z, x\}][x, X][z, Y] \triangleleft_z && \text{Prop. 7.2.4 (ii), (iii)} \\
&= \triangleright_z [z, Y][y, X \sqcup \{z\}][y, \{z, x\}][z, Y][x, X] \triangleleft_z \\
&= \triangleright_z [z, Y][y, X \sqcup \{z\}][z, Y][y, \{z, x\}][y, Y \sqcup \{x\}][x, X] \triangleleft_z && \text{Prop. 7.2.4 (ii), (iii)} \\
&= \triangleright_z [z, Y][z, Y][y, X \sqcup \{z\}][y, X \cup Y][y, \{z, x\}][y, Y \sqcup \{x\}][x, X] \triangleleft_z && \text{Ind. Hyp.} \\
&= \triangleright_z [y, X \sqcup \{z\}][y, X \cup Y][y, \{z, x\}][y, Y \sqcup \{x\}][x, X] \triangleleft_z \\
&= \triangleright_z [y, X \sqcup \{z\}][y, X \cup Y][y, \{z, x\}] \triangleleft_z [y, Y \sqcup \{x\}][x, X] \\
&= \triangleright_z [y, X \cup Y] \triangleleft_z [y, Y \sqcup \{x\}][x, X] && \text{Lemma 7.2.6 (ii)} \\
&= \triangleright_z \triangleleft_z [y, X \cup Y][y, Y \sqcup \{x\}][x, X]
\end{aligned}$$

$$= [y, X \cup Y][y, Y \sqcup \{x\}][x, X]$$

- Using Corollary 7.2.5, it suffices to observe:



□

7.3 TOF is a discrete inverse category

Next we prove that TOF is a discrete inverse category.

The diagonal map in TOF is the image of the diagonal map, $\Delta : n \rightarrow n \otimes n$, in CNOT under the canonical functor $\text{CNOT} \rightarrow \text{TOF}$:

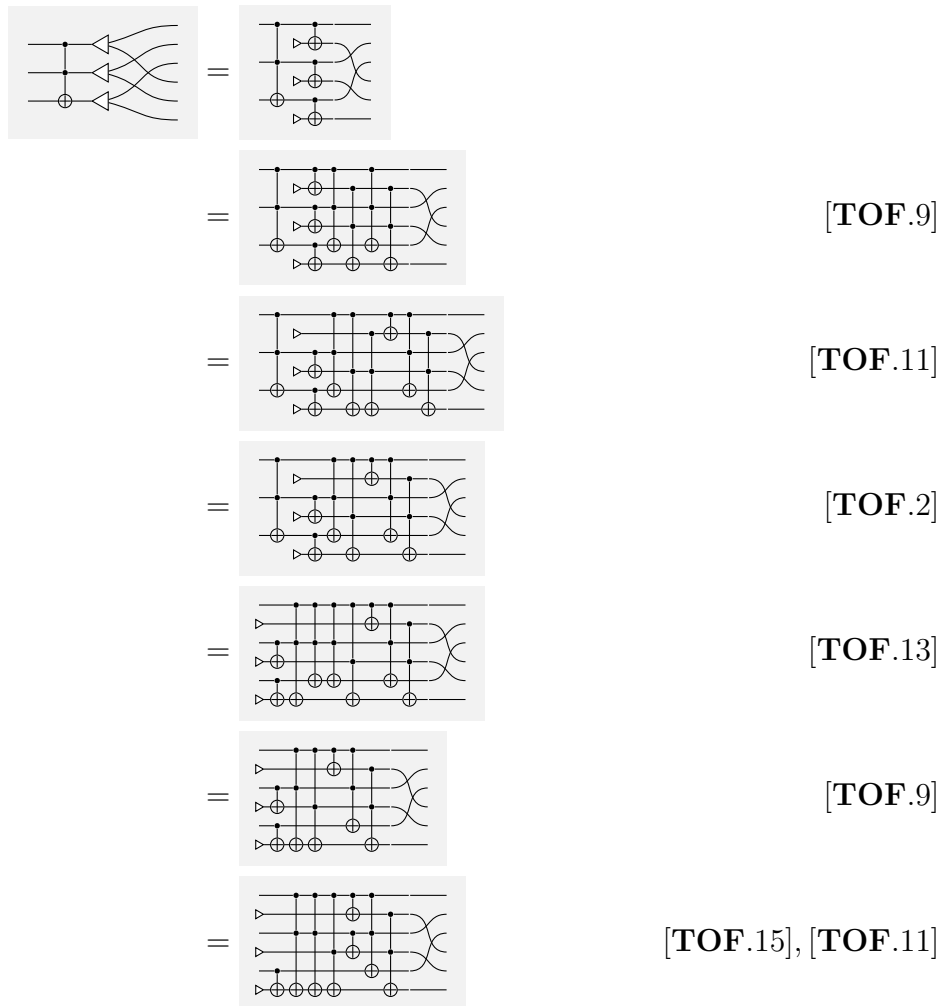
Definition 7.3.1. Define a family of maps $\Delta := \{\Delta_n\}_{n \in \mathbb{N}}$ with the **cnot** gate and 0-ancillary bits inductively, as in Chapter 5, such that $\Delta_0 = 1_0 = (u_0^L)^{-1} = (u_0^R)^{-1}$,

$$\Delta_1 := \begin{array}{c} \text{---} \end{array} \begin{array}{c} \diagup \\ \diagdown \end{array} := \begin{array}{c} \text{---} \\ \oplus \end{array} \quad \text{and for all } n > 1: \quad \Delta_n := \begin{array}{c} \text{---} \\ \oplus \end{array} \begin{array}{c} \diagup \\ \diagdown \end{array} := \begin{array}{c} \text{---} \\ \oplus \end{array} \begin{array}{c} \text{---} \\ \oplus \end{array} \begin{array}{c} \text{---} \\ \oplus \end{array} \begin{array}{c} \diagup \\ \diagdown \end{array}$$

Much of the proofs are inherited from Chapter 5 using Lemma 7.1.1; extending the results of CNOT to TOF.

Lemma 7.3.2. Δ is a natural transformation in TOF.

