

UNIVERSITY OF CALGARY

An investigation of quantum and reversible computing

by

Brett Gordon Giles

A DISSERTATION

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

August, 2013

© Brett Gordon Giles 2013

Abstract

Acknowledgements

Table of Contents

Abstract	i
Acknowledgements	ii
Table of Contents	iii
List of Tables	vi
List of Figures	vii
List of Symbols	viii
1 Introduction	4
2 Abstract Computability	5
2.1 Categories	5
2.1.1 Enrichment of categories	7
2.1.2 Examples of categories	7
2.1.3 Properties of maps	9
2.1.4 Limits and colimits in categories	11
2.1.5 Functors and natural transformations	12
2.1.6 Categories with additional structure	14
2.2 Restriction categories	14
2.2.1 Enrichment and meets	16
2.2.2 Range categories	19
2.2.3 Partial monics, sections and isomorphisms	21
2.2.4 Split restriction categories	23
2.2.5 Partial Map Categories	27
2.2.6 Restriction products and Cartesian restriction categories	28
2.2.7 Graphic Categories	30
2.3 Turing Categories	33
3 Inverse categories	34
3.1 Inverse products	34
3.1.1 Inverse categories with restriction products	34
3.1.2 Inverse products	36
3.1.3 Discrete inverse categories	38
3.1.4 The inverse subcategory of a discrete restriction category	44
3.2 Completing a discrete inverse category	47
3.2.1 The restriction category $\widetilde{\mathbb{X}}$	47
3.2.2 The category $\widetilde{\mathbb{X}}$ is a discrete restriction category	56
3.2.3 Equivalence of categories	60
3.2.4 Examples of the $\widetilde{(-)}$ construction	65
3.3 Coproducts in restriction categories	66
3.3.1 Coproducts	66
3.3.2 Inverse categories with restriction coproducts	69
3.4 Disjointness in an inverse category	69
3.4.1 Disjointness relations	69
3.4.2 Disjoint joins	76
3.4.3 Monoidal Tensors for disjointness	80

3.5	Inverse sum categories	94
3.5.1	Inverse sums	94
3.5.2	Inverse sum tensor	100
3.6	Completing a distributive inverse category	110
3.6.1	Distributive restriction categories	110
3.6.2	Distributive inverse categories	110
3.7	Inverse Turing Categories	111
3.8	Reversible computation	111
3.8.1	Reversible Turing machines	112
3.8.2	Reversible automata and linear combinatory algebras	117
4	Quantum computation	125
4.1	Linear algebra	125
4.1.1	Basic definitions	125
4.1.2	Matrices	126
4.2	Quantum computation overview	128
4.2.1	Density matrix representation	133
4.3	Dagger categories	135
4.3.1	Definitions	135
4.3.2	Examples of dagger categories	138
4.4	Semantics of quantum computation	140
4.4.1	Semantics of QPL	140
4.4.2	Semantics of pure quantum computations	153
4.4.3	Bases and Frobenius Algebras	155
4.4.4	Quantum and classical data	156
4.4.5	Complete positivity	158
5	Frobenius Algebras and Quantum Computation	161
5.1	The category of Commutative Frobenius Algebras	161
6	$D[\omega]$ based \dagger -categories	162
6.1	Toy quantum semantics	162
6.2	Introduction to synthesis	162
6.3	Algebraic background	162
6.3.1	Conjugate and norm	163
6.3.2	Denominator exponents	164
6.3.3	Residues	164
6.4	Exact synthesis of single qubit operators	167
6.4.1	Existence	169
6.4.2	T -Optimality	172
6.4.3	Uniqueness	172
6.4.4	The Matsumoto-Amano decomposition algorithm	176
6.4.5	A characterization of Clifford+ T on the Bloch sphere	177
6.4.6	Alternative normal forms	181
6.4.7	Matsumoto-Amano normal forms and $U(2)$	185
6.5	Exact synthesis of multi-qubit operators	188
6.5.1	Decomposition into two-level matrices	189
6.5.2	Main result	195

6.5.3	The no-ancilla case	197
6.5.4	Complexity	199
7	Conclusions and future work	201
	Bibliography	202

List of Tables

3.1	Structural maps for the tensor in $Inv(\mathbb{X})$	45
4.1	Interpretation of QPL operations	150
6.1	Some operations on residues	165

List of Figures and Illustrations

4.1	Classical flowcharts	140
4.2	General flowcharts	141
4.3	Quantum flowcharts	141
4.4	Example of a subroutine and loop	142
4.5	Unwinding a loop	143
6.1	The action of Matsumoto-Amano normal forms on k -parities. All matrices are written modulo the right action of the Clifford group, i.e., modulo a permutation of the columns.	174
6.2	Transitions of residue matrices in U2 when applying the Matsumoto-Amano algorithm	186

List of Symbols, Abbreviations and Nomenclature

Symbol	Definition
U of C	University of Calgary
\mathbb{N}	The set of natural numbers, i.e., $\{0, 1, 2, \dots\}$
\mathbb{Z}	The ring of integers numbers, i.e., $\{0, \pm 1, \pm 2, \dots\}$
\mathbb{C}	The field of complex numbers

Overview of thesis chapters

Introduction

This chapter will give a brief introduction to and an explanation of both reversible and quantum computing. I will discuss the relationship of both of these, along with a brief introduction to their categorical semantics. I will discuss the equivalence of reversible Turing machines and standard Turing machines.

Abstract Computability

This chapter includes a basic introduction to category theory. Specific areas include definitions of categories, natural transformations, functors and tensors. It introduces limits and co-limits, focussing on products and co-products.

This chapter then moves on to introduce restriction categories and discuss how Cartesian restriction categories can be used to model standard computing. I will introduce Turing categories and show how this can be used to create a Partial Combinatory Algebra.

Inverse categories and Reversible computing

An inverse category is a specific type of restriction category, in which each map has a partial inverse. An inverse category corresponds to a restriction category in the same way a groupoid corresponds to a category.

Inverse categories are explored, along with some basic results regarding products and idempotent splitting. The inverse product is introduced, along with the concept of a discrete inverse category and the relationship between a discrete inverse category and a Cartesian restriction category.

Next, I explore the inverse sum, disjointness, and the disjoint join of maps in an inverse

category. These provide a way to work with objects which behave like co-products in the inverse category. The interaction of the inverse sum and inverse product will be explained.

The chapter will define inverse Turing Categories.

The foregoing is based on two papers which are in preparation (joint work with R. Cockett).

To conclude the chapter, I will provide a detailed proof of the equivalence of reversible Turing machines and standard Turing machines and connect this to inverse categories.

Quantum Computation

This chapter introduces quantum computation, focussing on the semantics of quantum computing as described using \dagger -categories. This will include examples of “Toy” quantum semantics.

Frobenius Algebras in \dagger -Categories

In this chapter, I highlight the connection between the model of reversible computing (inverse categories) introduced in this thesis, and that of quantum computing. Frobenius Algebras provide a way of describing the basis used in quantum computation.

$D[\omega]$ based \dagger -categories and exact synthesis

I will give an example of a toy quantum semantics on matrices over $D[\omega]$ (where $D = \mathbb{Z}[\frac{1}{2}]$). The primary result of this chapter is that these matrices are equivalent to the Clifford Group $+ T$ over multiple qubits.

In preparation for the main result, I will discuss the issue of gate synthesis, where an arbitrary quantum transform is to be expressed in terms of a set of base gates. An overview of the history of both approximate and exact synthesis will be given.

Then, the chapter continues with an algorithm for exact synthesis of single-qubit transforms over the Clifford group $+ T$, together with a normal form and characterization of these. This is based on a paper that is an extension of work done by Matsumoto and Amano (joint work with P. Selinger.)

Finally, I present the algorithm for exact synthesis over the Clifford group of multi-qubit transforms and characterize those transforms that may be exactly synthesized. This is based on a paper published in the journal Physical Review A (joint work with P. Selinger).

Chapter 1

Introduction

Chapter 2

Abstract Computability

2.1 Categories

A category as a mathematical object can be defined in a variety of equivalent ways. As much of our work will involve the exploration of partial and reversible maps, their domains and ranges, we choose a definition that highlights the algebraic nature of these. Note that ranges are normally referred to as codomains in category theory and we will use the codomain terminology in this section.

Definition 2.1.1. A *category* \mathbb{A} is a collection of maps together with two functions, D and C , from \mathbb{A} to \mathbb{A} and a partial associative composition of maps (written by juxtaposing maps), such that:

$$[\mathbf{C.1}] \quad D(f)f \text{ is defined and equals } f,$$

$$[\mathbf{C.2}] \quad fC(f) \text{ is defined and equals } f,$$

$$[\mathbf{C.3}] \quad fg \text{ is defined iff } C(f) = D(g) \text{ and } D(fg) = D(f) \text{ and } C(fg) = C(g),$$

$$[\mathbf{C.4}] \quad (fg)h = f(gh) \text{ whenever either side is defined,}$$

$$[\mathbf{C.5}] \quad D(C(x)) = C(x), C(D(x)) = D(x) \text{ and } C, D \text{ are both idempotent.}$$

A more familiar definition, often used in introducing categories, is given next.

Definition 2.1.2. A *category* \mathbb{A} is a directed graph consisting of objects A_o and maps A_m . Each $f \in A_m$ has two associated objects in A_o , called the domain and codomain. When f has domain X and codomain Y we will write $f : X \rightarrow Y$. For $f, g \in A_m$, if $f : X \rightarrow Y$ and $g : Y \rightarrow Z$, there is a map called the *composite* of f and g , written fg such that $fg : X \rightarrow Z$.

For any $W \in A_o$ there is an *identity* map $1_W : W \rightarrow W$. Additionally, these two axioms must hold:

$$[\mathbf{C'}.1] \text{ for } f : X \rightarrow Y, 1_X f = f = f 1_Y,$$

$$[\mathbf{C'}.2] \text{ given } f : X \rightarrow Y, g : Y \rightarrow Z \text{ and } h : Z \rightarrow W, \text{ then } f(gh) = (fg)h.$$

Lemma 2.1.3. *A category as defined in Definition 2.1.1 is equivalent to a category as defined in Definition 2.1.2 and vice versa.*

Proof. Assume \mathbb{A} is as in Definition 2.1.1. Then set A_o to the collection of all $D(f)$ and $C(f)$. Set A_m to all the maps in \mathbb{A} . The domain of any map $f \in A_m$ is $D(f)$ and the codomain is $C(f)$. By $[\mathbf{C}.3]$, for $f : X \rightarrow Y$ and $g : Y \rightarrow Z$ the composite fg is defined. The identity map of the object $D(f)$ is the map $D(f)$ and the identity map of the object $C(f)$ is $C(f)$. By $[\mathbf{C}.5]$, we see $[\mathbf{C'}.1]$ is satisfied. By $[\mathbf{C}.4]$, we see $[\mathbf{C'}.2]$ is satisfied. Therefore, \mathbb{A} satisfies Definition 2.1.2.

Conversely, assume \mathbb{Z} is as in Definition 2.1.2. Then, we already have the collection of maps, Z_m . For each $f : A \rightarrow B \in Z_m$, set $D(f) = 1_A$ and $C(f) = 1_B$. By the definition of the identity maps and $[\mathbf{C'}.1]$, we see $[\mathbf{C}.1]$, $[\mathbf{C}.2]$ and $[\mathbf{C}.5]$ are all satisfied. From the composition requirements on \mathbb{Z} and $[\mathbf{C'}.2]$, it follows that $[\mathbf{C}.4]$ is satisfied. For $[\mathbf{C}.3]$, assume fg is defined. Then for some $A, B, C \in Z_o$, $f : A \rightarrow B$ and $g : B \rightarrow C$. This gives us $1_B = C(f) = D(g)$, $1_A = D(fg) = D_f$ and $1_B = C(fg) = C(g)$. Next, assume we have $C(f) = D(g)$, $D(fg) = D(f)$ and $C(fg) = C(g)$. This tells us the codomain of f is some object B which is also the domain of g , hence we may form the composition fg which will have domain A , the domain of f and codomain C , the codomain of g . \square

As we have shown the two definitions are equivalent, it will be convenient to reference either definition and manner of referring to a category throughout this thesis. Essentially, we will use whichever definition seems the most appropriate to use at any point.

We may also consider the notion of containment between categories.

Definition 2.1.4. Given the categories \mathbb{C} and \mathbb{D} , we may say the following:

- (i) \mathbb{C} is a *sub-category* of \mathbb{D} when each object of \mathbb{C} is an object of \mathbb{D} and when each map of \mathbb{C} is a map of \mathbb{D} .
- (ii) \mathbb{C} is a *full sub-category* of \mathbb{D} when it is a sub-category and given A, B objects in \mathbb{C} and $f : A \rightarrow B$ in \mathbb{D} , then f is a map in \mathbb{C} .

2.1.1 Enrichment of categories

Definition 2.1.5. If \mathbb{X} is a category, then $\mathbb{X}(A, B)$ is called a *hom-collection* of \mathbb{X} and consists of all arrows f with $D(f) = A$ and $C(f) = B$.

In the case where the hom-objects of a category \mathbb{X} are all sets, we call them hom-sets. Additionally, we say \mathbb{X} is *enriched* in SETS. We may extend this to any mathematical structure, e.g., enriched in partial orders, enriched in groups, etc..

Specific types of enrichment may force a specific structure on a category. For example, if \mathbb{X} is enriched in sets of cardinality of 0 or 1, then \mathbb{X} must be a preorder.

2.1.2 Examples of categories

In this section, we will offer a few examples of categories. As Definition 2.1.2 tends to be a more succinct way to present the data of a category, this section will give the examples in terms of objects and maps rather than the “object-free” definition.

Categories based on SETS

There are three primary categories of interest to us where the objects are the collection of sets. The first is SETS, where the maps are given by all set functions. The second is PAR, where the maps are all partial maps. In each case, the standard definition of functions suffices to ensure identities, compositions and associativity are all satisfied. Domain and codomain are given by the domain and range respectively.

A third example, often of interest in quantum programming language semantics is REL:

Objects: Sets

Maps: Relations: $R : X \rightarrow Y$

Identity: $1_X = \{(x, x) | x \in X\}$

Composition: $RS = \{(x, z) | \exists y, (x, y) \in R \text{ and } (y, z) \in S\}$

Note that **REL** is enriched in posets, via set inclusion. **PAR** can be viewed as a subcategory of **REL**, with the same objects, but only allowing maps which are functions, i.e., if $(x, y), (x, y') \in R$, then $y = y'$. **PAR** is also enriched in posets, via the same inclusion ordering as in **REL**.

Matrix categories

Given a rig R (i.e., a ring minus negatives, e.g., the positive rationals), one may form the category **MAT** (R).

Objects: \mathbb{N}

Maps: $[r_{ij}] : n \rightarrow m$ where $[r_{ij}]$ is an $n \times m$ matrix over R

Identity: I_n

Composition: Matrix multiplication

Dual categories

Given a category \mathbb{C} , we may form the *dual* of \mathbb{C} , written \mathbb{C}^{op} as the following category:

Objects: The objects of \mathbb{C}

Maps: $f^{op} : B \rightarrow A$ in \mathbb{C}^{op} when $f : A \rightarrow B$ in \mathbb{C} .

Identity: The identity maps of \mathbb{C}

Composition: If $fg = h$ in \mathbb{C} , $g^{op}f^{op} = h^{op}$

2.1.3 Properties of maps

Many interesting properties of maps are generalizations of notions that have been found useful in considering sets and functions. We present a few of these in a tabular format, together with their categorical definition. Throughout the table, e, f, g are maps in a category C with $e : A \rightarrow A$ and $f, g : A \rightarrow B$.

Sets	Categorical Property	Definition
Injective	Monic	f is monic whenever $hf = kf$ means that $h = k$.
Surjective	Epic	The dual notion to monic, g is epic whenever $gh = gk$ means that $h = k$. A map that is both monic and epic is called <i>bijic</i> .
Left Inverse	Section	f is a section when there is a map f^* such that $ff^* = 1_A$. f is also referred to as the <i>left inverse</i> of f^* .
Right Inverse	Retraction	f is a retraction when there is a map f_* such that $f_*f = 1_B$. f is also referred to as the <i>right inverse</i> of f_* . A map that is both a section and a retraction is called an <i>isomorphism</i> .
Idempotent	Idempotent	An endomap e is idempotent whenever $ee = e$.

We state without proof a number of properties of maps.

Lemma 2.1.6. *In a category \mathbb{C} ,*

- (i) *If f, g are monic, then fg is monic.*
- (ii) *If fg is monic, then f is monic.*
- (iii) *f being a section means it is monic.*

(iv) f, g sections implies that fg is a section.

(v) fg a section means f is a section.

Lemma 2.1.7. *If $f : A \rightarrow B$ is both a section and a retraction, then $f^* = f_*$.*

Lemma 2.1.8. *f is an isomorphism if and only if it is an epic section.*

Note there are corresponding properties for epics and retractions, obtained by dualizing the statements of Lemma 2.1.6 and Lemma 2.1.8.

Suppose $f : A \rightarrow B$ is a retraction with left inverse $f_* : B \rightarrow A$. Note that ff_* is idempotent as $ff_*ff_* = f1_Bf_* = ff_*$. If we are given an idempotent e , we say e is *split* if there is a retraction f with $e = ff_*$.

In general, not all idempotents in a category will split. The following construction allows us to create a category based on the original one in which all idempotents do split.

Definition 2.1.9. Given a category \mathbb{C} we define $Split(\mathbb{C})$ as the following category:

Objects: (A, e) , where A is an object of \mathbb{C} , $e : A \rightarrow A$ and $e \in E$.

Maps: $f_{d,e} : (A, d) \rightarrow (B, e)$ is given by $f : A \rightarrow B$ in \mathbb{C} , where $f = dfe$.

Identity: The map $e_{e,e}$ for (A, e) .

Composition: Inherited from \mathbb{C} .

Lemma 2.1.10. *Given a category \mathbb{C} , then it is a full sub-category of $Split(\mathbb{C})$ and all idempotents split in $Split(\mathbb{C})$.*

Proof. We identify each object A in \mathbb{C} with the object $(A, 1)$ in $Split(\mathbb{C})$. The only maps between $(A, 1)$ and $(B, 1)$ in $Split(\mathbb{C})$ are the maps between A and B in \mathbb{C} , hence we have a full sub-category.

Suppose we have the map $d_{e,e} : (A, e) \rightarrow (A, e)$ with $dd = d$, i.e., it is idempotent in \mathbb{C} and $Split(\mathbb{C})$. In $Split(\mathbb{C})$, we have the map $d_{e,d} : (A, e) \rightarrow (A, d)$ and $d_{d,e} : (A, d) \rightarrow (A, e)$ where $d_{d,e}d_{e,d} = d_{d,d} = 1_{(A,d)}$ and $d_{e,d}d_{d,e} = d_{e,e}$, hence it is a splitting of the map $d_{e,e}$. \square

2.1.4 Limits and colimits in categories

We shall discuss only a few basic limits/colimits in categories. First we discuss initial and terminal objects.

Definition 2.1.11. An *initial object* in a category \mathbb{C} is an object which has exactly one map to each other object in the category. The dual notion is *terminal object* which has exactly one map from each other object in the category.

Lemma 2.1.12. Suppose I, J are initial objects in \mathbb{C} . Then there is a unique isomorphism $i : I \rightarrow J$.

Proof. First, note that by definition there is only one map from I to I — which must be the identity map. As I is initial there is a map $i : I \rightarrow J$. As J is initial there is a map $j : J \rightarrow I$. But this means $ij : I \rightarrow I = 1$ and $ji : J \rightarrow J = 1$ and hence i is the unique isomorphism from I to J . \square

Dually, we have the corresponding result of Lemma 2.1.12 for terminal objects — they are also unique up to a unique isomorphism.

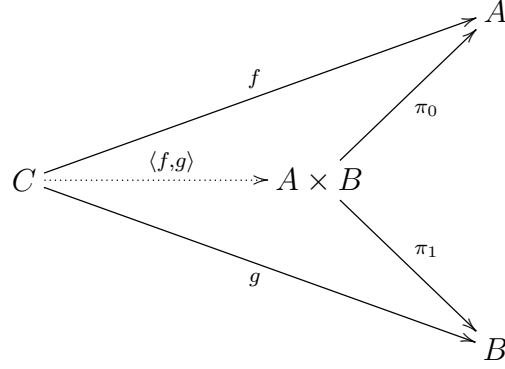
In categories, we normally designate the initial object by 0 and the terminal object by 1 .

We now turn to products and co-products.

Definition 2.1.13. Let A, B be objects of the category \mathbb{C} . Then the object $A \times B$ is a *product* of A and B when:

- There exist maps π_0, π_1 with $\pi_0 : A \times B \rightarrow A$, $\pi_1 : A \times B \rightarrow B$;
- Given an object C with maps $f : C \rightarrow A$ and $g : C \rightarrow B$ there exists an

unique map $\langle f, g \rangle$ such that the following diagram commutes:



2.1.5 Functors and natural transformations

Definition 2.1.14. A map $F : \mathbb{X} \rightarrow \mathbb{Y}$ between categories (as in Definition 2.1.1 is called a *functor*, provided it satisfies the following:

$$[\mathbf{F.1}] \quad F(D(f)) = D(F(f)) \text{ and } F(C(f)) = C(F(f));$$

$$[\mathbf{F.2}] \quad F(fg) = F(f)F(g);$$

Lemma 2.1.15. *The collection of categories and functors form the category CAT.*

Proof. **Objects:** Categories.

Maps: Functors.

Identity: The identity functor which takes a map to the same map.

Composition: $FG(x) = F(G(x))$ which is clearly associative.

□

We will often restrict ourselves to specific classes of functors which either *preserve* or *reflect* certain characteristics of the domain category or codomain category. To be more precise, we provide some definitions.

Definition 2.1.16. A *diagram* in a category is a collection of objects and maps between those objects which satisfy categorical composition rules. More precisely: Given a category \mathbb{C} , a diagram in a category \mathbb{C} of *shape* \mathbb{J} is a functor $D : \mathbb{J} \rightarrow \mathbb{C}$.

In practice, diagrams are pictorially represented by drawing the objects and the maps between them.

Definition 2.1.17. A *property* of a diagram D , written $P(D)$ is a logical relation expressed using the objects and maps of the diagram D .

Example 2.1.18. $P(f : A \rightarrow B) = \exists h : B \rightarrow A. hf = 1_A$ expresses that f is a retraction.

Definition 2.1.19. A functor F *preserves* the property P over maps f_i and objects A_j when $P(f_1, \dots, f_n, A_1, \dots, A_m)$ implies $P(F(f_1), \dots, F(f_n), F(A_1), \dots, F(A_m))$.

Definition 2.1.20. A functor F *reflects* the property P over maps f_i and objects A_j when $P(F(f_1), \dots, F(f_n), F(A_1), \dots, F(A_m))$ implies $P(f_1, \dots, f_n, A_1, \dots, A_m)$.

For example, all functors preserve the properties of being an idempotent or a retraction or section, but in general, not the property of being monic.

A functor $F : \mathbb{C} \rightarrow \mathbb{D}$ induces a map between hom-objects in \mathbb{C} and hom-objects in \mathbb{D} . For each object A, B in \mathbb{C} we have the map:

$$F_{AB} : \mathbb{C}(A, B) \rightarrow \mathbb{D}(F(A), F(B)).$$

Definition 2.1.21. Given a functor $F : \mathbb{C} \rightarrow \mathbb{D}$, we say:

- F is *faithful* when for all A, B , F_{AB} is an injective function;
- F is *full* when for all A, B , F_{AB} is a surjective function.

Definition 2.1.22. Given functors $F, G : \mathbb{X} \rightarrow \mathbb{Y}$, a *natural transformation* $\alpha : F \Rightarrow G$ is a collection of maps in \mathbb{Y} , $\alpha_X : F(X) \rightarrow G(X)$, indexed by the objects of \mathbb{X} such that for all $f : X_1 \rightarrow X_2$ in \mathbb{X} the following diagram in \mathbb{Y} commutes:

$$\begin{array}{ccc} F(X_1) & \xrightarrow{F(f)} & F(X_2) \\ \alpha_{X_1} \downarrow & & \downarrow \alpha_{X_2} \\ G(X_1) & \xrightarrow{G(f)} & G(X_2) \end{array}$$

2.1.6 Categories with additional structure

Definition 2.1.23 (Symmetric Monoidal Category). A *symmetric monoidal category* \mathbf{D} is a category equipped with a monoid \oplus (a bi-functor $\oplus : \mathbf{D} \times \mathbf{D} \rightarrow \mathbf{D}$) together with three families of natural isomorphisms: $a_{A,B,C} : A \otimes (B \otimes C) \rightarrow (A \otimes B) \otimes C$, $u_A : A \rightarrow A \oplus I$ and $c_{A,B} : A \oplus B \rightarrow B \oplus A$, which satisfy specific coherence diagrams. The isomorphisms are referred to as the *structure isomorphisms* for the symmetric monoidal category. I is the unit of the monoid.

For details on the coherence diagrams, please see e.g., [8] or [22]. The essence of the coherence diagrams is that any diagram composed solely of the structure isomorphisms will commute.

Definition 2.1.24 (Compact Closed Category). A *compact closed category* \mathbf{D} is a symmetric monoidal category with monoid \otimes where each object A has a dual A^* and there exist families of maps $\eta_A : I \rightarrow A^* \otimes A$ (the *unit*) and $\epsilon_A : A \otimes A^* \rightarrow I$ (the *counit*) such that

$$\begin{array}{ccccc} A & \xrightarrow{u_A} & A \otimes I & \xrightarrow{1 \otimes \eta_A} & A \otimes (A^* \otimes A) \\ \parallel & & & & \downarrow a_{A,A^*,A} \\ A & \xleftarrow{u_A^{-1}} & I \otimes A & \xleftarrow{\otimes \epsilon_B \otimes 1} & (A \otimes A^*) \otimes A \end{array}$$

commutes and so does the similar one based on A^* .

Given a map $f : A \rightarrow B$ in a compact closed category, define the map $f^* : B^* \rightarrow A^*$ as

$$\begin{array}{ccccc} B^* & \xrightarrow{u_{B^*}} & I \otimes B^* & \xrightarrow{\eta_A \otimes 1} & A^* \otimes A \otimes B^* \\ f^* \downarrow & & & & \downarrow 1 \otimes f \otimes 1 \\ A^* & \xleftarrow{u_{A^*}^{-1}} & A^* \otimes I & \xleftarrow{1 \otimes \epsilon_B} & A^* \otimes B \otimes B^* \end{array}$$

2.2 Restriction categories

Restriction categories were introduced in [27] as a convenient axiomatization of partial maps.

Definition 2.2.1. A *restriction category* is a category \mathbb{X} together with a *restriction operator* on maps:

$$\frac{f : A \rightarrow B}{\overline{f} : A \rightarrow A}$$

where f is an map of \mathbb{X} and A, B are objects of \mathbb{X} , such that the following four *restriction identities* hold, whenever the compositions¹ are defined.

$$[\mathbf{R.1}] \quad \overline{f}f = f$$

$$[\mathbf{R.2}] \quad \overline{g}\overline{f} = \overline{fg}$$

$$[\mathbf{R.3}] \quad \overline{\overline{f}g} = \overline{f}\overline{g}$$

$$[\mathbf{R.4}] \quad f\overline{g} = \overline{fg}f$$

Definition 2.2.2. A *restriction functor* is a functor which preserves the restriction. That is, given a functor $F : \mathbb{X} \rightarrow \mathbb{Y}$ with \mathbb{X} and \mathbb{Y} restriction categories, F is a restriction functor if:

$$F(\overline{f}) = \overline{F(f)}.$$

Any map such that $r = \overline{r}$ is an idempotent, as $\overline{r}r = \overline{\overline{r}r} = \overline{r}$, and is called a *restriction idempotent*. All maps \overline{f} are restriction idempotents as $\overline{f} = \overline{\overline{f}}$. Below, we record some basic facts for restriction categories shown in [27] pp 4-5:

Lemma 2.2.3. *In a restriction category \mathbb{X} ,*

$$(i) \quad \overline{f} \text{ is idempotent};$$

$$(v) \quad \overline{f}\overline{g} = \overline{\overline{f}g};$$

$$(ii) \quad \overline{fg} = \overline{f}g\overline{f};$$

$$(vi) \quad f \text{ monic implies } \overline{f} = 1;$$

$$(iii) \quad \overline{fg} = \overline{f}\overline{g};$$

$$(vii) \quad f = \overline{g}f \implies \overline{g}\overline{f} = \overline{f}.$$

$$(iv) \quad \overline{\overline{f}} = \overline{f};$$

A map $f : A \rightarrow B$ in a restriction category is said to be *total* when $\overline{f} = 1_A$. The total maps in a restriction category form a subcategory $Total(\mathbb{X}) \subseteq \mathbb{X}$.

¹Note that composition is written in diagrammatic order throughout this paper.

An example of a restriction category is PAR, the category with objects sets and arrows the partial functions between sets. In PAR, the restriction of $f : A \rightarrow B$ is:

$$\bar{f}(x) = \begin{cases} x & \text{if } f(x) \text{ is defined,} \\ \uparrow & \text{if } f(x) \text{ is } \uparrow. \end{cases}$$

(The symbol \uparrow means that the function is undefined at that element). In PAR, the total maps correspond precisely to the functions that are defined on all elements of the domain.

2.2.1 Enrichment and meets

In any restriction category, there is a partial order on each hom-set, given by $f \leq g$ iff $\bar{f}g = f$, where $f, g : A \rightarrow B$.

Lemma 2.2.4. *In a restriction category \mathbb{X} :*

- (i) \leq as defined above is a partial order on each hom-set;
- (ii) $f \leq g \implies \bar{f} \leq \bar{g}$;
- (iii) $f \leq g \implies hf \leq hg$;
- (iv) $f \leq g \implies fh \leq gh$;
- (v) $f \leq 1 \iff f = \bar{f}$.

Proof.

- (i) With f, g, h parallel maps in \mathbb{X} , each of the requirements for a partial order is shown below:

Reflexivity: $\bar{f}f = f$ and therefore, $f \leq f$.

Anti-Symmetry: Given $\bar{f}g = f$ and $\bar{g}f = g$, it follows:

$$f = \bar{f}f = \overline{\bar{f}g}f = \bar{f}\bar{g}f = \bar{g}\bar{f}f = \bar{g}f = g.$$

Transitivity: Given $f \leq g$ and $g \leq h$,

$$\overline{f}h = \overline{\overline{f}gh} = \overline{f}\overline{g}h = \overline{f}g = f$$

showing that $f \leq h$.

(ii) The premise is that $\overline{f}g = f$. From this, $\overline{f}\overline{g} = \overline{\overline{f}g} = \overline{f}$, showing $\overline{f} \leq \overline{g}$.

(iii) $\overline{h}\overline{f}hg = h\overline{f}g = hf$ and therefore $hf \leq hg$.

(iv) $\overline{f}g = f$, this shows $\overline{f}hgh = \overline{\overline{f}gh}gh = \overline{f}\overline{g}hgh = \overline{f}gh = fh$ and therefore $fh \leq gh$.

(v) As $f \leq 1$ means precisely $\overline{f}1 = f$.

□

Lemma 2.2.4 on the preceding page shows that restriction categories are enriched in partial orders.

Definition 2.2.5. A restriction category has *meets* if there is an operation \cap on parallel maps:

$$\frac{A \xrightarrow{f} B}{A \xrightarrow{g} B} \quad A \xrightarrow{f \cap g} B$$

such that $f \cap g \leq f, f \cap g \leq g, f \cap f = f, h(f \cap g) = hf \cap hg$.

Meets were introduced in [12]. The following are basic results on meets:

Lemma 2.2.6. *In a restriction category \mathbb{X} with meets, where f, g, h are maps in \mathbb{X} , the following are true:*

(i) $f \leq g$ and $f \leq h \iff f \leq g \cap h$;

(ii) $f \cap g = g \cap f$;

(iii) $\overline{f \cap 1} = f \cap 1$;

- (iv) $(f \cap g) \cap h = f \cap (g \cap h)$;
- (v) $r(f \cap g) = rf \cap g$ where $r = \bar{r}$ is a restriction idempotent;
- (vi) $(f \cap g)r = fr \cap g$ where $r = \bar{r}$ is a restriction idempotent;
- (vii) $\overline{f \cap g} \leq \bar{f}$ (and therefore $\overline{f \cap g} \leq \bar{g}$);
- (viii) $(f \cap 1)f = f \cap 1$;
- (ix) $e(e \cap 1) = e$ where e is idempotent.

Proof.

- (i) $f \leq g$ and $f \leq h$ means precisely $f = \bar{f}g$ and $f = \bar{f}h$. Therefore,

$$\bar{f}(g \cap h) = \bar{f}g \cap \bar{f}h = f \cap f = f$$

and so $f \leq g \cap h$. Conversely, given $f \leq g \cap h$, we have $f = \bar{f}(g \cap h) = \bar{f}g \cap \bar{f}h \leq \bar{f}g$. But $f \leq \bar{f}g$ means $f = \bar{f}\bar{f}g = \bar{f}g$ and therefore $f \leq g$. Similarly, $f \leq h$.

- (ii) From (i) on the previous page, as by definition, $f \cap g \leq g$ and $f \cap g \leq f$.
- (iii) $f \cap 1 = \overline{f \cap 1}(f \cap 1) = (\overline{f \cap 1}f) \cap (\overline{f \cap 1}) \leq \overline{f \cap 1}$ from which the result follows.
- (iv) By definition and transitivity, $(f \cap g) \cap h \leq f, g, h$ therefore by (i) on the preceding page $(f \cap g) \cap h \leq f \cap (g \cap h)$. Similarly, $f \cap (g \cap h) \leq (f \cap g) \cap h$ giving the equality.

- (v) Given $rf \cap g \leq rf$, calculate:

$$rf \cap g = \overline{rf \cap g}rf = \overline{r(rf \cap g)}f = \overline{rrf \cap rgf} = \overline{r(f \cap g)}f = \overline{rf \cap gf} = r(f \cap g).$$

- (vi) Using the previous point with the restriction idempotent $\bar{f}r$,

$$\begin{aligned} fr \cap g &= f\bar{r} \cap g = \overline{f\bar{r}}f \cap g = \overline{f\bar{r}}(f \cap g) = \overline{f\bar{r}}\overline{f \cap g}f \\ &= \overline{f \cap g}\overline{f\bar{r}}f = \overline{f \cap g}f\bar{r} = (f \cap g)r. \end{aligned}$$

(vii) For the first claim,

$$\overline{f \cap g} \overline{f} = \overline{\overline{f}(f \cap g)} = \overline{(\overline{f}f) \cap g} = \overline{f \cap g}.$$

The second claim then follows by (ii) on page 17.

(viii) Given $f \cap 1 \leq f$:

$$f \cap 1 \leq f \iff \overline{f \cap 1} f = f \cap 1 \iff (f \cap 1)f = f \cap 1$$

where the last step is by item (iii) on page 17 of this lemma.

(ix) As e is idempotent, $e(e \cap 1) = (ee \cap e) = e$.

□

2.2.2 Range categories

Corresponding to Definition 2.2.1, which axiomatizes the concept of a domain of definition, we now introduce range categories which serve to axiomatize the concept of the range of a function.

Definition 2.2.7. A restriction category \mathbb{X} is a *range category* when it has an operator on all maps

$$\frac{f : A \rightarrow B}{\hat{f} : B \rightarrow B}$$

where the operator satisfies the following:

$$[\mathbf{RR.1}] \quad \overline{\hat{f}} = \hat{f}$$

$$[\mathbf{RR.2}] \quad f \hat{f} = f$$

$$[\mathbf{RR.3}] \quad \widehat{f \overline{g}} = \hat{f} \overline{g}$$

$$[\mathbf{RR.4}] \quad \widehat{\hat{f} g} = \hat{f} \hat{g}$$

whenever the compositions are defined.

Lemma 2.2.8. *In a range category \mathbb{X} , the following hold:*

$$\begin{array}{ll}
(i) \quad \hat{g}\hat{f} = \hat{f}\hat{g}; & (v) \quad \hat{f}\hat{f} = \hat{f}; \\
(ii) \quad \overline{f}\hat{g} = \hat{g}\overline{f}; & (vi) \quad \hat{\hat{f}} = \hat{f}; \\
(iii) \quad \widehat{f\hat{g}} = \hat{f}\hat{g}; & (vii) \quad \widehat{\hat{f}} = \overline{f}; \\
(iv) \quad \hat{f} = 1 \text{ when } f \text{ is epic, hence} & (viii) \quad \widehat{\hat{g}\hat{f}\hat{g}} = \widehat{\hat{f}\hat{g}}; \\
\hat{1} = 1; & (ix) \quad \widehat{\hat{f}\hat{g}} = \hat{f}\hat{g}.
\end{array}$$

Proof. See, e.g., [16]. □

Lemma 2.2.9. *In a range category:*

$$\begin{array}{ll}
(i) \quad \overline{f\hat{g}} \leq \overline{f}; & (iii) \quad f' \leq f \text{ implies } \overline{f'} \leq \overline{f}; \\
(ii) \quad \widehat{h\hat{f}} \leq \hat{f}; & (iv) \quad f' \leq f \text{ implies } \hat{f'} \leq \hat{f}.
\end{array}$$

Proof.

- (i) This is immediate from Lemma 2.2.3[(ii)].
- (ii) Noting that $\widehat{h\hat{f}}\hat{f} = \widehat{h\hat{f}\hat{f}} = \widehat{h\hat{f}}\hat{f} = \widehat{h\hat{f}}$, we see $\widehat{h\hat{f}} \leq \hat{f}$.
- (iii) $\overline{f'}\overline{f} = \overline{f'f} = \overline{f'}$, therefore $\overline{f'} \leq \overline{f}$.
- (iv) Calculating $\widehat{\hat{f'}}\hat{f} = \hat{f'}\hat{f} = \widehat{\hat{f'}}\hat{f}\hat{f} = \widehat{\hat{f'}}\hat{f}\hat{f} = \widehat{\hat{f'}}\hat{f} = \hat{f'}$, we see $\hat{f'} \leq \hat{f}$.

□

Remark 2.2.10. Note that unlike restrictions, a range is a *property* of a restriction category.

To see this, assume we have two ranges $\widehat{(-)}$ and $\widetilde{(-)}$. Then,

$$\hat{f} = \widehat{\hat{f}} = \hat{f}\hat{f} = \widetilde{\hat{f}} = \widetilde{\hat{f}}\hat{f} = \hat{f}.$$

Lemma 2.2.11. *An inverse category \mathbb{X} is a range category, where $\hat{f} = f^{(-1)}f = \overline{f^{(-1)}}$.*

Proof.

$$[\mathbf{RR.1}] \quad \widehat{f} = \overline{\overline{f^{(-1)}}} = \overline{f^{(-1)}} = \widehat{f};$$

$$[\mathbf{RR.2}] \quad f\widehat{f} = f\overline{f^{(-1)}} = f\overline{f^{(-1)}}f = \overline{f}f = f;$$

$$[\mathbf{RR.3}] \quad \widehat{f\overline{g}} = \overline{(f\overline{g})^{(-1)}} = \overline{\overline{g^{(-1)}}f^{(-1)}} = \overline{\overline{g}f^{(-1)}} = \overline{\overline{g}}\overline{f^{(-1)}} = \overline{f^{(-1)}}\overline{g} = \widehat{f\overline{g}};$$

$$[\mathbf{RR.4}] \quad \widehat{\widehat{f}g} = \overline{(f^{(-1)}g)^{(-1)}} = \overline{g^{(-1)}\overline{f^{(-1)}}^{(-1)}} = \overline{g^{(-1)}\overline{\overline{f^{(-1)}}}} = \overline{g^{(-1)}f^{(-1)}} = \overline{g^{(-1)}f^{(-1)}} = \overline{(fg)^{(-1)}} = \widehat{fg}$$

□

2.2.3 Partial monics, sections and isomorphisms

Partial isomorphisms play a central role in this paper and below we develop some their basic properties.

Definition 2.2.12. A map f in a restriction category \mathbb{X} is said:

- To be a *partial isomorphism* when there is a *partial inverse*, written $f^{(-1)}$ with $f f^{(-1)} = \overline{f}$ and $f^{(-1)} f = \overline{f^{(-1)}}$;
- To be a *partial monic* if $hf = kf \implies h\overline{f} = k\overline{f}$;
- To be a *partial section* if there exists an h such that $fh = \overline{f}$;
- To be a *restriction monic* if it is a section s with a retraction r such that $rs = \overline{rs}$.

Lemma 2.2.13. *In a restriction category:*

- (i) f, g partial monic implies fg is partial monic;
- (ii) f a partial section implies f is partial monic;
- (iii) f, g partial sections implies fg is a partial section;
- (iv) The partial inverse of f , when it exists, is unique;
- (v) If f, g have partial inverses and fg exists, then fg has a partial inverse;
- (vi) A restriction monic s is a partial isomorphism.

Proof.

(i) Suppose $hfg = kfg$. As g is partial monic, $hf\bar{g} = kf\bar{g}$. Therefore:

$$h\overline{f\bar{g}}f = k\overline{f\bar{g}}f \quad [\mathbf{R.4}]$$

$$h\overline{f\bar{g}}\bar{f} = k\overline{f\bar{g}}\bar{f} \quad f \text{ partial monic}$$

$$h\overline{f\bar{g}} = k\overline{f\bar{g}} \quad \text{Lemma 2.2.3, (ii)}$$

(ii) Suppose $gf = kf$. Then, $g\bar{f} = gfh = kfh = k\bar{f}$.

(iii) We have $fh = \bar{f}$ and $gh' = \bar{g}$. Therefore,

$$fgh'h = f\bar{g}h \quad g \text{ partial section}$$

$$= \overline{f\bar{g}}fh \quad [\mathbf{R.4}]$$

$$= \overline{f\bar{g}}\bar{f} \quad f \text{ partial section}$$

$$= \overline{f\bar{f}\bar{g}} \quad [\mathbf{R.2}]$$

$$= \overline{\bar{f}f\bar{g}} \quad [\mathbf{R.3}]$$

$$= \overline{f\bar{g}} \quad [\mathbf{R.1}]$$

(iv) Suppose both $f^{(-1)}$ and f^* are partial inverses of f . Then,

$$\begin{aligned} f^{(-1)} &= \overline{f^{(-1)}}f^{(-1)} = f^{(-1)}ff^{(-1)} = f^{(-1)}\bar{f} = f^{(-1)}ff^* = f^{(-1)}f\bar{f}^*f^* \\ &= \overline{f^{(-1)}\bar{f}^*f^*} = \overline{f^*f^{(-1)}}f^* = f^*f\overline{f^{(-1)}}f^* = f^*ff^{(-1)}ff^* = f^*ff^* = f^* \end{aligned}$$

(v) For $f : A \rightarrow B$, $g : B \rightarrow C$ with partial inverses $f^{(-1)}$ and $g^{(-1)}$ respectively, the partial inverse of fg is $g^{(-1)}f^{(-1)}$. Calculating $fgg^{(-1)}f^{(-1)}$ using all the restriction identities:

$$fgg^{(-1)}f^{(-1)} = f\bar{g}f^{(-1)} = \overline{f\bar{g}}ff^{(-1)} = \overline{f\bar{g}}\bar{f} = \overline{f\bar{f}\bar{g}} = \overline{\bar{f}f\bar{g}} = \overline{f\bar{g}}.$$

The calculation of $g^{(-1)}f^{(-1)}fg = \overline{g^{(-1)}f^{(-1)}}$ is similar.

(vi) The partial inverse of s is $\overline{rs}r$. First, note that $\overline{\overline{rs}r} = \overline{rs}\overline{r} = \overline{r}\overline{rs} = \overline{\overline{r}rs} = \overline{rs}$.

Then, it follows that $(\overline{rs}r)s = rs = \overline{rs} = \overline{\overline{rs}r}$ and $s(\overline{rs}r) = sr\overline{s} = \overline{s}$.

□

A restriction category in which every map is a partial isomorphism is called an *inverse category*.

An interesting property of inverse categories:

Lemma 2.2.14. *In an inverse category, all idempotents are restriction idempotents.*

Proof. Given an idempotent e ,

$$\overline{e} = ee^{(-1)} = eee^{(-1)} = e\overline{e} = \overline{e}e = \overline{e}e = e.$$

□

2.2.4 Split restriction categories

The split restriction category, $K_E(\mathbb{X})$ is defined as:

Objects: (A, e) , where A is an object of \mathbb{X} , $e : A \rightarrow A$ and $e \in E$.

Maps: $f : (A, d) \rightarrow (B, e)$ is given by $f : A \rightarrow B$ in \mathbb{X} , where $f = dfe$.

Identity: The map e for (A, e) .

Composition: inherited from \mathbb{X} .

This is the standard idempotent splitting construction, also known as the Karoubi envelope.

Note that for $f : (A, d) \rightarrow (B, e)$, by definition, in \mathbb{X} we have $f = dfe$, giving

$$df = d(dfe) = ddfe = dfe = f \quad \text{and} \quad fe = (dfe)e = dfee = dfe = f.$$

When \mathbb{X} is a restriction category, there is an immediate candidate for a restriction in $K_E(\mathbb{X})$.

If $f \in K_E(\mathbb{X})$ is e_1fe_2 in \mathbb{X} , then define \overline{f} as given by $e_1\overline{f}$ in \mathbb{X} . Note that for $f : (A, d) \rightarrow (B, e)$, in \mathbb{X} we have:

$$d\overline{f} = \overline{d}fd = \overline{f}d.$$

Proposition 2.2.15. *If \mathbb{X} is a restriction category and E is a set of idempotents, then the restriction as defined above makes $K_E(\mathbb{X})$ a restriction category.*

Proof. The restriction takes $f : (A, e_1) \rightarrow (B, e_2)$ to an endomorphism of (A, e_1) . The restriction is in $K_E(\mathbb{X})$ as

$$e_1(e_1\bar{f})e_1 = e_1\bar{f}e_1 = \overline{e_1\bar{f}}e_1 = \overline{e_1\bar{f}}e_1 = e_1\bar{f}.$$

Checking the 4 restriction axioms:

$$[\mathbf{R.1}] \llbracket \bar{f}f \rrbracket = e_1\bar{f}f = e_1f = \llbracket f \rrbracket$$

$$[\mathbf{R.2}] \llbracket \bar{g}\bar{f} \rrbracket = e_1\bar{g}e_1\bar{f} = e_1e_1\bar{g}\bar{f} = e_1e_1\bar{f}\bar{g} = e_1\bar{f}e_1\bar{g} = \llbracket \bar{f}\bar{g} \rrbracket$$

$$[\mathbf{R.3}] \llbracket \bar{f}\bar{g} \rrbracket \equiv e_1\overline{e_1\bar{f}\bar{g}} = e_1\overline{e_1\bar{f}}\bar{g}e_1 = \overline{e_1\bar{f}}\bar{g}e_1 = e_1\overline{\bar{f}\bar{g}} = e_1\bar{f}\bar{g} = e_1e_1\bar{f}\bar{g} = e_1\bar{f}e_1\bar{g} = \llbracket \bar{f}\bar{g} \rrbracket$$

$$[\mathbf{R.4}] \llbracket \bar{f}\bar{g} \rrbracket = e_1f e_2\bar{g} = \overline{e_1f e_2\bar{g}}e_1f e_2 = \overline{e_1e_1f e_2\bar{g}}e_1f e_2 \\ = e_1\overline{e_1f e_2\bar{g}}e_1f e_2 = e_1\bar{f}\bar{g}e_1f e_2 = \llbracket \bar{f}\bar{g}f \rrbracket$$

□

Given this, provided all identity maps are in E , $K_E(\mathbb{X})$ is a restriction category with \mathbb{X} as a full sub-restriction category, via the embedding defined by taking an object A in \mathbb{X} to the object $(A, 1)$ in $K_E(\mathbb{X})$. Furthermore, the property of being an inverse category is preserved by splitting.

Lemma 2.2.16. *When \mathbb{X} is an inverse category, $K_E(\mathbb{X})$ is an inverse category.*

Proof. The inverse of $f : (A, e_1) \rightarrow (B, e_2)$ in $K_E(\mathbb{X})$ is $e_2f^{(-1)}e_1$ as

$$\llbracket ff^{(-1)} \rrbracket = e_1f e_2e_2f^{(-1)}e_1 = e_1e_1f e_2f^{(-1)}e_1 = e_1ff^{(-1)}e_1 = e_1e_1\bar{f}e_1 = e_1\bar{f} = \llbracket \bar{f} \rrbracket$$

and

$$\llbracket f^{(-1)}f \rrbracket = e_2f^{(-1)}e_1e_1f e_2 = e_2f^{(-1)}e_1f e_2e_2 = e_2f^{(-1)}f e_2 \\ = e_2e_2\overline{f^{(-1)}}e_2 = e_2\overline{f^{(-1)}} = \llbracket \overline{f^{(-1)}} \rrbracket$$

□

Proposition 2.2.17. *In a restriction category \mathbb{X} , with meets, let R be the set of restriction idempotents. Then, $K(\mathbb{X}) \cong K_R(\mathbb{X})$ (where $K(\mathbb{X})$ is the split of \mathbb{X} over all idempotents). Furthermore, $K_R(\mathbb{X})$ has meets.*

Proof. The proof below first shows the equivalence of the two categories, then addresses the claim that $K_R(\mathbb{X})$ has meets.

For equivalence, we require two functors,

$$U : K_R(\mathbb{X}) \rightarrow K(\mathbb{X}) \text{ and } V : K(\mathbb{X}) \rightarrow K_R(\mathbb{X}),$$

with:

$$UV \cong I_{K_R(\mathbb{X})} \tag{2.1}$$

$$VU \cong I_{K(\mathbb{X})}. \tag{2.2}$$

U is the standard inclusion functor. V will take the object (A, e) to $(A, e \cap 1)$ and the map $f : (A, e_1) \rightarrow (B, e_2)$ to $(e_1 \cap 1)f$.

V is a functor as:

Well Defined: If $f : (A, e_1) \rightarrow (B, e_2)$, then $(e_1 \cap 1)f$ is a map in \mathbb{X} from A to B and

$$(e_1 \cap 1)(e_1 \cap 1)f(e_2 \cap 1) = (e_1 \cap 1)(fe_2 \cap f) = (e_1 \cap 1)(f \cap f) = (e_1 \cap 1)f,$$

therefore, $V(f) : V((A, e_1)) \rightarrow V((B, e_2))$.

Identities: $V(e) = (e \cap 1)e = e \cap 1$ by lemma 2.2.6 on page 17.

Composition: $V(f)V(g) = (e_1 \cap 1)f(e_2 \cap 1)g = (e_1 \cap 1)fe_2(e_2 \cap 1)g = (e_1 \cap 1)f(e_2 \cap e_2)g = (e_1 \cap 1)fe_2g = (e_1 \cap 1)fg = V(fg)$.

Recalling from Lemma 2.2.6 on page 17, $(e \cap 1)$ is a restriction idempotent. Using this fact, the commutativity of restriction idempotents and the general idempotent identities from 2.2.6 on page 17, the composite functor UV is the identity on $K_r(\mathbb{X})$ as when e is a restriction idempotent, $e = e(e \cap 1) = (e \cap 1)e = (e \cap 1)$.

For the other direction, note that for a particular idempotent $e : A \rightarrow A$, this gives the maps $e : (A, e) \rightarrow (A, e \cap 1)$ and $e \cap 1 : (A, e \cap 1) \rightarrow (A, e)$, again by [2.2.6 on page 17](#). These maps give the natural isomorphism between I and VU as

$$\begin{array}{ccc} (A, e) & \xrightarrow{e} & (A, e \cap 1) \\ & \searrow e & \downarrow e \cap 1 \\ & & (A, e) \end{array} \quad \text{and} \quad \begin{array}{ccc} (A, e \cap 1) & \xrightarrow{e \cap 1} & (A, e) \\ & \searrow e \cap 1 & \downarrow e \\ & & (A, e \cap 1) \end{array}$$

both commute. Therefore, $UV = I$ and $VU \cong I$, giving an equivalence of the categories.

For the rest of this proof, the bolded functions, e.g., \mathbf{f} are in $K_R(\mathbb{X})$. Italic functions, e.g., f are in \mathbb{X} .

To show that $K_R(\mathbb{X})$ has meets, designate the meet in $K_R(\mathbb{X})$ as \cap_K and define $\mathbf{f} \cap_K \mathbf{g}$ as the map given by the \mathbb{X} map $f \cap g$, where $\mathbf{f}, \mathbf{g} : (A, d) \rightarrow (B, e)$ in $K_R(\mathbb{X})$ and $f, g : A \rightarrow B$ in \mathbb{X} . This is a map in $K_R(\mathbb{X})$ as $d(f \cap g)e = (df \cap dg)e = (f \cap g)e = (fe \cap g) = f \cap g$ where the penultimate equality is by [2.2.6 on page 17](#). By definition $\overline{\mathbf{f} \cap_K \mathbf{g}}$ is $\overline{df \cap dg}$.

It is necessary to show \cap_K satisfies the four meet properties.

- $\mathbf{f} \cap_K \mathbf{g} \leq \mathbf{f}$: We need to show $\overline{\mathbf{f} \cap_K \mathbf{g}} \mathbf{f} = \mathbf{f} \cap_K \mathbf{g}$. Calculating now in \mathbb{X} :

$$\begin{aligned} \overline{df \cap dg} f &= \overline{d(f \cap g)} df \\ &= \overline{df \cap dg} df \\ &= \overline{f \cap g} f \\ &= f \cap g \end{aligned}$$

which is the definition of $\mathbf{f} \cap_K \mathbf{g}$.

- $\mathbf{f} \cap_K \mathbf{g} \leq \mathbf{g}$: Similarly and once again calculating in \mathbb{X} ,

$$\begin{aligned} \overline{df \cap dg} g &= \overline{d(f \cap g)} dg \\ &= \overline{df \cap dg} dg \\ &= \overline{f \cap g} g \\ &= f \cap g \end{aligned}$$

which is the definition of $\mathbf{f} \cap_K \mathbf{g}$.

- $\mathbf{f} \cap_K \mathbf{f} = \mathbf{f}$: From the definition, this is $f \cap f = f$ which is just \mathbf{f} .
- $\mathbf{h}(\mathbf{f} \cap_K \mathbf{g}) = \mathbf{hf} \cap_K \mathbf{hg}$: From the definition, this is given in \mathbb{X} by $h(f \cap g) = hf \cap hg$ which in $K_R(\mathbb{X})$ is $\mathbf{hf} \cap_K \mathbf{hg}$.

□

2.2.5 Partial Map Categories

In [27], it is shown that split restriction categories are equivalent to *partial map categories*.

The main definitions and results related to partial map categories are given below.

Definition 2.2.18. A collection \mathcal{M} of monics is a *stable system of monics* when it includes all isomorphisms, is closed under composition and is pullback stable.

Stable in this definition means that if $m : A \rightarrow B$ is in \mathcal{M} , then for arbitrary b with codomain B , the pullback

$$\begin{array}{ccc} A' & \xrightarrow{a} & A \\ m' \downarrow & & \downarrow m \\ B' & \xrightarrow{b} & B \end{array}$$

exists and $m' \in \mathcal{M}$. A category that has a stable system of monics is referred to as an \mathcal{M} -category.

Lemma 2.2.19. *If $nm \in \mathcal{M}$, a stable system of monics, and m is monic, then $n \in \mathcal{M}$.*

Proof. The commutative square

$$\begin{array}{ccc} A & \xrightarrow{1} & A \\ n \downarrow & & \downarrow nm \\ A' & \xrightarrow{m} & B \end{array}$$

is a pullback.

□

Given a category \mathbb{C} and a stable system of monics, the *partial map category*, $\text{Par}(\mathbb{C}, \mathcal{M})$ is:

Objects: $A \in \mathbb{C}$

Equivalence Classes of Maps: $(m, f) : A \rightarrow B$ with $m : A' \rightarrow A$ is in \mathcal{M} and $f : A' \rightarrow B$

is a map in \mathbb{C} . i.e.,

$$\begin{array}{ccc} & A' & \\ m \swarrow & & \searrow f \\ A & & B \end{array}.$$

Identity: $1_A, 1_A : A \rightarrow A$

Composition: via a pullback, $(m, f)(m', g) = (m''m, f'g)$ where

$$\begin{array}{ccccc} & & A'' & & \\ & m'' \swarrow & & \searrow f' & \\ & A' & \text{(pb)} & B' & \\ m \swarrow & & & & \searrow g \\ A & & f \searrow & B & \swarrow m' \end{array}$$

Restriction: $\overline{(m, f)} = (m, m)$

For the maps, $(m, f) \sim (m', f')$ when there is an isomorphism $\gamma : A'' \rightarrow A'$ such that $\gamma m' = m$ and $\gamma f' = f$.

In [28], it is shown that:

Theorem 2.2.20 (Cockett-Lack). *Every restriction category is a full subcategory of a partial map category.*

2.2.6 Restriction products and Cartesian restriction categories

Restriction categories have analogues of products and terminal objects.

Definition 2.2.21. In a restriction category \mathbb{X} a *restriction product* of two objects X, Y is an object $X \times Y$ equipped with *total* projections $\pi_0 : X \times Y \rightarrow X, \pi_1 : X \times Y \rightarrow Y$ where:

$\forall f : Z \rightarrow X, g : Z \rightarrow Y, \exists$ a unique $\langle f, g \rangle : Z \rightarrow X \times Y$ such that

- $\langle f, g \rangle \pi_0 \leq f$,
- $\langle f, g \rangle \pi_1 \leq g$ and

- $\overline{\langle f, g \rangle} = \bar{f} \bar{g} (= \bar{g} \bar{f})$.

Definition 2.2.22. In a restriction category \mathbb{X} a *restriction terminal object* is an object \top such that $\forall X$, there is a unique total map $!_X : X \rightarrow \top$ and the diagram

$$\begin{array}{ccc} X & \xrightarrow{\bar{f}} & X \xrightarrow{!_X} \top \\ \downarrow f & \nearrow !_Y & \\ Y & & \end{array}$$

commutes. That is, $f !_Y = \bar{f} !_X$. Note this implies that a restriction terminal object is unique up to a unique isomorphism.

Definition 2.2.23. A restriction category \mathbb{X} is *Cartesian* if it has all restriction products and a restriction terminal object.

Definition 2.2.24. An object A in a Cartesian restriction category is *discrete* when the diagonal map,

$$\Delta : A \rightarrow A \times A$$

is a partial isomorphism.

A Cartesian restriction category is *discrete* when every object is discrete.

Theorem 2.2.25. A Cartesian restriction category \mathbb{X} is discrete if and only if it has meets.

Proof. If \mathbb{X} has meets, then

$$\Delta(\pi_0 \cap \pi_1) = \Delta\pi_0 \cap \Delta\pi_1 = 1 \cap 1 = 1$$

and as $\langle \pi_0, \pi_1 \rangle$ is identity,

$$\begin{aligned} \overline{\pi_0 \cap \pi_1} &= \overline{\pi_0 \cap \pi_1} \langle \pi_0, \pi_1 \rangle \\ &= \langle \overline{\pi_0 \cap \pi_1} \pi_0, \overline{\pi_0 \cap \pi_1} \pi_1 \rangle \\ &= \langle \pi_0 \cap \pi_1, \pi_0 \cap \pi_1 \rangle \\ &= (\pi_0 \cap \pi_1) \Delta \end{aligned}$$

and therefore, $\pi_0 \cap \pi_1$ is $\Delta^{(-1)}$.

For the other direction, set $f \cap g = \langle f, g \rangle \Delta^{(-1)}$. By the definition of the restriction product:

$$f \cap g = \langle f, g \rangle \Delta^{(-1)} = \langle f, g \rangle \Delta^{(-1)} \Delta \pi_0 = \langle f, g \rangle \overline{\Delta^{(-1)}} \pi_0 \leq \langle f, g \rangle \pi_0 \leq f$$

Similarly, substituting π_1 for π_0 above, this gives $f \cap g \leq g$. For the left distributive law,

$$h(f \cap g) = h \langle f, g \rangle \Delta^{(-1)} = \langle hf, hg \rangle \Delta^{(-1)} = hf \cap hg$$

and finally an intersection of a map with itself is

$$f \cap f = \langle f, f \rangle \Delta^{(-1)} = (f \Delta) \Delta^{(-1)} = f \overline{\Delta} = f$$

as Δ is total. This shows that \cap as defined above is a meet for the Cartesian restriction category \mathbb{X} .

□

We shall refer to a Cartesian restriction category in which every object is discrete as simply a discrete restriction category.

2.2.7 Graphic Categories

In a Cartesian restriction category, a map $A \xrightarrow{f} B$ is called *graphic* when the maps

$$A \xrightarrow{\langle f, 1 \rangle} B \times A \quad \text{and} \quad A \xrightarrow{\langle \bar{f}, 1 \rangle} A \times A$$

have partial inverses. A Cartesian restriction category is *graphic* when all of its maps are graphic.

Lemma 2.2.26. *In a Cartesian restriction category:*

- (i) *Graphic maps are closed under composition;*
- (ii) *Graphic maps are closed under the restriction;*

(iii) An object is discrete if and only if its identity map is graphic.

Proof.

- (i) To show closure, it is necessary to show that $\langle fg, 1 \rangle$ has a partial inverse. By Lemma 2.2.13 on page 21, the uniqueness of the partial inverse gives

$$(\langle f, 1 \rangle; \langle g, 1 \rangle \times 1)^{(-1)} = \langle g, 1 \rangle^{(-1)} \times 1; \langle f, 1 \rangle^{(-1)}.$$

By the definition of the restriction product, $\overline{\langle fg, 1 \rangle} = \overline{fg}$. Additionally, a straightforward calculation shows that $\overline{\langle f, 1 \rangle; \langle g, 1 \rangle \times 1} = \overline{\langle f \langle g, 1 \rangle, 1 \rangle} = \overline{f; \langle g, 1 \rangle} = \overline{\langle f; g, f \rangle} = \overline{fgf} = \overline{fg}$ where the last equality is by [R.2], [R.3] and finally [R.1].

Consider the diagram

$$\begin{array}{ccccc} A & \xrightarrow{\langle f, 1 \rangle} & B \times A & \xrightarrow{\langle g, 1 \rangle \times 1} & C \times B \times A \\ & \searrow \langle fg, 1 \rangle & & & \uparrow 1 \times \langle f, 1 \rangle \\ & & & & C \times A \end{array}$$

From this:

$$\begin{aligned} \langle fg, 1 \rangle (1 \times \langle f, 1 \rangle) (\langle g, 1 \rangle^{(-1)} \times 1) \langle f, 1 \rangle^{(-1)} &= \langle f, 1 \rangle (\langle g, 1 \rangle \times 1) (\langle g, 1 \rangle^{(-1)} \times 1) \langle f, 1 \rangle^{(-1)} \\ &= \langle f, 1 \rangle (\overline{g \times 1}) \langle f, 1 \rangle^{(-1)} \\ &= \overline{\langle f, 1 \rangle (g \times 1) \langle f, 1 \rangle \langle f, 1 \rangle^{(-1)}} \\ &= \overline{\langle f, 1 \rangle (g \times 1) \langle f, 1 \rangle} \\ &= \overline{\langle f, 1 \rangle \langle f, 1 \rangle (g \times 1)} \\ &= \overline{\langle f, 1 \rangle (g \times 1)} \\ &= \overline{\langle fg, 1 \rangle} (= \overline{fg}) \end{aligned}$$

showing that $1 \times \langle f, 1 \rangle (\langle g, 1 \rangle^{(-1)} \times 1) \langle f, 1 \rangle^{(-1)}$ is a right inverse for $\langle fg, 1 \rangle$.

For the other direction, note that in general $hk^{(-1)} = k^{(-1)}h^{(-1)}$ and that we have $\langle fg, 1 \rangle = \langle f, 1 \rangle (\langle g, 1 \rangle \times 1) (1 \times \langle f, 1 \rangle^{(-1)})$, thus $(1 \times \langle f, 1 \rangle) (\langle g, 1 \rangle^{(-1)} \times 1) \langle f, 1 \rangle^{(-1)}$ will also be a left inverse and $\langle fg, 1 \rangle$ is a restriction isomorphism.

- (ii) This follows from the definition of graphic and that $\overline{\langle f, 1 \rangle} = \bar{f} = \overline{\bar{f}}$.
- (iii) Given a discrete object A , the map 1_A is graphic as $\langle 1_A, 1 \rangle = \Delta$ and therefore $\langle 1, 1 \rangle^{(-1)} = \Delta^{(-1)}$. Conversely, if $\langle 1_A, 1 \rangle$ has an inverse, then $\Delta = \langle 1_A, 1 \rangle$ has that same inverse and therefore the object is discrete.

□

Lemma 2.2.27. *A discrete restriction category is precisely a graphic Cartesian restriction category.*

Proof. The requirement is that $\langle f, 1 \rangle$ (and $\langle \bar{f}, 1 \rangle$) each have partial inverses. For $\langle f, 1 \rangle$, the inverse is $\overline{(1 \times f) \Delta^{(-1)}} \pi_1$.

To show this, calculate the two compositions. First,

$$\langle f, 1 \rangle \overline{(1 \times f) \Delta^{(-1)}} \pi_1 = \overline{\langle f, f \rangle \Delta^{(-1)}} \langle f, 1 \rangle \pi_1 = \overline{f \Delta \Delta^{(-1)}} \langle f, 1 \rangle \pi_1 = \bar{f} \langle f, 1 \rangle \pi_1 = \bar{f}.$$

The other direction is:

$$\begin{aligned} \overline{(1 \times f) \Delta^{(-1)}} \pi_1 \langle f, 1 \rangle &= \langle \overline{(1 \times f) \Delta^{(-1)}} \pi_1 f, \overline{(1 \times f) \Delta^{(-1)}} \pi_1 \rangle \\ &= \langle \overline{(1 \times f) \Delta^{(-1)}} (1 \times f) \pi_1, \overline{(1 \times f) \Delta^{(-1)}} \pi_1 \rangle \\ &= \langle \overline{(1 \times f) \Delta^{(-1)}} \pi_1, \overline{(1 \times f) \Delta^{(-1)}} \pi_1 \rangle \\ &= \langle \overline{(1 \times f) \Delta^{(-1)}} \pi_0, \overline{(1 \times f) \Delta^{(-1)}} \pi_1 \rangle \\ &= \langle \overline{(1 \times f) \Delta^{(-1)}} (1 \times f) \pi_0, \overline{(1 \times f) \Delta^{(-1)}} \pi_1 \rangle \\ &= \langle \overline{(1 \times f) \Delta^{(-1)}} \pi_0, \overline{(1 \times f) \Delta^{(-1)}} \pi_1 \rangle \\ &= \overline{(1 \times f) \Delta^{(-1)}} \langle \pi_0, \pi_1 \rangle \\ &= \overline{(1 \times f) \Delta^{(-1)}} \end{aligned}$$

The one tricky step is to realize

$$\begin{aligned}
\overline{\Delta^{(-1)}}\pi_1 &= \Delta^{(-1)}\Delta\pi_1 \\
&= \Delta^{(-1)} \\
&= \Delta^{(-1)}\Delta\pi_0 \\
&= \overline{\Delta^{(-1)}}\pi_0
\end{aligned}$$

For $\langle \bar{f}, 1 \rangle$, the inverse is $\overline{(1 \times \bar{f})\Delta^{(-1)}}\pi_1$. Similarly to above,

$$\langle \bar{f}, 1 \rangle \overline{1 \times \bar{f}\Delta^{(-1)}}\pi_1 = \overline{\langle \bar{f}, \bar{f} \rangle \Delta^{(-1)}}\langle \bar{f}, 1 \rangle \pi_1 = \overline{\bar{f}\Delta\Delta^{(-1)}}\langle \bar{f}, 1 \rangle \pi_1 = \bar{\bar{f}}\langle \bar{f}, 1 \rangle \pi_1 = \bar{f}.$$

The other direction follows the same pattern as for $\langle f, 1 \rangle$. □

2.3 Turing Categories

Chapter 3

Inverse categories

3.1 Inverse products

Our goal is now to add “products”, to an inverse category. Because an inverse category that has a restriction product is a restriction preorder, what is meant by “product” must be specialized for the inverse setting. These we call *inverse products*, which are defined in sub-section [sub-section 3.1.2 on page 36](#) below.

Inverse products are given by a tensor product which supports a diagonal, but lack projections. The diagonal map is required to give a natural Frobenius structure to each object.

3.1.1 Inverse categories with restriction products

We start by showing than an inverse category with restriction products is a restriction preorder. Thus simply using restriction products provides a notion which is too narrow.

Definition 3.1.1. Two parallel maps $f, g : A \rightarrow B$ in a restriction category are *compatible*, written as $f \smile g$, when $\overline{f}g = \overline{g}f$.

Definition 3.1.2. A restriction category \mathbb{X} is a *restriction preorder* when all parallel pairs of maps are compatible.

Lemma 3.1.3. *Given an inverse category \mathbb{X} , if it has restriction products, it is a restriction preorder. That is,*

$$A \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} B \implies f \smile g.$$

Proof. Notice,

$$\begin{aligned}
\pi_1^{(-1)} &= \Delta \pi_1 \pi_1^{(-1)} \\
&= \Delta \overline{\pi_1} \\
&= \Delta.
\end{aligned}$$

This gives $\overline{\pi_1^{(-1)}} = 1$ and therefore π_1 (and similarly, π_0) is an isomorphism.

Starting with the product map $\langle f, g \rangle$,

$$\begin{aligned}
&\overline{\langle f, g \rangle} = \langle f, g \rangle \\
&\overline{\langle f, g \rangle \pi_1 \pi_1^{(-1)}} = \overline{\langle f, g \rangle \pi_0 \pi_0^{(-1)}} \\
&\overline{\overline{f} g \pi_1^{(-1)}} = \overline{\overline{g} f \pi_0^{(-1)}} \\
&\overline{\overline{f} g \Delta} = \overline{\overline{g} f \Delta} \\
&\overline{\overline{f} g} = \overline{\overline{g} f}
\end{aligned}$$

which shows that f and g are compatible. □

Corollary 3.1.4. *\mathbb{X} is an Cartesian inverse category if and only if $Total(K_r(\mathbb{X}))$ is a meet preorder.*

Proof. $Total(\mathbb{X})$, the subcategory of total maps on \mathbb{X} , has products and therefore every pair of parallel maps is compatible. However, total compatible maps are simply equal, therefore there is at most one map between any two objects. Hence, it is a preorder with the meet being the product.

Similarly, from [27] and [29], $Total(K_r(\mathbb{X}))$ is an inverse category and has products and is therefore also a meet preorder.

When $Total(K_r(\mathbb{X}))$ is a meet preorder, define the product as the meet of the maps and the terminal object as the supremum of all maps. □

Corollary 3.1.5. *Every Cartesian inverse category is a full subcategory of a partial map category of a meet semi-lattice.*

3.1.2 Inverse products

An *inverse product* on a restriction category \mathbb{X} is given by a tensor \otimes together with a natural “Frobenius” diagonal map, Δ . The data for the tensor is:

$$- \otimes - : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{X} \quad (\text{a restriction functor})$$

$$1 : \mathbf{1} \rightarrow \mathbb{X}$$

$$u_{\otimes}^l : 1 \otimes A \rightarrow A$$

$$u_{\otimes}^r : A \otimes 1 \rightarrow A$$

$$a_{\otimes} : (A \otimes B) \otimes C \rightarrow A \otimes (B \otimes C)$$

$$c_{\otimes} : A \otimes B \rightarrow B \otimes A$$

where $u_{\otimes}^l, u_{\otimes}^r, a_{\otimes}, c_{\otimes}$ are all natural isomorphisms and the standard symmetric monoidal equations and coherence diagrams hold (see, e.g., [22]). Note that as all the coherence maps are isomorphisms, they are total. Additionally, we define the map $ex_{\otimes} : (A \otimes B) \otimes (C \otimes D) \rightarrow (A \otimes C) \otimes (B \otimes D)$

$$ex_{\otimes} = a_{\otimes}(1 \otimes a_{\otimes}^{(-1)})(1 \otimes (c_{\otimes} \otimes 1))(1 \otimes a_{\otimes})a_{\otimes}^{(-1)}.$$

The diagonal map $\Delta_A : A \rightarrow A \otimes A$ must be total and must satisfy the following:

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

Co-commutative

$$\begin{array}{ccc}
A & \xrightarrow{\Delta} & A \otimes A \\
\Delta \downarrow & & \downarrow 1 \otimes \Delta \\
A \otimes A & \xrightarrow{\Delta \otimes 1} & (A \otimes A) \otimes A \\
& \searrow \Delta \otimes 1 & \nearrow a_{\otimes} \\
& (A \otimes A) \otimes A &
\end{array}$$

Co-associative

$$\begin{array}{ccc}
A \otimes B & \xrightarrow{\Delta \otimes \Delta} & (A \otimes A) \otimes (B \otimes B) \\
\Delta \downarrow & & \downarrow ex_{\otimes} \\
(A \otimes B) \otimes (A \otimes B) & \xlongequal{\quad\quad\quad} & (A \otimes B) \otimes (A \otimes B)
\end{array}$$

Exchange

$$\begin{array}{ccccc}
A \otimes A & \xrightarrow{(\Delta \otimes 1)a_{\otimes}} & A \otimes (A \otimes A) & & \\
\downarrow (1 \otimes \Delta)a_{\otimes}^{(-1)} & \searrow \Delta^{(-1)} & \downarrow 1 \otimes \Delta^{(-1)} & & \\
(A \otimes A) \otimes A & \xrightarrow{\Delta^{(-1)} \otimes 1} & A \otimes A & & \\
& \nearrow \Delta & & &
\end{array}$$

Frobenius

Thus, Δ is a co-commutative, coassociative map which together with $\Delta^{(-1)}$ forms a Frobenius algebra.

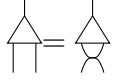
Remark 3.1.6. Note also, co-commutativity implies that $c_{\otimes} \Delta^{(-1)} = \Delta^{(-1)}$. One can see this as:

$$\begin{aligned}
\Delta(c_{\otimes} \Delta^{(-1)}) &= (\Delta c_{\otimes}) \Delta^{(-1)} = \Delta \Delta^{(-1)} = \overline{\Delta} \text{ and} \\
(c_{\otimes} \Delta^{(-1)}) \Delta &= (c_{\otimes} \Delta^{(-1)}) (\Delta c_{\otimes}) = \overline{c_{\otimes} \Delta^{(-1)}}.
\end{aligned}$$

But this means that both $\Delta^{(-1)}$ and $c_{\otimes}\Delta^{(-1)}$ are partial inverses for Δ and are therefore equal.

Similarly, one can show that $(\Delta^{(-1)} \otimes 1)\Delta^{(-1)} = a_{\otimes}(\Delta^{(-1)} \otimes 1)\Delta^{(-1)}$.

Diagrammatic language



Inverse products are extra structure on an inverse category, rather than a property. A concrete category showing this is given in the following example.

Example 3.1.7 (Showing that inverse product is additional structure.).

Any discrete category (i.e., a category with only the identity arrows) is a trivial inverse category. To create an inverse product on the category, add a commutative, associative, idempotent multiplication, with a unit, on the objects.

Label the four objects of \mathbb{D} as a, b, c and d . Then, define two different inverse product tensors by:

\otimes	a	b	c	d
a	a	a	a	a
b	a	b	b	b
c	a	b	c	c
d	a	b	c	d

\otimes	a	b	c	d
a	a	a	a	a
b	a	b	a	b
c	a	a	c	c
d	a	b	c	d

The fact that these operations are idempotent(commutative and associative) implies there is a trivial Frobenius structure.

3.1.3 Discrete inverse categories

An inverse category with inverse products is a *discrete inverse category*. This paper will now present some properties of discrete inverse categories. These properties will be used later when describing a functor that lifts the inverse category to a Cartesian restriction category.

Lemma 3.1.8. *In a discrete inverse category \mathbb{X} with the tensor \otimes and Δ defined as above, where $e = \bar{e}$ is a restriction idempotent and f, g, h are arrows in \mathbb{X} , the following are true:*

- (i) $e = \Delta(e \otimes 1)\Delta^{(-1)}$.
- (ii) $e\Delta(f \otimes g) = \Delta(e f \otimes g)$ (and $= \Delta(f \otimes e g)$ and $= \Delta(e f \otimes e g)$.)
- (iii) $(f \otimes g e)\Delta^{(-1)} = (f \otimes g)\Delta^{(-1)}e$ (and $= (f e \otimes g)\Delta^{(-1)}$ and $= (f e \otimes g e)\Delta^{(-1)}$.)
- (iv) $\overline{\Delta(f \otimes g)\Delta^{(-1)}} = \Delta(1 \otimes g f^{(-1)})\Delta^{(-1)}$.
- (v) If $\Delta(h \otimes g)\Delta^{(-1)} = \overline{\Delta(h \otimes g)\Delta^{(-1)}}$ then $(\Delta(h \otimes g)\Delta^{(-1)})h = \Delta(h \otimes g)\Delta^{(-1)}$.
- (vi) $\Delta(f \otimes 1) = \Delta(g \otimes 1) \implies f = g$.
- (vii) $(f \otimes 1) = (g \otimes 1) \implies f = g$.

Proof.

(i) This is shown by proving both sides equal $\Delta(e \otimes 1)\Delta^{(-1)}\Delta(e \otimes 1)\Delta^{(-1)}$.

$$\begin{aligned}
\Delta(e \otimes 1)\Delta^{(-1)}\Delta(e \otimes 1)\Delta^{(-1)} &= \Delta(e \otimes 1)\Delta^{(-1)}\Delta(1 \otimes e)\Delta^{(-1)} && \text{cocommutativity} \\
&= \Delta(e\Delta \otimes 1)(1 \otimes \Delta^{(-1)}e)\Delta^{(-1)} && \text{Frobenius} \\
&= \Delta(\Delta \otimes 1)(e \otimes e \otimes 1)(1 \otimes \Delta^{(-1)}e)\Delta^{(-1)} && \Delta \text{ natural} \\
&= \Delta(\Delta \otimes 1)(e \otimes e \otimes 1)(1 \otimes e \otimes e)(1 \otimes \Delta^{(-1)})\Delta^{(-1)} && \Delta^{(-1)} \text{ natural} \\
&= \Delta(\Delta \otimes 1)(e \otimes e \otimes e)(1 \otimes \Delta^{(-1)})\Delta^{(-1)} && e \text{ idempotent} \\
&= \Delta(\Delta \otimes 1)(e \otimes \Delta^{(-1)}e)\Delta^{(-1)} && \Delta^{(-1)} \text{ natural} \\
&= \Delta(\Delta \otimes 1)(1 \otimes \Delta^{(-1)})\Delta^{(-1)}e && \Delta^{(-1)} \text{ natural} \\
&= \Delta\Delta^{(-1)}\Delta\Delta^{(-1)}e && \text{Frobenius} \\
&= e && \Delta \text{ total.}
\end{aligned}$$

At the same time,

$$\begin{aligned}
\Delta(e \otimes 1)\Delta^{(-1)}\Delta(e \otimes 1)\Delta^{(-1)} &= \Delta(e\Delta \otimes 1)(e \otimes \Delta^{(-1)}1)\Delta^{(-1)} && \text{Frobenius} \\
&= \Delta(\Delta \otimes 1)(e \otimes e \otimes 1)(e \otimes \Delta^{(-1)})\Delta^{(-1)} && \Delta_{\text{natural}} \\
&= \Delta(\Delta \otimes 1)(e \otimes e \otimes 1)(e \otimes 1 \otimes 1)(1 \otimes \Delta^{(-1)})\Delta^{(-1)} && \Delta^{(-1)} \text{ natural} \\
&= \Delta(\Delta \otimes 1)(e \otimes e \otimes 1)(1 \otimes \Delta^{(-1)})\Delta^{(-1)} && e \text{ idempotent} \\
&= \Delta(e\Delta \otimes 1)(1 \otimes \Delta^{(-1)})\Delta^{(-1)} && \Delta_{\text{natural}} \\
&= \Delta(e \otimes 1)\Delta^{(-1)}\Delta\Delta^{(-1)} && \text{Frobenius} \\
&= \Delta(e \otimes 1)\Delta^{(-1)} && \Delta \text{ total}
\end{aligned}$$

which gives $e = \Delta(e \otimes 1)\Delta^{(-1)}$.

(ii) This equality starts by using the previous equality:

$$\begin{aligned}
e\Delta(f \otimes g) &= \Delta(e \otimes 1)\Delta^{(-1)}\Delta(f \otimes g) && \text{by part (i)} \\
&= \Delta(e \otimes 1)\overline{\Delta^{(-1)}}(f \otimes g) \\
&= \Delta\overline{\Delta^{(-1)}}(e \otimes 1)(f \otimes g) && [\mathbf{R.2}] \text{ as } e \otimes 1 \text{ is a restriction idempotent} \\
&= \Delta(e f \otimes g) && (f f^{(-1)} = f).
\end{aligned}$$

The second and third equalities follow by cocommutativity, naturality of Δ and e being a restriction idempotent.

(iii) As in ((ii) on the preceding page), details are only given for the first equality.

$$\begin{aligned}
(f \otimes g)\Delta^{(-1)}e &= (f \otimes g)\Delta^{(-1)}\Delta(1 \otimes e)\Delta^{(-1)} && \text{part (i)} \\
&= (f \otimes g)\overline{\Delta^{(-1)}}(1 \otimes e)\Delta^{(-1)} \\
&= (f \otimes g)(1 \otimes e)\overline{\Delta^{(-1)}}\Delta^{(-1)} && [\mathbf{R.2}] \\
&= (f \otimes g e)\Delta^{(-1)} && [\mathbf{R.1}]
\end{aligned}$$

The other equalities follow from co-commutativity, naturality of Δ and e being a restriction idempotent.

(iv) Here, we start by using the fact all maps have a partial inverse:

$$\begin{aligned}
& \overline{\Delta(f \otimes g) \Delta^{(-1)}} \\
&= \Delta(f \otimes g) \Delta^{(-1)} \Delta(f^{(-1)} \otimes g^{(-1)}) \Delta^{(-1)} \\
&= \Delta(g \otimes f) \Delta^{(-1)} \Delta(g^{(-1)} \otimes f^{(-1)}) \Delta^{(-1)} && \text{co-commutative} \\
&= \Delta(g \Delta \otimes f) (g^{(-1)} \otimes \Delta^{(-1)} f^{(-1)}) \Delta^{(-1)} && \text{Frobenius} \\
&= \Delta(\Delta \otimes 1) (g \otimes g \otimes f) (g^{(-1)} \otimes \Delta^{(-1)} f^{(-1)}) \Delta^{(-1)} && \Delta \text{ natural} \\
&= \Delta(\Delta \otimes 1) (g \otimes g \otimes f) (g^{(-1)} \otimes f^{(-1)} \otimes f^{(-1)}) (1 \otimes \Delta^{(-1)}) \Delta^{(-1)} && \Delta^{(-1)} \text{ natural} \\
&= \Delta(\Delta \otimes 1) (\bar{g} \otimes g f^{(-1)} \otimes \bar{f}) (1 \otimes \Delta^{(-1)}) \Delta^{(-1)} && \text{combine maps} \\
&= \Delta(\Delta \otimes 1) (\bar{g} \otimes \bar{g} g f^{(-1)} \bar{f} \otimes \bar{f}) (1 \otimes \Delta^{(-1)}) \Delta^{(-1)} && \text{restriction axioms} \\
&= \Delta(\bar{g} \Delta \otimes 1) (1 \otimes g f^{(-1)} \bar{f} \otimes \bar{f}) (1 \otimes \Delta^{(-1)}) \Delta^{(-1)} && \Delta \text{ natural} \\
&= \Delta(\bar{g} \Delta \otimes 1) (1 \otimes g f^{(-1)} \otimes 1) (1 \otimes \Delta^{(-1)} \bar{f}) \Delta^{(-1)} && \Delta^{(-1)} \text{ natural} \\
&= \Delta(\Delta \otimes 1) (1 \otimes \bar{g} g f^{(-1)} \otimes 1) (1 \otimes \Delta^{(-1)} \bar{f}) \Delta^{(-1)} && \text{This lemma((ii))} \\
&= \Delta(\Delta \otimes 1) (1 \otimes \bar{g} g f^{(-1)} \bar{f} \otimes 1) (1 \otimes \Delta^{(-1)}) \Delta^{(-1)} && \text{This lemma((iii))} \\
&= \Delta(\Delta \otimes 1) (1 \otimes g f^{(-1)} \otimes 1) (1 \otimes \Delta^{(-1)}) \Delta^{(-1)} && \text{restriction axioms} \\
&= \Delta c_{A,A} (\Delta \otimes 1) (1 \otimes g f^{(-1)} \otimes 1) (1 \otimes \Delta^{(-1)}) \Delta^{(-1)} && \text{co-commutative} \\
&= \Delta(1 \otimes \Delta) c_{A,A \otimes A} (1 \otimes g f^{(-1)} \otimes 1) (1 \otimes \Delta^{(-1)}) \Delta^{(-1)} && c_{\otimes} \text{natural} \\
&= \Delta(1 \otimes \Delta) (1 \otimes 1 \otimes g f^{(-1)}) c_{A,A \otimes A} (1 \otimes \Delta^{(-1)}) \Delta^{(-1)} && c_{\otimes} \text{natural} \\
&= \Delta(1 \otimes \Delta) (1 \otimes 1 \otimes g f^{(-1)}) (\Delta^{(-1)} \otimes 1) c_{A,A} \Delta^{(-1)} && c_{\otimes} \text{natural} \\
&= \Delta(1 \otimes \Delta) (1 \otimes 1 \otimes g f^{(-1)}) (\Delta^{(-1)} \otimes 1) \Delta^{(-1)} && c_{\otimes} \text{co-commutative} \\
&= \Delta \Delta^{(-1)} \Delta (1 \otimes g f^{(-1)}) \Delta^{(-1)} && \text{Frobenius} \\
&= \Delta(1 \otimes g f^{(-1)}) \Delta^{(-1)} && \Delta \text{ total}
\end{aligned}$$

Note the pattern in the last few lines of using the co-commutativity of Δ ,

the naturality of the commutativity isomorphism and finishing with the co-commutativity of $\Delta^{(-1)}$. In future proofs, these steps will be combined to a single line and referred to as commutativity.