Proof. We prove that Δ is a natural transformation by structural induction. We have only to prove the inductive case for **tof** as the cases for the 1-ancillary bits are proven in Lemma 5.3.2:



$$\begin{aligned}
&= \text{[TOF.9]} \\
&= \text{[TOF.13]} \\
&=
\end{aligned}$$

□

To prove that **TOF** is a discrete inverse category with respect to \dagger -functor $(-)^{\circ} : \mathbf{TOF}^{\text{op}} \rightarrow \mathbf{TOF}$ it must also be shown that **[INV.1]**, **[INV.2]** and **[INV.3]** hold. As $(-)^{\circ} : \mathbf{TOF}^{\text{op}} \rightarrow \mathbf{TOF}$ is a \dagger -functor **[INV.1]** is immediate. It remains to prove **[INV.2]** and **[INV.3]**:

Lemma 7.3.3. For all maps f in **TOF** **[INV.2]** holds, that is $ff^{\circ}f = f$.

Proof. We prove that **[INV.2]** holds by structural induction. Again we have only to prove the inductive case for **tof** as the cases for the ancillary bits are proven in Lemma 5.3.9. Suppose inductively that $ff^{\circ}f = f$ for some circuit f , then we need to show that we can extend f by a **tof** gate and preserve the property. This is almost immediate by **[TOF.9]**:

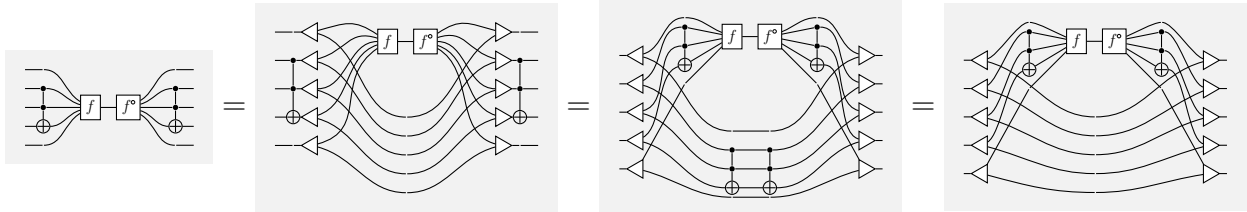
$$\text{[Diagram 1]} = \text{[Diagram 2]} = \text{[Diagram 3]}$$

□

To prove that **[INV.3]** holds, again, we identify the restriction idempotents of **TOF** with latchable circuits. We already know that latchable circuits commute with each other, by Lemma 4.1.13; therefore, to prove that all circuits of the form ff° are latchable implies that **[INV.3]** holds.

Lemma 7.3.4. Circuits of the form ff° in **TOF** are latchable and, thus, **[INV.3]** holds.

Proof. We prove that circuits of the form ff° in TOF are latchable by structural induction. We have only to prove the inductive case for **tof** as the cases for the 1-ancillary bits are proven in 5.3.9. Suppose that some circuit f is latchable, then we must show inductively that conjugating a Toffoli gate will result in a latchable circuit:



□

Therefore:

Proposition 7.3.5. TOF is a discrete inverse category with this structure.

Proof. Lemmas 7.3.3 and 7.3.4 show that TOF is an inverse category. Lemma 7.3.2 and the fact that the semi-Frobenius identities hold in CNOT imply that $(n, \Delta_n, \Delta_n^\circ)$ forms a natural special, commutative, semi-Frobenius algebra for all $n \in \mathbb{N}$. Thus, TOF is a discrete inverse category. □

7.4 The points of TOF

The (total) points of TOF are iterated tensors of the input ancillary bits ($|0\rangle$ and $|1\rangle$) and can be represented using ket notation $|b_1, \dots, b_n\rangle := |b_1\rangle \otimes \dots \otimes |b_n\rangle$, as in Chapter 5. We start by observing that the Toffoli gate in TOF behaves as expected on these points:

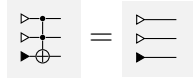
Lemma 7.4.1. $\text{tof}_\circ |x_1, x_2, x_3\rangle = |x_1, x_2, x_1 \cdot x_2 + x_3\rangle$

Proof. $\text{tof}_\circ |000\rangle = |000\rangle$:



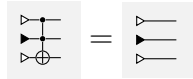
[TOF.2]

$\text{tof}_\circ|001\rangle = |001\rangle:$



[TOF.2]

$\text{tof}_\circ|010\rangle = |010\rangle:$



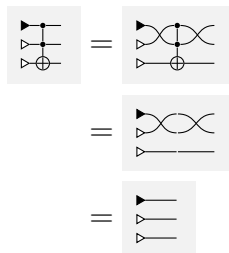
[TOF.2]

$\text{tof}_\circ|011\rangle = |011\rangle:$



[TOF.2]

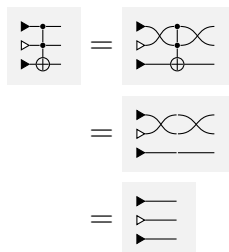
$\text{tof}_\circ|100\rangle = |100\rangle:$



[TOF.15]

[TOF.2]

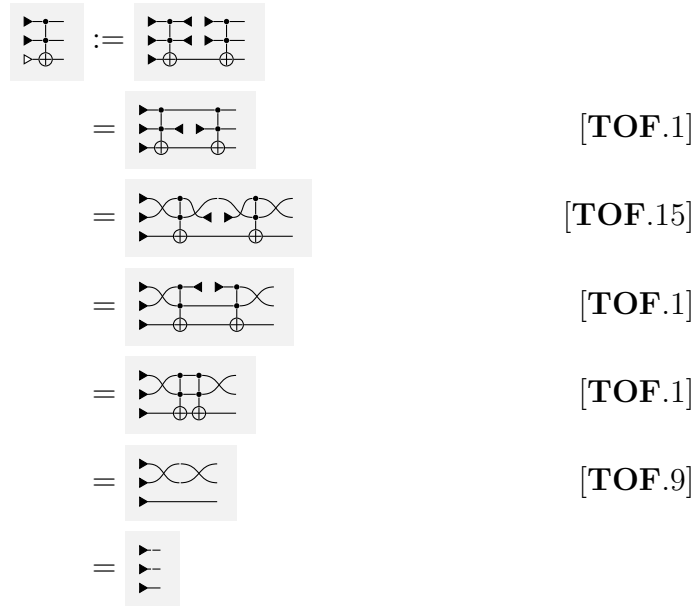
$\text{tof}_\circ|101\rangle = |101\rangle:$