(v) Beginning with the assumption,

$$\begin{aligned}
(\Delta(h \otimes g)\Delta^{(-1)})h &= \overline{\Delta(h \otimes g)\Delta^{(-1)}}h \\
&= \Delta(1 \otimes gh^{(-1)})\Delta^{(-1)}h && \text{This lemma((iv))} \\
&= \Delta(1 \otimes gh^{(-1)})\Delta^{(-1)}\Delta(h \otimes h)\Delta^{(-1)} && \Delta \text{ total and natural} \\
&= \Delta(1 \otimes gh^{(-1)})(\Delta \otimes 1)(1 \otimes \Delta^{(-1)})(h \otimes h)\Delta^{(-1)} && \text{Frobenius} \\
&= \Delta(\Delta \otimes 1)(1 \otimes 1 \otimes gh^{(-1)})(1 \otimes \Delta^{(-1)})(h \otimes h)\Delta^{(-1)} && \Delta \text{ natural} \\
&= \Delta(\Delta \otimes 1)(1 \otimes 1 \otimes gh^{(-1)})(h \otimes h \otimes h)(1 \otimes \Delta^{(-1)})\Delta^{(-1)} && \Delta^{(-1)} \text{ natural} \\
&= \Delta(\Delta \otimes 1)(h \otimes h \otimes gh^{(-1)}h)(1 \otimes \Delta^{(-1)})\Delta^{(-1)} && \text{combine terms} \\
&= \Delta(h \otimes g\overline{h^{(-1)}})(\Delta \otimes 1)(1 \otimes \Delta^{(-1)})\Delta^{(-1)} && \Delta \text{ natural} \\
&= \Delta(h \otimes g\overline{h^{(-1)}})\Delta^{(-1)}\Delta\Delta^{(-1)} && \text{Frobenius} \\
&= \Delta(h \otimes g\overline{h^{(-1)}})\Delta^{(-1)} && \Delta \text{ total} \\
&= \Delta(h \otimes g)\Delta^{(-1)}\overline{h^{(-1)}} && \text{part ((ii))} \\
&= \Delta(\overline{hh^{(-1)}} \otimes g)\Delta^{(-1)} && \text{part ((ii))} \\
&= \Delta(h \otimes g)\Delta^{(-1)} && \text{property of inverse.}
\end{aligned}$$

(vi) As Δ is total and natural, we start with:

$$\begin{aligned}
f &= \Delta(f \otimes f)\Delta^{(-1)} \\
&= \Delta(f \otimes 1)(1 \otimes f)\Delta^{(-1)} \\
&= \Delta(g \otimes 1)(1 \otimes f)\Delta^{(-1)} && \text{assumption} \\
&= \Delta(1 \otimes f)(g \otimes 1)\Delta^{(-1)} && \text{Identities commute} \\
&= \Delta(1 \otimes g)(g \otimes 1)\Delta^{(-1)} && \text{assumption, co-commutative} \\
&= \Delta(g \otimes g)\Delta^{(-1)} \\
&= g\Delta\Delta^{(-1)} && \Delta \text{ natural} \\
&= g && \Delta \text{ total.}
\end{aligned}$$

(vii) Immediate from part (vi) on page 39.

□

Proposition 3.1.9. *A discrete inverse category has meets, where $f \cap g = \Delta(f \otimes g)\Delta^{(-1)}$.*

Proof. $f \cap g \leq f$:

$$\begin{aligned}
f \cap g &= \Delta(f \otimes g)\Delta^{(-1)} && \text{Definition of } \cap \\
&= \Delta(\overline{f f^{(-1)}} \otimes g)\Delta^{(-1)} && \text{property of inverse} \\
&= \Delta(f \otimes g \overline{f^{(-1)}})\Delta^{(-1)} && \text{by lemma 3.1.8((iii))} \\
&= \Delta(f \otimes g f^{(-1)} f)\Delta^{(-1)} && \text{definition of inverse} \\
&= \Delta(1 \otimes g f^{(-1)})\Delta^{(-1)} f && \Delta^{(-1)} \text{ natural} \\
&= \overline{f \cap g} f && \text{by lemma 3.1.8((iv))}
\end{aligned}$$

$$f \cap f = f:$$

$$\begin{aligned} f \cap f &= \Delta(f \otimes f) \Delta^{(-1)} \\ &= f \Delta \Delta^{(-1)} && \Delta \text{ natural} \\ &= f && \Delta \text{ total.} \end{aligned}$$

$$h(f \cap g) = hf \cap hg:$$

$$\begin{aligned} h(f \cap g) &= h \Delta(f \otimes g) \Delta^{(-1)} && \text{Definition of } \cap \\ &= \Delta(h \otimes h)(f \otimes g) \Delta^{(-1)} && \Delta \text{ natural} \\ &= \Delta(hf \otimes hg) \Delta^{(-1)} && \text{compose maps} \\ &= hf \cap hg && \text{Definition of } \cap. \end{aligned}$$

□

3.1.4 The inverse subcategory of a discrete restriction category

Given a discrete restriction category, one can pick out the maps which are partial isomorphisms. Using results from the previous sub-section and from sub-section [sub-section 2.2.7 on page 30](#), this section will show that these maps form a restriction subcategory and in fact, form a discrete inverse category.

Lemma 3.1.10. *Given \mathbb{X} is a discrete restriction category, the invertible maps of \mathbb{X} , together with the objects of \mathbb{X} form a sub restriction category which is a discrete inverse category, denoted by $Inv(\mathbb{X})$.*

Proof. As shown in Lemma [2.2.13 on page 21](#), partial isomorphisms are closed under composition. The identity maps are in $Inv(\mathbb{X})$. Trivially, restrictions of partial isomorphisms are also partial isomorphisms.

The product on the discrete restriction category \mathbb{X} becomes the tensor product of the restriction category $Inv(\mathbb{X})$. Table [table 3.1 on the following page](#) shows how each of the

elements of the tensor are defined. Note that the last definition makes explicit use of the fact we are in a discrete restriction category and hence the Δ of \mathbb{X} possesses a partial inverse.

\mathbb{X}	$Inv(\mathbb{X})$	Inverse map
$A \times B$	$A \otimes B$	
\top	1	
$\pi_1: \top \times A \rightarrow A$	$u_{\otimes}^l: 1 \otimes A \rightarrow A$	$\langle !, 1 \rangle$
$\pi_0: A \times \top \rightarrow A$	$u_{\otimes}^r: A \otimes 1 \rightarrow A$	$\langle 1, ! \rangle$
$\langle \pi_0 \pi_0, \langle \pi_0 \pi_1, \pi_1 \rangle \rangle: (A \times B) \times C \rightarrow A \times (B \times C)$	$a_{\otimes}: (A \otimes B) \otimes C \rightarrow A \otimes (B \otimes C)$	$\langle \langle \pi_0, \pi_1 \pi_0 \rangle, \pi_1 \pi_1 \rangle$
$\langle \pi_1, \pi_0 \rangle: A \times B \rightarrow B \times A$	$c_{\otimes}: A \otimes B \rightarrow B \otimes A$	$\langle \pi_1, \pi_0 \rangle$
$\Delta_{\mathbb{X}}: A \rightarrow A \times A$	$\Delta: A \rightarrow A \otimes A$	$\Delta_{\mathbb{X}}^{(-1)}$

Table 3.1: Structural maps for the tensor in $Inv(\mathbb{X})$

The monoid coherence diagrams and Δ being total follow directly from the characteristics of the product in \mathbb{X} . It remains to show co-commutativity, co-associativity and the Frobenius condition.

Co-commutativity requires $\Delta c_{\otimes} = c_{\otimes}$. From the definitions, this means we need

$$\Delta_{\mathbb{X}} \langle \pi_1, \pi_0 \rangle = \Delta_{\mathbb{X}}.$$

Once again, this follows immediately from the definition of restriction product.

Co-associativity requires $\Delta(1 \otimes \Delta) = \Delta(\Delta \otimes 1)a_{\otimes}$. Expressing this in \mathbb{X} , we require

$$\Delta_{\mathbb{X}}(1 \times \Delta_{\mathbb{X}}) = \Delta_{\mathbb{X}}(\Delta_{\mathbb{X}} \times 1)(\langle \pi_0 \pi_0, \langle \pi_0 \pi_1, \pi_1 \rangle \rangle).$$

Again each is equal based on the properties of the restriction product.

The Frobenius requirement is two-fold:

$$\Delta^{(-1)} \Delta = (\Delta \otimes 1)a_{\otimes}(1 \otimes \Delta^{(-1)}) \quad (3.1)$$

$$\Delta^{(-1)} \Delta = (1 \otimes \Delta)a_{\otimes}^{(-1)}(\Delta^{(-1)} \otimes 1), \quad (3.2)$$

but in \mathbb{X} , this becomes:

$$\Delta_{\mathbb{X}}^{(-1)} \Delta_{\mathbb{X}} = (\Delta_{\mathbb{X}} \times 1) \langle \pi_0 \pi_0, \langle \pi_0 \pi_1, \pi_1 \rangle \rangle (1 \times \Delta_{\mathbb{X}}^{(-1)}) \quad (3.3)$$

$$\Delta_{\mathbb{X}}^{(-1)} \Delta_{\mathbb{X}} = (1 \times \Delta_{\mathbb{X}}) \langle \langle \pi_0, \pi_1 \pi_0 \rangle, \pi_1 \pi_1 \rangle (\Delta_{\mathbb{X}}^{(-1)} \times 1). \quad (3.4)$$

We will detail the proof of equation [equation \(3.3\) on the previous page](#). Equation [equation \(3.4\) on the preceding page](#) is proved similarly.

To show the equation, note first that $\Delta(1 \times !)$ (and $\Delta(! \times 1)$) is the identity and secondly that maps to a product of objects may be split into a product map — e.g. if $f : A \rightarrow B \times B$, then $f = \langle f(1 \times !), f(! \times 1) \rangle$.

Using this we see that the left hand side of equation [equation \(3.3\) on the previous page](#) computes as follows:

$$\begin{aligned}\Delta_{\mathbb{X}}^{(-1)}\Delta_{\mathbb{X}} &= \langle \Delta_{\mathbb{X}}^{(-1)}\Delta_{\mathbb{X}}(1 \times !), \Delta_{\mathbb{X}}^{(-1)}\Delta_{\mathbb{X}}(! \times 1) \rangle \\ &= \langle \Delta_{\mathbb{X}}^{(-1)}, \Delta_{\mathbb{X}}^{(-1)} \rangle\end{aligned}$$

Similarly, removing the associativity maps, the right hand side of the same equation becomes:

$$\begin{aligned}(\Delta_{\mathbb{X}} \times 1)(1 \times \Delta_{\mathbb{X}}^{(-1)}) &= \langle (\Delta_{\mathbb{X}} \times 1)(1 \times \Delta_{\mathbb{X}}^{(-1)})(1 \times !), (\Delta_{\mathbb{X}} \times 1)(1 \times \Delta_{\mathbb{X}}^{(-1)})(! \times 1) \rangle \\ &= \langle (\Delta_{\mathbb{X}} \times 1)(1 \times \Delta_{\mathbb{X}}^{(-1)})(1 \times !), \Delta_{\mathbb{X}}^{(-1)} \rangle \\ &= \langle (\Delta_{\mathbb{X}} \times 1)(1 \times \Delta_{\mathbb{X}}^{(-1)})(1 \times \Delta_{\mathbb{X}})(1 \times ! \times !), \Delta_{\mathbb{X}}^{(-1)} \rangle \\ &= \langle (\Delta_{\mathbb{X}} \times 1)(1 \times \overline{\Delta_{\mathbb{X}}^{(-1)}})(1 \times ! \times !), \Delta_{\mathbb{X}}^{(-1)} \rangle \\ &= \langle (\Delta_{\mathbb{X}} \times 1)\overline{1 \times \Delta_{\mathbb{X}}^{(-1)}}(1 \times ! \times !), \Delta_{\mathbb{X}}^{(-1)} \rangle \\ &= \langle \overline{(\Delta_{\mathbb{X}} \times 1)(1 \times \Delta_{\mathbb{X}}^{(-1)})}(\Delta_{\mathbb{X}} \times 1)(1 \times ! \times !), \Delta_{\mathbb{X}}^{(-1)} \rangle \\ &= \langle \overline{(\Delta_{\mathbb{X}} \times 1)(1 \times \Delta_{\mathbb{X}}^{(-1)})}(1 \times !), \Delta_{\mathbb{X}}^{(-1)} \rangle \\ &= \langle \overline{(\Delta_{\mathbb{X}} \times 1)(1 \times \Delta_{\mathbb{X}}^{(-1)})}(! \times 1)(1 \times !), \Delta_{\mathbb{X}}^{(-1)} \rangle \\ &= \langle \overline{\Delta_{\mathbb{X}}^{(-1)}}(1 \times !), \Delta_{\mathbb{X}}^{(-1)} \rangle \\ &= \langle \Delta_{\mathbb{X}}^{(-1)}\Delta_{\mathbb{X}}(1 \times !), \Delta_{\mathbb{X}}^{(-1)} \rangle \\ &= \langle \Delta_{\mathbb{X}}^{(-1)}, \Delta_{\mathbb{X}}^{(-1)} \rangle\end{aligned}$$

and therefore we see that the first equation for the Frobenius condition is satisfied. Thus, $Inv(\mathbb{X})$ is a discrete inverse category. □

3.2 Completing a discrete inverse category

The purpose of this section is to prove that the category of discrete inverse categories is equivalent to the the category of discrete restriction categories. In order to prove this, we show how to construct a discrete restriction category, $\widetilde{\mathbb{X}}$, from a discrete inverse category, \mathbb{X} .

3.2.1 The restriction category $\widetilde{\mathbb{X}}$

Definition 3.2.1. When \mathbb{X} is an inverse category, define $\widetilde{\mathbb{X}}$ as:

Objects: objects as in \mathbb{X}

Maps: equivalence classes of maps (the equivalence class is defined below in Definition 3.2.2 on the following page) with the following structure in \mathbb{X} :

$$\frac{A \xrightarrow{(f,C)} B \text{ in } \widetilde{\mathbb{X}}}{A \xrightarrow{f} B \otimes C \text{ in } \mathbb{X}}$$

Identity: by

$$\frac{A \xrightarrow{(u_{\otimes}^r(-1),1)} A}{A \xrightarrow{u_{\otimes}^r(-1)} A \otimes 1}$$

Composition: given by

$$\frac{\frac{A \xrightarrow{(f,B')} B \xrightarrow{(g,C')} C}{A \xrightarrow{f(g \otimes 1)a_{\otimes}} C \otimes (C' \otimes B')}}{A \xrightarrow{(f(g \otimes 1)a_{\otimes}, C' \otimes B')} C}$$

When considering an $\widetilde{\mathbb{X}}$ map $(f, C) : A \rightarrow B$ in \mathbb{X} , we occasionally use the notation $f : A \rightarrow B|_C (\equiv f : A \rightarrow B \otimes C)$.

Equivalence classes of maps in \mathbb{X}

Definition 3.2.2. In a discrete inverse category \mathbb{X} as defined above, the map f is equivalent to f' in \mathbb{X} when $\bar{f} = \bar{f}'$ in \mathbb{X} and the below diagram commutes for some map h :

$$\begin{array}{ccccc}
 & & B \otimes C & \xrightarrow{(\Delta \otimes 1) a_{\otimes}} & B \otimes (B \otimes C) \\
 & \nearrow f & & & \downarrow 1 \otimes h \\
 A & & & & B \otimes (B \otimes C') \\
 & \searrow f' & & \xleftarrow{a_{\otimes}^{(-1)} (\Delta^{(-1)} \otimes 1)} & \\
 & & B \otimes C' & &
 \end{array}$$

Notation 3.2.3. When f is equivalent to g via the mediating map h , this is written as

$$f \stackrel{h}{\simeq} g.$$

Lemma 3.2.4. Definition 3.2.2 gives a symmetric, reflexive equivalence class of maps in \mathbb{X} .

Proof.

Reflexivity: Choose h as the identity map.

Symmetry: Suppose $f \stackrel{h}{\simeq} g$. Then, $\bar{f} = \bar{g}$ and $fk = g$ where

$$k = (\Delta \otimes 1) a_{\otimes} (1 \otimes h) a_{\otimes}^{(-1)} (\Delta^{(-1)} \otimes 1).$$

Applying $k^{(-1)}$, which is

$$(\Delta \otimes 1) a_{\otimes} (1 \otimes h^{(-1)}) a_{\otimes}^{(-1)} (\Delta^{(-1)} \otimes 1),$$

we have

$$gk^{(-1)} = fkk^{(-1)} = f\bar{k} = \bar{f}k f = \bar{g}f = \bar{f}f = f.$$

Thus, $g \stackrel{h^{(-1)}}{\simeq} f$.

Transitivity: Suppose $f \xrightarrow{h} f'$ and $f' \xrightarrow{k} f''$. Then, consider the compositions of the mediating portions of the equivalences:

$$\ell = ((\Delta \otimes 1)a_{\otimes}(1 \otimes h)a_{\otimes}^{(-1)}(\Delta^{(-1)} \otimes 1))((\Delta \otimes 1)a_{\otimes}(1 \otimes k)a_{\otimes}^{(-1)}(\Delta^{(-1)} \otimes 1)).$$

By pasting the diagrams which give the above equivalences, we see that $f\ell = f''$. However, it is not in the form of a mediating map as presented.

The claim is that ℓ is the actual mediating map for f and f'' . That is, that we have $f(\Delta \otimes 1)a_{\otimes}(1 \otimes \ell)a_{\otimes}^{(-1)}(\Delta^{(-1)} \otimes 1) = f''$. In the interest of some brevity, this is shown below with the associativity maps elided from the equations.

We need to show that $(\Delta \otimes 1)(1 \otimes \ell)(\Delta^{(-1)} \otimes 1) = \ell$.

$$\begin{aligned} & (\Delta \otimes 1)(1 \otimes \ell)(\Delta^{(-1)} \otimes 1) \\ &= (\Delta \otimes 1)(1 \otimes \Delta \otimes 1)(1 \otimes 1 \otimes h)(1 \otimes \Delta^{(-1)} \otimes 1) \\ & \quad (1 \otimes \Delta \otimes 1)(1 \otimes 1 \otimes k)(1 \otimes \Delta^{(-1)} \otimes 1)(\Delta^{(-1)} \otimes 1) \\ &= (\Delta \otimes 1)(\Delta \otimes 1 \otimes 1)(1 \otimes 1 \otimes h)(1 \otimes \Delta^{(-1)} \otimes 1) \\ & \quad (1 \otimes \Delta \otimes 1)(1 \otimes 1 \otimes k)(\Delta^{(-1)} \otimes 1 \otimes 1)(\Delta^{(-1)} \otimes 1) \quad \text{co-associativity} \\ &= (\Delta \otimes 1)(1 \otimes h)(\Delta \otimes 1 \otimes 1)(1 \otimes \Delta^{(-1)} \otimes 1) \\ & \quad (1 \otimes \Delta \otimes 1)(\Delta^{(-1)} \otimes 1 \otimes 1)(1 \otimes k)(\Delta^{(-1)} \otimes 1) \quad \text{Naturality} \\ &= (\Delta \otimes 1)(1 \otimes h)(\Delta^{(-1)} \otimes 1)(\Delta \otimes 1) \\ & \quad (\Delta^{(-1)} \otimes 1)(\Delta \otimes 1)(1 \otimes k)(\Delta^{(-1)} \otimes 1) \quad \text{Frobenius} \\ &= (\Delta \otimes 1)(1 \otimes h)(\Delta^{(-1)} \otimes 1)(\Delta \otimes 1)(1 \otimes k)(\Delta^{(-1)} \otimes 1) \quad \Delta \text{ Total} \\ &= \ell \end{aligned}$$

and therefore $f \xrightarrow{\ell} f''$. □

Corollary 3.2.5. *If $\bar{f} = \bar{g}$ in \mathbb{X} , a discrete inverse category, and the diagram*

$$\begin{array}{ccc} & & B \otimes C \\ & \nearrow f & \downarrow 1 \otimes h \\ A & & B \otimes C' \\ & \searrow g & \end{array}$$

commutes for some h , then there is a h' such that $f \stackrel{h'}{\simeq} g$.

Proof. Consider

$$\begin{aligned} (\Delta \otimes 1) a_{\otimes} (1 \otimes (1 \otimes h)) a_{\otimes}^{(-1)} (\Delta^{(-1)} \otimes 1) & \\ = (\Delta \otimes 1) ((1 \otimes 1) \otimes h) a_{\otimes} a_{\otimes}^{(-1)} (\Delta^{(-1)} \otimes 1) & \text{Naturality} \\ = (\Delta \otimes 1) ((1 \otimes 1) \otimes h) (\Delta^{(-1)} \otimes 1) & \text{Isomorphism Inverse} \\ = (\Delta(1 \otimes 1) \Delta^{(-1)}) \otimes h & \text{Naturality of } \otimes \\ = (1 \otimes h) & \Delta \Delta^{(-1)} = 1 \end{aligned}$$

and therefore we can set $h' = 1 \otimes h$. □

Lemma 3.2.6. $\widetilde{\mathbb{X}}$ as defined above is a category.

Proof. The maps are well defined, as shown in lemma 3.2.4 on page 48. The existence of the identity map is due to the tensor \otimes being defined on \mathbb{X} , an inverse category, hence $u_{\otimes}^{r(-1)}$ is defined.

It remains to show the composition is associative and that $(u_{\otimes}^{r(-1)}, 1)$ acts as an identity in $\widetilde{\mathbb{X}}$.

Associativity: Consider

$$A \xrightarrow{(f, B')} B \xrightarrow{(g, C')} C \xrightarrow{(h, D')} D.$$

To show the associativity of this in $\widetilde{\mathbb{X}}$, we need to show in \mathbb{X} that

$$\overline{(f(g \otimes 1) a_{\otimes})(h \otimes 1) a_{\otimes}} = \overline{f(((g(h \otimes 1) a_{\otimes}) \otimes 1) a_{\otimes})}$$

From this, we can conclude

$$(f(g \otimes 1)a_{\otimes})(h \otimes 1)a_{\otimes} \stackrel{1 \otimes a_{\otimes}^{(-1)}}{\simeq} f(((g(h \otimes 1)a_{\otimes}) \otimes 1)a_{\otimes})$$

which gives us that composition in $\widetilde{\mathbb{X}}$ is associative.

Identity: This requires:

$$(f, C)(u_{\otimes}^{r(-1)}, 1) = (f, C) = (u_{\otimes}^{r(-1)}, 1)(f, C)$$

for all maps $A \xrightarrow{(f, C)} B$ in $\widetilde{\mathbb{X}}$.

First, we see $\overline{f(u_{\otimes}^{r(-1)} \otimes 1)a_{\otimes}} = \overline{f}$ by Lemma 2.2.3 on page 15. Then, calculating in \mathbb{X} , we have a mediating map of $1 \otimes u_{\otimes}^l$ as shown below.

$$\begin{array}{ccccccc}
 A & \xrightarrow{f} & B \otimes C & \xrightarrow{u_{\otimes}^{r(-1)} \otimes 1} & (B \otimes 1) \otimes C & \xrightarrow{a_{\otimes}} & B \otimes (1 \otimes C) \\
 & & & \searrow & \searrow & & \downarrow 1 \otimes u_{\otimes}^l \\
 & & & & & & B \otimes C \\
 & \searrow f & & & & & \\
 & & & & & &
 \end{array}$$

$\xrightarrow{1 \otimes u_{\otimes}^l} B \otimes C$

Next, $\overline{u_{\otimes}^{r(-1)}(f \otimes 1)a_{\otimes}} = \overline{f}$ by the naturality of $u_{\otimes}^{r(-1)}$ and Lemma 2.2.3 on page 15. The diagram below

$$\begin{array}{ccccccc}
 A & \xrightarrow{u_{\otimes}^{r(-1)}} & A \otimes 1 & \xrightarrow{f \otimes 1} & (B \otimes C) \otimes 1 & \xrightarrow{a_{\otimes}} & B \otimes (C \otimes 1) \\
 & \searrow f & & \nearrow u_{\otimes}^{r(-1)} & \nearrow 1 \otimes u_{\otimes}^{r(-1)} & & \downarrow 1 \otimes u_{\otimes}^r \\
 & & B \otimes C & & & & B \otimes C \\
 & \searrow f & & & & & \\
 A & \xrightarrow{f} & B \otimes C & & & &
 \end{array}$$

shows our mediating map is $1 \otimes u_{\otimes}^r$. □

Defining the restriction on $\widetilde{\mathbb{X}}$

Define the restriction in $\widetilde{\mathbb{X}}$ as follows:

$$\frac{\frac{A \xrightarrow{(f,C)} B}{A \xrightarrow{(f,C)} A}}{A \xrightarrow{\bar{f}u_{\otimes}^{r(-1)}} A \otimes 1 \text{ in } \mathbb{X}}$$

Lemma 3.2.7. *The category $\widetilde{\mathbb{X}}$ with restriction defined as above is a restriction category.*

Proof. Given the above definition, the four restriction axioms must now be checked. (Diagrams are in \mathbb{X}).

[R.1] ($\bar{f}f = f$) Calculating the restriction of the left hand side in \mathbb{X} , we have:

$$\begin{aligned} \overline{\bar{f}u_{\otimes}^{r(-1)}(f \otimes 1)a_{\otimes}} &= \overline{\bar{f}u_{\otimes}^{r(-1)}(f \otimes 1)} && a_{\otimes} \text{ iso, Lemma 2.2.3} \\ &= \overline{\bar{f}fu_{\otimes}^{r(-1)}} && u_{\otimes}^{r(-1)} \text{ natural} \\ &= \overline{fu_{\otimes}^{r(-1)}} && [\text{R.1}] \text{ in } \mathbb{X} \\ &= \bar{f} && u_{\otimes}^{r(-1)} \text{ iso, Lemma 2.2.3.} \end{aligned}$$

Then, the following diagram

$$\begin{array}{ccccccc} A & \xrightarrow{\bar{f}u_{\otimes}^{r(-1)}} & A \otimes 1 & \xrightarrow{f \otimes 1} & (A \otimes B) \otimes 1 & \xrightarrow{a_{\otimes}} & A \otimes (B \otimes 1) \\ & \searrow \bar{f}f & & \nearrow u_{\otimes}^{r(-1)} & \searrow u_{\otimes}^r & & \downarrow 1 \otimes u_{\otimes}^r \\ & & & A \otimes B & & & A \otimes B \\ & \searrow f & & & & & \downarrow \\ & & & & & & A \otimes B \end{array}$$

shows $\bar{f}u_{\otimes}^{r(-1)}(f \otimes 1)a_{\otimes} \xrightarrow{1 \otimes u_{\otimes}^r} f$ in \mathbb{X} and therefore $\bar{f}f = f$ in $\widetilde{\mathbb{X}}$.

[R.2] ($\bar{g}\bar{f} = \bar{f}\bar{g}$) The restriction of the left hand side equals the restriction of the right hand side as seen below:

$$\begin{aligned} \overline{\bar{f}u_{\otimes}^{r(-1)}((\bar{g}u_{\otimes}^{r(-1)}) \otimes 1)a_{\otimes}} &= \overline{\bar{f}(\bar{g}u_{\otimes}^{r(-1)})u_{\otimes}^{r(-1)}a_{\otimes}} && u_{\otimes}^{r(-1)} \text{ natural} \\ &= \overline{\bar{g}\bar{f}u_{\otimes}^{r(-1)}u_{\otimes}^{r(-1)}a_{\otimes}} && [\text{R.2}] \text{ in } \mathbb{X} \\ &= \overline{\bar{g}u_{\otimes}^{r(-1)}((\bar{f}u_{\otimes}^{r(-1)}) \otimes 1)a_{\otimes}} && u_{\otimes}^{r(-1)} \text{ natural.} \end{aligned}$$

The below diagram commutes by the naturality of u_{\otimes}^r and the tensor coherence,

$$\begin{array}{ccccc}
A & \xrightarrow{\bar{g}u_{\otimes}^{r(-1)}} & A \otimes 1 & \xrightarrow{(\bar{f}u_{\otimes}^{r(-1)}) \otimes 1} & (A \otimes 1) \otimes 1 \xrightarrow{a_{\otimes}} A \otimes (1 \otimes 1) \\
\downarrow \bar{f}u_{\otimes}^{r(-1)} & \searrow \bar{g}\bar{f} & & \nearrow u_{\otimes}^r u_{\otimes}^r & \downarrow 1 \otimes id \\
A \otimes 1 & & A & \nwarrow u_{\otimes}^{r(-1)} u_{\otimes}^{r(-1)} & \\
\downarrow (\bar{g}u_{\otimes}^{r(-1)}) \otimes 1 & \nearrow u_{\otimes}^r u_{\otimes}^r & & & \\
(A \otimes 1) \otimes 1 & \xrightarrow{a_{\otimes}} & A \otimes (1 \otimes 1) & &
\end{array}$$

which allows us to conclude $\bar{f}\bar{g} = \bar{g}\bar{f}$ in $\widetilde{\mathbb{X}}$.

R.3 ($\overline{\bar{f}g} = \bar{f}\bar{g}$). As above, the first step is to show that the restrictions of each side are the same. Computing the restriction of the left hand side in \mathbb{X} :

$$\begin{aligned}
\overline{(\bar{f}u_{\otimes}^{r(-1)})(g \otimes 1)a_{\otimes}u_{\otimes}^{r(-1)}} &= \overline{(\bar{f}u_{\otimes}^{r(-1)})(g \otimes 1)a_{\otimes}} && u_{\otimes}^{r(-1)} \text{ iso, Lemma 2.2.3} \\
&= \overline{(\bar{f}u_{\otimes}^{r(-1)})(g \otimes 1)a_{\otimes}} && \text{Lemma 2.2.3} \\
&= \overline{\bar{f}gu_{\otimes}^{r(-1)}a_{\otimes}} && u_{\otimes}^{r(-1)} \text{ natural} \\
&= \overline{\bar{f}g} && u_{\otimes}^{r(-1)}, a_{\otimes} \text{ iso, Lemma 2.2.3} \\
&= \bar{f}\bar{g} && [\mathbf{R.3}] \text{ in } \mathbb{X}.
\end{aligned}$$

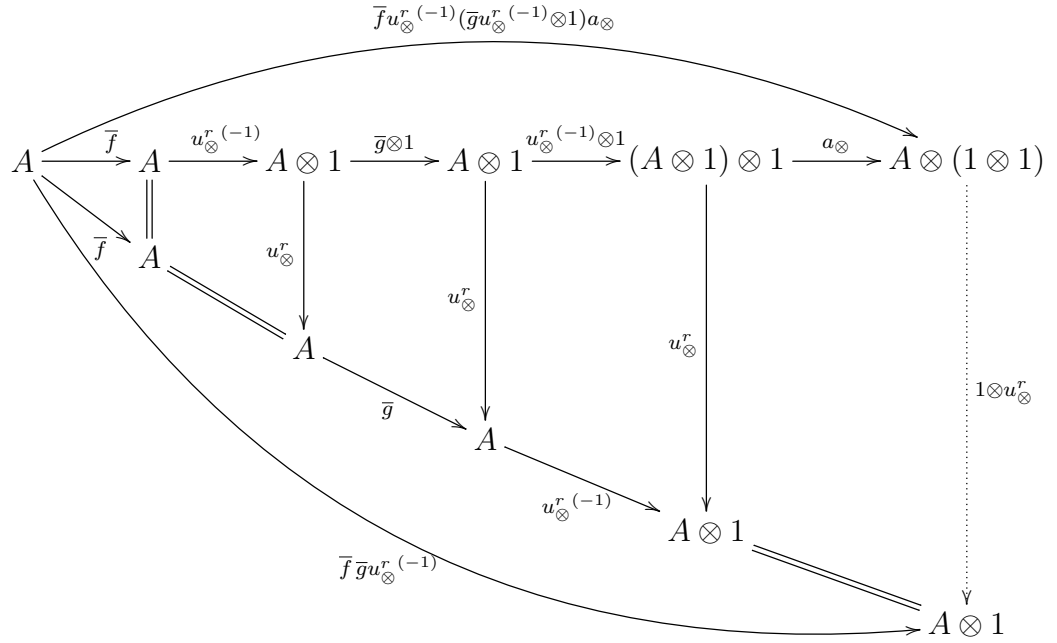
The restriction of the right hand side computes in \mathbb{X} as:

$$\begin{aligned}
\overline{(\bar{f}u_{\otimes}^{r(-1)})(\bar{g}u_{\otimes}^{r(-1)} \otimes 1)a_{\otimes}} &= \overline{(\bar{f}u_{\otimes}^{r(-1)})(\bar{g}u_{\otimes}^{r(-1)} \otimes 1)} && a_{\otimes} \text{ iso, Lemma 2.2.3} \\
&= \overline{\bar{f}\bar{g}u_{\otimes}^{r(-1)}u_{\otimes}^{r(-1)}} && u_{\otimes}^{r(-1)} \text{ natural} \\
&= \overline{\bar{f}\bar{g}} && u_{\otimes}^{r(-1)}u_{\otimes}^{r(-1)} \text{ iso, Lemma 2.2.3} \\
&= \bar{f}\bar{g} && \text{Lemma 2.2.3.}
\end{aligned}$$

Additionally, we see $\overline{\bar{f}g}$ in $\widetilde{\mathbb{X}}$ is expressed in \mathbb{X} as:

$$\begin{aligned}
\overline{(\bar{f}u_{\otimes}^{r(-1)})(g \otimes 1)a_{\otimes}u_{\otimes}^{r(-1)}} &= \bar{f}u_{\otimes}^{r(-1)}\overline{g \otimes 1} && [\mathbf{R.3}], [\mathbf{R.4}], a_{\otimes} \text{ iso} \\
&= \bar{f}\bar{g}u_{\otimes}^{r(-1)} && \otimes \text{a restriction bi-functor, } u_{\otimes}^{r(-1)} \text{ natural.}
\end{aligned}$$

The following diagram in \mathbb{X} follows the above right hand side with the top curved arrow and the left hand side with the bottom curved arrow. Note that we are using that $\overline{(\bar{f}u_{\otimes}^{r(-1)})(g \otimes 1)a_{\otimes}} = \bar{f}\bar{g}$ as shown above.



Hence, in \mathbb{X} , $\overline{(\bar{f}u_{\otimes}^{r(-1)})(g \otimes 1)a_{\otimes}u_{\otimes}^{r(-1)}} \stackrel{1 \otimes u_{\otimes}^r}{\simeq} \overline{(\bar{f}u_{\otimes}^{r(-1)})(\bar{g}u_{\otimes}^{r(-1)} \otimes 1)a_{\otimes}}$ and therefore $\overline{\bar{f}g} = \bar{f}\bar{g}$ in $\tilde{\mathbb{X}}$.

R.4 $f\bar{g} = \overline{f\bar{g}}f$ The restriction of the left hand side is:

$$\begin{aligned}
 \overline{f(\bar{g}u_{\otimes}^{r(-1)} \otimes 1)a_{\otimes}} &= \overline{f(\bar{g}u_{\otimes}^{r(-1)} \otimes 1)} && a_{\otimes} \text{ iso, Lemma 2.2.3} \\
 &= \overline{f\bar{g}u_{\otimes}^{r(-1)}} \otimes \bar{f} && \otimes \text{ restriction functor} \\
 &= \overline{f\bar{g}} \otimes \bar{f} && u_{\otimes}^{r(-1)} \text{ iso, Lemma 2.2.3} \\
 &= \overline{f(\bar{g} \otimes 1)}
 \end{aligned}$$

and the restriction of the right hand side is:

$$\begin{aligned}
\overline{f(g \otimes 1)u_{\otimes}^{r(-1)}(f \otimes 1)a_{\otimes}} &= \overline{f(g \otimes 1)u_{\otimes}^{r(-1)}(f \otimes 1)} && a_{\otimes} \text{ iso, Lemma 2.2.3} \\
&= \overline{f(g \otimes 1)fu_{\otimes}^{r(-1)}} && u_{\otimes}^{r(-1)} \text{ natural} \\
&= \overline{f(\bar{g} \otimes 1)u_{\otimes}^{r(-1)}} && [\mathbf{R.4}] \text{ for } \mathbb{X} \\
&= \overline{f(\bar{g} \otimes 1)u_{\otimes}^{r(-1)}} && \otimes \text{ is a restriction functor} \\
&= \overline{f(\bar{g} \otimes 1)} && u_{\otimes}^{r(-1)} \text{ iso, Lemma 2.2.3}
\end{aligned}$$

Computing the right hand side in \mathbb{X} ,

$$\begin{aligned}
\overline{f(g \otimes 1)a_{\otimes}u_{\otimes}^{r(-1)}(f \otimes 1)a_{\otimes}} &= \overline{f(g \otimes 1)fu_{\otimes}^{r(-1)}a_{\otimes}} && a_{\otimes} \text{ iso, } u_{\otimes}^{r(-1)} \text{ natural.} \\
&= \overline{f(\bar{g} \otimes 1)u_{\otimes}^{r(-1)}a_{\otimes}} && [\mathbf{R.3}], \otimes \text{ a restriction functor.}
\end{aligned}$$

$$\begin{array}{ccccccc}
A & \xrightarrow{f} & B \otimes C & \xrightarrow{\bar{g}u_{\otimes}^{r(-1)} \otimes 1} & (B \otimes 1) \otimes C & \xrightarrow{a_{\otimes}} & B \otimes (1 \otimes C) \\
& \searrow f & & & & & \downarrow 1 \otimes c_{\otimes} \\
& & B \otimes C & \xrightarrow{\bar{g} \otimes 1} & B \otimes C & \xrightarrow{u_{\otimes}^{r(-1)}} & (B \otimes C) \otimes 1 \xrightarrow{a_{\otimes}} B \otimes (C \otimes 1)
\end{array}$$

and hence, $\tilde{\mathbb{X}}$ is a restriction category. \square

3.2.2 The category $\tilde{\mathbb{X}}$ is a discrete restriction category

Lemma 3.2.8. *The unit of the inverse product in \mathbb{X} is the terminal object in $\tilde{\mathbb{X}}$.*

Proof. The unique map to the terminal object for any object A in $\tilde{\mathbb{X}}$ is the equivalence class of maps represented by $(u_{\otimes}^{l(-1)}, A)$. For this to be a terminal object, the diagram

$$\begin{array}{ccccc}
X & \xrightarrow{\overline{(f,C)}} & X & \xrightarrow{!_X} & \top \\
\downarrow (f,C) & & & \nearrow !_Y & \\
Y & & & &
\end{array}$$

must commute for all choices of f . Translating this to \mathbb{X} , this is the same as requiring

$$\begin{array}{ccccccc}
X & \xrightarrow{\bar{f}} & X & \xrightarrow{u_{\otimes}^{r(-1)}} & X \otimes 1 & \xrightarrow{u_{\otimes}^{l(-1)}} & 1 \otimes X \otimes 1 \\
\downarrow f & & & & & \swarrow 1 \otimes (u_{\otimes}^r f) & \\
Y \otimes C & \xrightarrow{u_{\otimes}^{l(-1)}} & 1 \otimes Y \otimes C & & & &
\end{array}$$

commute, which is true by [R.1] and from the coherence diagrams for the inverse product tensor. \square

Next, we show that the category $\widetilde{\mathbb{X}}$ has restriction products, given by the action of $(\widetilde{})$ on the \otimes tensor in \mathbb{X} .

First, define total maps π_0, π_1 in $\widetilde{\mathbb{X}}$ by:

$$\pi_0 : A \otimes B \xrightarrow{(1, B)} A \quad (3.5)$$

$$\pi_1 : A \otimes B \xrightarrow{(c_\otimes, A)} B \quad (3.6)$$

Given the maps $Z \xrightarrow{(f, C)} A$ and $Z \xrightarrow{(g, C')} B$, define $\langle (f, C), (g, C') \rangle$ as

$$Z \xrightarrow{(\Delta(f \otimes g)(1 \otimes c_\otimes \otimes 1), C \otimes C')} A \otimes B \quad (3.7)$$

where associativity is assumed as needed. Note that with the associativity maps, this is actually:

$$Z \xrightarrow{(\Delta(f \otimes g)a_\otimes(1 \otimes a_\otimes^{(-1)})(1 \otimes (c_\otimes \otimes 1))(1 \otimes a_\otimes)a_\otimes^{(-1)}, C \otimes C')} A \otimes B \quad (3.8)$$

Lemma 3.2.9. *On $\widetilde{\mathbb{X}}$, \otimes is a restriction product with projections π_0, π_1 with the product of maps f, g being $\langle f, g \rangle$.*

Proof. From the definition above, as 1 and c_\otimes are isomorphisms, the maps π_0, π_1 are total.

In order to show that $\overline{\langle f, g \rangle} = \overline{f} \overline{g}$, first reduce the left hand side:

$$\begin{aligned} \overline{\langle f, g \rangle} &= \overline{\Delta(f \otimes g)(1 \otimes c_\otimes \otimes 1)u_\otimes^{r(-1)}} && \text{in } \mathbb{X}, \text{ definition of restriction} \\ &= \overline{\Delta(f \otimes g)u_\otimes^{r(-1)}} && c_\otimes \text{ is iso} \\ &= \overline{\Delta(\overline{f} \otimes \overline{g})u_\otimes^{r(-1)}} && \text{from Lemma 2.2.3} \\ &= \overline{\Delta(\overline{f} \otimes \overline{g})u_\otimes^{r(-1)}} && \otimes \text{ is a restriction functor} \\ &= \overline{\overline{f} \overline{g} \Delta(1 \otimes 1)u_\otimes^{r(-1)}} && \text{Lemma 3.1.8(ii) twice} \\ &= \overline{\overline{f} \overline{g} u_\otimes^{r(-1)}} && \text{Lemma 2.2.3} \\ &= \overline{f} \overline{g} u_\otimes^{r(-1)} && \text{Lemma 2.2.3.} \end{aligned}$$

Then, the right hand side reduces as:

$$\begin{aligned}\overline{f\bar{g}} &= \overline{f}u_{\otimes}^r{}^{(-1)}(\overline{g}u_{\otimes}^r{}^{(-1)} \otimes 1)a_{\otimes} && \text{in } \mathbb{X} \text{ by definitions} \\ &= \overline{f\bar{g}}u_{\otimes}^r{}^{(-1)}u_{\otimes}^r{}^{(-1)}a_{\otimes} && u_{\otimes}^r{}^{(-1)} \text{ natural.}\end{aligned}$$

The restriction of the left hand side and the right hand side, in \mathbb{X} , is $\overline{f\bar{g}}$. This is done by applying Lemma 2.2.3 on page 15 once on the left and thrice on the right.

Thus, this shows $\overline{\langle f, g \rangle} = \overline{f\bar{g}}$ in $\widetilde{\mathbb{X}}$ where the mediating map in \mathbb{X} is $1 \otimes u_{\otimes}^r$.

Next, to show $\langle f, g \rangle \pi_0 \leq f$ (and $\langle f, g \rangle \pi_1 \leq g$), it is required to show $\overline{\langle f, g \rangle \pi_0} f = \langle f, g \rangle \pi_0$.

Calculating the left side, we see:

$$\begin{aligned}\overline{\langle f, g \rangle \pi_0} f &= \overline{\langle f, g \rangle \pi_0} f && \text{Lemma 2.2.3} \\ &= \overline{\langle f, g \rangle} f && \pi_0 \text{ is total} \\ &= \overline{f\bar{g}} f && \text{by above} \\ &= \overline{g\bar{f}} f && [\mathbf{R.2}] \\ &= \overline{g} f && [\mathbf{R.1}].\end{aligned}$$

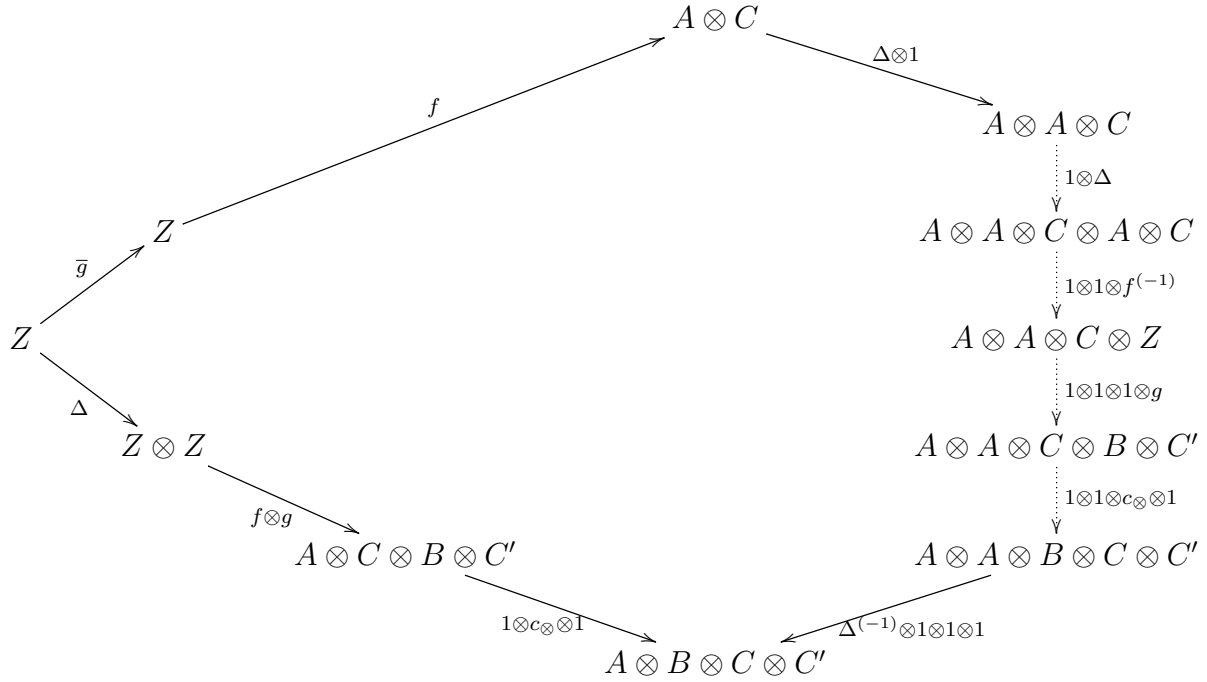
Now, turning to the right hand side:

$$\langle f, g \rangle \pi_0 = \Delta(f \otimes g)(1 \otimes c_{\otimes} \otimes 1)1 \quad \text{in } \mathbb{X}, \text{ by definition.}$$

To show these are equal in $\widetilde{\mathbb{X}}$, we need to first show the restrictions are the same in \mathbb{X} and then show there is a mediating map between the images in \mathbb{X} . The restriction of $\overline{g\bar{f}}$ is $\overline{f\bar{g}}$ immediately by $[\mathbf{R.3}]$ and $[\mathbf{R.2}]$. For the right hand side, calculate in \mathbb{X} :

$$\begin{aligned}\overline{\Delta(f \otimes g)(1 \otimes c_{\otimes} \otimes 1)} &= \overline{\Delta(f \otimes g)} && \text{Lemma 2.2.3} \\ &= \Delta(f \otimes g)(f^{(-1)} \otimes g^{(-1)})\Delta^{(-1)} && \mathbb{X} \text{ is an inverse category} \\ &= \Delta(\overline{f} \otimes \overline{g})\Delta^{(-1)} \\ &= \overline{f\bar{g}}\Delta\Delta^{(-1)} && \text{Lemma 3.1.8(ii) twice} \\ &= \overline{f\bar{g}}.\end{aligned}$$

The diagram below, shows the required mediating map.



□

At this point, we have shown that $\tilde{\mathbb{X}}$ is a restriction category with restriction products. This leads us to the following theorem:

Theorem 3.2.10. *For any inverse category \mathbb{X} , the category $\tilde{\mathbb{X}}$ is a discrete restriction category.*

Proof. The fact that $\tilde{\mathbb{X}}$ is a Cartesian restriction category is immediate from lemmas 3.2.6 on page 50, 3.2.7 on page 53, 3.2.8 on page 56 and 3.2.9 on page 57.

To show that it is discrete, we need only show that the map $(\Delta u_{\otimes}^{r(-1)}, 1)$ is in the same equivalence class as $\tilde{\mathbb{X}}$'s $\Delta (= \langle 1, 1 \rangle = \langle (u_{\otimes}^{r(-1)}, 1), (u_{\otimes}^{r(-1)}, 1) \rangle)$. As both Δ and $u_{\otimes}^{r(-1)}$ are total, the restriction of each side is the same, namely 1. The diagram below uses Corollary

3.2.5 and shows that the two maps are in the same equivalence class.

$$\begin{array}{ccc}
 & & A \otimes A \otimes 1 \\
 & \nearrow \Delta u_{\otimes}^{r(-1)} & \downarrow u_{\otimes}^{r(-1)} \\
 A & \xrightarrow{\Delta(u_{\otimes}^{r(-1)} \otimes u_{\otimes}^{r(-1)})(1 \otimes c_{\otimes} \otimes 1)} & A \otimes A \otimes 1 \otimes 1
 \end{array}$$

□

3.2.3 Equivalence of categories

This section will show that the category of discrete inverse categories (maps being restriction functors that preserve the inverse tensor) is equivalent to the category of discrete restriction categories (maps being the restriction functors which preserve the product). In the following, \mathbb{X} will always be a discrete inverse category, \mathbb{D} and \mathbb{C} will be discrete restriction categories.

We approach the equivalence proof by exhibiting the universal property for discrete inverse categories for the functor **INV** from discrete restriction categories to discrete inverse categories. The functor **INV** maps a discrete restriction category to its inverse subcategory and maps functors between discrete restriction categories to a functor having the same action on the partial inverses. That is, given $G : \mathbb{C} \rightarrow \mathbb{D}$, then:

$$\mathbf{INV}(G) : \mathbf{INV}(\mathbb{C}) \rightarrow \mathbf{INV}(\mathbb{D})$$

$$\mathbf{INV}(G)(A) = GA \quad (\text{all objects of } \mathbb{D} \text{ are in } \text{Inv}(\mathbb{D}))$$

$$\mathbf{INV}(G)(f) = G(f) \quad (\text{restriction functors preserve partial inverse})$$

We continue by showing the η and ε of the universal property are isomorphisms. First, let $\eta : \mathbb{X} \rightarrow \mathbf{INV}(\tilde{\mathbb{X}})$ be an identity on objects functor. For maps f in \mathbb{X} , $\eta(f) = (fu_{\otimes}^{r(-1)}, 1)$.

Next, consider a functor $F : \mathbb{X} \rightarrow \mathbf{INV}(\mathbb{D})$ defined as follows:

$$\text{Objects: } F^{\#} : A \mapsto F(A)$$

$$\text{Arrows: } F^{\#} : (f, C) \mapsto F(f)\pi_0$$

This allows us to write the diagram:

$$\begin{array}{ccc}
 \mathbb{X} & \xrightarrow{\eta} & \mathbf{INV}(\tilde{\mathbb{X}}) \\
 & \searrow F & \downarrow \mathbf{INV}(F^\#) \\
 & & \mathbf{INV}(\mathbb{D})
 \end{array} \tag{3.9}$$

In order to show this is a universal diagram, we proceed with a series of lemmas building to the result.

Lemma 3.2.11. *For any discrete inverse category \mathbb{X} , all invertible maps $(g, C) : A \rightarrow B$ in $\tilde{\mathbb{X}}$ are in the equivalence class of $(fu_\otimes^r{}^{(-1)}, 1)$ for some $f : A \rightarrow B$.*

Proof. As (g, C) is invertible in $\tilde{\mathbb{X}}$, the map $(g, C)^{(-1)} : B \rightarrow A$ exists. $(g, C)^{(-1)}$ must be in the equivalence class of some map $k : B \rightarrow A \otimes D$, and also note that $\overline{(g, C)}$ is by construction the equivalence class of the map $\bar{g}u_\otimes^r{}^{(-1)} : A \rightarrow A \otimes 1$ in \mathbb{X} . This means, diagramming in \mathbb{X} , there is an n such that

$$\begin{array}{ccccc}
 B & \xrightarrow{k} & A \otimes D & \xrightarrow{f \otimes 1} & B \otimes C \otimes D \\
 & & & & \downarrow \Delta \otimes 1 \\
 & & & & B \otimes B \otimes C \otimes D \\
 & & & & \vdots 1 \otimes n \\
 & & & & B \otimes B \otimes 1 \\
 & & & & \downarrow \Delta^{(-1)} \otimes 1 \\
 & & & & B \otimes 1 \\
 & \searrow \bar{g}u_\otimes^r{}^{(-1)} & & &
 \end{array}$$

commutes.

Starting with $g : A \rightarrow B \otimes C$, construct the map f in \mathbb{X} with the following diagram:

$$\begin{array}{ccc}
 A & \xrightarrow{g} & B \otimes C \\
 & \searrow f & \downarrow \Delta \otimes 1 \\
 & & B \otimes B \otimes C \\
 & & \downarrow 1 \otimes \Delta \otimes 1 \\
 & & B \otimes B \otimes B \otimes C \\
 & & \downarrow 1 \otimes 1 \otimes k \otimes 1 \\
 & & B \otimes B \otimes A \otimes D \otimes C \\
 & & \downarrow 1 \otimes 1 \otimes g \otimes 1 \otimes 1 \\
 & & B \otimes B \otimes B \otimes C \otimes D \otimes C \\
 & & \downarrow 1 \otimes \Delta^{(-1)} \otimes 1 \otimes c_{\otimes} \\
 & & B \otimes B \otimes C \otimes C \otimes D \\
 & & \downarrow 1 \otimes 1 \otimes \Delta^{(-1)} \otimes 1 \\
 & & B \otimes B \otimes C \otimes D \\
 & & \downarrow 1 \otimes n \\
 & & B \otimes B \otimes 1 \\
 & & \downarrow (\Delta^{(-1)} \otimes 1) u_{\otimes}^l \\
 & & B
 \end{array}$$

By its construction, $f : A \rightarrow B$ in \mathbb{X} and $(fu_{\otimes}^{r(-1)}, 1)$ is in the same equivalence class as (g, C) .

□

Lemma 3.2.12. *Diagram (equation (3.9)) above is a commutative diagram.*

Proof. Chasing maps around the diagram, we have:

$$\begin{array}{ccc}
 f & \xrightarrow{\eta} & (fu_{\otimes}^{r(-1)}, 1) \\
 \searrow F & & \downarrow \mathbf{INV}(F\#) \\
 & & F(f) \equiv F(fu_{\otimes}^{r(-1)})\pi_0
 \end{array}$$

As η is identity on the objects, diagram equation (3.9) on the preceding page commutes. □

Lemma 3.2.13. *The functor \mathbf{INV} from the category of discrete restriction categories to the category of discrete inverse categories is full and faithful.*

Proof. To show fullness, we must show **INV** is surjective on hom-sets. Given a functor between two categories in the image of **INV**, i.e., $G : \mathbf{INV}(\mathbb{C}) \rightarrow \mathbf{INV}(\mathbb{D})$, construct a functor $H : \mathbb{C} \rightarrow \mathbb{D}$ as follows:

Action on objects: $H(A) = G(A)$,

Objects on maps: $H(f) = G(\langle f, 1 \rangle)\pi_0$.

H is well defined as we know $\langle f, 1 \rangle$ is an invertible map and therefore in the domain of G .

To see H is a functor:

$$H(1) = G(\langle 1, 1 \rangle)\pi_0 = \Delta_{\mathbb{D}}\pi_0 = 1$$

$$H(fg) = G(\langle fg, 1 \rangle)\pi_0 = G(\langle f, 1 \rangle)\pi_0 G(\langle g, 1 \rangle)\pi_0 = H(f)H(g)$$

But on any invertible map, $H(f) = G(\langle f, 1 \rangle)\pi_0 = \langle G(f), 1 \rangle\pi_0 = G(f)$ and therefore $\mathbf{INV}(\cdot)H = G$, so **INV** is full.

Next, assume we have $F, G : \mathbb{C} \rightarrow \mathbb{D}$ with $\mathbf{INV}(F) = \mathbf{INV}(G)$. Considering $F(f)$ and $F(g)$, we know $F(\langle f, 1 \rangle) = G(\langle f, 1 \rangle)$ as $\langle f, 1 \rangle$ is invertible. Thus, as the functors preserve the product structure, we have

$$F(f) = F(\langle f, 1 \rangle)F(\pi_0) = G(\langle f, 1 \rangle)G(\pi_0) = G(f).$$

Thus, **INV** is faithful. □

Corollary 3.2.14. *The functor $F^\#$ in diagram [equation \(3.9\) on page 61](#) is unique.*

Proof. This follows immediately from lemma [3.2.13 on the preceding page](#), **INV** is faithful. □

Corollary 3.2.15. *The category $\tilde{\mathbb{X}}$ and functor $\eta : \mathbb{X} \rightarrow \mathbf{INV}(\tilde{\mathbb{X}})$ is a universal pair for the functor **INV**.*

Proof. Immediate from Corollary [3.2.14](#) and Lemma [3.2.12 on the preceding page](#). □

Lemma 3.2.16. *The functor $\eta : \mathbb{X} \rightarrow \mathbf{INV}(\widetilde{\mathbb{X}})$ is an isomorphism.*

Proof. As η is an identity on objects functor, we need only show that it is full and faithful. Referring to Lemma 3.2.11 on page 61 above, we immediately see that η is full. For faithful, if we assume $(fu_{\otimes}^{r(-1)}, 1)$ is equal in $\widetilde{\mathbb{X}}$ to $(gu_{\otimes}^{r(-1)}, 1)$. This means in \mathbb{X} , that $\bar{f} = \bar{g}$ and there is a h such that

$$\begin{array}{ccccc}
 & & B \otimes 1 & \xrightarrow{(\Delta \otimes 1) a_{\otimes}} & B \otimes (B \otimes 1) \\
 & \nearrow fu_{\otimes}^{r(-1)} & & & \downarrow 1 \otimes h \\
 A & & & & B \otimes (B \otimes 1) \\
 & \searrow gu_{\otimes}^{r(-1)} & & & \uparrow a_{\otimes}^{(-1)} (\Delta^{(-1)} \otimes 1) \\
 & & B \otimes 1 & &
 \end{array}$$

This simplifies out to $g = f\Delta(1 \otimes h)\Delta^{(-1)}$. But by Lemma 3.1.8 on page 39, part (iv) on page 39, $\Delta(1 \otimes h)\Delta^{(-1)} = \overline{\Delta(1 \otimes h)\Delta^{(-1)}}$. Setting $\Delta(1 \otimes h)\Delta^{(-1)}$ as k , we have $g = f\bar{k}$. But this gives us:

$$g = f\bar{k} = \overline{f\bar{k}}f = \overline{f\bar{k}}f = \bar{g}f = \bar{f}f = f.$$

This shows η is faithful and hence an isomorphism between \mathbb{X} and $\mathbf{INV}(\widetilde{\mathbb{X}})$. \square

Theorem 3.2.17. *The category of discrete inverse categories (objects are discrete inverse categories, maps are inverse tensor preserving functors) is equivalent to the category of discrete restriction categories (objects are discrete restriction categories, maps are the Cartesian restriction functors).*

Proof. From the above lemmas, we have shown that we have an adjoint:

$$(\eta, \varepsilon) : \mathbf{T} \vdash \mathbf{INV} : D_{ic} \rightarrow D_{rc} \quad (3.10)$$

By lemma 3.2.16 we know η is an isomorphism. But this means the functor \mathbf{T} is full and faithful, as shown in, e.g., Proposition 2.2.6 of [26]. From lemma 3.2.13 we know that \mathbf{INV}

is full and faithful. But again by the previous reference, this means ε is an isomorphism. Thus, by Corollary 3.2.15 and Proposition 2.2.7 of [26] we have the equivalence of the two categories. \square

3.2.4 Examples of the $\widetilde{(-)}$ construction

Example 3.2.18 (Completing a finite discrete inverse category).

Continuing from example 3.1.7 on page 38, recall the discrete category of 4 elements with two different tensors. Completing these gives two different lattices. They are either the straight line lattice, or the diamond semilattice. Below are the details of these constructions.

Recall \mathbb{D} has four elements a, b, c and d , and there are two possible inverse product tensors:

\otimes	a	b	c	d
a	a	a	a	a
b	a	b	b	b
c	a	b	c	c
d	a	b	c	d

\otimes	a	b	c	d
a	a	a	a	a
b	a	b	a	b
c	a	a	c	c
d	a	b	c	d

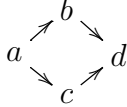
Define Δ as the identity map. Then, for the first tensor, $\widetilde{\mathbb{D}}$ has the following maps

$$\begin{array}{llll}
 a \xrightarrow{(id,a) \ (\equiv(id,b)\equiv(id,c)\equiv(id,d))} a, & a \xrightarrow{(id,a)} b, & a \xrightarrow{(id,a)} c, & a \xrightarrow{(id,a)} d \\
 b \xrightarrow{(id,b) \ (\equiv(id,c)\equiv(id,d))} b, & b \xrightarrow{(id,b)} c, & b \xrightarrow{(id,b)} d & \\
 c \xrightarrow{(id,c) \ (\equiv(id,d))} c, & c \xrightarrow{(id,c)} d & & \\
 d \xrightarrow{(id,d)} d & & &
 \end{array}$$

resulting in the straight-line $(a \rightarrow b \rightarrow c \rightarrow d)$ lattice. The tensor in \mathbb{D} becomes the meet and hence is a categorical product in $\widetilde{\mathbb{D}}$. Note that the only partial inverses in $\widetilde{\mathbb{D}}$ are the identity functions and that for all maps f , $\langle f, 1 \rangle = id$.

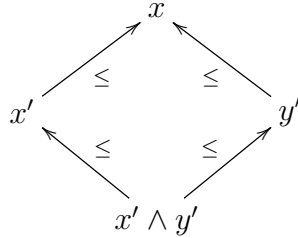
With the second tensor table, we have:

$$\begin{array}{llll}
a \xrightarrow{(id,a) \ (\equiv(id,b)\equiv(id,c)\equiv(id,d))} a, & a \xrightarrow{(id,a)} b, & a \xrightarrow{(id,a)} c, & a \xrightarrow{(id,a)} d \\
& b \xrightarrow{(id,b) \ (\equiv(id,d))} b, & b \xrightarrow{(id,b)} d & \\
& c \xrightarrow{(id,c) \ (\equiv(id,d))} c, & c \xrightarrow{(id,c)} d & \\
& & d \xrightarrow{(id,d)} d &
\end{array}$$

resulting in the “diamond” lattice, . Once again, the tensor in \mathbb{D} is the meet.

Example 3.2.19. Lattice completion. Suppose we have a set together with an idempotent, commutative, associative operation \wedge on the set, giving us a lattice, \mathbb{L} . Further suppose the set is partially ordered via \leq with the order being compatible with \wedge .

Then, we may create a pullback square for any $x' \leq x$, $y' \leq x$ with



Considering \mathbb{L} as a category, we see that all maps are monic and therefore, we may create a partial map category $\text{Par}(\mathbb{L}, \mathcal{M})$ where the stable system of monics are all the maps.

Then $\widetilde{\text{Par}(\mathbb{L}, \mathcal{M})}$ becomes the completion of the lattice over \wedge .

3.3 Coproducts in restriction categories

3.3.1 Coproducts

Restriction categories may also have coproducts and initial objects.

Definition 3.3.1. In a restriction category \mathbb{X} , a coproduct is a *restriction coproduct* when the embeddings Π_0 and Π_1 are total.

Lemma 3.3.2. *The definition of restriction coproduct implies the following:*

- (i) $\overline{f + g} = \overline{f} + \overline{g}$ which means $+$ is a restriction functor.
- (ii) $\nabla : A + A \rightarrow A$ is total.
- (iii) $? : 0 \rightarrow A$ is total, where 0 is the initial object in the restriction category.

Proof.

(i) **$+$ is a restriction functor.** Consider the diagram:

$$\begin{array}{ccccc}
 A & \xrightarrow{f} & A' & & \\
 \searrow \Pi_0 & & \searrow \Pi'_0 & & \\
 & A + B & \xrightarrow{f+g} & A' + B' & \\
 \nearrow \Pi_1 & & \nearrow \Pi'_1 & & \\
 B & \xrightarrow{g} & B' & &
 \end{array}$$

In order to show $\overline{f + g} = \overline{f} + \overline{g}$, it suffices to show that $\Pi_0 \overline{f + g} = \Pi_0(\overline{f} + \overline{g}) = \overline{f} \Pi_0$.

$$\begin{aligned}
 \Pi_0 \overline{f + g} &= \overline{\Pi_0(f + g)} \Pi_0 && \text{[R.4]} \\
 &= \overline{f \Pi'_0} \Pi_0 && \text{coproduct diagram} \\
 &= \overline{f \Pi'_0} \Pi_0 && \text{Lemma 2.2.3[(iii)]} \\
 &= \overline{f} \Pi_0 && \Pi'_0 \text{ total}
 \end{aligned}$$

(ii) **$\nabla : A + A \rightarrow A$ is total.** By the definition of ∇ ($= \langle 1|1 \rangle$) and the co-product, the following diagram commutes,

$$\begin{array}{ccc}
 & A + A & \\
 \Pi_0 \nearrow & \downarrow \nabla & \nwarrow \Pi_1 \\
 A & = & A = A
 \end{array}$$

resulting in:

$$\begin{aligned}
\Pi_0 \overline{\nabla} &= \overline{\Pi_0 \nabla} \Pi_0 \\
&= \overline{1} \Pi_0 \\
&= \Pi_0
\end{aligned}$$

Similarly, $\Pi_1 \overline{\nabla} = \Pi_1$, hence, the restriction of ∇ is 1 and therefore ∇ is total.

(iii) $? : 0 \rightarrow A$ is **total**. This follows from

$$\begin{array}{ccc}
0 & \xrightarrow{\Pi_1} & A + 0 \\
& \searrow ? & \parallel \\
& & A
\end{array}$$

so $?$ can be defined as the total coproduct injection.

□

Recall that when an object is both initial and terminal, it is referred to as a zero object and denoted as 0. This gives rise to the zero map $0_{A,B} : A \rightarrow 0 \rightarrow B$ between any two objects.

Definition 3.3.3. Given a restriction category \mathbb{X} with a zero object, then 0 is a *restriction zero* when for each object A in \mathbb{X} we have $\overline{0_{A,A}} = 0_{A,A}$.

Lemma 3.3.4 (Cockett-Lack). *For a restriction category \mathbb{X} , the following are equivalent:*

- (i) \mathbb{X} has a restriction zero;
- (ii) \mathbb{X} has an initial object 0 and terminal object 1 and each initial map z_A is a restriction monic;
- (iii) \mathbb{X} has a terminal object 1 and each terminal map t_A is a restriction retraction.

3.3.2 Inverse categories with restriction coproducts

Proposition 3.3.5. *An inverse category \mathbb{X} with restriction coproducts is a preorder.*

Proof. By Lemma 3.3.2, we know ∇ is total and therefore $\nabla\nabla^{(-1)} = 1$. From the coproduct diagrams, we have $\Pi_0\nabla = 1$ and $\Pi_1\nabla = 1$. But this gives us $\nabla^{(-1)}\Pi_0^{(-1)} = \Pi_0\nabla^{(-1)} = 1$ and $\nabla^{(-1)}\Pi_1^{(-1)} = 1$. Hence, $\nabla^{(-1)} = \Pi_0$ and $\nabla^{(-1)} = \Pi_1$.

This means for parallel maps $f, g : A \rightarrow B$, we have

$$f = \Pi_0\langle f|g \rangle = \nabla^{(-1)}\langle f|g \rangle = \Pi_1\langle f|g \rangle = g$$

and therefore \mathbb{X} is a preorder. □

3.4 Disjointness in an inverse category

In the following, we will add two related structures to an inverse category with a restriction zero. This structure is meant to be evocative of the concept of *join* in a restriction category.

3.4.1 Disjointness relations

In this subsection, we will define a disjointness relationship between maps and explore alternate characterizations of this relation on the restriction idempotents of objects.

Definition 3.4.1. In an inverse category \mathbb{X} with a restriction zero, the relation \perp between two parallel maps $f, g : A \rightarrow B$ is called a *disjointness relation* when it satisfies the following

properties:

[Dis.1] For all $f : A \rightarrow B$, $f \perp 0$;

[Dis.2] $f \perp g$ implies $\overline{f}g = 0$;

[Dis.3] $f \perp g$, $f' \leq f$, $g' \leq g$ implies $f' \perp g'$;

[Dis.4] $f \perp g$ implies $g \perp f$;

[Dis.5] $f \perp g$ implies $hf \perp hg$; (Stable)

[Dis.6] $f \perp g$ implies $\overline{f} \perp \overline{g}$ and $\hat{f} \perp \hat{g}$.

[Dis.7] $\overline{f} \perp \overline{g}$, $\hat{h} \perp \hat{k}$ implies $fh \perp gk$;

Lemma 3.4.2. *In Definition 3.4.1, provided we retain [Dis.1-5] we may replace [Dis.6] and [Dis.7] by:*

[Dis.6'] $f \perp g$ if and only if $\overline{f} \perp \overline{g}$ and $\hat{f} \perp \hat{g}$.

Proof. Given [Dis.6] and [Dis.7], the *only if* direction of [Dis.6'] is immediate. To show the *if* direction, assume $\overline{f} \perp \overline{g}$ and $\hat{f} \perp \hat{g}$. This also means that $\overline{\overline{f}} \perp \overline{\overline{g}}$. Then, by [Dis.7], $\overline{f}f \perp \overline{g}g$ and therefore $f \perp g$.

Conversely, assume we are given [Dis.6']. Then, [Dis.6] follows immediately. To show [Dis.7], assume we have $\overline{f} \perp \overline{g}$, $\hat{h} \perp \hat{k}$. As $\overline{fh} \leq \overline{f}$ and $\overline{gk} \leq \overline{g}$, by [Dis.3], we know that $\overline{fh} \perp \overline{gk}$. Similarly, $\widehat{fh} \leq \hat{h}$ and $\widehat{gk} \leq \hat{k}$, giving us $\widehat{fh} \perp \widehat{gk}$. Then, from [Dis.6'] we may conclude $fh \perp gk$, showing [Dis.7] holds. \square

Lemma 3.4.3. *In an inverse category \mathbb{X} with \perp a disjointness relation:*

(i) $f \perp g$ if and only if $f^{(-1)} \perp g^{(-1)}$;

(ii) $f \perp g$ implies $fh \perp gh$ (Universal);

(iii) $f \perp g$ implies $f\hat{g} = 0$;

(iv) if m, n are monic, then $fm \perp gn$ implies $\overline{f} \perp \overline{g}$;

(v) if m, n are monic, then $m^{(-1)}f \perp n^{(-1)}g$ implies $\hat{f} \perp \hat{g}$;

Proof.

- (i) Assume $f \perp g$. Then we know that $\overline{f} \perp \overline{g}$ and $\hat{f} \perp \hat{g}$. But since $\hat{f} = \overline{f^{(-1)}}$ and $\overline{f} = \widehat{f^{(-1)}}$, this means $\overline{f^{(-1)}} \perp \overline{g^{(-1)}}$ and $\widehat{f^{(-1)}} \perp \widehat{g^{(-1)}}$ and again by the first item of this lemma, we have $f^{(-1)} \perp g^{(-1)}$. The converse follows with a similar argument.
- (ii) Assume $f \perp g$. By the previous item, we have $f^{(-1)} \perp g^{(-1)}$. By [Dis.5], $h^{(-1)}f^{(-1)} \perp h^{(-1)}g^{(-1)}$, giving us $(fh)^{(-1)} \perp (gh)^{(-1)}$. Again by the previous item, we now have $fh \perp gh$.
- (iii) Assume $f \perp g$. From item (i) and reflexivity, we know that $g^{(-1)} \perp f^{(-1)}$ and therefore $\overline{g^{(-1)}}f^{(-1)} = \hat{g}f^{(-1)} = 0$. However, in an inverse category, $0^{(-1)} = 0$ and therefore $0 = (\hat{g}f^{(-1)})^{(-1)} = f\hat{g}^{(-1)} = f\hat{g}$.
- (iv) Assume we have $fm \perp gn$ where m, n are monic. By [Dis.6], $\overline{fm} \perp \overline{gn}$. By Lemma 2.2.3, $\overline{fm} = \overline{f\overline{m}} = \overline{f}1 = \overline{f}$ and therefore $\overline{f} \perp \overline{g}$.
- (v) This is a corollary to the previous item. By assumption, we have $m^{(-1)}f \perp n^{(-1)}g$ and therefore $f^{(-1)}m \perp g^{(-1)}n$. By the previous item, this means $\overline{f^{(-1)}} \perp \overline{g^{(-1)}}$ and hence $\hat{f} \perp \hat{g}$.

□

We may define the disjointness relation via its action in $\mathcal{O}(a)$.

Definition 3.4.4. Given an inverse category \mathbb{X} , a relation $\perp_A \subseteq \mathcal{O}(A)^2$ for each $A \in \text{ob}(\mathbb{X})$,

is an *open disjointness* relation when for all $e, e' \in \mathcal{O}(A)$

$$[\mathbf{Odis.1}] \quad 1 \perp_A 0;$$

$$[\mathbf{Odis.2}] \quad e \perp_A e' \text{ implies } e' \perp_A e;$$

$$[\mathbf{Odis.3}] \quad e \perp_A e' \text{ implies } ee' = 0;$$

$$[\mathbf{Odis.4}] \quad e \perp_A e' \text{ implies } \overline{fe} \perp_B \overline{fe'} \text{ for all } f : B \rightarrow A;$$

$$[\mathbf{Odis.5}] \quad e \perp_A e' \text{ implies } \widehat{eg} \perp_C \widehat{e'g} \text{ for all } g : A \rightarrow C;$$

$$[\mathbf{Odis.6}] \quad e \perp_A e', \quad e_1 \leq e, \quad e'_1 \leq e' \text{ implies } e_1 \perp_A e'_1.$$

We will normally write \perp rather than \perp_A where the object is either clear or not germane to the point under discussion.

Proposition 3.4.5. *If \perp is a disjointness relation in \mathbb{X} , it is an open disjointness relation on the restriction idempotents.*

Proof.

[**Odis.1**] This follows immediately from [**Dis.1**] by taking $f = 1$.

[**Odis.2**] Reflexivity follows directly from [**Dis.4**].

[**Odis.3**] By [**Dis.2**], $0 = \bar{e}e' = ee'$.

[**Odis.4**] Given $e \perp e'$, we have $fe \perp fe'$ by [**Dis.5**]. Then, by [**Dis.6**] we may conclude $\overline{fe} \perp \overline{fe'}$.

[**Odis.5**] This follows from the above item, using $g^{(-1)}$ for f . This means we have $\overline{g^{(-1)}e} \perp \overline{g^{(-1)}e'}$. But this gives us $\overline{(eg)^{(-1)}} \perp \overline{(e'g)^{(-1)}}$. Recalling from Lemma 2.2.11 that $\hat{k} = \overline{k^{(-1)}}$, we may conclude $\widehat{eg} \perp \widehat{e'g}$.

[**Odis.6**] Assuming $e \perp e'$ and $e_1 \leq e, e'_1 \leq e'$, by [**Dis.3**], $e_1 \perp e'_1$.

Therefore, \perp acts as an open disjointness relation on $\mathcal{O}(A)^2$.

□

Definition 3.4.6. If \perp is an open disjointness relation in \mathbb{X} , then we may define a relation ${}_A\perp_B \in \mathbb{X}(A, B)^2$ by

$$\frac{f, g : A \rightarrow B, \bar{f} \perp \bar{g}, \hat{f} \perp \hat{g}}{f {}_A\perp_B g}.$$

We call ${}_A\perp_B$ an *extended disjointness relation*.

Proposition 3.4.7. *If \perp is an extended disjointness relation based on \perp in \mathbb{X} , then \perp is a disjointness relation in \mathbb{X} .*

Proof.

[Dis.1] We need to show $f \perp 0$ for any f . We know that $1 \perp 0$ and therefore $\bar{f} \perp 0$ and $\hat{f} \perp 0$, as $\bar{f} \leq 1$ and $\hat{f} \leq 1$. This gives us $f \perp 0$.

[Dis.2] Assume $f \perp g$, i.e., $\bar{f} \perp \bar{g}$. Then, $\bar{f}g = \bar{f}\bar{g}g = 0g = 0$.

[Dis.3] We are given $f \perp g$, $f' \leq f$ and $g' \leq g$. By lemma 2.2.9[(iii)] $\bar{f}' \leq \bar{f}$ and $\bar{g}' \leq \bar{g}$. Then, by [Odis.6], as $\bar{f} \perp \bar{g}$ we have $\bar{f}' \perp \bar{g}'$. By 2.2.9[(iv)], we have $\hat{f}' \leq \hat{f}$ and $\hat{g}' \leq \hat{g}$. Then, by [Odis.6], as $\hat{f} \perp \hat{g}$ we have $\hat{f}' \perp \hat{g}'$. This means $f' \perp g'$.

[Dis.4] Reflexivity of \perp follows immediately from the reflexivity of \perp .

[Dis.5] Assume $f \perp g$, i.e., $\bar{f} \perp \bar{g}$ and $\hat{f} \perp \hat{g}$. Then we have $\overline{hf} \perp \overline{hg}$ by [Odis.4]. By lemma 2.2.9[(ii)] we have $\widehat{hf} \leq \hat{f}$ and $\widehat{hg} \leq \hat{g}$. Therefore we have $\widehat{hf} \perp \widehat{hg}$ by [Odis.6] and therefore $hf \perp hg$.

[Dis.6] This follows directly from definition 3.4.6.

[Dis.7] We assume $\bar{f} \perp \bar{g}$ and $\hat{h} \perp \hat{k}$. By definition 3.4.6 we have $\bar{f} \perp \bar{g}$ and $\hat{h} \perp \hat{k}$. By lemma 2.2.9[(i)], we have $\overline{fh} \leq \bar{f}$ and $\overline{gk} \leq \bar{g}$. Therefore, $\overline{fh} \perp \overline{gk}$ by [Odis.6]. By 2.2.9[(ii)], $\widehat{fh} \leq \hat{h}$ and $\widehat{gk} \leq \hat{k}$, giving us $\widehat{fh} \perp \widehat{gk}$ also by [Odis.6]. This means $fh \perp gk$.

□

Theorem 3.4.8. *To give a disjointness relation \perp on \mathbb{X} is to give an open disjointness relation $\underline{\perp}$ on \mathbb{X} .*

Proof. Suppose we are given the disjointness relation \perp . By Proposition 3.4.5, this is an open disjointness relation on each of the sets of idempotents, $\mathcal{O}(A)$. We will label that relation $\underline{\perp}$.

Use Definition 3.4.6 to create an extended disjointness relation based on $\underline{\perp}$, signify it by $\underline{\underline{\perp}}$. By Proposition 3.4.7, $\underline{\underline{\perp}}$ is a disjointness relation on \mathbb{X} .

Assume $f \perp g$. Then we have $\bar{f} \underline{\underline{\perp}} \bar{g}$ and $\hat{f} \underline{\underline{\perp}} \hat{g}$ by [Dis.6] and Proposition 3.4.5. Then, from Definition 3.4.6, we have $f \underline{\underline{\perp}} g$.

Assume $f \underline{\underline{\perp}} g$. Then we must have had $\bar{f} \underline{\underline{\perp}} \bar{g}$ and $\hat{f} \underline{\underline{\perp}} \hat{g}$ by Definition 3.4.6 and therefore $\bar{f} \perp \bar{g}$ and $\hat{f} \perp \hat{g}$. Then, by Proposition 3.4.3, we have $f \perp g$.

Now, suppose we are given the open disjointness relation $\underline{\perp}$. Similar to above, we can construct the extended disjointness relation $\underline{\underline{\perp}}$ by Definition 3.4.6. From the disjointness relation $\underline{\underline{\perp}}$, we have the open disjointness relation $\overline{\underline{\underline{\perp}}}$ by Lemma 3.4.5.

Assume $e \underline{\underline{\perp}} e'$. As this means both $\bar{e} \underline{\underline{\perp}} \bar{e}'$ and $\hat{e} \underline{\underline{\perp}} \hat{e}'$, we have $e \perp e'$. By Proposition 3.4.5 this means $e \overline{\underline{\underline{\perp}}} e'$.

If we are given that $e \overline{\underline{\underline{\perp}}} e'$, then we know that $e \perp e'$ by Proposition 3.4.5. From Definition 3.4.6, this requires that $\bar{e} \underline{\underline{\perp}} \bar{e}'$ and $\hat{e} \underline{\underline{\perp}} \hat{e}'$, but that just means $e \underline{\underline{\perp}} e'$. \square

Note that while we have worked with binary disjointness throughout this section, one may extend the concept to lists of maps simply by considering disjointness pairwise. I.e., we have $\perp [f_1, f_2, \dots, f_n]$ if and only if $f_i \perp f_k$ whenever $i \neq j$.

Disjointness is additional structure on a restriction category, i.e., it is possible to have more than one disjointness relation on the category.

Example 3.4.9. Consider the restriction category INJ . Here, the objects are sets and maps are the partial injective set functions, where $\bar{f} = id|_{\text{dom}(f)}$. The restriction zero is the empty map (i.e., $\text{dom}(0) = \text{range}(0) = \emptyset$).

We may define the disjointness relation \perp by $f \perp g$ if and only if $\text{dom}(f) \cap \text{dom}(g) = \emptyset$ and $\text{range}(f) \cap \text{range}(g) = \emptyset$. It is reasonably straightforward to verify [Dis.1] through [Dis.7]. For example, take [Dis.7]:

Proof. We are given $\bar{f} \perp \bar{g}$ and $\hat{h} \perp \hat{k}$. This means

$$\text{dom } f \cap \text{dom } g = \emptyset \text{ and } \text{range } h \cap \text{range } k = \emptyset.$$

Note that in general for partial injective functions m and n we have $\text{dom } mn \subseteq \text{dom } m$ and that $\text{range } mn \subseteq \text{range } n$. Hence we have

$$\begin{aligned} \text{dom } fh \cap \text{dom } gk &\subseteq \text{dom } f \cap \text{dom } g = \emptyset \\ \text{range } fh \cap \text{range } gk &\subseteq \text{range } h \cap \text{range } k = \emptyset. \end{aligned}$$

Therefore, $fh \perp gk$. □

We may define a different disjointness relation, \perp' , on the same restriction category. Define $f \perp' g$ if and only if one of f or g is the restriction 0, \emptyset . As $0 = \bar{0} = \hat{0} = h0 = 0k$, all of the seven disjointness axioms are easily verifiable.

Although disjointness is additional structure on a restriction category, one can use the disjointness structure of a base category (or categories) to define a disjointness structure on derived categories, such as the product category.

Lemma 3.4.10. *If \mathbb{X} and \mathbb{Y} are inverse categories with restriction zeros and respective disjointness relations \perp and \perp' , then we may construct a disjointness relation \perp_{\times} on $\mathbb{X} \times \mathbb{Y}$.*

Proof. Recall that product categories are defined component-wise. These definitions extend to the restriction, the inverse and the restriction zero. That is:

- If (f, g) is a map in $\mathbb{X} \times \mathbb{Y}$, then $(f, g)^{(-1)} = (f^{(-1)}, g^{(-1)})$;
- If (f, g) is a map in $\mathbb{X} \times \mathbb{Y}$, then $\overline{(f, g)} = (\bar{f}, \bar{g})$;

- The map $(0_X, 0_Y)$ is the restriction zero in $\mathbb{X} \times \mathbb{Y}$.

Following this pattern, for (f, g) and (h, k) maps in $\mathbb{X} \times \mathbb{Y}$, $(f, g) \perp_{\times} (h, k)$ iff $f \perp h$ and $g \perp' k$.

Verifying the disjointness axioms is straightforward, we show axioms 2 and 5. Proofs of the others are similar.

[Dis.2] : Given $(f, g) \perp_{\times} (h, k)$, we have $\overline{(f, g)}(h, k) = (\overline{f}, \overline{g})(h, k) = (\overline{f}h, \overline{g}k) = (0, 0) = 0$.

[Dis.5] : We are given $(f, g) \perp_{\times} (h, k)$. Consider the map $z = (x, y)$ in $\mathbb{X} \times \mathbb{Y}$. We know that $xf \perp xh$ and $yg \perp yk$, therefore we have $z(f, g) = (xf, yg) \perp_{\times} (xh, yk) = z(h, k)$.

□

3.4.2 Disjoint joins

We now consider additional structure on the inverse category, dependant upon the disjointness relation.

Definition 3.4.11. An *inverse category with disjoint joins* is an inverse category \mathbb{X} , with a restriction 0, a disjointness relation \perp and a binary operator on disjointness parallel maps:

$$\frac{f : A \rightarrow B, \ g : A \rightarrow B, \ f \perp g}{f \sqcup g : A \rightarrow B}$$

where the following hold:

$$\textbf{[DJ.1]} \quad f \leq f \sqcup g \text{ and } g \leq f \sqcup g;$$

$$\textbf{[DJ.2]} \quad f \leq h, \ g \leq h \text{ and } f \perp g \text{ implies } f \sqcup g \leq h;$$

$$\textbf{[DJ.3]} \quad h(f \sqcup g) = hf \sqcup hg. \text{ (Stable)}$$

$$\textbf{[DJ.4]} \quad \perp [f, g, h] \text{ if and only if } f \perp (g \sqcup h).$$

The binary operator, \sqcup , is referred to as the *disjoint join*.

Note that [DJ.1] with [DJ.2] immediately gives us that there is only one disjoint join given a specific disjointness relation.

Lemma 3.4.12. *Suppose \mathbb{X} in an inverse category with disjoint joins, with the join \sqcup and that it has a second disjoint join, \square . Then $f \sqcup g = f \square g$ for all maps f, g in \mathbb{X} .*

Proof. The first axiom tells us:

$$f, g \leq f \sqcup g \text{ and } f, g \leq f \square g.$$

Using the second axiom, we may therefore conclude $f \sqcup g \leq f \square g$ and $f \square g \leq f \sqcup g$, hence $f \sqcup g = f \square g$. \square

Lemma 3.4.13. *In an inverse category with disjoint joins, the disjoint join respects the restriction and is universal. Additionally, it is a partial associative and commutative operation, with identity 0. That is, the following hold:*

- (i) $\overline{f \sqcup g} = \overline{f} \sqcup \overline{g}$;
- (ii) $(f \sqcup g)k = fk \sqcup gk$ (Universal);
- (iii) $f \perp g, g \perp h, f \perp h$ implies that $(f \sqcup g) \sqcup h = f \sqcup (g \sqcup h)$;
- (iv) $f \perp g$ implies $f \sqcup g = g \sqcup f$;
- (v) $f \sqcup 0 = f$.

Proof.