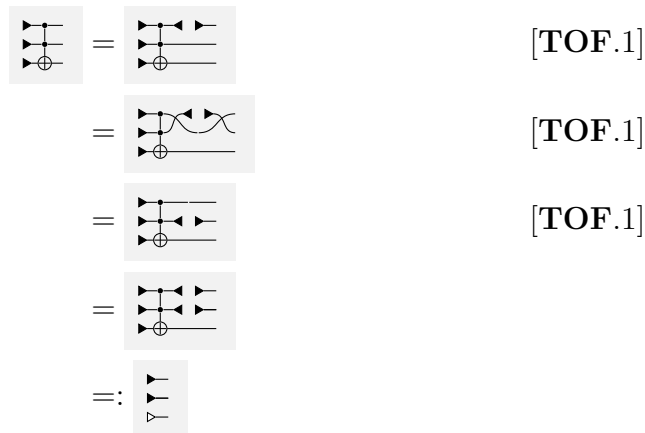
[TOF.15]

[TOF.2]

$\text{tof}_\circ|110\rangle = |111\rangle$:



$\text{tof}_\circ|111\rangle = |110\rangle$:



□

The points above which are expressed in ket notation are clearly total. However, not all maps with domain 0 are total: in particular, as in CNOT, the maps $\emptyset_{0,m}$ for $m \in \mathbb{N}$ are not total:

Definition 7.4.2. Define $\emptyset := \langle 0|_\circ|1\rangle$, and $\emptyset_{n,m} = |1\rangle^{\otimes m} \emptyset_\circ \langle 1|^{\otimes n}$.

When any map is tensored with such a map it becomes one of them:

Lemma 7.4.3. For all circuits $f \in \text{TOF}(n, m)$, $f \otimes \emptyset = \emptyset_{n,m}$.

Proof. By the functionality of the interpretation of CNOT into TOF and [14, Lemma A.2] note that \emptyset is idempotent. Therefore, use [CNOT.9] to cut all of the wires around all Toffoli gates. Notice that when the wires round a Toffoli gate is cut this results in the map:

$$\langle 111 | \text{tof} | 111 \rangle = \langle 11 | \text{cnot} | 11 \rangle = \langle 1 | \text{not} | 1 \rangle = \langle 1 | 0 \rangle =: \emptyset$$

When all wires of a circuit are cut it will therefore take the form:

$$|1\rangle^{\otimes m} \langle 1 | \otimes |1\rangle^{\otimes \ell} \emptyset^{\otimes k} \langle 1 |^{\otimes n} = |1\rangle^{\otimes m} \langle 1_0 |^{\otimes \ell} \emptyset^{\otimes k} \langle 1 |^{\otimes n} = |1\rangle^{\otimes m} \emptyset \langle 1 |^{\otimes n} = \emptyset_{n,m}$$

□

Just as before, this gives the following result:

Corollary 7.4.4. $\emptyset_{n,m}$ are zero maps.

There are only two sorts of points, those that are expressible as bras—and maps of the form $\emptyset_{0,n}$ for some $n \in \mathbb{N}$:

Lemma 7.4.5. For all $n \in \mathbb{N}$ and $f \in \text{TOF}(0, n)$, $f = \emptyset_{0,n}$ or $f = |b_1, \dots, b_n\rangle$ for some $b_1, \dots, b_n \in \mathbb{Z}_2$.

Proof. Given any circuit $f \in \text{TOF}(0, n)$, pull all of the $|1\rangle$ gates to the left of the circuit and all of the $\langle 1 |$ gate to the right. In the middle, there will only be **tof** gates; thus, apply the $|1\rangle$ gates to the **tof** gates using Lemma 7.4.1. This will result in a series of $|1\rangle$ and $|0\rangle$ gates on the left. Repeatedly apply these $|1\rangle$ and $|0\rangle$ gates to the **tof** gates using Lemma 7.4.1 until there are only a series of $|1\rangle$ and $|0\rangle$ gates on the left and a series of $\langle 1 |$ gates on the right, and nothing in the middle. If a $|1\rangle$ and $\langle 1 |$ gate meet, they compose to form the identity on

0. If this process can eliminate all of the $\langle 1|$ gates, then we are done. Otherwise, if $|0\rangle$ and $\langle 1|$ gate meet, they compose to form \emptyset ; therefore, by Lemma 7.4.3, $f = \emptyset_{0,n}$. \square

Therefore:

Lemma 7.4.6. $\tilde{h}_0 : \text{TOF} \rightarrow \text{Pinj}$ is a strong symmetric monoidal functor.

The proof is the same as Lemma 5.5.4, for CNOT.

This implies, in conjunction with Lemma 4.2.4, that:

Lemma 7.4.7. $\tilde{h}_0 : \text{TOF} \rightarrow \text{Pinj}$ is a discrete inverse functor.

7.5 Partial injective functions and TOF

Recall the representable restriction functor $h_x : \mathbb{X} \rightarrow \text{Par}$ given in Definition 4.2.1.

Fix $\mathbb{X} = \text{TOF}$ and $x = 0$. Recall that $h_0 : \text{TOF} \rightarrow \text{Par}$ can be lifted to an inverse-product preserving functor $\tilde{h}_0 : \text{TOF} \rightarrow \text{Pinj}$.

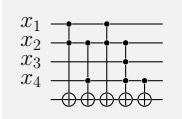
Define FPinj_2 to be the full subcategory of Pinj (sets and partial isomorphisms) with objects finite powers of the two element set. By Lemma 7.4.5, the object $\tilde{h}_0(n)$ corresponds to the set $\{|0\rangle, |1\rangle\}^n$ in $\text{FPinj}_2 \subset \text{Pinj}$; therefore, by restricting the codomain of \tilde{h}_0 we obtain a functor $\tilde{H}_0 : \text{TOF} \rightarrow \text{FPinj}_2$. We will prove that this functor is an equivalence of categories.

7.6 A normal form for the idempotents of TOF

As in Chapter 5, our objective is to reduce the fullness and faithfulness of $\tilde{H}_0 : \text{TOF} \rightarrow \text{FPinj}_2$ to its fullness and faithfulness on restriction idempotents. cnot_n gates will be used to define a class of circuits which will allow a normal form for the idempotents of TOF to be established. It is also worth noting that the following class of circuits corresponds to the canonical form of Iwama et al. [33, Definition 4]:

Definition 7.6.1. A circuit $f : n \rightarrow n$ is said to be in **polynomial form** when it is the composition of circuits $f = c_1 \cdots c_k$ where each c_i is a cnot_j gate with control the first $n - 1$ wires and target the last wire.

For example, the following circuit corresponding to the polynomial $x_1x_2 + x_2x_4 + x_1x_2 + x_1x_2 + x_2x_3x_4 + x_4$ is in polynomial form:

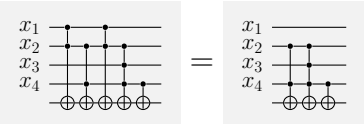


Clearly all polynomials (in normal form) can be represented by a circuit in polynomial form, where each monomial corresponds to a cnot_n gate. Moreover, this correspondence is bijective, as cnot_n gates targeting the same wire commute, and cnot_n gates are self-inverse by Lemma 7.2.2.