- (i) As $\overline{f}, \overline{g} \leq \overline{f \sqcup g}$, we immediately have $\overline{f} \sqcup \overline{g} \leq \overline{f \sqcup g}$. To show the other direction, consider

$$\begin{aligned} \overline{f}(\overline{f \sqcup g})(f \sqcup g) &= (\overline{f} \overline{f} \sqcup \overline{f} \overline{g})(f \sqcup g) && \text{[DJ.3]} \\ &= \overline{f}(f \sqcup g) && \text{Lemma 2.2.3, [Dis.2]} \\ &= f. \end{aligned}$$

Hence, we have $f \leq (\bar{f} \sqcup \bar{g})(f \sqcup g)$ and similarly, so is g . By [DJ.2] and that $\bar{f} \sqcup \bar{g}$ is a restriction idempotent, we then have

$$f \sqcup g \leq (\bar{f} \sqcup \bar{g})(f \sqcup g) \leq f \sqcup g$$

and therefore $f \sqcup g = (\bar{f} \sqcup \bar{g})(f \sqcup g)$. By Lemma 2.2.4, $\overline{f \sqcup g} \leq \bar{f} \sqcup \bar{g}$ and so $\overline{f \sqcup g} = \bar{f} \sqcup \bar{g}$.

- (ii) First consider when f, g and k are restriction idempotents, say e_0, e_1 and e_2 . Then, we have $(e_0 \sqcup e_1)e_2 = e_2(e_0 \sqcup e_1) = e_2e_0 \sqcup e_2e_1 = e_0e_2 \sqcup e_1e_2$. Next, note that for general f, g, h , we have $fk \sqcup gk \leq (f \sqcup g)k$ as both $fk, gk \leq (f \sqcup g)k$. By Lemma 2.2.4, we need only show that their restrictions are equal:

$$\begin{aligned}
\overline{(f \sqcup g)k} &= \overline{\bar{f} \sqcup \bar{g}(f \sqcup g)k} & [\text{R.1}] \\
&= \overline{\bar{f} \sqcup \bar{g}(f \sqcup g)k} & [\text{R.3}] \\
&= (\bar{f} \sqcup \bar{g})\overline{(f \sqcup g)k} & \text{previous item} \\
&= \bar{f}\overline{(f \sqcup g)k} \sqcup \bar{g}\overline{(f \sqcup g)k} & \text{idempotent universal} \\
&= \overline{\bar{f}(f \sqcup g)k} \sqcup \overline{\bar{g}(f \sqcup g)k} & [\text{R.3}] \\
&= \overline{fk} \sqcup \overline{gk} \\
&= \overline{fk \sqcup gk}.
\end{aligned}$$

Therefore, as the restrictions are equal, we have shown $(f \sqcup g)k = fk \sqcup gk$.

- (iii) *Associativity*: Note that [DJ.4] shows that both sides of the equation exist. To show they are equal, we show that they are less than or equal to each other. From the definitions, we know that $f \sqcup g, h \leq (f \sqcup g) \sqcup h$, which also means $f, g \leq (f \sqcup g) \sqcup h$. Similarly, $g \sqcup h \leq (f \sqcup g) \sqcup h$ and then $f \sqcup (g \sqcup h) \leq (f \sqcup g) \sqcup h$. Conversely, $f, g, h \leq f \sqcup (g \sqcup h)$ and therefore $(f \sqcup g) \sqcup h \leq f \sqcup (g \sqcup h)$ and both sides are equal.

- (iv) *Commutativity*: Note first that both f and g are less than or equal to both $f \sqcup g$ and $g \sqcup f$, by [DJ.1]. By [DJ.2], we have $f \sqcup g \leq g \sqcup f$ and $g \sqcup f \leq f \sqcup g$ and we may conclude $f \sqcup g = g \sqcup f$.
- (v) *Identity*: By [DJ.1], $f \leq f \sqcup 0$. As $0 \leq f$ and $f \leq f$, by [DJ.2], $f \sqcup 0 \leq f$ and we have $f = f \sqcup 0$.

□

Note that the previous lemma and proof of associativity allows a simple inductive argument which shows that having binary disjoint joins extends unambiguously to disjoint joins of an arbitrary finite collection of disjoint maps.

We will write $[f_i]$ to signify a list of maps, where each $f_i : A \rightarrow B$. For disjointness, $\perp [f_i]$ will mean that $f_j \perp f_k$ where $j \neq k$ and $f_j, f_k \in [f_i]$. Finally, $\sqcup[f_i]$ will mean the disjoint join of all maps f_i , i.e., $f_1 \sqcup f_2 \sqcup \cdots \sqcup f_n$.

Lemma 3.4.14. *In an inverse category with disjoint joins, $\perp [f_i]$ if and only if $\sqcup[f_i]$ is defined unambiguously.*

Proof. Using [Dj.4], proceed as in the proof of Lemma 3.4.13[(iii)], inducting on n . □

Lemma 3.4.15. *Given \mathbb{X} is an inverse category with a disjoint join, then if $f_i, g_j : A \rightarrow B$ and $\perp [f_i]$ and $\perp [g_j]$, then $\sqcup[f_i] \perp \sqcup[g_j]$ if and only if $f_i \perp g_j$ for all i, j ;*

Proof. Assume $\sqcup[f_i] \perp \sqcup[g_j]$. Then by [Dj.4] and associativity, we have $\sqcup[f_i] \perp g_j$ for each j . Then, applying the reflexivity of \perp , [Dj.4] and associativity, we have $f_i \perp g_j$ for each i and j .

Assume $f_i \perp g_j$ for each i and j . Then by [Dj.4] and associativity, $f_i \perp \sqcup[g_j]$ for each i . Applying [Dj.4] again, we have $\sqcup[f_i] \perp \sqcup[g_j]$. □

Following the same method as in the previous section, we show that the product of two inverse categories with disjoint joins has a disjoint join.

Lemma 3.4.16. *Given \mathbb{X}, \mathbb{Y} are inverse categories with disjoint joins, \sqcup and \sqcup' respectively, then the category $\mathbb{X} \times \mathbb{Y}$ is an inverse category with disjoint joins.*

Proof. From Lemma 3.4.10, we know $\mathbb{X} \times \mathbb{Y}$ has a disjointness relation that is defined point-wise. We therefore define \sqcup_{\times} the disjoint join on $\mathbb{X} \times \mathbb{Y}$ by

$$(f, g) \sqcup_{\times} (h, k) = (f \sqcup h, g \sqcup' k) \quad (3.11)$$

We now prove each of the axioms in Definition 3.4.11 hold.

[DJ.1] From Equation (equation (3.11)), we see that since $f, h \leq f \sqcup h$ and $g, k \leq g \sqcup' k$, we have $(f, g) \leq (f, g) \sqcup_{\times} (h, k)$ and $(h, k) \leq (f, g) \sqcup_{\times} (h, k)$.

[DJ.2] Suppose $(f, g) \leq (x, y)$, $(h, k) \leq (x, y)$ and $(f, g) \perp_{\times} (h, k)$. Then regarding it point-wise, we have $(f, g) \sqcup_{\times} (h, k) = (f \sqcup h, g \sqcup' k) \leq (x, y)$.

[DJ.3] $(x, y) ((f, g) \sqcup_{\times} (h, k)) = (x(f \sqcup h), y(g \sqcup' k)) = (xf \sqcup xh, yg \sqcup' yk) = (xf, yg) \sqcup_{\times} (xh, yk) = ((x, y)(f, g)) \sqcup_{\times} ((x, y)(h, k))$.

[DJ.4] Given $\perp_{\times} [(f, g), (h, k), (x, y)]$, we know $f \perp (h \sqcup x)$ and $g \perp' (k \sqcup' y)$. Hence, $(f, g) \perp_{\times} ((h, k) \sqcup_{\times} (x, y))$. The opposite direction is similar.

□

3.4.3 Monoidal Tensors for disjointness

Suppose we are given a monoidal tensor \oplus on \mathbb{X} , an inverse category with a restriction zero. Under certain conditions, it is possible to define disjointness based upon the action of the tensor. Note that throughout, we are assuming the following naming for the standard

monoidal tensor isomorphisms.

$$u_{\oplus}^l : 0 \oplus A \rightarrow A$$

$$u_{\oplus}^r : A \oplus 0 \rightarrow A$$

$$a_{\oplus} : (A \oplus B) \oplus C \rightarrow A \oplus (B \oplus C)$$

$$c_{\oplus} : A \oplus B \rightarrow B \oplus A.$$

Note we also require the tensor isomorphisms above be natural.

Definition 3.4.17. Suppose we are given an inverse category \mathbb{X} with restriction zero and a symmetric monoidal tensor \oplus . Then \oplus is a *disjointness tensor* when:

- It is a restriction functor — i.e., $-\oplus- : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{X}$.
- The unit is the restriction zero. ($0 : \mathbf{1} \rightarrow \mathbb{X}$ picks out the restriction zero in \mathbb{X}).
- Define $\Pi_0 := (1 \oplus 0)u_{\oplus}^r : A \oplus B \rightarrow A$ and $\Pi_1 := (0 \oplus 1)u_{\oplus}^l : A \oplus B \rightarrow B$. Then Π_0 and Π_1 are jointly monic. That is, whenever $f\Pi_0 = g\Pi_0$ and $f\Pi_1 = g\Pi_1$ then $f = g$.
- Define $\Pi_0 = u_{\oplus}^r{}^{(-1)}(1 \oplus 0) : A \rightarrow A \oplus A$ and $\Pi_1 = u_{\oplus}^l{}^{(-1)}(0 \oplus 1) : A \rightarrow A \oplus A$. Then Π_0 and Π_1 are jointly epic. That is, if $\Pi_0 f = \Pi_0 g$ and $\Pi_1 f = \Pi_1 g$, then $f = g$.

Lemma 3.4.18. *Given an inverse category \mathbb{X} with restriction zero and disjointness tensor \oplus , then the map $0 \oplus 0 : A \oplus B \rightarrow C \oplus D$ is the map $0 : A \oplus B \rightarrow C \oplus D$.*

Proof. Recall the zero map factors through the restriction zero, i.e. $0 : A \rightarrow B$ is the same as saying $A \xrightarrow{!} 0 \xrightarrow{?} B$. Additionally, as objects, $0 \oplus 0 \cong 0$ — the restriction zero.

Therefore the map $0 \oplus 0 : A \oplus B \rightarrow C \oplus D$ is writable as

$$A \oplus B \xrightarrow{!\oplus!} 0 \oplus 0 \xrightarrow{?\oplus?} C \oplus D,$$

which may then be rewritten as

$$A \oplus B \xrightarrow{! \oplus !} 0 \oplus 0 \xrightarrow{u_{\oplus}^l} 0 \xrightarrow{u_{\oplus}^{l(-1)}} 0 \oplus 0 \xrightarrow{? \oplus ?} C \oplus D.$$

But by the properties of the restriction zero, $(! \oplus !)u_{\oplus}^l = !$ and $u_{\oplus}^{l(-1)}(? \oplus ?) = ?$ and therefore the map $0 \oplus 0 : A \oplus B \rightarrow C \oplus D$ is the same as the map $0 : A \oplus B \rightarrow C \oplus D$. \square

Lemma 3.4.19. *Given an inverse category \mathbb{X} with a restriction zero and a disjointness tensor, the map Π_0 is natural in the left component and Π_1 is natural in the right, up to isomorphism. This means:*

$$\Pi_0(f \oplus g) = f \Pi_0 \quad \text{and} \quad \Pi_1(f \oplus g) = g \Pi_1.$$

Proof. For the left and right naturality, we see:

$$\Pi_0(f \oplus g) = u_{\oplus}^{r(-1)}(1 \oplus 0)(f \oplus g) = u_{\oplus}^{r(-1)}(f \oplus 0) = f u_{\oplus}^{r(-1)}(1 \oplus 0) = f \Pi_0,$$

and

$$\Pi_1(f \oplus g) = u_{\oplus}^{l(-1)}(0 \oplus 1)(f \oplus g) = u_{\oplus}^{l(-1)}(0 \oplus g) = g u_{\oplus}^{l(-1)}(0 \oplus 1) = g \Pi_1.$$

\square

Lemma 3.4.20. *Given an inverse category \mathbb{X} with restriction zero and disjointness tensor \oplus , then the following hold:*

1. $\Pi_i \Pi_i = \overline{\Pi_i}$ and $\Pi_i \Pi_i = \overline{\Pi_i} = 1$;
2. $\overline{\Pi_0} \Pi_1 = 0$ and $\overline{\Pi_1} \Pi_0 = 0$;
3. $\Pi_1 \Pi_0 = 0$, $\Pi_1 \overline{\Pi_0} = 0$, $\Pi_0 \Pi_1 = 0$ and $\Pi_0 \overline{\Pi_1} = 0$;
4. the maps Π_0 and Π_1 are monic.

Proof. For item 1, recalling that the restriction zero is its own partial inverse, we see from their definitions that $\Pi_i = \Pi_i^{(-1)}$. Calculating the restriction of Π_0 ,

$$\Pi_0 \Pi_0 = u_{\oplus}^{r(-1)}(1 \oplus 0)(1 \oplus 0)u_{\oplus}^r = (u_{\oplus}^{r(-1)}(1 \oplus 0))u_{\oplus}^r = 1 u_{\oplus}^{r(-1)}u_{\oplus}^r = 1.$$

The calculation for Π_1 and Π_1 are analogous.

For the second item,

$$\overline{\Pi_0}\Pi_1 = \overline{(1 \oplus 0)u_{\oplus}^r}(0 \oplus 1)u_{\oplus}^l = \overline{1 \oplus 0}(0 \oplus 1)u_{\oplus}^l = (1 \oplus 0)(0 \oplus 1)u_{\oplus}^l = (0 \oplus 0)u_{\oplus}^l = 0,$$

and

$$\overline{\Pi_1}\Pi_0 = \overline{(0 \oplus 1)u_{\oplus}^l}(1 \oplus 0)u_{\oplus}^r = (0 \oplus 1)(1 \oplus 0)u_{\oplus}^r = (0 \oplus 0)u_{\oplus}^r = 0.$$

For the third item, we see

$$\Pi_0\Pi_1 = (u_{\oplus}^r)^{(-1)}(1 \oplus 0)(0 \oplus 1)u_{\oplus}^l = u_{\oplus}^r{}^{(-1)}(0 \oplus 0)u_{\oplus}^l = 0$$

and

$$\Pi_1\Pi_0 = (u_{\oplus}^l)^{(-1)}(0 \oplus 1)(1 \oplus 0)u_{\oplus}^r = u_{\oplus}^l{}^{(-1)}(0 \oplus 0)u_{\oplus}^r = 0.$$

As $\overline{\Pi_0} = 1 \oplus 0$ and $\overline{\Pi_1} = 0 \oplus 1$, we see the other two identities hold as well.

To prove Π_0 is monic, suppose $f\Pi_0 = g\Pi_0$. Therefore we must have

$$f = f(\Pi_0\Pi_0) = (f\Pi_0)\Pi_0 = (g\Pi_0)\Pi_0 = g(\Pi_0\Pi_0) = g.$$

The proof that Π_1 is monic is similar. □

Corollary 3.4.21. *In an inverse category \mathbb{X} with a restriction zero and disjointness tensor, the following hold:*

$$\begin{array}{ll} (i) \ \Pi_0 f \oplus g \Pi_0 = f; & (iii) \ \Pi_1 f \oplus g \Pi_0 = 0; \\ (ii) \ \Pi_0 f \oplus g \Pi_1 = 0; & (iv) \ \Pi_1 f \oplus g \Pi_1 = g. \end{array}$$

Additionally, if t is a map such that for $i \in \{0, 1\}$,

$$\Pi_i t \Pi_j = \begin{cases} t_i & : \ i \neq j \\ 0 & : \ i = j, \end{cases}$$

then $t = t_0 \oplus t_1$

Proof. The calculations for $f \oplus g$ follow from Lemma 3.4.19 and Lemma 3.4.20. For example, $\Pi_0 f \oplus g \Pi_0 = f \Pi_0 \Pi_0 = f$.

For the second claim, note that we have $\Pi_0(t\Pi_0) = t_0 = \Pi_0(t_0 \oplus t_1)\Pi_0$ and $\Pi_1(t\Pi_0) = 0 = \Pi_1(t_0 \oplus t_1)\Pi_0$, hence $t\Pi_0 = (t_0 \oplus t_1)\Pi_0$. Similarly, we see $t\Pi_1 = (t_0 \oplus t_1)\Pi_1$ and therefore $t = t_0 \oplus t_1 + 1$. \square

Definition 3.4.22. In an inverse category \mathbb{X} with a restriction zero and disjointness tensor, we define two partial operations on pairs of arrows in \mathbb{X} to another arrow in \mathbb{X} . First, for arrows $f : A \rightarrow B$ and $g : A \rightarrow C$, we define $f \nabla g$ as being the map that makes diagram (equation (3.12)) below commute, when it exists.

$$\begin{array}{ccccc}
 & & B & \xleftarrow{\Pi_0} & B \oplus C & \xrightarrow{\Pi_1} & C \\
 & & \swarrow g & & \uparrow f \nabla g & & \searrow f \\
 & & A & & & &
 \end{array} \tag{3.12}$$

Then for $h : B \rightarrow A$, $k : C \rightarrow A$, $h \triangle k$ is that map that makes diagram (equation (3.13)) commute, if it exists.

$$\begin{array}{ccccc}
 & & A & & \\
 & \swarrow h & \uparrow h \triangle k & \nwarrow k & \\
 B & \xrightarrow{\Pi_0} & B \oplus C & \xleftarrow{\Pi_1} & C
 \end{array} \tag{3.13}$$

Due to Π_0 and Π_1 being jointly monic, $f \nabla g$ is unique when it exists. Similarly, as Π_0 and Π_1 are jointly epic, $f \triangle g$ is unique when it exists.

We give a lemma exploring the behaviour of the two operations: ∇ and \triangle .

Lemma 3.4.23. *Given \mathbb{X} is an inverse category with a restriction zero and a disjointness tensor \oplus then the following relations hold for ∇ and \triangle :*

- (i) *If $f \nabla g$ exists, then $g \nabla f$ exists. If $f \triangle g$ exists, then $g \triangle f$ exists.*
- (ii) *$f \nabla 0$ and $f \triangle 0$ always exist.*

- (iii) When $f \nabla g$ exists, $\bar{f}(f \nabla g) = f \nabla 0$, $\bar{f}g = 0$, $\bar{g}(f \nabla g) = 0 \nabla g$ and $\bar{g}f = 0$.
- (iv) Dually to the previous item, when $f \triangle g$ exists, $(f \triangle g)\hat{f} = f \triangle 0$, $g\hat{f} = 0$, $(f \triangle g)\hat{g} = 0 \triangle g$ and $f\hat{g} = 0$.
- (v) When $f \nabla g$ exists, $f \nabla g(h \oplus k) = fh \nabla gk$.
- (vi) Dually, when $f \triangle g$ exists, $(h \oplus k)f \triangle g = hf \triangle kg$.
- (vii) When $f \nabla g$ exists, then $h(f \nabla g) = hf \nabla hg$ and when $f \triangle g$ exists, $(f \triangle g)h = fh \triangle gh$.
- (viii) If $\bar{f} \nabla \bar{g}$ exists, then $\bar{f} \triangle \bar{g}$ exists and is the partial inverse of $\bar{f} \nabla \bar{g}$.
- (ix) If $f \nabla g$ exists and $f' \leq f$, $g' \leq g$, then $f' \nabla g'$ exists.
- (x) When $f \triangle g$ exists, $(f \triangle g)(f \triangle g)^{(-1)} = \bar{f} \oplus \bar{g}$.
- (xi) Given $f \nabla g$ and $h \nabla k$ exist, then $(f \oplus h) \nabla (g \oplus k) = (f \nabla g) \oplus (h \nabla k)$. Dually, the existence of $f \triangle g$ and $h \triangle k$ implies $(f \oplus h) \triangle (g \oplus k) = (f \triangle g) \oplus (h \triangle k)$.

Proof.

- (i) $g \nabla f = (f \nabla g)c_{\oplus}$ and $g \triangle f = c_{\oplus}(f \triangle g)$.
- (ii) Consider $f\Pi_0$. Then $f\Pi_0\Pi_0 = f$ and $f\Pi_0\Pi_1 = f0 = 0$. Hence, $f\Pi_0 = f \nabla 0$.
Consider $\Pi_0 f$. Then $\Pi_0\Pi_0 f = f$ and $\Pi_1\Pi_0 f = 0f = 0$ and therefore $\Pi_0 f = (f \triangle 0)$.
- (iii) Using Lemma 3.4.20

$$\bar{f}g = \overline{(f \nabla g)\Pi_0}(f \nabla g)\Pi_1 = (f \nabla g)\bar{\Pi_0}\Pi_1 = 0.$$

Similarly, $\bar{g}f = f \nabla g\bar{\Pi_1}\Pi_0 = 0$.

Recall that Π_0 and Π_1 are jointly monic. We have $\bar{f}(f \nabla g)\Pi_0 = \bar{f}f = f = (f \nabla 0)\Pi_0$ and $\bar{f}(f \nabla g)\Pi_1 = \bar{f}g = 0 = (f \nabla 0)\Pi_1$. Therefore, $\bar{f}(f \nabla g) = f \nabla 0$. Similarly, $\bar{g}(f \nabla g) = 0 \nabla g$.

(iv) Using Lemma 3.4.20

$$\begin{aligned}
gf &= \Pi_1(f \triangle g)(\widehat{\Pi_0(f \triangle g)}) = \Pi_1(f \triangle g)\overline{(f \triangle g)^{(-1)}}\Pi_0 \\
&= \Pi_1(f \triangle g)\overline{(f \triangle g)^{(-1)}}\overline{\Pi_0} = \overline{\Pi_1(f \triangle g)\Pi_0}\Pi_1(f \triangle g) = \\
&\quad \overline{\Pi_1\Pi_0(f \triangle g)}\Pi_1(f \triangle g) = \overline{0}\Pi_1(f \triangle g) = 0
\end{aligned}$$

Similarly, $f\hat{g} = 0$.

Recall that Π_0 and Π_1 are jointly epic. We have $\Pi_0(f \triangle g)\hat{f} = f\hat{f} = f = \Pi_0(f \triangle 0)$ and $\Pi_1(f \triangle g)\hat{f} = g\hat{f} = 0 = \Pi_1(f \triangle 0)$. Therefore, $(f \triangle g)\hat{f} = f \triangle 0$. Similarly, $(f \triangle g)\hat{g} = 0 \triangle g$.

(v) Calculating, we have

$$f \nabla g(h \oplus k)\Pi_0 = f \nabla g\Pi_0h = fh \text{ and } f \nabla g(h \oplus k)\Pi_1 = f \nabla g\Pi_1k = gk,$$

but this means that $f \nabla g(h \oplus k) = fh \nabla gk$ by the joint monic property of Π_0 , Π_1 .

(vi) The proof for this is dual to the previous item, and depends on the joint epic property of Π_0 and Π_1 .

(vii) We are given $f \nabla g$ exists, therefore $f = (f \nabla g)\Pi_0$ and $g = (f \nabla g)\Pi_1$. But this means $hf = h(f \nabla g)\Pi_0$ and $hg = h(f \nabla g)\Pi_1$, from which we may conclude $hf \nabla hg = h(f \nabla g)$ by the fact that Π_0 and Π_1 are jointly monic. The proof of $(f \triangle g)h = fh \triangle gh$ is similar.

(viii) We are given $\bar{f} = \bar{f} \nabla \bar{g}\Pi_0$. Therefore,

$$\bar{f} = \bar{f}^{(-1)} = \Pi_0^{(-1)}(\bar{f} \nabla \bar{g})^{(-1)} = \Pi_0(\bar{f} \nabla \bar{g})^{(-1)}.$$

Similarly, $\bar{g} = \Pi_1(\bar{f} \nabla \bar{g})^{(-1)}$. But this means $(\bar{f} \nabla \bar{g})^{(-1)} = \bar{f} \triangle \bar{g}$.

(ix) Note that from item (v), we know that $f \nabla g = \bar{f} \nabla \bar{g}(f \oplus g)$. We are given $f' \leq f$ and $g' \leq g$. This gives us $\bar{f}'f = f'$, $\bar{g}'g = g'$, $\bar{f}'\bar{f} = \bar{f}'$ and $\bar{g}'\bar{g} = \bar{g}'$.

Consider the map $\bar{f} \nabla \bar{g}(\bar{f}' \oplus \bar{g}')(f \oplus g)$. Calculating, we see

$$\begin{aligned}
\bar{f} \nabla \bar{g}(\bar{f}' \oplus \bar{g}')(f \oplus g) &= \bar{f} \nabla \bar{g}(\bar{f}' \oplus \bar{g}')(\bar{f}' \oplus \bar{g}')(f \oplus g) \\
&= \bar{f} \nabla \bar{g}(\bar{f}' \oplus \bar{g}')(f' \oplus g') \\
&= \bar{f} \bar{f}' \nabla \bar{g} \bar{g}'(f' \oplus g') \\
&= \bar{f}' \bar{f} \nabla \bar{g}' \bar{g}(f' \oplus g') \\
&= \bar{f}' \nabla \bar{g}'(f' \oplus g') \\
&= f' \nabla g'.
\end{aligned}$$

(x) From our diagram for Δ , we know that $f^{(-1)} = (f \Delta g)^{(-1)} \Pi_0$ and $g^{(-1)} = (f \Delta g)^{(-1)} \Pi_1$. As well, we know that $\Pi_0(f \Delta g) = f$ and $\Pi_1(f \Delta g) = g$. Therefore, we have:

$$\Pi_0(f \Delta g)(f \Delta g)^{(-1)} \Pi_0 = \bar{f} \text{ and } \Pi_1(f \Delta g)(f \Delta g)^{(-1)} \Pi_1 = \bar{g}.$$

As $f \perp_{\oplus} g$, we know that $f g^{(-1)} = f \hat{g} g^{(-1)} = 0 g^{(-1)} = 0$ and therefore,

$$\Pi_0(f \Delta g)(f \Delta g)^{(-1)} \Pi_1 = 0 \text{ and } \Pi_1(f \Delta g)(f \Delta g)^{(-1)} \Pi_0 = 0.$$

By Corollary 3.4.21 this means $(f \Delta g)(f \Delta g)^{(-1)} = \bar{f} \oplus \bar{g}$.

(xi) As $(f \nabla g) \oplus (h \nabla k) \Pi_0 = (f \nabla g)$ and $(f \nabla g) \oplus (h \nabla k) \Pi_1 = (h \nabla k)$, we see that $(f \nabla g) \oplus (h \nabla k)$ satisfies the diagram for $(f \oplus h) \nabla (g \oplus k)$. Dually, as $\Pi_0(f \Delta g) \oplus (h \Delta k) = (f \Delta g)$ and $\Pi_1(f \Delta g) \oplus (h \Delta k) = (h \Delta k)$, $(f \Delta g) \oplus (h \Delta k)$ satisfies the diagram for $(f \oplus h) \Delta (g \oplus k)$.

□

Definition 3.4.24. Define $f \perp_{\oplus} g$ when $f, g : A \rightarrow B$ and both $f \nabla g$ and $f \Delta g$.

Lemma 3.4.25. If \mathbb{X} is an inverse category with a restriction zero and a disjointness tensor \oplus then the relation \perp_{\oplus} is a disjointness relation.

Proof. We need to show that \perp_{\oplus} satisfies the disjointness axioms. We will use [Dis.6'] in place of [Dis.6] and [Dis.7] as discussed in Lemma 3.4.2.

[Dis.1] We must show $f \perp_{\oplus} 0$. This follows immediately from Lemma 3.4.23, item (ii).

[Dis.2] Show $f \perp_{\oplus} g$ implies $\bar{f}g = 0$. This is a direct consequence of Lemma 3.4.23, item (iii).

[Dis.3] We require $f \perp_{\oplus} g, f' \leq f, g' \leq g$ implies $f' \perp_{\oplus} g'$. From Lemma 3.4.23, item (ix), we immediately have $f' \nabla g'$ exists. Using a similar argument to the proof of this item, we also have $f' \Delta g'$ exists and hence $f' \perp_{\oplus} g'$.

[Dis.4] Commutativity of \perp_{\oplus} follows from the symmetry of the two required diagrams, see Lemma 3.4.23, item (i).

[Dis.5] Show that if $f \perp_{\oplus} g$ then $hf \perp_{\oplus} hg$ for any map h . By Lemma 3.4.23, item (vii), we know that $hf \nabla hg$ exists. By item (vi), $(hf) \Delta (hg) = (h \oplus h)(f \Delta g)$ and therefore $hf \perp_{\oplus} hg$.

[Dis.6'] We need to show $f \perp_{\oplus} g$ if and only if $\bar{f} \perp_{\oplus} \bar{g}$ and $\hat{f} \perp_{\oplus} \hat{g}$. This follows directly from Lemma 3.4.23, items (v) and (vi), which give us $f \nabla g = \bar{f} \nabla \bar{g}(f \oplus g)$ and $f \Delta g = (f \oplus g)\hat{f} \Delta \hat{g}$, where the equalities hold if either side of the equation exists.

□

The operations ∇ and Δ are sufficient to define a disjointness relation on an inverse category. However, when we wish to extend this to a disjoint join, we run into problems when trying to prove [DJ.4]. Specifically, there is not enough information to show that $\perp_{\oplus}[f, g, h]$ implies $f \perp_{\oplus} (g \sqcup_{\oplus} h)$.

Therefore, we add one more assumption regarding our tensor in order to define disjointness.

Definition 3.4.26. Let \mathbb{X} be an inverse category with a disjointness tensor \oplus and a restriction zero. Consider diagrams [equation \(3.14\)](#) and [equation \(3.15\)](#).

$$(3.14)$$

$$\begin{array}{ccc} A & \xrightarrow{g} & X \oplus Z \\ \alpha \swarrow & & \downarrow \Pi_0 \\ X \oplus Y \oplus Z & \xrightarrow{\Pi_{0,2}} & X \oplus Z \\ f \searrow & & \downarrow \Pi_0 \\ X \oplus Y & \xrightarrow{\Pi_0} & X \end{array}$$

$$(3.15)$$

$$\begin{array}{ccc} X \oplus Y & \xrightarrow{\Pi_{0,1}} & X \oplus Y \oplus Z \\ \Pi_0 \uparrow & & \uparrow \Pi_{0,2} \\ X & \xrightarrow{\Pi_0} & X \oplus Y \\ h \nearrow & & \downarrow \beta \\ & & A \end{array}$$

Then \oplus is a *disjoint sum tensor* when the following two conditions hold:

- α exists if and only if $f\Pi_1 \nabla g\Pi_1$ exists;
- β exists if and only if $\Pi_1 h \Delta \Pi_1 k$ exists.

Lemma 3.4.27. Let \mathbb{X} be an inverse category with a disjoint sum tensor as in Definition [3.4.26](#) and we are given $f, g, h : A \rightarrow B$ with $\perp_{\oplus}[f, g, h]$. Then both $f \nabla (g \nabla h)$ and $f \Delta (g \Delta h)$ exist.

Proof. As all the maps are disjoint, we know that each pair's ∇ map and Δ maps exist.

Consider the diagram

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ \alpha \swarrow & & \uparrow \Pi_1 \\ B \oplus B \oplus B & \xrightarrow{\Pi_{0,2}} & B \oplus B \\ g \nabla h \searrow & & \downarrow \Pi_0 \\ B \oplus B & \xrightarrow{\Pi_0} & B \end{array}$$

where we claim $\alpha = (g \nabla h) \nabla f$.

The lower part of the diagram commutes as it fulfills the conditions of Definition [3.4.26](#).

The upper triangle of the diagram commutes by the definition of $g \nabla f$. Noting that $\Pi_{0,1} :$

$B \oplus B \oplus B \rightarrow B \oplus B$ is the same map as $\Pi_0 : (B \oplus B) \oplus B \rightarrow (B \oplus B)$ and $\Pi_{0,2}\Pi_1 : B \oplus B \oplus B \rightarrow B \oplus B \rightarrow B$ is the same map as $\Pi_1 : (B \oplus B) \oplus B \rightarrow B$, we see α does make the ∇ diagram for $g \nabla h$ and f commute. Therefore by Lemma 3.4.23, $f \nabla (g \nabla h)$ exists and is equal to $\alpha c_{\oplus\{01,2\}}$.

A dual diagram and reasoning shows $f \triangle (g \triangle h)$ exists. \square

Lemma 3.4.28. *In an inverse category \mathbb{X} with a disjoint sum tensor, when $\perp_{\oplus}[f, g, h]$, then:*

1. $f \nabla (g \nabla h) = ((f \nabla g) \nabla h)a_{\oplus}$ and both exist;
2. $f \triangle (g \triangle h) = ((f \triangle g) \triangle h)a_{\oplus}$ and both exist;

Proof. Consider the diagram

$$\begin{array}{ccccc}
 A & \xrightarrow{h} & B & & \\
 & \searrow^{f \nabla h} & \uparrow \Pi_1 & & \\
 & \searrow^{\alpha} & B \oplus B & \xrightarrow{\Pi_{0,2}} & B \oplus B \\
 & \searrow^{f \nabla g} & \downarrow \Pi_{0,1} & & \downarrow \Pi_0 \\
 & & B \oplus B & \xrightarrow{\Pi_0} & B
 \end{array}$$

which gives us $\alpha = (f \nabla g) \nabla h : A \rightarrow (B \oplus B) \oplus B$ and $\alpha a_{\oplus} : A \rightarrow B \oplus (B \oplus B)$. Next consider the diagram

$$\begin{array}{ccccc}
 A & \xrightarrow{f} & B & & \\
 & \searrow^{g \nabla f} & \uparrow \Pi_1 & & \\
 & \searrow^{\gamma} & B \oplus B & \xrightarrow{\Pi_{0,2}} & B \oplus B \\
 & \searrow^{g \nabla h} & \downarrow \Pi_{0,1} & & \downarrow \Pi_0 \\
 & & B \oplus B & \xrightarrow{\Pi_0} & B
 \end{array}$$

which gives us $\gamma c_{\oplus} = f \nabla (g \nabla h) : A \rightarrow B \oplus (B \oplus B)$.

Note from the diagrams we have:

$$\gamma c_{\oplus} \Pi_0 = f = \alpha a_{\oplus} \Pi_0$$

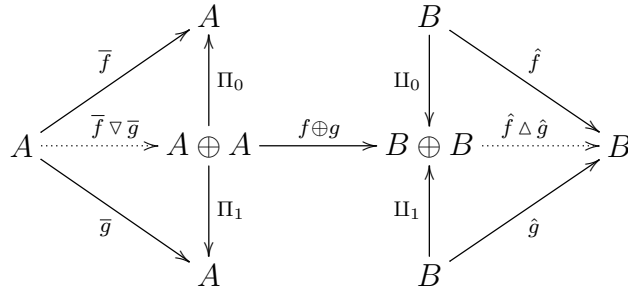
$$\gamma c_{\oplus} \Pi_1 \Pi_0 = g = \alpha a_{\oplus} \Pi_1 \Pi_0$$

$$\gamma c_{\oplus} \Pi_1 \Pi_1 = h = \alpha a_{\oplus} \Pi_1 \Pi_1.$$

Hence, by the assumption that Π_0, Π_1 are jointly monic, we have $\alpha = \gamma c_{\oplus} a_{\oplus}$ and hence $f \nabla (g \nabla h) = (f \nabla g) \nabla h$, up to the associativity isomorphism. \square

Definition 3.4.29. Let \mathbb{X} be an inverse category with a disjointness tensor and restriction zero. Assume we have two maps $f, g : A \rightarrow B$ with $f \perp_{\oplus} g$. Then define the map $f \sqcup_{\oplus} g = \bar{f} \nabla \bar{g} (f \oplus g) \hat{f} \Delta \hat{g}$.

For reference, the map $f \sqcup_{\oplus} g$ may be visualized as follows:



Using Lemma 3.4.23, we may rewrite this in a variety of equivalent ways:

$$\begin{aligned}
 f \sqcup_{\oplus} g &= \bar{f} \nabla \bar{g} (f \oplus g) \hat{f} \Delta \hat{g} \\
 &= f \nabla g \hat{f} \Delta \hat{g} \\
 &= \bar{f} \nabla \bar{g} f \Delta g \\
 &= f \nabla g (f^{(-1)} \oplus g^{(-1)}) f \Delta g
 \end{aligned}$$

In particular, note that $\bar{f} \sqcup_{\oplus} \bar{g} = (\bar{f} \nabla \bar{g})(\bar{f} \Delta \bar{g})$ as $\hat{g} = \bar{g}$.

Lemma 3.4.30. Let \mathbb{X} be an inverse category with a disjointness tensor and restriction zero.

Let \mathbb{X} have the maps $f, g : A \rightarrow B$ with $f \perp_{\oplus} g$. Then \sqcup_{\oplus} has the following properties.

(i) For all maps $h : A \rightarrow B$, $\bar{f} h \sqcup_{\oplus} \bar{g} h = (\bar{f} \sqcup_{\oplus} \bar{g}) h$.

(ii) $\bar{f} \sqcup_{\oplus} \bar{g} = \overline{f \sqcup_{\oplus} g}$.

Proof. (i) By Lemma 3.4.3, item (ii), we know that $\bar{f} h \perp_{\oplus} \bar{g} h$, hence we can form

$\bar{f} h \sqcup_{\oplus} \bar{g} h$. Also, noting that

$$\widehat{h f h} = \overline{h h^{(-1)} f} = \overline{h h^{(-1)} f h} = \overline{\bar{h} f h} = \bar{f} h h = \bar{f} h,$$

we may then calculate from the left hand side as follows:

$$\begin{aligned}
\overline{f}h \sqcup_{\oplus} \overline{g}h &= (\overline{f}h \nabla \overline{g}h)(\widehat{\overline{f}h} \Delta \widehat{\overline{g}h}) \\
&= (\overline{f} \nabla \overline{g})(h\widehat{\overline{f}h} \Delta h\widehat{\overline{g}h}) \\
&= (\overline{f} \nabla \overline{g})(\overline{f}h \Delta \overline{g}h) \\
&= (\overline{f} \nabla \overline{g})(\overline{f} \Delta \overline{g})h \\
&= (\overline{f} \sqcup_{\oplus} \overline{g})h.
\end{aligned}$$

(ii) Using Lemma 3.4.23, item (x), we can compute:

$$\begin{aligned}
\overline{f \sqcup_{\oplus} g} &= f \sqcup_{\oplus} g (f \sqcup_{\oplus} g)^{(-1)} \\
&= ((\overline{f} \nabla \overline{g})(f \Delta g)) \left((f \nabla g)^{(-1)} (\overline{f} \nabla \overline{g})^{(-1)} \right) \\
&= \overline{f} \nabla \overline{g} (f \nabla g) (f \nabla g)^{(-1)} \overline{f} \Delta \overline{g} \\
&= \overline{f} \nabla \overline{g} (\overline{f} \oplus \overline{g}) \overline{f} \Delta \overline{g} \\
&= \overline{f} \nabla \overline{g} \overline{f} \Delta \overline{g} \\
&= \overline{f} \sqcup_{\oplus} \overline{g}
\end{aligned}$$

□

Proposition 3.4.31. *Let \mathbb{X} be an inverse category with a disjoint sum tensor and restriction zero. Assume we have two maps f, g with $f \perp_{\oplus} g$. Then the map $f \sqcup_{\oplus} g$ from Definition 3.4.29 is a disjoint join.*

Proof. [DJ.1] We must show $f, g \leq f \sqcup_{\oplus} g$. Computing,

$$\begin{aligned}
\overline{f} (\overline{f} \nabla \overline{g}) f \triangle g &= (\overline{f} \nabla \overline{g}) \Pi_0 (\overline{f} \nabla \overline{g}) f \triangle g \\
&= \overline{(\overline{f} \nabla \overline{g}) \Pi_0 (\overline{f} \nabla \overline{g}) f \triangle g} \\
&= (\overline{f} \nabla \overline{g}) \overline{\Pi_0} f \triangle g \\
&= (\overline{f} \nabla \overline{g}) \Pi_0 \Pi_0 f \triangle g \\
&= ((\overline{f} \nabla \overline{g}) \Pi_0) (\Pi_0 (f \triangle g)) \\
&= \overline{f} f \\
&= f
\end{aligned}$$

we see $f \leq f \sqcup_{\oplus} g$. Showing $g \leq f \sqcup_{\oplus} g$ proceeds in the same manner.

[DJ.2] We must show that $f \leq h$, $g \leq h$ and $f \perp_{\oplus} g$ implies $f \sqcup_{\oplus} g \leq h$. First, note that

$$\begin{aligned}
\overline{f \sqcup_{\oplus} g} h &= \overline{\overline{f} h \sqcup_{\oplus} \overline{g} h} h \\
&= \overline{(\overline{f} \sqcup_{\oplus} \overline{g}) h} h \\
&= \overline{(\overline{f} \sqcup_{\oplus} \overline{g}) h} h \\
&= \overline{(\overline{f} \sqcup_{\oplus} \overline{g}) h (\overline{f} \sqcup_{\oplus} \overline{g}) h} \\
&= \overline{(\overline{f} \sqcup_{\oplus} \overline{g}) h (\overline{f} \sqcup_{\oplus} \overline{g}) h} \\
&= (\overline{f} \sqcup_{\oplus} \overline{g}) h \\
&= (\overline{f} h \sqcup_{\oplus} \overline{g} h) \\
&= (f \sqcup_{\oplus} g)
\end{aligned}$$

[DJ.3] We must show stability of \sqcup_{\oplus} , i.e., that $h(f \sqcup_{\oplus} g) = hf \sqcup_{\oplus} hg$.

$$\begin{aligned}
h(f \sqcup_{\oplus} g) &= h((\bar{f} \nabla \bar{g})(f \triangle g)) \\
&= (h\bar{f} \nabla h\bar{g})(f \triangle g) \\
&= (\overline{hf} \nabla \overline{hg})(f \triangle g) \\
&= (\overline{hf} \nabla \overline{hg})(h \oplus h)(f \triangle g) \\
&= (\overline{hf} \nabla \overline{hg})(hf \triangle hg) \\
&= hf \sqcup_{\oplus} hg
\end{aligned}$$

[DJ.4] We need to show $\perp_{\oplus}[f, g, h]$ if and only if $f \perp_{\oplus}(g \sqcup_{\oplus} h)$. For the right to left implication, note that the existence of $g \sqcup_{\oplus} h$ implies $g \perp_{\oplus} h$. We also know $g, h \leq g \sqcup_{\oplus} h$ by item 1 of this lemma. This gives us that $f \perp_{\oplus} g$ and $f \perp_{\oplus} h$, hence $\perp_{\oplus}[f, g, h]$. For the left to right implication, we use Lemma 3.4.27. As we have $\perp_{\oplus}[f, g, h]$, we know $f \nabla (g \nabla h)$ and $f \triangle (g \triangle h)$.

Recall that $g \sqcup_{\oplus} h = (g \nabla h)(\hat{g} \triangle \hat{h})$. Then the map

$$A \xrightarrow{f \nabla (g \nabla h)} B \oplus B \oplus B \xrightarrow{1 \oplus (\hat{g} \triangle \hat{h})} B \oplus B$$

makes the diagram for $f \nabla (g \sqcup_{\oplus} h)$ commute.

Recalling that $g \sqcup_{\oplus} h = (\bar{g} \nabla \bar{h})(g \triangle h)$, we also see that

$$A \oplus A \xrightarrow{1 \oplus (\bar{g} \nabla \bar{h})} A \oplus A \oplus A \xrightarrow{f \triangle (g \triangle h)} B$$

provides the witness map for $f \triangle (g \sqcup_{\oplus} h)$ and hence $f \perp_{\oplus}(g \sqcup_{\oplus} h)$.