For example, the following identity holds in $\mathbb{Z}_2[x_1, x_2, x_3, x_4]$

$$x_1x_2 + x_2x_4 + x_1x_2 + x_2x_3x_4 + x_4 = x_2x_4 + x_2x_3x_4 + x_4$$

and this corresponds to the following identity in TOF:

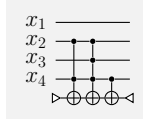


Since the restriction idempotents in FPin_j are determined by Boolean equations or, equivalently, by the zeros of multivariate polynomials over \mathbb{Z}_2 , we use polynomial form circuits to construct a normal form for the restriction idempotents of TOF. By restricting the value of the bottom wire to be 0, the evaluation of this circuit on total points is defined if and only if the corresponding tuple of bits is in the kernel of the function corresponding to polynomial evaluation:

Definition 7.6.2. A circuit $e : n \rightarrow n$ in TOF is a **polyform** if $e = (1_n \otimes \langle 0 |) \circ q \circ (1_n \otimes | 0 \rangle)$

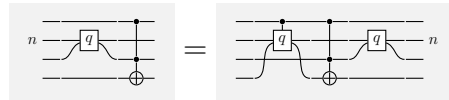
for some $q : n + 1 \rightarrow n + 1$ in polynomial form.

For example the following circuit corresponding to the equation $x_2x_4 + x_2x_3x_4 + x_4 = 0$ is a polyform:



Given a polynomial form circuit q , let the circuit \overline{q} denote the circuit where every cnot_n gate of q is additionally controlled by wire with the black dot. Note that this is well-defined by Corollary 7.2.5.

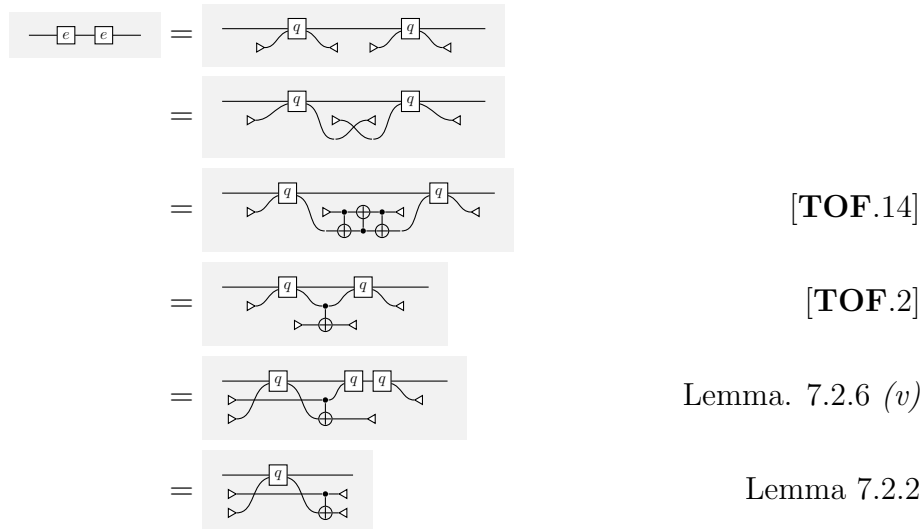
Given a polynomial form circuit $q : n + 1 \rightarrow n + 1$, by the repeated application of Lemma 7.2.6 (v) for each constituent cnot_n gate, it is useful to observe:



Our aim is now to show that the idempotents of TOF are always equivalent to a polyform so these circuits provide a normal form for the idempotents.

Lemma 7.6.3. Polyforms are idempotents (and therefore restriction idempotents).

Proof. Consider some map $e := (1_n \otimes \langle 0 |) q (1_n \otimes | 0 \rangle)$ a polyform, as above, then:



$$\begin{aligned}
&= \text{[Diagram: A box containing a gate labeled } g \text{ with two control lines and one target line.]} \\
&= \text{[Diagram: A box containing a gate labeled } e \text{ with one control line and one target line.]} \qquad \text{[TOF.2]}
\end{aligned}$$

□

We prove that the converse also holds using 3 lemmas:

Proposition 7.6.4. All idempotents in TOF are equivalent to a circuit which is a polyform.

Given a polyform, f , by structural induction, it suffices to show that sandwiching f with a generating gate, g , to obtain $(1 \otimes g \otimes 1)f(1 \otimes g^\circ \otimes 1)$, always results in a circuit which has a polyform. There are 3 cases, one for each generating circuit:

Lemma 7.6.5. If $f : n \rightarrow n$ is a polyform then sandwiching f by a **tof** gate results in a circuit with a polyform.

Proof. By Lemma 7.2.6 (v), push g through f until it meets the other $g^\circ = g$ and then annihilate both generalized controlled-not gates by Lemma 7.2.2. This circuit is still a polyform. □

Lemma 7.6.6. If $f : n \rightarrow n$ is a polyform then sandwiching f by a $|1\rangle$ gate results in a circuit equivalent to a polyform.

Proof. It suffices to observe for the inductive step that:

$$\text{[Diagram: A circuit with 4 horizontal lines. The top two lines have control dots. The bottom line has a target dot. The circuit consists of a CNOT gate from line 1 to line 2, followed by a CNOT gate from line 2 to line 3, followed by a CNOT gate from line 3 to line 4.]} = \text{[Diagram: A circuit with 4 horizontal lines. The top two lines have control dots. The bottom line has a target dot. The circuit consists of a CNOT gate from line 2 to line 1, followed by a CNOT gate from line 2 to line 3, followed by a CNOT gate from line 3 to line 4.]} \qquad \text{[TOF.15], [TOF.1]}$$

$$= \text{[Diagram: A circuit with 4 horizontal lines. The top two lines have control dots. The bottom line has a target dot. The circuit consists of a CNOT gate from line 2 to line 1, followed by a CNOT gate from line 2 to line 3, followed by a CNOT gate from line 3 to line 4.]} \qquad \text{Lemma 7.2.3(ii)}$$

$$= \text{[Diagram: A circuit with 4 horizontal lines. The top two lines have control dots. The bottom line has a target dot. The circuit consists of a CNOT gate from line 1 to line 2, followed by a CNOT gate from line 2 to line 3, followed by a CNOT gate from line 3 to line 4.]} \qquad \text{[TOF.9]}$$

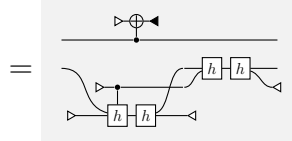
$$\begin{aligned}
&= \text{[TOF.2]} \\
&=
\end{aligned}$$

□

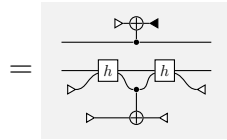
Lemma 7.6.7. If $f : n \rightarrow n$ is a polyform then sandwiching f by a $|1\rangle$ and $\langle 1|$ gate results in a circuit with a polyform.