□

3.5 Inverse sum categories

3.5.1 Inverse sums

Definition 3.5.1. In an inverse category with disjoint joins, an object X is the *inverse sum* of A and B when there exist maps i_0, i_1, x_0, x_1 such that:

- (i) i_0 and i_1 are monic;
- (ii) $i_0 : A \rightarrow X$, $i_1 : B \rightarrow X$, $x_0 : X \rightarrow A$ and $x_1 : X \rightarrow B$.
- (iii) $i_0^{(-1)} = x_0$ and $i_1^{(-1)} = x_1$.
- (iv) $i_0^{(-1)}i_0 \perp i_1^{(-1)}i_1$ and $i_0^{(-1)}i_0 \sqcup i_1^{(-1)}i_1 = 1_X$.

i_0 and i_1 will be referred to as the *injection* maps of the inverse sum.

Lemma 3.5.2. *The inverse sum X of A and B is unique up to isomorphism.*

Proof. Assume we have two inverse sums over A and B :

$$A \begin{array}{c} \xrightarrow{i_0} \\ \xleftarrow{x_0} \end{array} X \begin{array}{c} \xleftarrow{i_1} \\ \xrightarrow{x_1} \end{array} B \quad \text{and} \quad A \begin{array}{c} \xrightarrow{j_0} \\ \xleftarrow{y_0} \end{array} Y \begin{array}{c} \xleftarrow{j_1} \\ \xrightarrow{y_1} \end{array} B.$$

We will show that the map $x_0j_0 \sqcup x_1j_1 : X \rightarrow Y$ is an isomorphism.

Note by the fact that i_1 is monic, we may conclude from the definition that $0 = \overline{x_0i_0x_1}$ and therefore $0 = x_0i_0x_1$. Then, given that x_0 is the inverse of the monic i_0 , we may calculate $0 = \hat{0} = \widehat{x_0i_0x_1} = \overline{x_1^{(-1)}i_0^{(-1)}i_0} = \overline{x_1^{(-1)}i_0^{(-1)}} = \widehat{i_0x_1}$. From this we see $i_0x_1 = 0$. Similarly, we have $i_1x_0 = 0$, $j_0y_1 = 0$ and $j_1y_0 = 0$.

Next, by Lemma 3.4.3, we know that $\overline{x_0} \perp \overline{x_1}$ as both i_0 and i_1 are monic. By the same lemma, $\hat{j}_0 \perp \hat{j}_1$ as y_0, y_1 are the inverses of monic maps. Then, from [Dis.7], we have $x_0j_0 \perp x_1j_1$, hence we may form $x_0j_0 \sqcup x_1j_1 : X \rightarrow Y$.

Similarly, we may form the map $y_0i_0 \sqcup y_1i_1 : Y \rightarrow X$. Computing their composition:

$$\begin{aligned} (x_0j_0 \sqcup x_1j_1)(y_0i_0 \sqcup y_1i_1) &= (x_0j_0(y_0i_0 \sqcup y_1i_1)) \sqcup (x_1j_1(y_0i_0 \sqcup y_1i_1)) \\ &= x_0j_0y_0i_0 \sqcup x_0j_0y_1i_1 \sqcup x_1j_1y_0i_0 \sqcup x_1j_1y_1i_1 \\ &= x_01i_0 \sqcup x_00i_1 \sqcup x_10i_0 \sqcup x_11i_1 \\ &= x_0i_0 \sqcup x_1i_1 = 1. \end{aligned}$$

Computing the other direction,

$$\begin{aligned}
(y_0 i_0 \sqcup y_1 i_1)(x_0 j_0 \sqcup x_1 j_1) &= (y_0 i_0(x_0 j_0 \sqcup x_1 j_1)) \sqcup (y_1 i_1(x_0 j_0 \sqcup x_1 j_1)) \\
&= y_0 i_0 x_0 j_0 \sqcup y_0 i_0 x_1 j_1 \sqcup y_1 i_1 x_0 j_0 \sqcup y_1 i_1 x_1 j_1 \\
&= y_0 1 j_0 \sqcup y_0 0 j_1 \sqcup y_1 0 j_0 \sqcup y_1 1 j_1 \\
&= y_0 j_0 \sqcup y_1 j_1 = 1.
\end{aligned}$$

This shows that the map between any two inverse sums is an isomorphism.

□

Definition 3.5.3. Suppose \mathbb{X} is an inverse category with disjoint joins \sqcup based on a disjointness relation \perp and a restriction zero. If every pair of objects has an inverse sum as in Definition 3.5.1, we call the category an *inverse sum* category. For any two objects A, B in \mathbb{X} , we write their inverse sum as $A + B$.

Lemma 3.5.4. *Let \mathbb{X} be an inverse category with a restriction 0 and a disjoint sum tensor \oplus . Then \mathbb{X} is an inverse sum category.*

Proof. We claim that setting $i_i = \Pi_i$ and $x_i = \Pi_i$ and setting $X = A \oplus B$ produces inverse sums in \mathbb{X} and show this satisfies the four conditions of Definition 3.5.1.

- (i) From Lemma 3.4.20, we know that Π_0 and Π_1 are monic maps.
- (ii) $\Pi_0 : A \rightarrow A \oplus B$, $\Pi_1 : B \rightarrow A \oplus B$, $\Pi_0 : A \oplus B \rightarrow A$ and $\Pi_1 : A \oplus B \rightarrow B$.
- (iii) $\Pi_0^{(-1)} = \Pi_0$ and $\Pi_1^{(-1)} = \Pi_1$.
- (iv) $i_0^{(-1)} i_0 = 1 \oplus 0 \perp_{\oplus} 0 \oplus 1 = i_1^{(-1)} i_1$ as $1 \oplus 0 \nabla 0 \oplus 1 = (u_{\oplus}^r{}^{(-1)} \oplus u_{\oplus}^l{}^{(-1)})$ and $1 \oplus 0 \triangle 0 \oplus 1 = (\Pi_0 \oplus \Pi_1)$. For their join, $(1 \oplus 0) \sqcup_{\oplus} (0 \oplus 1) = (u_{\oplus}^r{}^{(-1)} \oplus u_{\oplus}^l{}^{(-1)})(\Pi_0 \oplus \Pi_1) = u_{\oplus}^r{}^{(-1)} \Pi_0 \oplus u_{\oplus}^l{}^{(-1)} \Pi_1 = 1 \oplus 1 = 1$.

□

Lemma 3.5.5. *If A is an object in \mathbb{X} , an inverse sum category, then $A + 0$ is isomorphic to A .*

Proof. We write the inverse sum diagram:

$$A \begin{array}{c} \xrightarrow{1} \\ \xleftarrow{1} \end{array} A \begin{array}{c} \xleftarrow{0} \\ \xrightarrow{0} \end{array} 0.$$

□

Lemma 3.5.6. *Suppose \mathbb{X} is an inverse sum category and \mathbb{Y} is an inverse category with a restriction zero. Further, suppose $F : \mathbb{X} \rightarrow \mathbb{Y}$ is a restriction functor which preserves disjoint joins. Then, F preserves inverse sums.*

Proof. In \mathbb{X} , consider the inverse sum over A and B ,

$$A \begin{array}{c} \xrightarrow{i_0} \\ \xleftarrow{x_0} \end{array} X \begin{array}{c} \xleftarrow{i_1} \\ \xrightarrow{x_1} \end{array} B.$$

The functor F maps this as follows:

$$F(A) \begin{array}{c} \xrightarrow{F(i_0)} \\ \xleftarrow{F(x_0)} \end{array} F(X) \begin{array}{c} \xleftarrow{F(i_1)} \\ \xrightarrow{F(x_1)} \end{array} F(B).$$

As F is a restriction functor, we immediately have $F(x_0) = F(i_0^{(-1)}) = F(i_0)^{(-1)}$ and $F(x_1) = F(i_1)^{(-1)}$. Since F preserves the disjoint join, we also have $F(i_0)^{(-1)}F(i_0) \perp F(i_1)^{(-1)}F(i_1)$ and $F(i_0)^{(-1)}F(i_0) \sqcup F(i_1)^{(-1)}F(i_1) = 1$.

Finally, as F is a restriction functor, it preserves monics, hence $F(i_0)$ and $F(i_1)$ are both monic and therefore $F(X)$ is the inverse sum of $F(A)$ and $F(B)$.

□

Lemma 3.5.7. *Given \mathbb{X} an inverse sum category and maps $f : A \rightarrow C$ and $g : B \rightarrow D$ in \mathbb{X} . Then $i_0^{(-1)}fi_0 \perp i_1^{(-1)}gi_1 : A + B \rightarrow A + B$.*

Proof. Note that $\overline{i_0^{(-1)}fi_0} = \overline{i_0^{(-1)}f} \leq \overline{i_0^{(-1)}}$ and similarly $\overline{i_1^{(-1)}gi_1} \leq \overline{i_1^{(-1)}}$. Then, by [Dis.3], we have $\overline{i_0^{(-1)}fi_0} \perp \overline{i_1^{(-1)}gi_1}$.

Then, as $\widehat{i_0^{(-1)}fi_0} \leq \widehat{i_0}$ and $\widehat{i_1^{(-1)}gi_1} \leq \widehat{i_1}$, this means we have $\widehat{i_0^{(-1)}fi_0} \perp \widehat{i_1^{(-1)}gi_1}$ and by Lemma 3.4.3, this means $i_0^{(-1)}fi_0 \perp i_1^{(-1)}gi_1$. \square

Lemma 3.5.8. *Given \mathbb{X} is an inverse sum category. Denote the inverse sum of objects A, B of \mathbb{X} by $A + B$. Then for objects A, B and X with maps $f : A \rightarrow X$ and $g : B \rightarrow X$ such that $\hat{f} \perp \hat{g}$, there exists a unique map h making the following diagram commute.*

$$\begin{array}{ccc}
 A & & \\
 \downarrow i_0 & \searrow f & \\
 A + B & \xrightarrow{h} & X \\
 \uparrow i_1 & \nearrow g & \\
 B & &
 \end{array}$$

We use the notation $f \hat{+} g$ for the unique map h .

Proof. As $\hat{f} \perp \hat{g}$ and $\overline{i_0^{(-1)}} \perp \overline{i_1^{(-1)}}$ we may form the map $h' = i_0^{(-1)}f \sqcup i_1^{(-1)}g$. By its construction, h' is a map from $A + B$ to X which makes the diagram commute. Suppose now that both maps v and w are such maps. Then we have

$$(i_0^{(-1)}i_0)v = (i_0^{(-1)}i_0)w \quad \text{and} \quad (i_1^{(-1)}i_1)v = (i_1^{(-1)}i_1)w.$$

As $i_0^{(-1)}i_0 \perp i_1^{(-1)}i_1$, by Lemmas 3.4.3 and 3.4.13, we know that $(i_0^{(-1)}i_0)v \perp (i_1^{(-1)}i_1)v$ and $(i_0^{(-1)}i_0)w \perp (i_1^{(-1)}i_1)w$ allowing us to form their respective disjoint joins. As the disjoint joins of equal maps remains equal, we have

$$(i_0^{(-1)}i_0)v \sqcup (i_1^{(-1)}i_1)v = (i_0^{(-1)}i_0)w \sqcup (i_1^{(-1)}i_1)w$$

$$(i_0^{(-1)}i_0 \sqcup i_1^{(-1)}i_1)v = (i_0^{(-1)}i_0 \sqcup i_1^{(-1)}i_1)w$$

$$(1)v = (1)w$$

$$v = w.$$

\square

Corollary 3.5.9. *Given \mathbb{X} is an inverse sum category. Then for objects A, B and X with maps $f : X \rightarrow A$ and $g : X \rightarrow B$ such that $\bar{f} \perp \bar{g}$, there exists a unique map h making the following diagram commute.*

$$\begin{array}{ccc}
 & & A \\
 & \nearrow f & \downarrow i_0 \\
 X & \xrightarrow{\quad h \quad} & A + B \\
 & \searrow g & \uparrow i_1 \\
 & & B
 \end{array}$$

We use the notation $f \overline{+} g$ for the unique map h .

Proof. This is simply the dual of Lemma 3.5.8. The unique map h in this case is $f i_0 \sqcup g i_1$. \square

Corollary 3.5.10. *Suppose \mathbb{X} is an inverse sum category. Then for objects A, B, C and D with maps $f : A \rightarrow C$ and $g : B \rightarrow D$, there exists a unique map h making the following diagram commute.*

$$\begin{array}{ccc}
 A & \xrightarrow{\quad f \quad} & C \\
 \downarrow i_0 & & \downarrow i_0 \\
 A + B & \xrightarrow{\quad h \quad} & C + D \\
 \uparrow i_1 & & \uparrow i_1 \\
 B & \xrightarrow{\quad g \quad} & D
 \end{array}$$

We use the notation $f + g$ for the map h .

Proof. This follows directly from Lemma 3.5.8 by setting $X = C + D$. The unique map in this case is $i_0^{(-1)} f i_0 \sqcup i_1^{(-1)} g i_1$. \square

Lemma 3.5.11. *Suppose \mathbb{X} and \mathbb{Y} are inverse sum categories and $F : \mathbb{X} \rightarrow \mathbb{Y}$ is a restriction functor which preserves inverse sums. Then, F preserves disjoint joins.*

Proof. By stating that F preserves the inverse sum, we mean it preserves diagrams derived via the properties of the inverse sum, and specifically, it will preserve the diagrams of Lemma 3.5.8 and Corollaries 3.5.9 and 3.5.10.

Suppose we are given $f, g : A \rightarrow B$ with $f \perp g$. In the inverse sum category, we know that $f \sqcup g = (\bar{f}i_0 \sqcup \bar{g}i_1)(i_0^{(-1)}fi_0 \sqcup i_1^{(-1)}gi_1)(i_0^{(-1)}\hat{f} \sqcup i_1^{(-1)}\hat{g})$, as this follows by:

1. Apply Corollary 3.5.9 to \bar{f} and \bar{g} ;
2. then apply Corollary 3.5.10 to f, g ;
3. finally apply Lemma 3.5.8 to \hat{f}, \hat{g} .

In the notation above, we have that $f \sqcup g = (\bar{f} \bar{\perp} \bar{g})(f + g)(\hat{f} \hat{+} \hat{g})$. This gives us, as F preserves the inverse sum:

$$\begin{aligned}
F(f \sqcup g) &= F(\bar{f} \bar{\perp} \bar{g})F(f + g)F(\hat{f} \hat{+} \hat{g}) \\
&= (F(\bar{f}) \bar{\perp} F(\bar{g}))(F(f) + F(g))(F(\hat{f}) \hat{+} F(\hat{g})) \\
&= (\overline{F(f)} \bar{\perp} \overline{F(g)})(F(f) + F(g))(\widehat{F(f)} \hat{+} \widehat{F(g)}) \\
&= F(f) \sqcup F(g).
\end{aligned}$$

The last line in the above is due to \mathbb{Y} being an inverse sum category as well.

□

3.5.2 Inverse sum tensor

Inverse sum tensor definitions

Definition 3.5.12. An *inverse sum tensor* in an inverse category \mathbb{X} with disjoint joins \sqcup based on a disjointness relation \perp and a restriction zero is given by a tensor combined with

two restriction monics, Π_0 and Π_1 . The data for the tensor is:

$$_-\oplus_- : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{X} \quad (\text{a restriction functor preserving disjoint joins})$$

$$0 : \mathbf{1} \rightarrow \mathbb{X}$$

$$u_\oplus^l : 0 \oplus A \rightarrow A$$

$$u_\oplus^r : A \oplus 0 \rightarrow A$$

$$a_\oplus : (A \oplus B) \oplus C \rightarrow A \oplus (B \oplus C)$$

$$c_\oplus : A \oplus B \rightarrow B \oplus A$$

$$\Pi_0 : A \rightarrow A \oplus B$$

$$\Pi_1 : B \rightarrow A \oplus B$$

where $u_\oplus^l, u_\oplus^r, a_\oplus, c_\oplus$ are all isomorphisms and the standard symmetric monoidal equations and coherence diagrams hold. The unit of the tensor, $0 : \mathbf{1} \rightarrow \mathbb{X}$, is the restriction zero of the category. We specifically note that preserving disjoint joins means the tensor obeys the following two equations:

$$f \perp g, h \perp k \text{ implies } f \oplus h \perp g \oplus k \tag{3.16}$$

$$f \perp g, h \perp k \text{ implies } (f \sqcup g) \oplus (h \sqcup k) = (f \oplus h) \sqcup (g \oplus k). \tag{3.17}$$

Disjointness and the inverse sum tensor

Lemma 3.5.13. *Given an inverse category \mathbb{X} with a disjoint sum tensor \oplus as in Definition 3.4.26, then \oplus is an inverse sum tensor.*

Proof. From the data of the disjoint sum tensor, the only thing remaining to show is that the tensor preserves the disjoint join.

Suppose we have $f \perp_\oplus g$ and $h \perp_\oplus k$. From Lemma 3.4.23, item (xi), we know both $(f \oplus h) \nabla (g \oplus k)$ and $(f \oplus h) \Delta (g \oplus k)$ exist, hence $(f \oplus h) \perp_\oplus (g \oplus k)$. This shows condition (equation (3.16)).

For condition (equation (3.17)), we compute from the right hand side:

$$\begin{aligned}
(f \oplus h) \sqcup_{\oplus} (g \oplus k) &= (f \oplus h) \nabla (g \oplus k) (\widehat{f \oplus h}) \triangle (\widehat{g \oplus k}) \\
&= ((f \nabla g) \oplus (h \nabla k)) \left((\hat{f} \oplus \hat{h}) \triangle (\hat{g} \oplus \hat{k}) \right) \\
&= ((f \nabla g) \oplus (h \nabla k)) \left((\hat{f} \triangle \hat{g}) \oplus (\hat{h} \triangle \hat{k}) \right) \\
&= \left((f \nabla g)(\hat{f} \triangle \hat{g}) \right) \oplus \left((h \nabla k)(\hat{h} \triangle \hat{k}) \right) \\
&= (f \sqcup_{\oplus} g) \oplus (h \sqcup_{\oplus} k).
\end{aligned}$$

The second and third lines above again use Lemma 3.4.23, item (xi). □

Inverse sums and the inverse sum tensor

Lemma 3.5.14. *If \oplus is an inverse sum tensor in the inverse category \mathbb{X} , then $A \oplus B \cong A + B$, an inverse sum of A and B .*

Proof. As \oplus is a restriction functor from $\mathbb{X} \times \mathbb{X}$ to \mathbb{X} , this actually follows immediately from Lemma 3.5.6. It may also be proven directly:

Draw the inverse sum diagram:

$$\begin{array}{ccccc}
A & \xrightarrow{i_0 = u_{\oplus}^r (-1)(1 \oplus 0)} & A \oplus B & \xleftarrow{i_1 = u_{\oplus}^l (-1)(0 \oplus 1)} & B \\
& \searrow x_0 = (1 \oplus 0)u_{\oplus}^r & & \swarrow x_1 = (0 \oplus 1)u_{\oplus}^l & \\
& & & &
\end{array}$$

Therefore, we have $i_0^{(-1)}i_0 = (1 \oplus 0)u_{\oplus}^r u_{\oplus}^r (-1)(1 \oplus 0) = (1 \oplus 0)(1 \oplus 0) = (1 \oplus 0)$. Similarly, $i_1^{(-1)}i_1 = (0 \oplus 1)$. Since $0 \perp 1$, we have $i_0^{(-1)}i_0 \perp i_1^{(-1)}i_1$.

By the functoriality of \oplus and that it preserves disjoint joins, we have $(1 \oplus 0) \sqcup (0 \oplus 1) = (1 \sqcup 0) \oplus (0 \sqcup 1) = 1 \oplus 1 = 1_{A \oplus B}$. Hence $A \oplus B$ is an inverse sum of A and B and by Lemma 3.5.2 it is isomorphic to $A + B$. □

Conversely, we can show that given a tensor which produces inverse sums, that tensor will be an inverse sum tensor.

Lemma 3.5.15. *Given an inverse category \mathbb{X} with restriction zero, a disjointness relation \perp , a disjoint join \sqcup and a symmetric monoidal tensor \oplus , with natural restriction monics $\Pi_0 : A \rightarrow A \oplus B$ and $\Pi_1 : B \rightarrow A \oplus B$ such that $A \oplus B$ is an inverse sum under Π_0 and Π_1 , then when $f, g : A \rightarrow B$ and $h, k : C \rightarrow D$ with $f \perp g$ and $h \perp k$, then $f \oplus h \perp g \oplus k$ and $(f \oplus h) \sqcup (g \oplus k) = (f \sqcup g) \oplus (h \sqcup k)$.*

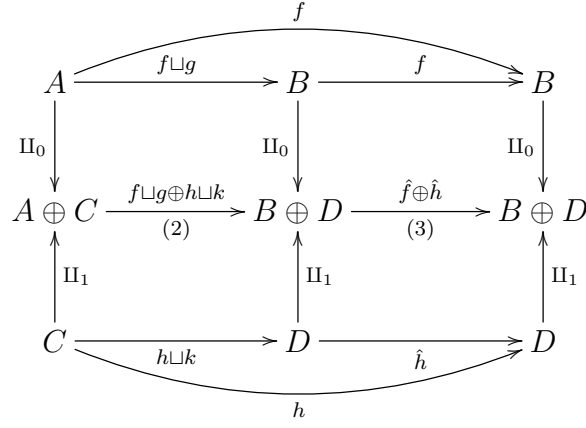
Proof. Similarly, this follows immediately from Lemma 3.5.11. We show it directly below:

$$\begin{array}{ccccc}
 & & f & & \\
 & \curvearrowright & & \curvearrowright & \\
 A & \xrightarrow{\bar{f}} & A & \xrightarrow{f \sqcup g} & B \\
 \Pi_0 \downarrow & & \Pi_0 \downarrow & & \Pi_0 \downarrow \\
 A \oplus C & \xrightarrow[\text{(1)}]{\bar{f} \oplus \bar{h}} & A \oplus C & \xrightarrow[\text{(2)}]{f \sqcup g \oplus h \sqcup k} & B \oplus D \\
 \Pi_1 \uparrow & & \Pi_1 \uparrow & & \Pi_1 \uparrow \\
 C & \xrightarrow{\bar{h}} & C & \xrightarrow{h \sqcup k} & D \\
 & \curvearrowleft & & \curvearrowleft & \\
 & & h & &
 \end{array}$$

Consider $\Pi_0^{(-1)} \bar{f} \Pi_0$. As this is idempotent and we are in an inverse category, we know that $\Pi_0^{(-1)} \bar{f} \Pi_0 = \overline{\Pi_0^{(-1)} \bar{f} \Pi_0} = \overline{\Pi_0^{(-1)} \bar{f}} = \widehat{\bar{f} \Pi_0}$. Similarly, $\Pi_1^{(-1)} \bar{h} \Pi_1 = \widehat{\bar{h} \Pi_1}$. By [Dis.5] and [Dis.6], we know that $\widehat{\bar{f} \Pi_0} \perp \widehat{\bar{g} \Pi_0}$ and $\widehat{\bar{h} \Pi_1} \perp \widehat{\bar{k} \Pi_1}$. Additionally, as shown in the proof of Lemma 3.5.2, we know $\widehat{\Pi_0} \perp \widehat{\Pi_1}$. Hence, by [Dis.3], we have $\widehat{\bar{f} \Pi_0} \perp \widehat{\bar{g} \Pi_0}$ for any maps x, y .

Hence, we can form the map $\widehat{\bar{f} \Pi_0} \sqcup \widehat{\bar{h} \Pi_1}$. Referring to the commutative diagram above, by Corollary 3.5.10 there is an unique map at location (1) which makes the diagram commute — currently given as $\bar{f} \oplus \bar{h}$. But, the map $\widehat{\bar{f} \Pi_0} \sqcup \widehat{\bar{h} \Pi_1}$ also satisfies this, hence we have $\widehat{\bar{f} \Pi_0} \sqcup \widehat{\bar{h} \Pi_1} = \bar{f} \oplus \bar{h}$. Similarly, $\widehat{\bar{g} \Pi_0} \sqcup \widehat{\bar{k} \Pi_1} = \bar{g} \oplus \bar{k}$. But, by Lemma 3.4.15, this means $\overline{\bar{f} \oplus \bar{h}} \perp \overline{\bar{g} \oplus \bar{k}}$.

Using a similar argument based on the diagram



we can show $\widehat{f \oplus h} \perp \widehat{g \oplus k}$ and therefore $f \oplus h \perp g \oplus k$.

This allows us to form the map $(f \oplus h) \sqcup (g \oplus k)$. Once again, as the objects are inverse sums, the map at (2) is unique. However, we see that both $f \sqcup g \oplus h \sqcup k$ and $(f \oplus h) \sqcup (g \oplus k)$ fulfill this requirement and hence they are equal. \square

Definition 3.5.16. An inverse category \mathbb{X} with restriction zero, a disjointness relation \perp , a disjoint join \sqcup and an inverse sum tensor \oplus is called an *inverse sum tensor category*.

Corollary 3.5.17. In an inverse sum tensor category, $f \oplus g$ is given by $i_0^{(-1)} f i_0 \sqcup i_1^{(-1)} g i_1$.

Proof. Recall that in the proof of Lemma 3.5.2 that we showed $\overline{i_0^{(-1)}} \perp \overline{i_1^{(-1)}}$ and $\widehat{i_0} \perp \widehat{i_1}$. Hence, by [Dis.7], we know that $i_0^{(-1)} f i_0 \perp i_1^{(-1)} g i_1$ and we can therefore form the disjoint join. \square

Matrices

In this section, we will show that when given an inverse sum category \mathbb{X} , one can define a type of matrix category based on \mathbb{X} . We will call this category $iMat(\mathbb{X})$. Furthermore, we will show that $iMat(\mathbb{X})$ has is an inverse category and that \mathbb{X} embeds within this category.

Definition 3.5.18. A matrix of maps $[f_{ij}]$ in an inverse sum category \mathbb{X} which satisfy the condition:

$$\text{For each } i, \text{ whenever } j \neq k, \text{ then } f_{ij} \perp f_{ik} \quad (3.18)$$

is called an *inverse sum matrix*.

In the following we will use the notation Π_i for the i^{th} injection map of the inverse sum.

Further, we define a monoid \diamond of any two such matrices by $[h_{ik}] = [f_{ij}] \diamond [g_{jk}]$ where the element h_{ik} of the matrix is given by:

$$h_{ik} = \bigsqcup_j f_{ij} \Pi_i g_{jk}$$

noting again that composition of maps is diagrammatic order. Of course, this is only defined when the domain of g_{jk} is the inverse sum of some objects in \mathbb{X} and the range of f_{ij} .

We also define \diamond for an inverse sum matrix $[f_{ij}]$ and a row (list) of objects $[A_i]$ in \mathbb{X} by $[B_j] = [A_i] \diamond [f_{ij}]$ is given by

$$B_j = \Sigma_i f_{ij} A_i$$

where Σ stands for the inverse sum operation of the category \mathbb{X} . Note this requires choosing a specific inverse sum for each pair of objects from the set of isomorphic inverse sums.

Definition 3.5.19. Given an inverse sum category \mathbb{X} , we define $iMat(\mathbb{X})$, the *inverse matrix category* of \mathbb{X} as follows:

Objects: Lists of the objects of \mathbb{X} .

Maps: The inverse sum matrix $[f_{ij}] : [A_i] \rightarrow [B_j]$ where each individual map $f_{ij} : A_i \rightarrow B'_{ij}$ is a map in \mathbb{X} . The B_j are the chosen inverse sums of the B'_{ij} . The result of $[f_{ij}][A_i]$ is given by $[A_i] \diamond [f_{ij}]$.

Identity: The inverse sum matrix I .

Composition: Given $[f_{ij}] : [A_i] \rightarrow [B_j]$ and $[g_{jk}] : [B_j] \rightarrow [C_k]$, then $[f_{ij}][g_{jk}] : [A_i] \rightarrow [C_k]$ is defined as $[g_{jk}] \diamond [f_{ij}]$.

Restriction: We set $\overline{[f_{ij}]}$ to be $[f'_{ij}]$ where $f'_{ij} = 0$ when $i \neq j$ and $f'_{ii} = \sqcup_j \overline{f_{ij}}$.

We will use the notation $\delta[d_1, d_2, \dots, d_n]$ for a diagonal $n \times n$ matrix with entries along the diagonal of $[d_1, d_2, \dots, d_n]$ and $\delta_j[d_j]$ for diagonal matrices where the j, j entry is d_j .

Lemma 3.5.20. *When \mathbb{X} is an inverse sum category, $iMat(\mathbb{X})$ is a restriction category.*

Proof. We need to show the following:

- Composition is well defined and associative;
- The restriction is well defined.

Composition is well defined: Consider $[h_{ik}] = [f_{ij}][g_{jk}] (= [g_{jk}] \diamond [f_{ij}])$ where

$$[f_{ij}] : [A_1, \dots, A_n] \rightarrow [B_1, \dots, B_m] \text{ and } [g_{jk}] : [B_1, \dots, B_m] \rightarrow [C_1, \dots, C_\ell].$$

By Definition 3.5.18, we know $h_{ik} = \sqcup_j f_{ij} g_{jk}$. Individually, for each j we know the composition $f_{ij} \amalg_i g_{jk}$ is defined and is from A_i to C'_{jk} . By the stability and universality of \perp , we know h_{ik} exists and by the definition of \sqcup , we have each $h_{ik} : A_i \rightarrow C'_{jk}$ and hence composition is well-defined.

Associativity of composition. We have

$$\begin{aligned} ([f_{ij}][g_{jk}])[h_{k\ell}] &= \left[\left(\sqcup_j f_{ij} \amalg_i g_{jk} \right) \right] [h_{k\ell}] \\ &= \left[\sqcup_k \left(\sqcup_j f_{ij} \amalg_i g_{jk} \right) \amalg_j h_{k\ell} \right] \\ &= \left[\sqcup_j f_{ij} \amalg_i \left(\sqcup_k g_{jk} \amalg_j h_{k\ell} \right) \right] \\ &= [f_{ij}]([g_{jk}][h_{k\ell}]) \end{aligned}$$

For the restriction axioms:

$$[\mathbf{R.1}] \quad \overline{[f_{ij}]}[f_{ij}] = \begin{bmatrix} (\sqcup_j \overline{f_{1j}})f_{11} & \cdots & (\sqcup_j \overline{f_{1j}})f_{1n} \\ & \vdots & \\ (\sqcup_j \overline{f_{mj}})f_{m1} & \cdots & (\sqcup_j \overline{f_{mj}})f_{mn} \end{bmatrix} = [f_{ij}].$$

[R.2] $\overline{[f_{ij}]} \overline{g_{ij}} = \overline{g_{ij}} \overline{[f_{ij}]}$ as diagonal matrices commute and \sqcup is also commutative.

$$\begin{aligned}
[\mathbf{R.3}] \quad \overline{[f_{ij}][g_{jk}]} &= \overline{\delta[\sqcup_j \overline{f_{1j}}, \dots, \sqcup_j \overline{f_{nj}}][g_{jk}]} \\
&= \overline{\begin{bmatrix} \sqcup_j \overline{f_{1j}} g_{11} & \dots & \sqcup_j \overline{f_{1j}} g_{1k} \\ & \vdots & \\ \sqcup_j \overline{f_{nj}} g_{n1} & \dots & \sqcup_j \overline{f_{nj}} g_{nk} \end{bmatrix}} \\
&= \delta[\sqcup_k (\sqcup_j (\overline{f_{1j}} g_{1k})), \dots, \sqcup_k (\sqcup_j (\overline{f_{nj}} g_{nk}))] \\
&= \delta[\sqcup_k (\sqcup_j (\overline{f_{1j}}) \overline{g_{1k}}), \dots, \sqcup_k (\sqcup_j (\overline{f_{nj}}) \overline{g_{nk}})] \\
&= \delta[(\sqcup_j (\overline{f_{1j}}) \sqcup_k \overline{g_{1k}}), \dots, (\sqcup_j (\overline{f_{nj}}) \sqcup_k \overline{g_{nk}})] \\
&= \overline{[f_{ij}][g_{jk}]}
\end{aligned}$$

$$\begin{aligned}
[\mathbf{R.4}] \quad [f_{ij}][\overline{g_{jk}}] &= [f_{ij}]\delta_j[\sqcup_k \overline{g_{jk}}] \\
&= \begin{bmatrix} f_{11} \sqcup_k \overline{g_{1k}} & \cdots & f_{1n} \sqcup_k \overline{g_{nk}} \\ & \vdots & \\ f_{m1} \sqcup_k \overline{g_{1k}} & \cdots & f_{mn} \sqcup_k \overline{g_{nk}} \end{bmatrix} \\
&= \begin{bmatrix} \sqcup_k f_{11} \overline{g_{1k}} & \cdots & \sqcup_k f_{1n} \overline{g_{nk}} \\ & \vdots & \\ \sqcup_k f_{m1} \overline{g_{1k}} & \cdots & \sqcup_k f_{mn} \overline{g_{nk}} \end{bmatrix} \\
&= \begin{bmatrix} \sqcup_k \overline{f_{11} g_{1k}} f_{11} & \cdots & \sqcup_k \overline{f_{1n} g_{nk}} f_{1n} \\ & \vdots & \\ \sqcup_k \overline{f_{m1} g_{1k}} f_{m1} & \cdots & \sqcup_k \overline{f_{mn} g_{nk}} f_{mn} \end{bmatrix} \\
&= \begin{bmatrix} \sqcup_j \sqcup_k \overline{f_{1j} g_{jk}} f_{11} & \cdots & \sqcup_j \sqcup_k \overline{f_{1j} g_{jk}} f_{1n} \\ & \vdots & \\ \sqcup_j \sqcup_k \overline{f_{mj} g_{jk}} f_{m1} & \cdots & \sqcup_j \sqcup_k \overline{f_{mj} g_{jk}} f_{mn} \end{bmatrix} \\
&= \overline{[f_{ij}][g_{jk}]}[f_{ij}].
\end{aligned}$$

□

Lemma 3.5.21. *In an inverse sum tensor category, any map $f : A \oplus B \rightarrow C \oplus D$ may be represented in a matrix form. Composition of maps may be computed by multiplication of the matrices, with composition taking the place of base level multiplication and djoin the place of addition.*

Proof. Recall from Lemma 3.5.14 that $A \oplus B$ and $C \oplus D$ are inverse sums. Referencing Definition 3.5.1, define $e_0 = i_0^{(-1)}i_0$ and $e_1 = i_1^{(-1)}i_1$ and recall that $e_0 \perp e_1$, $e_0 \sqcup e_1 = 1$.

Then given a function $f : A \oplus B \rightarrow C \oplus D$ define

$$f_M = \begin{bmatrix} e_0 f e_0 & e_0 f e_1 \\ e_1 f e_0 & e_1 f e_1 \end{bmatrix}.$$

Note first that since $e_0 \perp e_1$, the maps in the rows of f_M are disjoint by the stability of the disjointness relation. Similarly, the maps in the columns are disjoint by universality. We have $e_0 f e_0 \sqcup e_0 f e_1 = e_0 f$ and $e_1 f e_0 \sqcup e_1 f e_1 = e_1 f$. Each of these maps are disjoint by universality. Finally, $e_0 f \sqcup e_1 f = (e_0 \sqcup e_1) f = f$ and hence we may recover the initial map whenever we have a matrix of this form. We will call this computation the distinct join of f_M .

Next, consider $f_M \times g_M$. As each e_i is idempotent, this is

$$f_M \times g_M = \begin{bmatrix} e_0 f e_0 g e_0 \sqcup e_0 f e_1 g e_0 & e_0 f e_0 g e_1 \sqcup e_0 f e_1 g e_1 \\ e_1 f e_0 g e_0 \sqcup e_1 f e_1 g e_0 & e_1 f e_0 g e_1 \sqcup e_1 f e_1 g e_1 \end{bmatrix} = \begin{bmatrix} e_0 f g e_0 & e_0 f g e_1 \\ e_1 f g e_0 & e_1 f g e_1 \end{bmatrix}$$

where the distinct joins are well defined due to the stability and universality of the join. We can see that the distinct join of $f_M \times g_M = f g$ and as such we have composition.

□

In particular, we note that we may represent $f : A \rightarrow B$ by the matrix

$$\begin{bmatrix} 1f1 & 1f0 \\ 0f1 & 0f0 \end{bmatrix} = \begin{bmatrix} f & 0 \\ 0 & 0 \end{bmatrix}$$

as $A \cong A \oplus 0$ and $B \cong B \oplus 0$.

We now turn to examining a category of specialized matrices over an inverse sum category. In general, the matrix category $\mathbf{iMat}(\mathbb{X})$ will have objects that are lists of objects in \mathbb{X} , $X = (X_1, \dots, X_m)$. Maps between lists will be matrices $[f_{ij}] : (X_1, \dots, X_m) \rightarrow (Y_1, \dots, Y_n)$. We will only consider maps whose matrices have disjoint rows, i.e., if $[f_{ij}]$ is a matrix, it must have $f_{ij} \perp f_{ik}$ for all i whenever $j \neq k$.

Lemma 3.5.22. *If \mathbb{X} is an inverse category with an inverse sum tensor, $\mathbf{iMat}(\mathbb{X})$, the category of inverse sum matrices over \mathbb{X} with composition as in Definition 3.5.18, has sums.*

Proof.

□

3.6 Completing a distributive inverse category

3.6.1 Distributive restriction categories

Definition 3.6.1. A Cartesian restriction category with a restriction zero and coproducts is called *distributive* when there is an isomorphism ρ such that

$$A \times (B + C) \xrightarrow{\rho} (A \times B) + (A \times C).$$

In a distributive inverse category, we lack:

$$A \xrightarrow{!} 1$$

and

$$A + A \xrightarrow{\nabla} A.$$

3.6.2 Distributive inverse categories

A *Distributive inverse category* has a sum tensor and product tensor as defined in sub-section [sub-section 3.6.1](#) where the product distributes over the sum in the following manner:

$$(A \otimes B) \oplus (A \otimes C) \xrightarrow{f} A \otimes (B \oplus C)$$

where f is some total function. As we are in an inverse category and f is total, we have $f f^{(-1)} = 1$. This also implies $1 = f^{(-1)} f$ where

$$A \otimes (B \oplus C) \xrightarrow{f^{(-1)}} (A \otimes B) \oplus (A \otimes C).$$

Suppose we have an inverse category \mathbb{X} with two tensors, \otimes and \oplus , as described above and we have:

$$\begin{array}{ccc} A \otimes (B \oplus C) & \xrightarrow{[1 \otimes \Pi_0, 1 \otimes \Pi_1]^{(-1)}} & (A \otimes B) \oplus (A \otimes C) \\ & \searrow & \downarrow [1 \otimes \Pi_0, 1 \otimes \Pi_1] \\ & & A \otimes (B \oplus C) \end{array}$$

Lemma 3.6.2. *The category \mathbb{X} as defined above is an inverse distributive category.*

Proof. We will show that the required function, f is $[1 \otimes \Pi_0, 1 \otimes \Pi_1]$.

$$(A \otimes B) \oplus (A \otimes C) \xrightarrow{[1 \otimes \Pi_0, 1 \otimes \Pi_1]} A \otimes (B \oplus C) \quad (3.19)$$

The map in [equation \(3.19\)](#) is total. To see this, consider the unique decomposition of the restriction of the map.

$$\begin{aligned} \overline{[1 \otimes \Pi_0, 1 \otimes \Pi_1]} &= \begin{bmatrix} \overline{1 \otimes \Pi_0} & 0 \\ 0 & \overline{1 \otimes \Pi_1} \end{bmatrix} \\ &= \begin{bmatrix} \bar{1} \otimes \overline{\Pi_0} & 0 \\ 0 & \bar{1} \otimes \overline{\Pi_1} \end{bmatrix} \\ &= \begin{bmatrix} 1 \otimes 1 & 0 \\ 0 & 1 \otimes 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ &= 1 \end{aligned}$$

□

3.7 Inverse Turing Categories

3.8 Reversible computation

Bennet, in [\[9\]](#), showed that it was possible to emulate a standard Turing machine via a reversible Turing machine and vice-versa. This showed the equivalence of standard and reversible Turing machines. We reproduce the essence of this proof below.

3.8.1 Reversible Turing machines

Turing machines consist of a tape, a read-write head positioned over the tape, a machine state and a set of instructions. The set of instructions may be given as a set of transitions determining the movement of the read-write head, what it writes and the resulting state of the machine.

Definition 3.8.1. Given an alphabet A which does not contain a space, a tape is in *standard format* when:

- [T.1] The tape head is positioned directly over a blank space;
- [T.2] The spaces to the left (the $+1$ direction) contain only elements of A .
- [T.3] All other spaces of the tape are blank.

Definition 3.8.2. A *turing quintuple* is a quintuple $(s, \alpha, \alpha', \delta, s')$ where:

- [Q.1] $s, s' \in S$, where S is a predefined set of states;
- [Q.2] $\alpha, \alpha' \in A$ is predefined set of glyphs;
- [Q.3] $\delta \in \{-1, 0, 1\}$.

Definition 3.8.3. A *standard turing quintuple set* Q consists of a set of turing quintuples such that:

- (i) If $q_1 = (s_1, \alpha_1, \alpha'_1, \delta_1, s'_1)$ and $q_2 = (s_2, \alpha_2, \alpha'_2, \delta_2, s'_2)$ are in Q , then either $s_1 \neq s_2$ or $\alpha_1 \neq \alpha_2$ or both are not equal.

- (ii) There are two special quintuples contained in Q :

(a) $(s_1, \sqcup, \sqcup, +1, s_2)$ ¹, the *start quintuple*;

(b) $(s_{t-1}, \sqcup, \sqcup, 0, s_t)$, the *end quintuple* where t is the number of states and is the final state of the machine.

¹Here, \sqcup is used to signify a blank.

Definition 3.8.4. A *standard Turing machine* is given by

- [TM.1] a standard turing quintuple set;
- [TM.2] a tape that starts in standard format;
- [TM.3] and the condition that and if the machine halts, it will halt in state s_t , the final state of the end quintuple and the output will be in standard format.

The turing quintuples may also be regarded as giving the data for a partial function in SETS: $\tau : S \times A \rightarrow A \times \{-1, 0, 1\} \times S$.

Remark 3.8.5. A multi-tape Turing machine with n tapes and read-write heads can be described by modifying definition 3.8.4 such that α is an n -tuple of the set of glyphs for the Turing machine and δ is an n -tuple of movement directions.

Example 3.8.6. Suppose $S = \{start, run, reset, done\}$, $A = \{0, 1, \sqcup\}$ and the Turing machine program is given by the quintuples

$$\begin{aligned}
 &(start, \sqcup, \sqcup, +1, run), \\
 &(run, 0, 1, +1, run), (run, 1, 0, +1, run), \\
 &(run, \sqcup, \sqcup, -1, reset), \\
 &(reset, 0, 0, -1, reset), (reset, 1, 1, -1, reset), \\
 &(reset, \sqcup, \sqcup, 0, done).
 \end{aligned}$$

This program will perform a “bit-flip” of all 0s and 1s on the tape until it reads a space, reposition the read head to the standard format and then it will halt.

As we see in example 3.8.6, it is *possible* that a Turing machine program is reversible. If we had chosen the second quintuple to be $(run, 0, 0, +1, run)$ instead, the program would not have been reversible.

The essential property that a Turing machine program needs to be reversible is that the function τ defined from the quintuples is injective. In order to simplify the discovery the function being injective, we reformulate the turing quintuples as quadruples.