Proof. Suppose $f : n \rightarrow n$ is a polyform $(1_n \otimes \langle 0| \otimes 1_m) \circ h \circ (1_n \otimes |0\rangle \otimes 1_m)$, then:

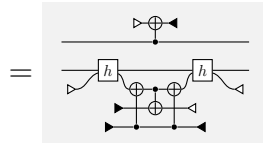
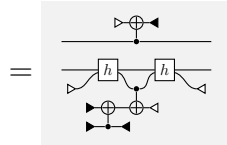
$$\begin{aligned}
&= \text{[TOF.14]} \\
&= \text{[TOF.1]} \\
&= \text{[TOF.1]} \\
&= \text{[TOF.11]} \\
&= \\
&= \text{Lemma 7.2.2}
\end{aligned}$$



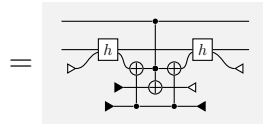
Lemma 7.2.6 (ii)



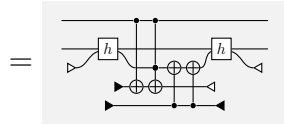
Lemma 7.2.6 (v)



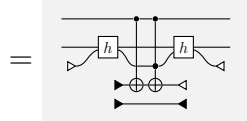
[TOF.11]



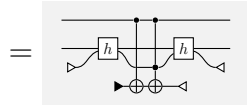
[TOF.7]



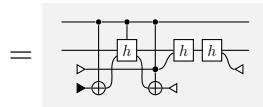
Lemma 7.2.6 (v)



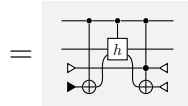
[TOF.9]



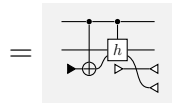
[TOF.1]



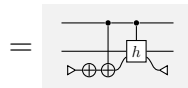
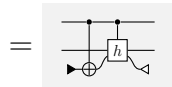
Lemma 7.2.6 (v)



Lemma 7.2.2



[TOF.2]



□

We have now established with Lemma 7.6.3 and Proposition 7.6.4:

Proposition 7.6.8. Polyforms are a normal form for the idempotents in TOF.

This implies, as the idempotents of FPinj_2 are determined by multivariate polynomials over \mathbb{Z}_2 :

Corollary 7.6.9. $\tilde{H}_0 : \text{TOF} \rightarrow \text{FPinj}_2$ is full and faithful on restriction idempotents.

Recall that \tilde{h}_0 is faithful in general, so this follows from the fullness on idempotents in specific.

7.6.1 $\tilde{H}_0 : \text{TOF} \rightarrow \text{FPinj}_2$ is a discrete inverse equivalence

We know that $\tilde{H}_0 : \text{TOF} \rightarrow \text{FPinj}_2$ is a discrete inverse functor. It remains to show that this functor is full and faithful.

Just as in Chapter 5, to prove the fullness of $\tilde{H}_0 : \text{TOF} \rightarrow \text{FPinj}_2$ we “simulate” all the total maps in FPinj_2 with extra outputs:

Lemma 7.6.10. For every total map $f \in \text{FPinj}_2(n, m)$, there is some $g \in \text{TOF}$ such that $\tilde{H}_0(g) = \langle 1_{\mathbb{Z}_2^n}, f \rangle$.

Proof. Consider a total $f \in \text{FPinj}_2(n, m)$. For any i such that $1 \leq i \leq m$, observe that $f\pi_i$ corresponds to a polynomial in $\mathbb{Z}[x_1, \dots, x_n]$ and thus, there is a circuit $g_i : n + 1 \rightarrow n + 1$ in polynomial form such that $\tilde{H}_0(h_i) = \langle 1_{\mathbb{Z}_2^n}, f\pi_i \rangle$ where $h_i := g_i \circ (1_n \otimes |0\rangle)$.

Now, inductively define the circuit P_i for all i such that $0 \leq i \leq m$, such that: $P_0 = 1_n$ and for every i such that $0 \leq i < m$, $P_{i+1} := h_{i+1}(P_i \otimes 1_1)$.

Then $\tilde{H}_0(P_m) = \langle 1_{\mathbb{Z}_2^n}, f\pi_1, \dots, f\pi_m \rangle = \langle 1_n, f \rangle$. □

Using this result with the fullness on idempotents, this allows us to show:

Proposition 7.6.11. $\tilde{H}_0 : \text{TOF} \rightarrow \text{FPinj}_2$ is full.

Proof. Consider a map $f \in \text{FPinj}_2(\mathbb{Z}_2^n, \mathbb{Z}_2^m)$ for arbitrary $n, m \in \mathbb{Z}$. By Lemma 4.2.7, if we can simulate $\langle 1, f \rangle$ and $\langle 1, f^\circ \rangle$, we can simulate f . However, partial maps of the form $\langle 1, g \rangle : X \rightarrow X \times Y$ are restrictions of a total map, unless Y is empty. The case of Y being empty does not occur in FPinj_2 as the empty set is not an object. Thus, all such maps are restrictions of total maps. Therefore, by Lemma 7.6.10 there is some $h \in \text{TOF}$ such that $\tilde{H}_0(h) \geq \langle 1, g \rangle$ for any g . However, by Proposition 7.6.9, $\tilde{H}_0 : \text{TOF} \rightarrow \text{FPinj}_2$ is full on restriction idempotents, so there is some $e = \bar{e}$ such that $H_0(e) = \bar{g}$ and so $H_0(eh) = \langle 1, g \rangle$ which completes the proof. □

The faithfulness of \tilde{H}_0 is reduced to its faithfulness on restriction idempotents:

Proposition 7.6.12. $\tilde{H}_0 : \text{TOF} \rightarrow \text{FPinj}_2$ is faithful.

This implies:

Theorem 7.6.13. The identities of TOF are complete.

The proof is essentially the same as the proof of Theorem 5.5.22, for the completeness of CNOT .

Because $\tilde{H}_0 : \text{TOF} \rightarrow \text{FPinj}_2$ is a discrete inverse functor, which is full, faithful and essentially surjective:

Theorem 7.6.14. TOF is discrete inverse-equivalent to FPinj_2 .