Definition 3.8.7. A *turing quadruple* is given by a quadruple

$$(s, [b_1, b_2, \dots, b_n], [b'_1, b'_2, \dots, b'_n], s')$$

such that:

- (i) $s, s' \in S$, some set of states;
- (i) $b_j \in A \cup \{\phi\}$ where A is some alphabet;
- (i) $b'_j \in A + \{-1, 0, 1\}$;
- (i) $b'_j \in \{-1, 0, 1\}$ if and only if $b_j = \phi$.

In this definition, $b_j = \phi$ means that the value of tape j is ignored.

A turing quadruple explicitly splits the read/write action of the Turing machine away from the movement. In a particular step for tape k , the turing machine will either read and write an item or it will move.

Remark 3.8.8. Any turing quintuple q of n tapes may be split into two turing quadruples, q_r and q_m by the addition of a new state a'' in A . The quadruple q_r will consist of all the read-write operations and leave the Turing machine in state a'' . The quadruple q_m will start in state a'' with all the b_j set to ϕ and b'_j being movement on each of the n tapes.

Definition 3.8.9. A set of turing quadruples Q is called *reversible set of turing quadruples* when given $q_1, q_2 \in Q$, with $q_1 = (a, [b_j], [b'_j], a')$ and $q_2 = (c, [d_j], [d'_j], c')$:

[RTM.1] if $a = c$, then there is a k where $b_k, d_k \in A$ and $b_k \neq d_k$;

[RTM.2] if $a' = c'$, then there is a j with $b'_j, d'_j \in A$ and $b'_j \neq d'_j$.

Similarly to turing quintuples, turing quadruples may be taken as the data for a function in SETS:

$$\rho : S \times (A \cup \{\phi\}) \rightarrow (A + \{-1, 0, 1\}) \times S.$$

We can see by inspection that ρ is a reversible partial function when the set of turing quadruples that give ρ is a reversible set of turing quadruples.

Definition 3.8.10. A *reversible Turing machine* is one that is described by a set of reversible turing quadruples.

We will show that a reversible Turing machine with three tapes can emulate a Turing machine.

Theorem 3.8.11 (Bennet[9]). *Given a standard Turing Machine M , it may be emulated by a three tape reversible Turing machine R . In this case, emulated means:*

- (i) M halts on standard input I if and only if R halts on standard input (I, \sqcup, \sqcup) .
- (i) M halts on standard input I producing standard output O , if and only if R halts on input (I, \sqcup, \sqcup) producing standard output (I, \sqcup, O) .

Proof. (Sketch only).

The crux of the proof is to convert the quintuples of M to the quadruples of R as noted in remark 3.8.8 on the preceding page. Explicitly for a single tape machine, we have

$$(s, a, a, \delta, s') \mapsto ((s, a, a', s''), (s'', \phi, \delta, s')). \quad (3.20)$$

In equation (3.20), s'' is a new state for the machine M , not in the current set of states.

Assign an order to the n quintuples of M , where the start quintuple is the first in the order and the end quintuple comes last. Convert these to quadruples as in equation (3.20).

We then proceed to create three groups of quadruples for R . We call these *emulation*, *copy*, and *restore*.

To create the emulation phase quadruples, we examine the pairs of quadruples of M in the sorted order and produce a pair of quadruples for R .

$$\begin{aligned}
\text{Pair 1} \quad & (s_1, \sqcup, \sqcup, s_1'') \mapsto (s_1, [\sqcup, \phi, \sqcup], [\sqcup, +1, \sqcup], e_1) \\
& (s_1'', \phi, \delta, s_2) \mapsto (e_1, [\phi, \sqcup, \phi], [\delta, 1, 0], s_2) \\
& \vdots \\
\text{Pair } j \quad & (s_k, a_j, a_j', s_k'') \mapsto (s_k, [a_j, \phi, \sqcup], [a_j', +1, \sqcup], e_j) \\
& (s_k'', \phi, \delta, s_i) \mapsto (e_j, [\phi, \sqcup, \phi], [\delta_j, j, 0], s_i) \\
& \vdots \\
\text{Pair } n \quad & (s_\ell, \sqcup, \sqcup, s_\ell'') \mapsto (s_\ell, [\sqcup, \phi, \sqcup], [\sqcup, +1, \sqcup], e_n) \\
& (s_\ell'', \phi, 0, s_f) \mapsto (e_n, [\phi, \sqcup, \phi], [0, n, 0], s_f).
\end{aligned}$$

By inspection, one can see that even if the quadruples of M were not a reversible set, the set created for R is a reversible set, due to the writing of the quadruple index on tape 2. Upon completion of the emulation phase, tape 1 will be the same as M would have produced on its single tape, tape 2 will be $[1, 2, \dots, n]$ and tape 3 will be blanks.

For the copy phase, we create the following quadruples:

$$\begin{aligned}
& (s_f, [\sqcup, n, \sqcup], [\sqcup, n, \sqcup], c_1) \\
& (c_1, [\phi, \phi, \phi], [+1, 0, +1], c_1') \\
& (c_1', [x, n, \sqcup], [x, n, x], c_1) \quad \text{when } x \neq \sqcup \\
& (c_1', [\sqcup, n, \sqcup], [\sqcup, n, x], c_2) \\
& (c_2, [\phi, \phi, \phi], [-1, 0, -1], c_2') \\
& (c_2', [x, n, x], [x, n, x], c_2) \quad \text{when } x \neq \sqcup \\
& (c_2', [\sqcup, n, \sqcup], [\sqcup, n, \sqcup], r_\ell).
\end{aligned}$$

In these quadruples, the states $\{c_1, c_1', c_2, c_2'\}$ should be chosen to be distinct from the states

in the emulation phase. As an example, set them as follows:

$$c_1 = (\{c\}, s_1) \quad c'_1 = (\{c'\}, s_1) \quad c_2 = (\{c\}, s_f) \quad c'_1 = (\{c'\}, s_f).$$

At the completion of this phase, tapes 1 and 2 will be unchanged and tape 3 will be a copy of tape 1.

Finally we perform the restore phase where the history will be erased and tape 1 reset to the input. The quadruples that will accomplish this are:

$$\begin{aligned} \text{Pair } n & \quad (r_n, [\phi, n, \phi], [0, \sqcup, 0], r'_n) \\ & \quad (r'_n, [\sqcup, \phi, \sqcup], [\sqcup, -1, \sqcup], r_{n-1}) \\ & \quad \vdots \\ \text{Pair } j & \quad (r_k, [\phi, j, \phi], [-\delta_j, \sqcup, 0], r'_j) \\ & \quad (r'_j, [a'_j, \phi, \sqcup], [a_j, -1, \sqcup], r_i) \\ & \quad \vdots \\ \text{Pair } 1 & \quad (r_2, [\phi, 1, \phi], [-1, \sqcup, 0], r'_1) \\ & \quad (r'_1, [\sqcup, \phi, \sqcup], [\sqcup, -1, \sqcup], r_1). \end{aligned}$$

The r states are derived from the s states of the emulation phase.

$$r_j = (\{r\}, s_j) \quad r'_j = (\{r'\}, s_j).$$

In this restore phase, the indexes of the states r match up to the indexes of states s . The quadruples reverse the actions of the emulate phase on tape 1, erase the history on tape 2 and make no change to tape 3.

□

3.8.2 Reversible automata and linear combinatory algebras

While reversible Turing machines, as described in [sub-section 3.8.1 on page 112](#), show that reversible computing is as powerful as standard computing, they do not give us a sense of what may be considered to be happening at a higher level.

To accomplish that task we examine the results of the paper “A Structural Approach to Reversible Computation”[1]. In this paper, Abramsky gives a description of a reversible automaton together with a linear combinatory algebra. We will begin by revisiting some definitions and constructions necessary for discussing automata. The next subsection will introduce combinatory algebras, after which we will describe the reversible automata of [1] and add a short proof that it can emulate a reversible turing machine.

Automata

We will describe the automata as a term-rewriting system. This requires, of course, giving a few basic definitions. See, e.g., [7].

Definition 3.8.12. An *arity* is a function from a function to the natural numbers. The arity of F is the number of inputs (arguments) required by F .

Definition 3.8.13. A *signature* Σ is a set of *function symbols* F, G, \dots , each of which has an arity.

Remark 3.8.14. We refer to functions with low arity in the following ways:

- *Arity* = 0. These are known as *nullary* functions or constants.
- *Arity* = 1. These are known as *unary* functions.
- *Arity* = 2. These are known as *binary* functions.

Definition 3.8.15. A *term alphabet* is a set A containing a signature Σ and a countably infinite set X , the variables. Furthermore, $\Sigma \cap X = \phi$.

Definition 3.8.16. A *term algebra* of the term alphabet $\Sigma \cup X$ is denoted by $T_\Sigma(X)$ and defined as follows:

- $x \in V \implies x \in T_\Sigma$ and
- For any $F \in \Sigma$, with $\text{arity}(F) = n$, and $\{t_1, \dots, t_n\} \subseteq T_\Sigma$, then $F(t_1, \dots, t_n) \in T_\Sigma$. In the case where $\text{arity}(F) = 0$, we write $F \in T_\Sigma$.

Definition 3.8.17. The *ground terms* of a term algebra are those terms that do not contain any variable. The set of these terms is designated as T_Σ .

Remark 3.8.18. Note the ground terms consist of the constants and recursively applying the function symbols of Σ to them.

As we are considering rewrite systems, we will need to consider aspects of substitution and unification.

Definition 3.8.19. A *substitution* is a map $\sigma : T_\Sigma(X) \rightarrow T_\Sigma(X)$ which is natural for all function symbols in Σ . In particular if $\text{arity}(c) = 0$ then $\sigma(c) = c$.

Note that given the above definition a substitution σ is completely determined by its action on variables. If $\sigma : X \rightarrow X$ and is injective, we call σ a renaming. Moreover, if σ restricted to the variables in a term t is an injective map of X on those variables, we call *sigma* a renaming of t .

Substitution allows us to define a partial order on $T_\Sigma(X)$, as follows:

Definition 3.8.20. In $T_\Sigma(X)$, let $\sigma(t) = s$. Then we say s is an *instance* of t , written $s \preceq t$. Moreover, if σ is not just a renaming for t , then we write $s \prec t$. If σ is a renaming of t , we write $s \simeq t$.

Lemma 3.8.21. *Subsumption, as defined in 3.8.20 is a partial order, i.e., it is transitive and reflexive.*

Proof.

□

Lemma 3.8.22. *Given terms r, t such that there is at least one s with $s \preceq r$ and $s \preceq t$, then there exists a g such that $g \preceq r$ and $g \preceq t$ and for any s' with $s' \preceq r$ and $s' \preceq t$ we will have $s' \preceq g$.*

Proof.

1. Algorithm to compute supremum of p, q terms.
2. Strict \prec has no infinite ascending chains.
3. Shows main part - there exists.
4. Can now show it is unique up to renaming.

□

The subsumption ordering can be used to derive a similar ordering on substitutions:

Definition 3.8.23. $\sigma \preceq \tau$ if and only if there is a ρ with $\sigma = \tau\rho$, where $\tau\rho$ is the diagrammatic order composition of the two substitutions.

Definition 3.8.24. For terms s, t , if $\sigma(t) = \sigma(s)$, then the substitution σ is called a *unifier* for the terms s, t .

Lemma 3.8.25. *If s, t are terms with a unifier σ , there exists a substitution τ that unifies s, t such that $\tau \preceq \rho$ whenever ρ unifies s, t . ρ is called the most general unifier of s and t .*

Proof. Follows from [3.8.22 on the previous page](#). □

Notation 3.8.26. Following [1], we write $\mathcal{U}(t, u) \downarrow \sigma$ if σ is the most general unifier of terms t, u .

Combinatory Algebra

Definition 3.8.27. A *combinatory algebra* is an algebra with one binary operation, \cdot written in infix notation. The operation is not assumed to be associative. Multi-element expressions such as $a \cdot b \cdot c$ are to be taken as associating to the left, that is,

$$a \cdot b \cdot c = (a \cdot b) \cdot c.$$

The combinatory algebra may possess distinguished elements that are subject to specific rewrite rules.

Definition 3.8.28. *Combinatory logic* is the combinatory algebra with two distinguished elements, K and S , such that the following hold:

$$\begin{aligned} K \cdot x \cdot y &= x \\ S \cdot x \cdot y \cdot z &= x \cdot z \cdot (y \cdot z). \end{aligned}$$

Note that combinatory logic does not require a specific set that must be used for the algebra, simply that it has the two distinguished elements.

Combinatory logic was shown to be equivalent to the λ calculus by

For example, we may define the identity combinator I as $I = S \cdot K \cdot K$. Further combinators may be defined, such as the B combinator, defined by $B \cdot a \cdot b \cdot c = a \cdot (b \cdot c)$. The S and K combinators are complete, in that other combinators such as B may be defined from them. E.g., $B = S \cdot (K \cdot S) \cdot K$. In fact, we may define an alternate combinatory algebra that is equivalent to Combinatory Logic.

Definition 3.8.29. A *BCKW-Combinatory algebra* is a Combinatory Algebra with four distinguish elements, B , C , K , and W subject to the following equations:

$$\begin{aligned} B \cdot a \cdot b \cdot c &= a \cdot (b \cdot c) \\ C \cdot a \cdot b \cdot c &= a \cdot c \cdot b \\ K \cdot a \cdot b &= a \\ W \cdot a \cdot b &= a \cdot b \cdot b \end{aligned}$$

In fact, a BCKW-Combinatory algebra is equivalent to a Combinatory logic.

Lemma 3.8.30. *The distinguished elements of a BCKW-Combinatory algebra may be represented by S and K . Conversely, the S and K of a Combinatory logic may be created from B, C, K and W .*

Proof. For the first statement, we have:

$$\begin{aligned}
B &= S \cdot (K \cdot S) \cdot K \\
C &= S \cdot (S \cdot (K \cdot (S \cdot (K \cdot S) \cdot K)) \cdot S) \cdot (K \cdot K) \\
K &= K \\
W &= S \cdot S \cdot (S \cdot K).
\end{aligned}$$

Going the other direction, we have:

$$\begin{aligned}
I &= W \cdot K \\
K &= K \\
S &= B \cdot (B \cdot (B \cdot W) \cdot C) \cdot (B \cdot B) \text{ and} \\
&= B \cdot (B \cdot W) \cdot (B \cdot B \cdot C).
\end{aligned}$$

We show the computations of B and S in detail.

$$\begin{aligned}
B \cdot a \cdot b \cdot c &= S \cdot (K \cdot S) \cdot K \cdot a \cdot b \cdot c \\
&= (K \cdot S) \cdot a \cdot (K \cdot a) \cdot b \cdot c \\
&= S \cdot (K \cdot a) \cdot b \cdot c \\
&= K \cdot a \cdot c \cdot (b \cdot c) \\
&= a \cdot (b \cdot c)
\end{aligned}$$

$$\begin{aligned}
S \cdot a \cdot b \cdot c &= B \cdot (B \cdot W) \cdot (B \cdot B \cdot C) \cdot a \cdot b \cdot c \\
&= (B \cdot W) \cdot ((B \cdot B \cdot C) \cdot a) \cdot b \cdot c \\
&= B \cdot W \cdot ((B \cdot B \cdot C) \cdot a) \cdot b \cdot c \\
&= W \cdot (((B \cdot B \cdot C) \cdot a) \cdot b) \cdot c \\
&= (((B \cdot B \cdot C) \cdot a) \cdot b) \cdot c \cdot c \\
&= B \cdot B \cdot C \cdot a \cdot b \cdot c \cdot c \\
&= B \cdot (C \cdot a) \cdot b \cdot c \cdot c \\
&= (C \cdot a) \cdot (b \cdot c) \cdot c \\
&= C \cdot a \cdot (b \cdot c) \cdot c \\
&= a \cdot c \cdot (b \cdot c)
\end{aligned}$$

□

If we use the notation $a^n \cdot b$ to mean $a \cdot a \cdot \dots \cdot a \cdot b$ where a is repeated n times, then we can terms which correspond to the Church numbers of lambda calculus:

$$\bar{n} \equiv (S \cdot B)^n \cdot (K \cdot I)$$

Definition 3.8.31. A partial function $f : \mathbb{N} \rightarrow \mathbb{N}$ is *representable* in combinatory logic if there is a term M_f such that $M_f \cdot \bar{n} = \bar{m}$ whenever $f(n) = m$ and $M_f \cdot \bar{n}$ does not have a normal form if $f(n) \uparrow$.

When we say that combinatory logic with S and K is complete, we mean the following theorem:

Theorem 3.8.32. *The partial functions that are representable in combinatory logic are exactly the partial recursive functions.*

Linear Combinatory Algebra

Definition 3.8.33. A *Linear Combinatory Algebra* $(A, \cdot, !)$ is an algebra A with an applicative binary operation \cdot , an unary operator $! : A \rightarrow A$ and eight distinguished elements: B, C, I, K, D, δ , F and W in A which satisfy the following rules:

1. $B \cdot a \cdot b \cdot c = a \cdot (b \cdot c)$
2. $C \cdot a \cdot b \cdot c = a \cdot c \cdot b$
3. $I \cdot a = a$
4. $K \cdot a \cdot !b = a$
5. $D \cdot !a = a$
6. $\delta \cdot !a = !!a$
7. $F \cdot !a \cdot !b = !(a \cdot b)$
8. $W \cdot a \cdot !b = a \cdot !b \cdot !b$

Note that a Linear Combinatory Algebra always contains a BCKW-Combinatory algebra.

Define $D' = C \cdot (B \cdot B \cdot I) \cdot (B \cdot D \cdot I)$ and the binary operator \bullet on A such that $a \bullet b \equiv a \cdot !b$.

Then, define the following:

$$\begin{aligned}
 B_s &= C \cdot (B \cdot (B \cdot B \cdot B) \cdot (D' \cdot I)) \cdot (C \cdot ((B \cdot B) \cdot F) \cdot \delta) \\
 C_s &= D' \cdot C \\
 K_s &= D' \cdot K \\
 W_s &= D' \cdot W.
 \end{aligned}$$

Lemma 3.8.34. *Given and Linear Combinatory Algebra $(A, \cdot, !)$, then (A, \bullet) is a BCKW-Combinatory algebra with B, C, K, W set to B_s, C_s, K_s, W_s from above.*

Proof. We show the calculation for K_s , the others are similar.

$$\begin{aligned}
K_s \bullet a \bullet b &\equiv D' \cdot K \cdot !a \cdot !b \\
&= C \cdot (B \cdot B \cdot I) \cdot (B \cdot D \cdot I) \cdot K \cdot !a \cdot !b \\
&= (B \cdot B \cdot I \cdot K) \cdot (B \cdot D \cdot I) \cdot !a \cdot !b \\
&= B \cdot (I \cdot K) \cdot (B \cdot D \cdot I) \cdot !a \cdot !b \\
&= (I \cdot K) \cdot ((B \cdot D \cdot I) \cdot !a) \cdot !b \\
&= K \cdot ((B \cdot D \cdot I) \cdot !a) \cdot !b \\
&= (B \cdot D \cdot I) \cdot !a \\
&= D \cdot (I \cdot !a) \\
&= D \cdot !a \\
&= a
\end{aligned}$$

□

Chapter 4

Quantum computation

4.1 Linear algebra

Quantum computation requires familiarity with the basics of linear algebra. This section will give definitions of the terms used throughout this thesis.

4.1.1 Basic definitions

The first definition needed is that of a *vector space*.

Definition 4.1.1 (Vector Space). Given a field F , whose elements will be referred to as scalars, a *vector space* over F is a non-empty set V with two operations, *vector addition* and *scalar multiplication*. *Vector addition* is defined as $+$: $V \times V \rightarrow V$ and denoted as $\mathbf{v} + \mathbf{w}$ where $\mathbf{v}, \mathbf{w} \in V$. The set V must be an abelian group under $+$. *Scalar multiplication* is defined as \cdot : $F \times V \rightarrow V$ and denoted as $c\mathbf{v}$ where $c \in F, \mathbf{v} \in V$. Scalar multiplication distributes over both vector addition and scalar addition and is associative. F 's multiplicative identity is an identity for scalar multiplication.

The specific algebraic requirements are:

1. $\forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in V, (\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$;
2. $\forall \mathbf{u}, \mathbf{v} \in V, \mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$;
3. $\exists \mathbf{0} \in V$ such that $\forall \mathbf{v} \in V, \mathbf{0} + \mathbf{v} = \mathbf{v}$;
4. $\forall \mathbf{u} \in V, \exists \mathbf{v} \in V$ such that $\mathbf{u} + \mathbf{v} = \mathbf{0}$;
5. $\forall \mathbf{u}, \mathbf{v} \in V, c \in F, c(\mathbf{u} + \mathbf{v}) = c\mathbf{u} + c\mathbf{v}$;

$$6. \forall \mathbf{u} \in V, c, d \in F, (c + d)\mathbf{u} = c\mathbf{u} + d\mathbf{u};$$

$$7. \forall \mathbf{u} \in V, c, d \in F, (cd)\mathbf{u} = c(d\mathbf{u});$$

$$8. \forall \mathbf{u} \in V, 1\mathbf{u} = \mathbf{u}.$$

Examples of vector spaces over F are: $F^{n \times m}$ – the set of $n \times m$ matrices over F ; and F^n – the n –fold Cartesian product of F . $F^{n \times 1}$, the set of $n \times 1$ matrices over F is also called the space of column vectors, while $F^{1 \times n}$, the set of row vectors. Often, F^n is identified with $F^{n \times 1}$.

This thesis shall identify F^n with the column vector space over F .

Definition 4.1.2 (Linearly independent). A subset of vectors $\{\mathbf{v}_i\}$ of the vector space V is said to be *linearly independent* when no finite linear combination of them, $\sum a_j \mathbf{v}_j$ equals $\mathbf{0}$ unless all the a_j are zero.

Definition 4.1.3 (Basis). A *basis* of a vector space V is a linearly independent subset of V that generates V . That is, any vector $u \in V$ is a linear combination of the basis vectors.

4.1.2 Matrices

As mentioned above, the set of $n \times m$ matrices over a field is a vector space. Additionally, matrices compose and the tensor product of matrices is defined.

Matrix composition is defined as usual. That is, for $A = [a_{ij}] \in F^{m \times n}$, $B = [b_{jk}] \in F^{n \times p}$:

$$AB = \left[\left(\sum_j a_{ij} b_{jk} \right)_{ik} \right] \in F^{m \times p}.$$

Definition 4.1.4 (Diagonal matrix). A *diagonal matrix* is a matrix where the only non-zero entries are those where the column index equals the row index.

The diagonal matrix $n \times n$ with only 1's on the diagonal is the identity for matrix multiplication, and is designated by I_n .

Definition 4.1.5 (Transpose). The *transpose* of an $n \times m$ matrix $A = [a_{ij}]$ is an $m \times n$ matrix A^t with the i, j entry being a_{ji} .

When the base field of a matrix is \mathbb{C} , the complex numbers, the *conjugate transpose* (also called the *adjoint*) of an $n \times m$ matrix $A = [a_{ij}]$ is defined as the $m \times n$ matrix A^* with the i, j entry being \bar{a}_{ji} , where \bar{a} is the complex conjugate of $a \in \mathbb{C}$.

When working with column vectors over \mathbb{C} , note that $\mathbf{u} \in \mathbb{C}^n \implies \mathbf{u}^* \in \mathbb{C}^{1 \times n}$ and that $\mathbf{u}^* \times \mathbf{u} \in \mathbb{C}^{1 \times 1}$. This thesis will use the usual identification of \mathbb{C} with $\mathbb{C}^{1 \times 1}$. A column vector \mathbf{u} is called a *unit vector* when $\mathbf{u}^* \times \mathbf{u} = 1$.

Definition 4.1.6 (Trace). The *trace*, $Tr(A)$ of a square matrix $A = [a_{ij}]$ is $\sum a_{ii}$.

Tensor Product

The tensor product of two matrices is the usual Kronecker product:

$$U \otimes V = \begin{bmatrix} u_{11}V & u_{12}V & \cdots & u_{1m}V \\ u_{21}V & u_{22}V & \cdots & u_{2m}V \\ \vdots & \vdots & \ddots & \\ u_{n1}V & u_{n2}V & \cdots & u_{nm}V \end{bmatrix} = \begin{bmatrix} u_{11}v_{11} & \cdots & u_{12}v_{11} & \cdots & u_{1m}v_{1q} \\ u_{11}v_{21} & \cdots & u_{12}v_{21} & \cdots & u_{1m}v_{2q} \\ \vdots & \vdots & \vdots & \ddots & \\ u_{n1}v_{p1} & \cdots & u_{n2}v_{p1} & \cdots & u_{nm}v_{pq} \end{bmatrix}$$

Special matrices

When working with quantum values certain types of matrices over the complex numbers are of special interest. These are:

Unitary Matrix : Any $n \times n$ matrix A with $AA^* = I$ ($= A^*A$).

Hermitian Matrix : Any $n \times n$ matrix A with $A = A^*$.

Positive Matrix : Any Hermitian matrix A in $\mathbb{C}^{n \times n}$ where $\mathbf{u}^* A \mathbf{u} \geq 0$ for all vectors $\mathbf{u} \in \mathbb{C}^n$.

Note that for any Hermitian matrix A and vector u , $\mathbf{u}^* A \mathbf{u}$ is real.

Completely Positive Matrix : Any positive matrix A in $\mathbb{C}^{n \times n}$ where $I_m \otimes A$ is positive.

The matrix

$$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

is an example of a matrix that is *unitary*, *Hermitian*, *positive* and *completely positive*.

Superoperators

A *Superoperator* S is a matrix over \mathbb{C} with the following restrictions:

1. S is *completely positive*. This implies that S is positive as well.
2. For all positive matrices A , $Tr(S A) \leq Tr(A)$.

4.2 Quantum computation overview

Quantum computation proceeds via the application of reversible transformations — Unitary transformations.

The semantics of quantum computation can be defined as a \dagger -compact closed category as introduced in [2, 4] and completely positive maps as discussed in [32].

Definition 4.2.1 (Dagger Category). A *Dagger Category* [32] is a category \mathbb{C} together with an operation \dagger that is an involutive, identity on objects, contra-variant endofunctor on \mathbb{C} .

Recalling first that a *symmetric monoidal category* is a category \mathbb{B} with a bi-functor \otimes , an object I and natural isomorphisms:

$$a_{A,B,C} : (A \otimes B) \otimes C \rightarrow A \otimes (B \otimes C)$$

$$c_{A,B} : A \otimes B \rightarrow B \otimes A$$

$$ul_A : A \rightarrow I \otimes A$$

with standard coherence conditions, as in [22]. Note that we also have a map $ur_A : A \rightarrow A \otimes I$ given by $ur_A = ul_{A \otimes I, A}$. Furthermore, a *compact closed category* \mathbb{C} is a symmetric monoidal

category where each object A has a dual A^* together with the maps:

$$\eta_A : I \rightarrow A^* \otimes A$$

$$\epsilon_A : A \otimes A^* \rightarrow I$$

such that

$$\begin{array}{ccc} A & \xrightarrow{ur_A} A \otimes I \xrightarrow{A \otimes \eta_A} A \otimes (A^* \otimes A) & \text{and} & A^* & \xrightarrow{ul_{A^*}} I \otimes A^* \xrightarrow{\eta_{A^*} \otimes A^*} (A^* \otimes A) \otimes A^* \\ & \searrow & & \searrow & \\ & & \downarrow a^{-1} & & \downarrow a \\ & & (A \otimes A^*) \otimes A & & A^* \otimes (A \otimes A^*) \\ & & \downarrow \epsilon \otimes A & & \downarrow A^* \otimes \epsilon \\ & & I \otimes A & & A^* \otimes I \\ & & \downarrow ul^{-1} & & \downarrow ur^{-1} \\ & & A & & A \end{array}$$

From the above, we can define a *Dagger symmetric monoidal category* and a *Dagger compact closed category*. The latter is referred to as a *strongly compact closed category* in [2], where they were initially introduced. In each case, the \dagger functor is added in a way that retains coherence with the bi-functor \otimes and with the dualizing operator. The coherence implies that the $i^\dagger = i^{-1}$ for the SMC isomorphisms, that $(f \otimes g)^\dagger = f^\dagger \otimes g^\dagger$ for all maps f, g in the symmetric monoidal category and that

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

commutes for all objects A in the compact closed category.

Example 4.2.2 (REL). REL is a dagger compact closed category with the dual of an object A is A , \otimes is the cartesian product and for $R : A \rightarrow B$, we have $R^* = R^\dagger = \{(y, x) | (x, y) \in R\}$.

Example 4.2.3 (FDHILB). The category of finite dimensional Hilbert spaces, FDHILB is a dagger compact closed category with the dual of an object H is the normal Hilbert space dual H^* , the space of continuous linear functions from H to the base field. \otimes is the normal Hilbert space tensor and for $f : A \rightarrow B$, we have f^\dagger is the unique map such that $\langle fx | y \rangle = \langle y | f^\dagger x \rangle$ for all $x \in A, y \in B$.

Additionally, if one has a dagger compact closed category with biproducts where the biproducts and dagger interact such that $p_i^\dagger = q_i$, this is called a *biproduct dagger compact closed category*.

In [32], the author continues from this point: Starting with a biproduct dagger compact closed category \mathbb{C} , he creates a new category, $\text{CPM}(\mathbb{C})$ which has the same objects as \mathbb{C} , but morphisms $f : A \rightarrow B$ in $\text{CPM}(\mathbb{C})$ are given by maps $f : A^* \otimes A \rightarrow B^* \otimes B$ in \mathbb{C} which are *completely positive*. Note that REL and FDHILB are biproduct dagger compact closed categories.

From this, the category $\text{CPM}(\mathbb{C})^\oplus$, the free biproduct completion of $\text{CPM}(\mathbb{C})$ is formed, which is suitable for describing quantum computation semantics. For example, given FDHILB as our starting point, the tensor unit I is the field of complex numbers. The type of **qubit** (in FDHILB and by lifting, in $\text{CPM}(\text{FDHILB})^\oplus$) is given as $I \oplus I$. At this stage, the necessity of the CPM construction to model physical reality can be seen in the following as in FDHILB , the morphisms initialization of a qubit: $init : I \oplus I \rightarrow \mathbf{qubit}$ and destructive measure: $meas : \mathbf{qubit} \rightarrow I \oplus I$ are inverses. However, in $\text{CPM}(\text{FDHILB})^\oplus$, these same maps are given as

$$\mathbf{qubit}^* \otimes \mathbf{qubit} \xrightarrow{meas} I \oplus I \xrightarrow{init} \mathbf{qubit}^* \otimes \mathbf{qubit}$$

by the formulae:

$$meas \begin{pmatrix} a & b \\ c & d \end{pmatrix} = (a, d), \quad init(a, d) = \begin{pmatrix} a & 0 \\ 0 & d \end{pmatrix}.$$

Therefore, the maps are not inverses and reflect the physical reality.

Example 4.2.4 (Commutative Frobenius algebras). Let \mathbb{X} be a symmetric monoidal category and form $\text{CFrob}(\mathbb{X})$ as follows:

Objects: Commutative Frobenius algebras[20]: A quintuple $(X, \nabla, \eta, \Delta, \epsilon)$ where X is a k -algebra for some field k , and $\nabla : A \otimes A \rightarrow A$, $\eta : k \rightarrow A$, $\Delta : A \rightarrow A \otimes A$, $\epsilon : A \rightarrow k$ are

natural maps in the algebra. Additionally, these satisfy

$$\begin{array}{ccc}
A \otimes A & \xrightarrow{\Delta \otimes 1} & A \otimes (A \otimes A) \\
\downarrow 1 \otimes \Delta & \searrow \nabla & \downarrow 1 \otimes \nabla \\
(A \otimes A) \otimes A & \xrightarrow{\nabla \otimes 1} & A \otimes A
\end{array}$$

together with the additional property that $\Delta \nabla = 1$.

Maps: Multiplication (∇) and co-multiplication (Δ) preserving homomorphisms which do not necessarily preserve the unit.

Theorem 4.2.5. *When \mathbb{X} is a symmetric monoidal category, $CFrob(\mathbb{X})$ is a discrete inverse category.*

Proof. For $f : X \rightarrow Y$, define $f^{(-1)}$ as

$$Y \xrightarrow{1 \otimes \eta} Y \otimes X \xrightarrow{1 \otimes \Delta} Y \otimes X \otimes X \xrightarrow{1 \otimes f \otimes 1} Y \otimes Y \otimes X \xrightarrow{\nabla \otimes 1} Y \otimes X \xrightarrow{\epsilon \otimes 1} X$$

Using a result from [27], we need only show:

$$(f^{(-1)})^{(-1)} = f$$

$$f f^{(-1)} f = f$$

$$f f^{(-1)} g g^{(-1)} = g g^{(-1)} f f^{(-1)}$$

We also use the following two identities from [20]:

$$(1 \otimes \eta) \nabla = id \tag{4.1}$$

$$\Delta(1 \otimes \epsilon) = id. \tag{4.2}$$

$$\begin{aligned}
f^{(-1)^{(-1)}} &= (1 \otimes \eta)(1 \otimes \Delta)(1 \otimes (f^{(-1)} \otimes 1)(\nabla \otimes 1)(\epsilon \otimes 1) \\
&= (1 \otimes \eta)(1 \otimes \Delta)(1 \otimes ((1 \otimes \eta)(1 \otimes \Delta)(1 \otimes f \otimes 1)(\nabla \otimes 1)(\epsilon \otimes 1)) \otimes 1) \\
&\quad (\nabla \otimes 1)(\epsilon \otimes 1) \\
&= (1 \otimes \eta)(1 \otimes \Delta)(1 \otimes 1 \otimes \eta)(1 \otimes 1 \otimes f \otimes 1 \otimes 1)(1 \otimes \nabla \otimes 1 \otimes 1) \\
&\quad (1 \otimes \epsilon \otimes 1 \otimes 1)(\nabla \otimes 1)(\epsilon \otimes 1) \\
&= (\eta \otimes 1)(\Delta \otimes 1)(1 \otimes \nabla)(f \otimes 1)((\eta)(\Delta \otimes 1)(1 \otimes \nabla)(1 \otimes \epsilon)) \otimes 1)((1 \otimes \epsilon) \\
&= (1 \otimes \eta)\nabla\Delta(1 \otimes \epsilon)f(\eta \otimes 1)\nabla\Delta(1 \otimes \epsilon) \\
&= id_x id_x f id_y id_y \\
&= f
\end{aligned}$$

$$\begin{aligned}
ff^{(-1)}f &= f(1 \otimes \eta)(1 \otimes \Delta)(1 \otimes f \otimes 1)(\nabla \otimes 1)(\epsilon \otimes 1)f \\
&= (1 \otimes \eta)(1 \otimes \Delta)(f \otimes f \otimes 1)(\nabla \otimes 1)(1 \otimes f)(\epsilon \otimes 1) \\
&= (1 \otimes \eta)(1 \otimes \Delta)(\nabla \otimes 1)(f \otimes f)(\epsilon \otimes 1) \\
&= (1 \otimes \eta)\nabla\Delta(f \otimes f)(\epsilon \otimes 1) \\
&= \Delta(f \otimes f)(\epsilon \otimes 1) \\
&= f\Delta(\epsilon \otimes 1) \\
&= f
\end{aligned}$$

Finally, to show $ff^{(-1)}$ and $gg^{(-1)}$ commute:

$$\begin{aligned}
& f(1 \otimes \eta)(1 \otimes \Delta)(1 \otimes f \otimes 1)(\nabla \otimes 1)(\epsilon \otimes 1)g(1 \otimes \eta)(1 \otimes \Delta)(1 \otimes g \otimes 1)(\nabla \otimes 1)(\epsilon \otimes 1) \\
&= (1 \otimes \eta)(1 \otimes \Delta)(\nabla \otimes 1)(f \otimes 1)(\epsilon \otimes 1)(1 \otimes \eta)(1 \otimes \Delta)(\nabla \otimes 1)(g \otimes 1)(\epsilon \otimes 1) \\
&= (1 \otimes \eta)\nabla\Delta(f \otimes 1)(\epsilon \otimes 1)(1 \otimes \eta)\nabla\Delta(g \otimes 1)(\epsilon \otimes 1) \\
&= \Delta(f \otimes 1)(\epsilon \otimes 1)\Delta(g \otimes 1)(\epsilon \otimes 1) \\
&= \Delta(1 \otimes \Delta)(f \otimes g \otimes 1)(\epsilon \otimes \epsilon \otimes 1) \\
&= \Delta(1 \otimes \Delta)(g \otimes f \otimes 1)(\epsilon \otimes \epsilon \otimes 1) \quad \text{co-commutativity} \\
&= gg^{(-1)}ff^{(-1)}
\end{aligned}$$

□

4.2.1 Density matrix representation

An alternate representation of quantum states, both pure and mixed, is via *density matrices*. If the state of a system is represented by some column vector u , then the matrix uu^* is its density matrix. Note that if $u = \nu v$ for some complex scalar ν with norm 1, then $uu^* = (\nu v)(\nu v)^* = \nu \bar{\nu} vv^* = vv^*$. For the mixed state $\sum \nu_i \{v_i\}$, the density matrix is $\sum \nu_i v_i v_i^*$. Density matrices are positive hermitian matrices with trace ≤ 1 . Note that the trace of the density matrix is the probability the system has reached this particular value in the computation.

The result of applying the unitary transform U to a state u represented by the density matrix A is UAU^* . The measurement operation on a density matrix is derived from the measurement effects on the **qubit**. For example, consider the density matrix for $q = \alpha |0\rangle + \beta |1\rangle$, $\begin{pmatrix} \alpha \bar{\alpha} & \alpha \bar{\beta} \\ \beta \bar{\alpha} & \beta \bar{\beta} \end{pmatrix}$. Measuring this **qubit** gives either $\begin{pmatrix} \alpha \bar{\alpha} & 0 \\ 0 & 0 \end{pmatrix}$ with probability $|\alpha|^2$ or $\begin{pmatrix} 0 & 0 \\ 0 & \beta \bar{\beta} \end{pmatrix}$ with probability $|\beta|^2$. If the results of the measurement are not used, this will result

in the density matrix $\begin{pmatrix} \alpha\bar{\alpha} & 0 \\ 0 & \beta\bar{\beta} \end{pmatrix}$. This extends linearly so that if a **qubit** is measured in

the system whose density matrix is $\left(\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right)$, the result will be the mixed density matrix

$$\left(\begin{array}{c|c} A & 0 \\ \hline 0 & D \end{array} \right).$$

It is possible to create a complete partial order on density matrices.

Definition 4.2.6 (Löwner partial order). For square complex matrices A, B of the same size, define $A \leq B$ if $B - A$ is positive.

Lemma 4.2.7. *Designate D_n to be the density matrices of size $n \times n$, then the poset (D_n, \leq) is a complete partial order.*

Proof. See [30], pp 13–14. □

4.3 Dagger categories

Dagger categories generalize the concepts of Hilbert spaces that are required to model quantum computation. These were introduced in [3] as *strongly compact closed categories*, an additional structure only on compact closed categories.

4.3.1 Definitions

Although dagger categories were introduced in the context of compact closed categories, the concept of a dagger is definable independently. This was first done in [32].

Definition 4.3.1 (Dagger, dagger category). A dagger operator on a category \mathbf{D} is an involutive, identity on objects contravariant functor $\dagger : \mathbf{D} \rightarrow \mathbf{D}$. A dagger category is a category that has a dagger operator.

Typically, the dagger is written as a superscript on the morphism. So, if $f : A \rightarrow B$ is a map in \mathbf{D} , then $f^\dagger : B \rightarrow A$ is a map in \mathbf{D} and is called the *adjoint* of f . A map where $f^{-1} = f^\dagger$ is called *unitary* and a map $f : A \rightarrow A$ with $f = f^\dagger$ is called *self-adjoint* or *hermitian*.

Definition 4.3.2 (Dagger symmetric monoidal). A *dagger symmetric monoidal category* is a symmetric monoidal category \mathbf{D} with a dagger operator such that the dagger interacts coherently with the monoid to preserve the symmetric monoidal structure.

The coherence requirements in definition 4.3.2 are in addition to the standard coherence diagrams for a symmetric monoidal category. The additional coherence requirements are for all maps $f : A \rightarrow B$ and $g : C \rightarrow D$, it is required that $(f \otimes g)^\dagger = f^\dagger \otimes g^\dagger : B \otimes D \rightarrow A \otimes C$ and that the monoid structure isomorphisms $a_{A,B,C} : (A \otimes B) \rightarrow C$, $u_A : A \rightarrow I \otimes A$ and $c_{A,B} : A \otimes B \rightarrow B \otimes A$ are all unitary.

Definition 4.3.3 (Dagger compact closed). A *dagger compact closed category* \mathbf{D} is a dagger symmetric monoidal category that is compact closed where the diagram

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

commutes for all objects A in \mathbf{D} .

A category \mathbf{D} is said to have *finite biproducts* when it has a zero object $\mathbf{0}$ (an object that is both initial and terminal) and when each pair of objects A, B have a biproduct $A \oplus B$. In such a category the unique map $A \rightarrow \mathbf{0} \rightarrow B$ is designated as $\mathbf{0}_{A,B}$.

Note that a category with finite biproducts is enriched in commutative monoids, where if $f, g : A \rightarrow B$, define $f + g : A \rightarrow B$ as $\langle id_A, id_A \rangle (f \oplus g) [id_B, id_B]$. The unit for the addition is $\mathbf{0}_{A,B}$. In the future, $\langle id, id \rangle$ will be designated by Δ and $[id, id]$ will be designated by ∇ .

Lemma 4.3.4. *If \mathbf{D} is a dagger category with biproducts, with injections in_1, in_2 and projections p_1, p_2 , then the following are equivalent.*

1. $p_i^\dagger = in_i, i = 1, 2$,
2. $(f \oplus g)^\dagger = f^\dagger \oplus g^\dagger$ and $\Delta^\dagger = \nabla$,
3. $\langle f, g \rangle^\dagger = [f^\dagger, g^\dagger]$,
4. the below diagram commutes.

$$\begin{array}{ccc} A^\dagger \oplus B^\dagger & & \\ \downarrow id & \searrow [p_1^\dagger, p_2^\dagger] & \\ A \oplus B & \xrightarrow{id} & (A \oplus B)^\dagger \end{array}$$

Proof. **1** \implies **2** To show $\Delta^\dagger = \nabla$, draw the product cone for Δ ,

$$\begin{array}{ccccc} & & A & & \\ & \swarrow id & \downarrow \Delta & \searrow id & \\ A & \xleftarrow{p_1} & A \oplus A & \xrightarrow{p_2} & A \end{array}$$

and apply the dagger functor to it. As $p_i^\dagger = in_i$, and \dagger is identity on objects, this is now a coproduct diagram and therefore $\Delta^\dagger = \nabla$.

For $(f \oplus g)^\dagger = f^\dagger \oplus g^\dagger$, start with the diagram defining $f \oplus g$ as a product of the arrows:

$$\begin{array}{ccccc} A & \xleftarrow{p_1} & A \oplus B & \xrightarrow{p_2} & A \\ \downarrow f & & \downarrow f \otimes g & & \downarrow g \\ C & \xleftarrow{p_1} & C \oplus D & \xrightarrow{p_2} & D \end{array}$$

and once again apply the dagger functor. This is now the diagram defining the coproduct of maps and therefore $(f \oplus g)^\dagger = f^\dagger \oplus g^\dagger$.