Chapter 8

Conclusions and future work

8.1 Conclusions

In this thesis, complete sets of identities are given for both the affine and entire fragments of classical reversible circuits with ancillary bits. In both cases, a discrete inverse equivalence into a subcategory of sets and partial isomorphisms is exhibited.

The proofs for both fragments have the same structure. First, it is proven that the category of circuits forms a discrete inverse category. Next, a normal form for the restriction idempotents in this category is constructed. The normal form is used to show that the discrete inverse functor \tilde{h}_0 is full and faithful. To this end, two new insights for discrete inverse functors were necessary. In particular, Lemmas 4.2.7 and 4.2.6 help reduce the fullness/faithfulness of discrete inverse functors to the behaviour on idempotents. Therefore, by postcomposing \tilde{h}_0 with Barr's ℓ^2 we obtain both completeness results.

We also demonstrated an embedding from CNOT into the angle-free fragment of the ZX-calculus, ZX_π . We then extended the identities of CNOT by adding the Hadamard gate as a generator. This adds (co)units to the inverse products of CNOT: yielding a classical structure corresponding to the Pauli X observable.

8.2 Future work

An immediate direction for future work would be to extend these classification results for the controlled-not gate and Toffoli gate to qudit (d valued instead of binary) classical reversible computing. The qudit generalizations of the Toffoli and controlled-not gates would have order d . This seems easier when d is prime, or perhaps even when d is a power of a prime. This leads me to pose the following question; is the category generated by the obvious extension of the Toffoli gate to d -dimensions along with the ancillary bits for 1 equivalent to the full subcategory of sets and partial injections where the objects are sets whose cardinality is a power of d .

After the extension to qudits, it would be interesting to see if this result could be extended to all of finite sets and partial isomorphisms using the isomorphism of sets $\mathbb{Z}_p^n \times \mathbb{Z}_q^m \cong \mathbb{Z}_{p^n q^m}$; a similar question was posed for the ZX-calculus in [52]. Work has already been performed on interacting Frobenius algebras, with applications to the ZX-calculus [21], so it would be interesting to see if some of their results could be generalized to the case of semi-Frobenius algebras.

Another interesting direction would be to consider the Fredkin and not—or Toffoli gate. This presents a challenge, because such a category would not have all inverse products, but rather only ternary inverse products. This would require a reformulation of the notion of a discrete inverse category, and in particular, a generalized Frobenius law for ternary semigroups. Ternary Frobenius algebras in compact closed categories have previously been researched [27], however, a general symmetric monoidal axiomatization of ternary Frobenius algebras would be needed.

Although the calculus which is presented in Chapter 7 is universal for reversible circuits with ancillary bits; recall that it is not reversible for quantum computing in general. It would be interesting to extend TOF to an approximately universal fragment of quantum computing by adding the Hadamard gate [3], just as we extended CNOT to the real fragment of stabilizer quantum mechanics. The proof would likely involve a two way translation between either

ΔZX or ZH_π . ΔZX is the extension of ZX_π with the triangle generator as a generator [51]. The triangle generator is the map given by $|0\rangle\langle 0| + |1\rangle\langle 0| + |0\rangle\langle 1|$, and allows the Toffoli gate to be described relatively easily; although, the Axioms are hard to translate to Toffoli and Hadamard gates. The phase-free fragment of the ZH-calculus, ZH_π , on the other hand is the extension of ZX_π with “Hadamard spiders” [50]. Although ΔZX and ZH_π are sound, universal and complete for the same fragment of quantum mechanics, the identities of ZH_π appear to be easier to translate into Toffoli and Hadamard gates.

Bibliography

- [1] Scott Aaronson, Daniel Grier, and Luke Schaeffer. The classification of reversible bit operations. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:34. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.
- [2] Lowell Abrams. *Frobenius algebra structures in topological quantum field theory and quantum cohomology*. PhD thesis, Johns Hopkins University, 1997.
- [3] Dorit Aharonov. A simple proof that Toffoli and Hadamard are quantum universal. *arXiv preprint quant-ph/0301040*, 2003.
- [4] Matthew Amy, Jianxin Chen, and Neil J. Ross. A finite presentation of CNOT-dihedral operators. *arXiv preprint arXiv:1701.00140*, 2017.
- [5] Miriam Backens. The ZX-calculus is complete for stabilizer quantum mechanics. *New Journal of Physics*, 16(9):093021, 2014.
- [6] Miriam Backens. Making the stabilizer ZX-calculus complete for scalars. *arXiv preprint arXiv:1507.03854*, 2015.
- [7] Miriam Backens. *Completeness and the ZX-calculus*. PhD thesis, University of Oxford, 2016.
- [8] Miriam Backens and Aleks Kissinger. ZH: A complete graphical calculus for quantum computations involving classical non-linearity. *arXiv preprint arXiv:1805.02175*, 2018.

- [9] Miriam Backens, Simon Perdrix, and Quanlong Wang. A simplified stabilizer ZX-calculus. *arXiv preprint arXiv:1602.04744*, 2016.
- [10] Michael Barr. Algebraically compact functors. *Journal of Pure and Applied Algebra*, 82(3):211–231, 1992.
- [11] Wolfgang Bertram and Michael Kinyon. Associative geometries. I: Torsors, linear relations and grassmannians. *arXiv preprint arXiv:0903.5441*, 2009.
- [12] Albert Camus. *Le mythe de Sisyphe: essai sur l’absurde*. 1942.
- [13] Robin Cockett and Cole Comfort. The Category TOF. *Electronic Proceedings in Theoretical Computer Science*, 287:67–84, 2019.
- [14] Robin Cockett, Cole Comfort, and Priyaa Srinivasan. The Category CNOT. *Electronic Proceedings in Theoretical Computer Science*, 266:258–293, 2018.
- [15] Robin Cockett, Xiuzhan Guo, and Pieter Hofstra. Range categories II: Towards regularity. *Theory and Applications of Categories*, 26(18):453–500, 2012.
- [16] Robin Cockett and Stephen Lack. Restriction categories I: categories of partial maps. *Theoretical computer science*, 270(1):223–259, 2002.
- [17] Bob Coecke and Ross Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, 2011.
- [18] Bob Coecke and Aleks Kissinger. *Picturing quantum processes*. Cambridge University Press, 2017.
- [19] Bob Coecke, Dusko Pavlovic, and Jamie Vicary. A new description of orthogonal bases. *Mathematical Structures in Computer Science*, 23(3):555–567, 2013.
- [20] Cole Comfort. Circuit relations for real stabilizers: Towards TOF+H, 2019.

- [21] Ross Duncan and Kevin Dunne. Interacting Frobenius algebras are Hopf. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16*, pages 535–544, New York, NY, USA, 2016. ACM.
- [22] Ross Duncan and Simon Perdrix. Pivoting makes the ZX-calculus complete for real stabilizers. *arXiv preprint arXiv:1307.7048*, 2013.
- [23] Edward Fredkin and Tommaso Toffoli. Conservative logic. In *Collision-based computing*, pages 47–81. Springer, 2002.
- [24] Brett Giles. *An investigation of some theoretical aspects of reversible computing*. PhD thesis, University of Calgary, 2014.
- [25] Daniel Gottesman. *Stabilizer codes and quantum error correction*. PhD thesis, California Institute of Technology, 1997.
- [26] Daniel Gottesman. The Heisenberg representation of quantum computers. *arXiv preprint quant-ph/9807006*, 1998.
- [27] Marino Gran, Chris Heunen, and Sean Tull. Monoidal characterisation of groupoids and connectors. *arXiv preprint arXiv:1906.02056*, 2019.
- [28] Alexander S Green, Peter LeFanu Lumsdaine, Neil J Ross, Peter Selinger, and Benoît Valiron. An introduction to quantum programming in Quipper. In *International Conference on Reversible Computation*, pages 110–124. Springer, 2013.
- [29] Amar Hadzihasanovic. *The algebra of entanglement and the geometry of composition*. PhD thesis, University of Oxford, 2017.
- [30] Chris Heunen. On the functor $\mathbb{l}2$. In *Computation, Logic, Games, and Quantum Foundations. The Many Facets of Samson Abramsky*, pages 107–121. Springer, 2013.
- [31] Chris Heunen and Jamie Vicary. Introduction to categorical quantum mechanics.

- [32] Peter Hines. Quantum circuit oracles for abstract machine computations. *Theoretical Computer Science*, 411(11):1501 – 1520, 2010.
- [33] Kazuo Iwama, Yahiko Kambayashi, and Shigeru Yamashita. Transformation rules for designing CNOT-based quantum circuits. In *Proceedings of the 39th annual Design Automation Conference*, pages 419–424. ACM, 2002.
- [34] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. A complete axiomatisation of the ZX-calculus for Clifford+T quantum mechanics. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '18*, pages 559–568, New York, NY, USA, 2018. ACM.
- [35] Robin Kaarsgaard. *The logic of reversible computing*. PhD thesis, University of Copenhagen, 2017.
- [36] Aleks Kissinger and Vladimir Zamdzhiev. Quantomatic: A proof assistant for diagrammatic reasoning. In *International Conference on Automated Deduction*, pages 326–336. Springer, 2015.
- [37] Maxim Kontsevich. Operads and motives in deformation quantization. *Letters in Mathematical Physics*, 48(1):35–72, 1999.
- [38] Yves Lafont. Towards an algebraic theory of boolean circuits. *Journal of Pure and Applied Algebra*, 184(2-3):257–310, 2003.
- [39] Rolf Landauer. Irreversibility and heat generation in the computing process. *IBM journal of research and development*, 5(3):183–191, 1961.
- [40] Saunders MacLane. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1971. Graduate Texts in Mathematics, Vol. 5.
- [41] Karl Marx. *Grundrisse der Kritik der Politischen Ökonomie*.

- [42] T Monz, K Kim, W Hänsel, M Riebe, AS Villar, P Schindler, M Chwalla, M Hennrich, and R Blatt. Realization of the quantum Toffoli gate with trapped ions. *Physical review letters*, 102(4):040501, 2009.
- [43] Kang Feng Ng and Quanlong Wang. Completeness of the ZX-calculus for pure qubit Clifford+T quantum mechanics. *arXiv preprint arXiv:1801.07993*, 2018.
- [44] Michael A Nielsen and Isaac L Chuang. Quantum computation and quantum information.
- [45] Matthew D Reed, Leonardo DiCarlo, Simon E Nigg, Luyan Sun, Luigi Frunzio, Steven M Girvin, and Robert J Schoelkopf. Realization of three-qubit quantum error correction with superconducting circuits. *Nature*, 482(7385):382, 2012.
- [46] Peter Selinger. A survey of graphical languages for monoidal categories. In *New structures for physics*, pages 289–355. Springer, 2010.
- [47] Peter Selinger. Generators and relations for n-qubit Clifford operators. *Logical Methods in Computer Science*, 11, 2015.
- [48] Vivek V Shende, Aditya K Prasad, Igor L Markov, and John P Hayes. Synthesis of reversible logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(6):710–722, 2003.
- [49] Peter W Shor. Fault-tolerant quantum computation. In *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pages 56–65. IEEE, 1996.
- [50] John van de Wetering and Sal Wolffs. Completeness of the phase-free ZH-calculus, 2019.
- [51] Renaud Vilmart. A ZX-calculus with triangles for Toffoli-Hadamard, Clifford+T, and beyond. *arXiv preprint arXiv:1804.03084*, 2018.
- [52] Quanlong Wang. *Completeness of the ZX-calculus*. PhD thesis, University of Oxford, 2018.

- [53] John Watrous. Lecture 12: Grover's algorithm, March 2006. <https://cs.uwaterloo.ca/~watrous/LectureNotes/CPSC519.Winter2006/12.pdf>.

Appendix A

Matrices and Constants

Boltzmann's constant (k_B)

$$k_B := 1.380649 \times 10^{-23} J/K$$

$\pi/4$ -phase (ω)

$$\omega := e^{i\pi/4}$$

Plus state and effect

$$|+\rangle := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \langle +| := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \end{bmatrix}$$

Minus state and effect

$$|-\rangle := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad \langle -| := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \end{bmatrix}$$

Z spider (co)units

$$|0\rangle + |1\rangle = \sqrt{2}|+\rangle = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \langle 0| + \langle 1| = \sqrt{2}\langle +| = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

Zero ancillary bits

$$|0\rangle := \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \langle 0| := \begin{bmatrix} 1 & 0 \end{bmatrix}$$

X spider (co)units

$$|+\rangle + |-\rangle = 2|0\rangle = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad \langle +| + \langle -| = 2\langle 0| = \begin{bmatrix} 2 & 0 \end{bmatrix}$$

One ancillary bits

$$|1\rangle := \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \langle 1| := \begin{bmatrix} 0 & 1 \end{bmatrix}$$

Pauli X matrix/not gate/X spider π -phase

$$\text{not} := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Pauli Y matrix

$$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

π -phase-shift gate/Pauli Z matrix/Z spider π -phase

$$\begin{bmatrix} 1 & 0 \\ 0 & e^{\pi i} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Controlled-not/controlled- X gate

$$\text{cnot} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Controlled- Z gate

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

Toffoli gate

$$\text{tof} := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Swap gate ($c_{1,1}$)

$$\text{swap} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Hadamard gate

$$H := |+\rangle\langle 0| + |-\rangle\langle 1| = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$\pi/2$ phase-shift gate

$$S := \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

Z spider multiplication

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

X spider multiplication

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Appendix B

Basic Calculations for CNOT

Lemma B.0.1.

(i)

$$\begin{aligned}
 \begin{array}{c} \text{---} \oplus \text{---} \\ | \\ \text{---} \oplus \text{---} \\ | \\ \text{---} \oplus \text{---} \end{array} &= \begin{array}{c} \text{---} \oplus \text{---} \\ | \\ \text{---} \oplus \text{---} \\ | \\ \text{---} \oplus \text{---} \end{array} && [\text{CNOT.8}] \\
 &= \begin{array}{c} \text{---} \oplus \text{---} \\ | \\ \text{---} \oplus \text{---} \\ | \\ \text{---} \oplus \text{---} \end{array} && [\text{CNOT.5}] \\
 &= \begin{array}{c} \text{---} \oplus \text{---} \\ | \\ \text{---} \oplus \text{---} \\ | \\ \text{---} \oplus \text{---} \end{array} && [\text{CNOT.8}]
 \end{aligned}$$

(ii)

$$\begin{aligned}
 \Sigma &:= \begin{array}{c} \text{---} \oplus \text{---} \\ | \\ \text{---} \oplus \text{---} \\ | \\ \text{---} \oplus \text{---} \end{array} \\
 &= \begin{array}{c} \text{---} \oplus \text{---} \\ | \\ \text{---} \oplus \text{---} \\ | \\ \text{---} \oplus \text{---} \end{array} && [\text{CNOT.4}] \\
 &= \begin{array}{c} \text{---} \oplus \text{---} \\ | \\ \text{---} \oplus \text{---} \end{array} && [\text{CNOT.2}] \\
 &= && [\text{CNOT.6}]
 \end{aligned}$$

(iii)

$$\begin{aligned} \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} &= \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} && \text{[CNOT.2]} \\ &= \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} && \text{[CNOT.8]} \end{aligned}$$

(iv)

$$\begin{aligned} \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} &= \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} && \text{[CNOT.4]} \\ &= \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} && \text{[CNOT.8]} \\ &= \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} && \text{[CNOT.3]} \\ &= \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} && \text{[CNOT.8]} \\ &= \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} && \text{[CNOT.4]} \end{aligned}$$

Lemma B.0.2.

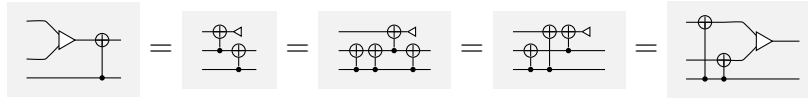
(i)

$$\begin{array}{c} \text{---} \\ \text{---} \end{array} = \begin{array}{c} \oplus \\ \text{---} \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \text{---} \end{array}$$

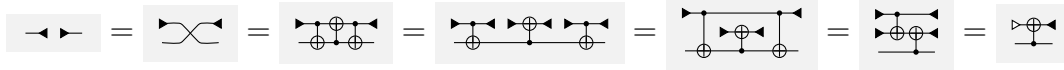
(ii)

$$\begin{array}{c} \text{---} \\ \text{---} \end{array} = \begin{array}{c} \oplus \\ \text{---} \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \\ \oplus \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \oplus \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \text{---} \end{array}$$

(iii)

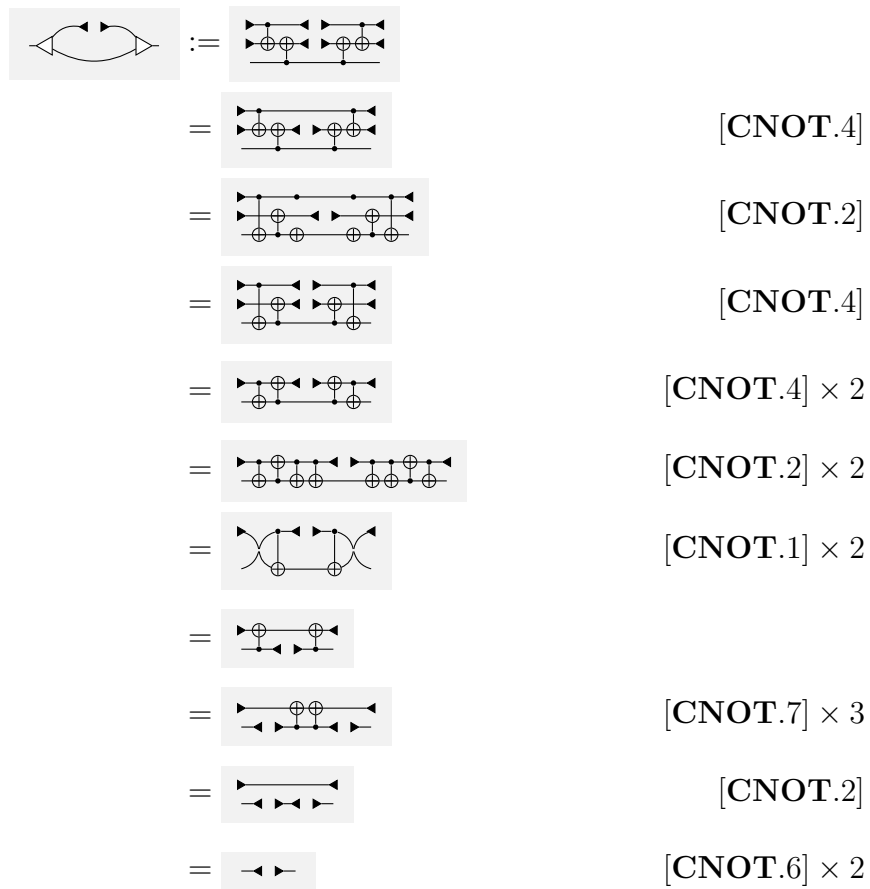


(iv)



Lemma B.0.3. $\nabla_1 \circ (|1\rangle\langle 1|) \otimes 1_1 \circ \Delta_1$

Proof.



□