2 \implies **3** The calculation showing this is

$$\begin{aligned} [f^\dagger, g^\dagger] &= \nabla; (f^\dagger \oplus g^\dagger) \\ &= \Delta^\dagger; (f^\dagger \oplus g^\dagger) \\ &= \Delta^\dagger; (f \oplus g)^\dagger \\ &= ((f \oplus g); \Delta)^\dagger \\ &= \langle f, g \rangle^\dagger \end{aligned}$$

3 \implies **4** Under the assumption,

$$\begin{aligned} [p_1^\dagger, p_2^\dagger] &= \langle p_1, p_2 \rangle^\dagger \\ &= id^\dagger \\ &= id \end{aligned}$$

and therefore the diagram commutes.

4 \implies **1** Using the injections and under the assumption, the following diagram commutes:

$$\begin{array}{ccc} A^\dagger \oplus B^\dagger & \xrightarrow{[in_1, in_2]} & A^\dagger \oplus B^\dagger \\ \downarrow id & \searrow [p_1^\dagger, p_2^\dagger] & \downarrow id \\ A \oplus B & \xrightarrow{id} & (A \oplus B)^\dagger \end{array}$$

and therefore, $p_1^\dagger = in_1$ and $p_2^\dagger = in_2$.

□

Definition 4.3.5. A *biproduct dagger compact closed category* is a dagger compact closed category with biproducts where the conditions of lemma 4.3.4 hold.

4.3.2 Examples of dagger categories

FDHILB The category of finite dimensional Hilbert spaces is the motivating example for the creation of the dagger and is, in fact, a biproduct dagger compact closed category. The biproduct is the direct sum of Hilbert spaces and the tensor for compact closure is the standard tensor of Hilbert spaces. The dual H^* of a space H is the space of all continuous linear functions from H to the base field. The dagger is defined via the adjoint as being the unique map $f^\dagger : B \rightarrow A$ such that $\langle fa|b \rangle = \langle a|f^\dagger b \rangle$ for all $a \in A, b \in B$.

REL The category **REL** of sets and relations has the tensor $S \otimes T = S \times T$, the cartesian product and the biproduct $S \oplus T = S + T$, the disjoint union. This is compact closed under $A^* = A$ and the dagger is the relational converse, that is if the relation $R = \{(s, t) | s \in S, t \in T\} : S \rightarrow T$, then $R^\dagger = \{(t, s) | (s, t) \in R\} (= R^*)$.

Inverse categories An inverse category \mathbb{X} is also a dagger category when the dagger is defined as the partial inverse. The unitary maps are the total maps which are isomorphisms. If the inverse category \mathbb{X} is also a symmetric monoidal category where the monoid \oplus is actually a restriction bi-functor, then \mathbb{X} is a dagger symmetric monoidal category. This follows from

$$(f \otimes g)(f \otimes g)^{(-1)} = \overline{f \otimes g} = \bar{f} \otimes \bar{g} = f f^{(-1)} \otimes g g^{(-1)} = (f \otimes g)(\bar{f} \otimes \bar{g})$$

but since the partial inverse of $f \otimes g$ is unique, $\overline{f \otimes g} = \bar{f} \otimes \bar{g}$. Finally, since all the structure isomorphisms are total maps, they are unitary and \mathbb{X} is a dagger symmetric monoidal restriction category.

Frobenius algebra

Definition 4.3.6 (Frobenius algebra). Given a symmetric monoidal category \mathbf{D} , a *Frobenius algebra* is an object X of \mathbf{D} and four maps, $\nabla : X \otimes X \rightarrow X$, $e : I \rightarrow X$, $\Delta : X \rightarrow X \otimes X$ and $\epsilon : X \rightarrow I$, with the conditions that (X, ∇, e) forms a commutative monoid, (X, Δ, ϵ) forms a commutative comonoid and the diagram

$$\begin{array}{ccccc}
 X \otimes X & \xrightarrow{X \otimes \Delta} & X \otimes X \otimes X & & \\
 \Delta \otimes X \downarrow & \searrow \nabla & & \searrow \nabla \otimes X & \\
 & X & & & \\
 & \searrow \Delta & & & \\
 X \otimes X \otimes X & \xrightarrow{X \otimes \nabla} & X \otimes X & &
 \end{array}$$

commutes. The Frobenius algebra is *special* when $\Delta; \nabla = 1_X$ and *commutative* when $\Delta; c_{X,X} = \Delta$

Definition 4.3.7 (\dagger -Frobenius algebra). A Frobenius algebra in a dagger symmetric monoidal category where $\Delta = \nabla^\dagger$ and $\epsilon = e^\dagger$ is a \dagger -Frobenius algebra.

For an example of a \dagger -Frobenius algebra, consider a finite dimensional Hilbert space H with an orthonormal basis $\{|\phi_i\rangle\}$ and define $\Delta : H \rightarrow H \otimes H : |\phi_i\rangle \mapsto |\phi_i\rangle \otimes |\phi_i\rangle$ and $\epsilon : H \rightarrow \mathbb{C} : |\phi_i\rangle \mapsto 1$. Then $(H, \nabla = \Delta^\dagger, e = \epsilon^\dagger, \Delta, \epsilon)$ forms a commutative special \dagger -Frobenius algebra.

4.4 Semantics of quantum computation

4.4.1 Semantics of QPL

QPL Basics

In [30] Dr. Selinger provides a denotational semantics for a quantum programming, QPL, with the slogan of “quantum data with classical control”. This slogan refers to the semantic representation described in the paper, where explicit classical branching based on a classical value is described by a **bit** value with specific probabilities of being 0 or 1.

QPL is defined via a collection of functional flowchart components, where “functional” specifically means that each flowchart is a function from its inputs to its outputs. These components describe the basic operations on **bits** and **qubits**. Edges between the components represent the data (**bits** and **qubits**). These edges are labelled with a typing context and annotated with a tuple of density matrices, describing the probability distribution of the classical data and the state of the quantum data. In the case of purely classical data, this annotation will be a tuple of probabilities, whereas in the case of purely quantum data, it will be a single density matrix.

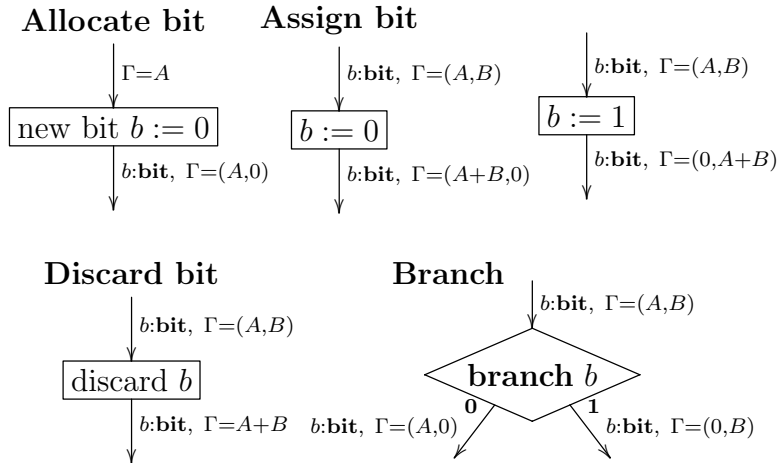


Figure 4.1: Classical flowcharts

In figure [figure 4.1](#), the annotation Γ consists of a tuple of probabilities, with n **bits**

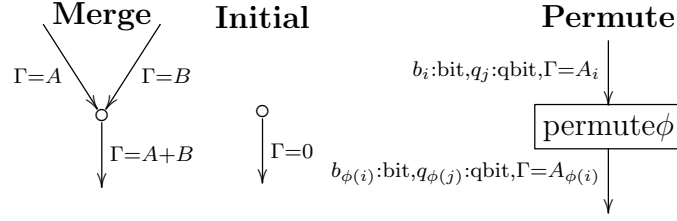


Figure 4.2: General flowcharts

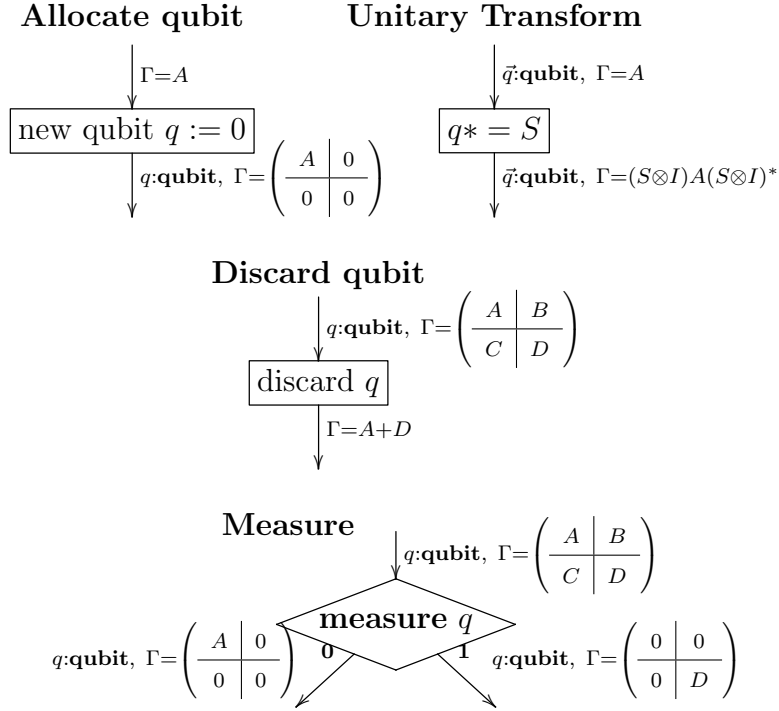


Figure 4.3: Quantum flowcharts

requiring 2^n probabilities for their description. In figure [figure 4.3](#), Γ will consist of a density matrix of size $2^m \times 2^m$ for m **qubits**. Note also that in figure [figure 4.3](#), the notation \vec{q} indicates an ordered set of **qubits**.

In QPL, the classical operations consist of: *Allocate bit*, *Assignment*, *Discard bit* and *Branch*. The quantum operations are: *Allocate qubit*, *Unitary Transform*, *Discard qubit* and *Measure*. The operations applicable to both types of data are *Merge*, *Initial* and *Permute*. These are found in Figure [figure 4.2](#).

When components are combined, the type annotation Γ consists of a tuple of density

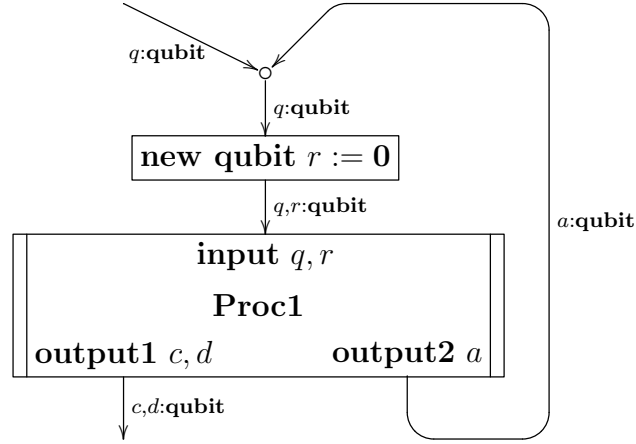


Figure 4.4: Example of a subroutine and loop

matrices. Flowchart components must be combined so that they are connected via edges with identical typing judgements. Flowcharts may have arbitrary numbers of input and output edges. By convention, component flow is from the top down and programs are read in the same manner.

The semantics of a component is the function that calculates the matrix tuple(s) of the output edges when given the matrix tuple of the input edges. Each of these functions is linear and preserves adjoints. They also preserve positivity and the sum of the traces of the output edges equals the sum of the traces of the input edges, which can be viewed as the probability of leaving a fragment is the same as the probability of entering a fragment.

Looping, subroutines and recursion

In the flow chart representation of QPL, looping occurs when one edge is connected to a component above the component originating the edge. Subroutines are represented by boxes with double left and right lines. A subroutine may have multiple input and output edges and is considered shorthand for the flowchart making up the subroutine. For example, see figure [figure 4.4](#), where the subroutine *Proc1* accepts two **qubits** q, r as input and produces either two **qubits** c, d or a single **qubit** q . In the case when the output is the single **qubit** q , the output is looped back to be merged with the original input.

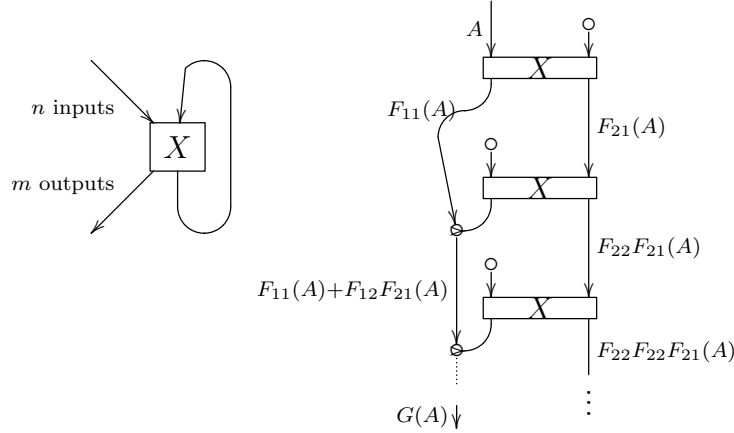


Figure 4.5: Unwinding a loop

Semantics of looping is based on “infinite unwinding”. It is interesting to note this is similar to the method used in [21] where a **while** program construction is unwound to

if $\neg B$ **then** I **else** S ; **while** B **do** S **od** **fi**;

and the semantics of **while** is given as a fixpoint W of the equation $W = e_{\neg B} + (e_B; S; W)$. Referring to figure 4.5, the input to X is $n + k$ density matrices, the output is $m + k$ density matrices, where the k matrices partake in the loop. This can be written as $F(A, C) = (B, D)$ with $A = (A_1, \dots, A_n)$, $B = (B_1, \dots, B_m)$, where F is the linear function giving the semantics of X . This allows creation of four component functions, where $F(A, 0) = (F_{11}(A), F_{21}(A))$ and $F(0, C) = (F_{12}(C), F_{22}(C))$.

Following the right hand side of figure 4.5, the state of the edges at the end are given by

$$G(A) = F_{11}(A) + \sum_{i=0}^{\infty} F_{12}(F_{22}^i(F_{21}(A))) \quad (4.3)$$

I will show later that this is a convergent sum.

The semantics of subroutines without recursion is the same as “in-lining” the subroutine at the place of its call. The first requirement for this is that the program handle renaming of variables as the formal parameters of the subroutine may have different names than the

calling parameters. For **bits**, renaming b to c may be accomplished by the fragment

```

new bit  $c := 0$ ;
branch  $b$   $0 \{ \}$   $1 \{ c := 1 \}$  ;
discard  $b$ .

```

For **qubits**, renaming q to r is done by the fragment

```

new qubit  $p := 0$ ;
 $q, p \text{ *=CNOT}$ ;
 $p, q \text{ *=CNOT}$ ;
discard  $q$ .

```

The second requirement is that the program needs to be able to extend the semantic context. That is, suppose that a subroutine X is defined with input typing Γ , with semantic function F . This means that starting with $\Gamma = A$, applying the subroutine X gives us the typing and context $\Gamma' = F(A)$. Inlining a subroutine requires that the addition of an arbitrary number of **bits** and **qubits** to the context and be able to derive the semantics. But, since each of the components of flow charts and looping are linear functions, this is a straightforward induction on the structure of the subroutine. The proof for one of the cases is below.

Lemma 4.4.1 (Context extension). *Given a subroutine X in context $\Gamma = A$ with semantics F (i.e., applying X to $\Gamma = A$ gives $\Gamma' = F(A)$),*

- *The result of X in context $b : \text{bit}$, $\Gamma = (A, B)$ is $b : \text{bit}$, $\Gamma' = (F(A), F(B))$.*
- *The result of X in context $q : \text{qubit}$, $\Gamma = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right)$ is $q : \text{qubit}$, $\Gamma' = \left(\begin{array}{c|c} F(A) & F(B) \\ \hline F(C) & F(D) \end{array} \right)$.*

Proof. **Case $X = \text{“allocate bit”}$** The semantics of allocate bit is $F(A) = (A, 0)$, where the number of 0 density matrices is the same as the number of density matrices in A . When the additional context is a **bit**, the starting context is $x : \text{bit}$, $\Gamma = (A, B)$, where

again the number and dimensions of density matrices in A and B agree. After applying X , $b : \text{bit}, x : \text{bit} \Gamma' = (A, B, 0, 0)$. Next, permute x and b , to retain x in the correct order and get $x : \text{bit}, b : \text{bit} \Gamma' = (A, 0, B, 0) = ((A, 0), (B, 0)) = (F(A), F(B))$.

When the additional context is a **qubit**, the starting context is $x : \text{qubit}, \Gamma = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right)$.

After allocation of a bit,

$$\begin{aligned} b : \text{bit}, x : \text{qubit}, \Gamma &= \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}, \begin{array}{c|c} 0 & 0 \\ \hline 0 & 0 \end{array} \right) \\ &= \left(\begin{array}{c|c} (A, 0) & (B, 0) \\ \hline (C, 0) & (D, 0) \end{array} \right) \\ &= \left(\begin{array}{c|c} F(A) & F(B) \\ \hline F(C) & F(D) \end{array} \right). \end{aligned}$$

□

As the semantics of each of the components of the flowchart language are linear functions, the technique used in the example case in Lemma 4.4.1 is applicable for each of these flowchart components. Furthermore, as the semantics are compositional, this extends to looping.

For recursive subroutines, a variant of the infinite unwinding is used. A recursive subroutine is one that calls itself in some way. If we have a subroutine X , let $X(Y)$ be the flowchart defined as X with the recursive call to itself replaced with a call to Y . As the semantics are compositional, there is some function Θ such that give the semantics of $X(Y)$ from the semantics of Y . Let Y_0 be a non-terminating program and define Y_i by the equation $Y_{i+1} = X(Y_i)$. Denote the semantics of Y_i by F_i . Then $F_0 = 0$ and $F_{i+1} = \Theta(F_i)$. From this, define

$$X = \lim_{i \rightarrow \infty} F_i. \tag{4.4}$$

The existence of this limit will be discussed below in section [sub-section 4.4.1](#).

An important point to note here is that the semantics of an arbitrary G may actually reduce the trace, that is, there may be a non-zero chance the program will not terminate.

Categorical semantics of QPL

While the exposition above referred to the functions described in the flowchart components as the semantics of the program, this section will give a formal definition of the categorical semantics.

Definition 4.4.2 (Signature). A *signature* is a list of positive non-zero integers, $\sigma = n_1, \dots, n_s$ which is associated with the complex vector space $V_\sigma = \mathbb{C}^{n_1 \times n_1} \times \dots \times \mathbb{C}^{n_s \times n_s}$

Designate the elements of V_σ by tuples of matrices, $A = (A_1, \dots, A_s)$. The trace of A will be the sum of the traces of the tuple matrices. A will be said to have a specific property when all matrices in the tuple have that property, e.g., positive, hermitian.

From the above, now define the category V with objects being signatures σ and maps from σ to τ being any complex linear function from V_σ to V_τ . Define \oplus by concatenation of signatures. Then, \oplus is both a product and coproduct in V . The co-pair map $[F, G] : \sigma \oplus \sigma' \rightarrow \tau$ is defined as $[F, G](A, B) = F(A) + F(B)$, while the pairing map $\langle F, G \rangle : \sigma \rightarrow \tau \oplus \tau'$ is defined as $\langle F, G \rangle(A) = (FA, GA)$. Additionally, define the tensor \otimes on $\sigma = n_1, \dots, n_s$ and $\tau = m_1, \dots, m_t$ as

$$\sigma \otimes \tau = n_1 m_1, n_1 m_2, \dots, n_s, m_t.$$

This tensor, together with the unit $I = 1$ makes V a symmetric monoidal category. Note that it is also distributive with $\tau \otimes (\sigma \oplus \sigma') = (\tau \otimes \sigma) \oplus (\tau \otimes \sigma')$.

As V is equivalent to the category of finite dimensional vector spaces, we will need to restrict the morphisms to those that can occur as programs in QPL. V has too many morphisms, for instance the signature $1, 1$ (which will be designated as **bit**) is isomorphic to the signature 2 (which will be designated as **qubit**).

Definition 4.4.3 (Superoperator). Given $F : V_\sigma \rightarrow V_\tau$, define:

- F as *positive* if $F(A)$ is positive for all positive A ;
- F as *completely positive* if $id_\rho \otimes F : V_{\rho \otimes \sigma} \rightarrow V_{\rho \otimes \tau}$ is positive for all ρ ;
- F as a *superoperator* if it is completely positive and $\text{tr } F(A) \leq \text{tr } A$ for all positive A .

The definition of a superoperator is trace *non-increasing* rather than trace *preserving* due to the possibility of non-termination in programs.

Considering superoperators in the category V , there a number of properties that hold. It is immediate to see that an identity map is a superoperator and that compositions of superoperators are again superoperators. The canonical injections $i_1 : \sigma \rightarrow \sigma \oplus \tau$ and $i_2 : \tau \rightarrow \sigma \oplus \tau$ are superoperators. The remain properties of interest are detailed in the following lemma.

Lemma 4.4.4. *In the category V , the following hold:*

1. *If $F : \sigma \rightarrow \tau$ and $G : \sigma' \rightarrow \tau$ are superoperators, so is $[F, G] : \sigma \oplus \sigma' \rightarrow \tau$.*
2. *If $F : \sigma \rightarrow \sigma'$ and $G : \tau \rightarrow \tau'$ are superoperators, then so are $F \oplus G$ and $F \otimes G$.*
3. *if $id_\rho \otimes F$ is positive for all one element signatures ρ , then F is completely positive.*
4. *Given U , a unitary $n \times n$ matrix, then $F : n \rightarrow n$ defined as $F(A) = UAU^*$ is a superoperator.*
5. *If T_1, T_2 are $n \times n$ matrices such that $T_1^*T_1 + T_2^*T_2 = I$, then $F : n \rightarrow n$, defined as $F(A) = (T_1AT_1^*, T_2AT_2^*)$ is a superoperator.*

Proof. For statement 1, as $\text{tr } (F(A) + F(B)) = (\text{tr } F(A)) + (\text{tr } F(B)) \leq \text{tr } A + \text{tr } B = \text{tr } (A, B)$, note that $[F, G]$ satisfies the trace condition. Secondly, because of distributivity $id_\rho \otimes [F, G] = [id_\rho \otimes F, id_\rho \otimes G]$ and the complete positivity follows. The first assertion of statement 2 follows in a similar manner. As $F \otimes G = (F \otimes id_\tau)(id_{\sigma'} \otimes G)$, note that

each element of the composition is a superoperator, hence so is $F \otimes G$. In statement 3, note that any signature ν of length n is equal to a coproduct of n single element signatures, $\nu_1 \oplus \dots \oplus \nu_n$. Then using distributivity, $id_\nu \otimes F = (id_{\nu_1} \otimes F) \oplus \dots \oplus (id_{\nu_n} \otimes F)$ which by assumption and statement 2 is positive. Hence, F is completely positive. For statement 4, as U is unitary, it is immediate that F is positive and preserves the trace. Note also that $(id_n \otimes F)(A) = (I \otimes U)A(I \otimes U)^*$ where I is the $n \times n$ identity matrix. However, $I \otimes U$ is also a unitary matrix, therefore $(id_n \otimes F)$ is positive for all n and by the previous point, F is completely positive and therefore a superoperator. For the last statement, by construction F preserves both positivity and trace and by a similar argument to the previous point, it is a superoperator. \square

At this point there is now sufficient machinery to define the category Q which will be used for the categorical semantics for QPL. Define Q as the subcategory of V having the same objects, but only superoperators as morphisms. By lemma 4.4.4, this is a valid subcategory, which inherits \oplus as a coproduct. It is not a product as the diagonal morphism increases the trace and is therefore not a superoperator.

Q is also a CPO enriched category. First, note that superoperators send density matrices to density matrices (positive hermitian matrices with trace ≤ 1). Designating D_σ to be the subset of density matrix tuples contained in V_σ , then for any superoperator F , it can be restricted to the density matrices. This restricted function preserves the Löwner order from definition 4.2.6 and it preserves the least upper bounds of sequences. From this, given signatures σ and τ , define a partial order on $Q(\sigma, \tau)$ by $F \leq G$ when $\forall \nu, A \in D_{\nu \otimes \sigma} : (id_\nu \otimes F)(A) \leq (id_\nu \otimes G)(A)$.

Lemma 4.4.5. *The poset $Q(\sigma, \tau)$ is a complete partial order. Composition, co-pairing and tensor are Scott-continuous and therefore Q is CPO-enriched.*

As Q is a CPO enriched category, it is possible to define a monoidal trace over the coproduct monoid. Recall that if $F : \sigma \oplus \tau \rightarrow \sigma' \oplus \tau$, then $tr F$ is a map, $\sigma \rightarrow \sigma'$. Given

such an F , construct the trace as follows:

- Define T_0 as the constant zero function.
- Define $T_{i+1} = F; [id_{\sigma'}, i_2 H_i] : \sigma \oplus \tau \rightarrow \sigma'$.

Then, $T_0 \leq T_1$ as 0 is the least element in the partial order. $T_i \leq T_{i+1}$ for all i as all the categorical operations are monotonic due to the CPO enrichment. Therefore, now define $T = \vee_i T_i : \sigma \oplus \tau \rightarrow \sigma'$. Finally, define $tr F = i_1; T : \sigma \rightarrow \sigma'$.

This trace construction may be compared to the loop semantics construction in section [sub-section 4.4.1](#). For F as above, we may decompose it into components $F_{11} : \sigma \rightarrow \sigma'$, $F_{21} : \sigma \rightarrow \tau$, $F_{12} : \tau \rightarrow \sigma'$ and $F_{22} : \tau \rightarrow \tau$. This gives us

$$\begin{aligned} T_0(A, 0) &= 0, \\ T_1(A, 0) &= F_{11}(A), \\ T_2(A, 0) &= F_{11}(A) + F_{12}F_{21}(A), \\ &\vdots \end{aligned}$$

which brings us to

$$(Tr F)(A) = T(A, 0) = F_{11}(A) + \sum_{i=0}^{\infty} F_{12}(F_{22}^i(F_{21}(A))).$$

This is the same construction as equation ([equation \(4.3\)](#)) and will be used for the interpretation of loops. In particular, this justifies the convergence of the infinite sum in that equation.

At this point, we now have the information required to give an interpretation of the quantum flow charts of QPL in the category \mathcal{Q} . There are two types, $\llbracket \mathbf{bit} \rrbracket = 1, 1$ and $\llbracket \mathbf{qubit} \rrbracket = 2$. The interpretation of basic operations is given in Table [table 4.1](#).

Additionally, if a type context Γ is $x_i : T_i$, then $\llbracket \Gamma \rrbracket = \otimes_i \llbracket A_i \rrbracket$. If $\bar{\Theta}$ is a list of typing context, Θ_i , then $\llbracket \bar{\Theta} \rrbracket = \oplus_i \llbracket \Theta_i \rrbracket$. For the various types of composite flowcharts, the interpretation is as follows:

Table 4.1: Interpretation of QPL operations

$\llbracket \text{new bit } b := 0 \rrbracket$	$= \text{newbit} : I \rightarrow \mathbf{bit} :$	$a \mapsto (a, 0)$
$\llbracket \text{discard } b \rrbracket$	$= \text{discardbit} : \mathbf{bit} \rightarrow I :$	$(a, b) \mapsto a + b$
$\llbracket b := 0 \rrbracket$	$= \text{set}_0 : \mathbf{bit} \rightarrow \mathbf{bit} :$	$(a, b) \mapsto (a + b, 0)$
$\llbracket b := 1 \rrbracket$	$= \text{set}_1 : \mathbf{bit} \rightarrow \mathbf{bit} :$	$(a, b) \mapsto (0, a + b)$
$\llbracket \text{branch } b \rrbracket$	$= \text{branch} : \mathbf{bit} \rightarrow \mathbf{bit} \oplus \mathbf{bit} :$	$(a, b) \mapsto (a, 0, 0, b)$
$\llbracket \text{new qbit } q := 0 \rrbracket$	$= \text{newqbit} : I \rightarrow \mathbf{qubit} :$	$a \mapsto \begin{pmatrix} a & 0 \\ 0 & 0 \end{pmatrix}$
$\llbracket \text{discard } q \rrbracket$	$= \text{discardqbit} : \mathbf{qubit} \rightarrow I :$	$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto a + d$
$\llbracket \vec{q}^* = U \rrbracket$	$= \text{unitary}_U : \mathbf{qubit}^n \rightarrow \mathbf{qubit}^n :$	$A \mapsto UAU^*$
$\llbracket \text{measure } q \rrbracket$	$= \text{measure} : \mathbf{qubit} \rightarrow \mathbf{qubit} \oplus \mathbf{qubit} :$	$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \mapsto \left(\begin{pmatrix} a & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & d \end{pmatrix} \right)$
$\llbracket \text{merge} \rrbracket$	$= \text{merge} : I \oplus I \rightarrow I :$	$(a, b) \mapsto (a + b)$
$\llbracket \text{initial} \rrbracket$	$= \text{initial} : 0 \rightarrow I :$	$() \mapsto 0$
$\llbracket \text{permute } \Phi \rrbracket$	$= \text{permute}_\Phi : \oplus_i A_i \rightarrow \oplus_i A_{\Phi(i)}$	

- If the context Γ is added to flowchart A , producing B , then $\llbracket B \rrbracket = \llbracket A \rrbracket \otimes \llbracket \Gamma \rrbracket$.
- If the outputs of flowchart A are connected to the inputs of B , giving flowchart C , then $\llbracket C \rrbracket = \llbracket A \rrbracket; \llbracket B \rrbracket$.
- If flowchart C is made up of parallel flowcharts A and B , then $\llbracket C \rrbracket = \llbracket A \rrbracket \oplus \llbracket B \rrbracket$.
- if flowchart C is a loop on flowchart A , then $\llbracket C \rrbracket = \text{tr}(\llbracket X \rrbracket)$.

For procedures, it is necessary to consider abstract variable flowcharts with specified types, $R_i : \overline{\Theta}_i \rightarrow \overline{\Theta}'_i$. With these variable flowcharts allowed, these may be interpreted in a specified environment κ which maps the R_i to specific morphisms of Q with the appropriate type. Then, $\llbracket R_i \rrbracket_\kappa = \kappa R_i$ and if A is a flow chart using R_i , its interpretation relative to κ may be built up inductively via the operations above, giving a function Ω_A which will map the environments to a specific map in Q .

For recursion, consider the recursive subroutine defined as $P = T(P)$ for a flowchart T . Then $\Omega_T : Q(\sigma, \tau) \rightarrow Q(\sigma, \tau)$ will be a Scott-continuous function. In this case, $\llbracket Y \rrbracket$ will be the least fixed point of Ω_T . There is an increasing sequence for all $i \geq 0$, $S_i \leq S_{i+1}$ given by $S_0 = 0$ and $S_{i+1} = \Omega_T(S_i)$. This gives the interpretation of P as

$$\llbracket P \rrbracket = \vee_i S_i = \lim_i S_i.$$

This corresponds to equation (equation (4.4)) above, which shows this is the correct interpretation for recursive procedures and since Q is a CPO enriched category with a least point, therefore this limit exists and therefore the the limit in equation (equation (4.4)) will exist.

Conclusions for QPL

Data types As can be seen by the proceeding pages, creating the categorical machinery for a semantic interpretation of QPL is quite detailed. The paper [30] goes on to prove soundness, completeness and provides some alternative syntaxes for QPL. The subject of structured types is discussed briefly. Tuple types (Γ, Λ) may be constructed as $\llbracket \Gamma \rrbracket \otimes \llbracket \Lambda \rrbracket$. This immediately provides types such as fixed length classical or quantum integers, characters, and so forth. Sum types can similarly be added as $\Gamma \oplus \Lambda$, noting that the “choice” between the two types remains classical. The one primary weakness in the type system is not allowing structured recursive types such as **List**. This weakness is addressed in a follow on paper, [18]. In this paper, the major change is that rather than restricting to a tuple of integers for the objects of V and Q , they consider arbitrary families of integers as the objects, defining a new category Q^∞ . Definitions such as positive, Hermitian, the Löwner order and trace follow in a straightforward manner, as does the definition $V_\sigma = \prod_{i \in |\sigma|} \mathbb{C}^{\sigma_i \times \sigma_i}$, noting this is now an infinite product. Note that in the infinite dimensional case there is no canonical basis for V_σ and therefore no canonical isomorphism between $V_{\sigma \otimes \tau}$ and $V_\sigma \otimes V_\tau$. To rectify this, the authors refine the allowed morphisms in the category Q^∞ . First, they define the category $\overline{Q^\infty}$ as having infinite signatures as objects, but maps $f : \sigma \rightarrow \tau$ are maps $f : D_\sigma \rightarrow D_\tau$ (D_σ are the density matrix tuples of V_σ). These f are called *positive operators*. They are required

to extend to linear maps $\bar{f} : V_\sigma \rightarrow V_\tau$ and be continuous for the Löwner order. These maps are definable as a matrix of maps over finite dimensional spaces $f_{ij} : \mathbb{C}^{\sigma_i \times \sigma_i} \rightarrow \mathbb{C}^{\tau_i \times \tau_i}$, calling this the *operator matrix*.

One can now define the tensor of two positive operators f, g by tensoring their respective operator matrices. Then, following the finite case, the positive operator $f : \sigma \rightarrow \tau$ with operator matrix F is a superoperator, if when ID_γ is the operator matrix for the identity on the signature γ , $ID_\gamma \otimes F$ is a positive operator. The infinite case superoperators follow the desired properties as in the finite case and the category Q^∞ is defined as the category with objects being infinite signatures and morphisms these superoperators.

From this, the authors show that any endofunctor definable via an “arithmetic” equation involving the coproduct and tensor will give rise to a data type in the category. In particular, one can define **qubit** lists as

$$QList = 1 \oplus (\mathbf{qubit} \otimes QList)$$

and trees of **qubits** as

$$QTree = \mathbf{qubit} \oplus (QTree \otimes QTree).$$

Quantum communication QPL makes no attempts to handle communication or transmission of quantum data. This will not be addressed in this thesis.

Higher order functions QPL is defined as a functional language. One of the expectations of modern functional languages is that programs themselves are first class objects, that is, they may be operated on by the program. Typical uses are partial evaluation and passing a subroutine of a specified type for use by another subroutine. In quantum computation, the primary issue with this is how does one guarantee the no-cloning, no-erasing rules with respect to quantum data. Work on a quantum lambda calculus, [35, 34], has attempted to address this, albeit primarily with operational rather than denotational semantics. In, [31], the author explores the use of cones rather than vector spaces to create a denotational

semantics, but finds that the candidates fail to provide the correct answer over the base types. We will not be considering the higher-order issues further in this paper.

4.4.2 Semantics of pure quantum computations

In [3], the authors approach the creation of a categorical semantics for quantum computation independently of a specific language. Rather, they use finitary quantum mechanics as their reference point.

Finitary quantum mechanics consists of the following:

1. The system's state space is represented by a finite dimensional Hilbert space H .
2. The basic type of the system is that of **qubit**— 2-dimensional Hilbert space — with the computational basis $\{|0\rangle, |1\rangle\}$.
3. Compound systems are tensor products of the components. This is what enables *entanglement* as the general form of the system $H \otimes J$ where H and J are Hilbert spaces is

$$\sum_{i=1}^n \alpha_i (u_i \otimes v_i)$$

where u_i is a basis element of H and v_i is a basis element of J .

4. The basic transforms are *unitary transformations*.
5. The measurements performable are *self-adjoint* (hermitian) operators - with two sub-steps:
 - (a) The actual act of measurement. (Preparation).
 - (b) The communication of the results of the measurement. (Observation).

The above definition does allow for the possibility of mixed states, as described in section [sub-section 4.2.1](#), but for the remainder of this section, it is assumed both steps of the measurement are carried out, resulting in pure states only.

[3] gives the interpretation of finitary quantum mechanics in the context of a biproduct dagger compact closed category, \mathbf{D} .

1. An n -dimensional state space S is an object of \mathbf{D} , together with a unitary isomorphism

$$base_A : \oplus^n I \rightarrow A.$$

2. A **qubit** is a 2 dimensional state space Q with the computational basis $base_Q : I \oplus I \rightarrow Q$.

3. Compound systems A, B are described by $A \otimes B$ and $base_{A \otimes B} = \phi(base_A \otimes base_B)$ where $\phi : \oplus^{nm} I \cong (\oplus^n I) \otimes (\oplus^m I)$ is the isomorphism obtained by repeated application of distributivity isomorphisms.

4. The basic transformations are unitary transformations, i.e., f , where $f^\dagger = f^{-1}$.

- 5a. A preparation is a morphism $P : I \rightarrow A$ which has a corresponding unitary morphism

$$f_P : \oplus^n I \rightarrow \oplus^n I \text{ and}$$

$$\begin{array}{ccc} I & \xrightarrow{P} & A \\ i_1 \downarrow & & \uparrow base_A \\ \oplus^n I & \xrightarrow{f_P} & \oplus^n I \end{array}$$

commutes.

- 5b. An observation is an isomorphism $O = \oplus^n O_i$ with components $O_i : A \rightarrow I$ which has an unitary automorphism $f_O : \oplus^n I \rightarrow \oplus^n I$ such that

$$\begin{array}{ccc} A & \xrightarrow{O_i} & I \\ base_A \uparrow & & \uparrow p_i \\ \oplus^n I & \xrightarrow{f_O} & \oplus^n I \end{array}$$

commutes for all $i = 1, \dots, n$. The observational branches are the individual $O_i : A \rightarrow I$.

Additionally, the biproduct \oplus represents distinct branches resulting from measurement. Accordingly, any operation on a biproduct must be an explicit biproduct, that is $f : A \oplus B \rightarrow C \oplus D$ will be $f_1 \oplus f_2$ with $f_1 : A \rightarrow C$ and $f_2 : B \rightarrow D$.

The authors go on to show how this interpretation is sufficient to model quantum teleportation, logic gate teleportation and entanglement swapping.

4.4.3 Bases and Frobenius Algebras

In [14], the authors provide an algebraic description of orthogonal bases in finite dimensional Hilbert spaces. As noted in the example at the end of section [sub-section 4.3.1](#), an orthonormal basis for such a space is a special commutative \dagger -Frobenius algebra. To show the other direction, given a commutative \dagger -Frobenius algebra, (H, ∇, u) and for each element $\alpha \in H$, define the right action of α as $R_\alpha := (id \otimes \alpha) \nabla : H \rightarrow H$. Note the use of the fact that elements $\alpha \in H$ can be considered as linear maps $\alpha : \mathbb{C} \rightarrow H : 1 \mapsto |\alpha\rangle$. The dagger of a right action is also a right action, $R_\alpha^\dagger = R_{\alpha'}$ where $\alpha' = u \nabla (id \otimes \alpha^\dagger)$, which is a consequence of the Frobenius identities.

The $(-)'$ construction is actually an involution:

$$\begin{aligned}
(\alpha')' &= u \nabla (id \otimes \alpha'^\dagger) \\
&= u \nabla (id \otimes (u \nabla (id \otimes \alpha^\dagger))^\dagger) \\
&= u \nabla (id \otimes ((id \otimes \alpha) \Delta \epsilon)) \\
&= (u \otimes \alpha)(\nabla \otimes id)(id \otimes \Delta)(id \otimes \epsilon) \\
&= (u \otimes \alpha)(id \otimes \Delta)(\nabla \otimes id)(id \otimes \epsilon) \\
&= (u \otimes \alpha)(id \otimes \epsilon) \\
&= \alpha
\end{aligned}$$

Lemma 4.4.6. *Any \dagger -Frobenius algebra in \mathbf{FDHILB} is a C^* -algebra.*

Proof. The endomorphism monoid of $\text{FDHILB}(H, H)$ is a C^* -algebra. From the proceeding,

$$H \cong \text{FDHILB}(\mathbb{C}, H) \cong R_{[\text{FDHILB}(\mathbb{C}, H)]} \subseteq \text{FDHILB}(H, H).$$

This inherits the algebra structure from $\text{FDHILB}(H, H)$. Since any finite dimensional involution-closed subalgebra of a C^* -algebra is also a C^* -algebra, this shows the \dagger -Frobenius algebra is a C^* -algebra. \square

Using the fact that the involution preserving homomorphisms from a finite dimensional commutative C^* -algebra to \mathbb{C} form a basis for the dual of the underlying vector space, write these homomorphisms as $\phi_i^\dagger : H \rightarrow \mathbb{C}$. Then their adjoints, $\phi_i : \mathbb{C} \rightarrow H$ will form a basis for the space H . These are the copyable elements in H .

This, together with continued applications of the Frobenius rules and linear algebra allow the authors to prove:

Theorem 4.4.7. *Every commutative \dagger -Frobenius algebra in FDHILB determines an orthogonal basis consisting of its copyable elements. Conversely, every orthogonal basis $\{|\phi_i\rangle\}_i$ determines a commutative \dagger -Frobenius algebra via*

$$\Delta : H \rightarrow H \otimes H : |\phi_i\rangle \mapsto |\phi_i\rangle \otimes |\phi_i\rangle \quad \epsilon : H \rightarrow \mathbb{C} : |\phi_i\rangle \mapsto 1$$

and these constructions are inverse to each other.

4.4.4 Quantum and classical data

In [13], the authors build on the results as related in the previous sections, to start from a \dagger -symmetric monoidal category and construct the minimal machinery needed to model quantum and classical computations. For the rest of this section, \mathbf{D} will be assumed to be such a category, with \otimes the monoid tensor and I the unit of the monoid.

Definition 4.4.8. A compact structure on an object A in the category \mathbf{D} is given by the object A , an object A^* called its *dual* and the maps $\eta : I \rightarrow A^* \otimes A$, $\epsilon : A \otimes A^* \rightarrow I$ such

that the diagrams

$$\begin{array}{ccccc}
& & A & \xrightarrow{A \otimes \eta} & A \otimes A^* \otimes A \\
& \swarrow & \searrow & \searrow & \downarrow \epsilon \otimes A \\
A^* & & & & A \\
\downarrow \eta \otimes A^* & \searrow id & & \searrow id & \\
A^* \otimes A \otimes A^* & \xrightarrow{A^* \otimes \epsilon} & A^* & &
\end{array}$$

commute.

Definition 4.4.9 (Quantum Structure). A *quantum structure* is an object A and map $\eta : I \rightarrow A \otimes A$ such that $(A, A, \eta, \eta^\dagger)$ form a compact structure.

Note that A is self-dual in definition 4.4.9.

This allows the creation of the category \mathbf{D}_q which has as objects quantum structures and maps are the maps in \mathbf{D} between the objects in the quantum structures.

In the category \mathbf{D}_q , it is now possible to define the upper $*$ operations on maps, such that $(f_*)^* = (f^*)_* = f^\dagger$.

$$\begin{aligned}
f^* &:= (\eta_A \otimes 1)(1 \otimes f \otimes 1)(1 \otimes \eta_B^\dagger) \\
f_* &:= (\eta_B \otimes 1)(1 \otimes f^\dagger \otimes 1)(1 \otimes \eta_A^\dagger)
\end{aligned}$$

Interestingly, \mathbf{D}_q possesses enough structure to be axiomatized in the same manner as above in section sub-section 4.4.2, excepting the portions dependent upon biproducts.

Next, define a classical structure on \mathbf{D} .

Definition 4.4.10 (Classical structure). A *classical structure* in \mathbf{D} is an objects X and two maps, $\Delta : X \rightarrow X \otimes X$, $\epsilon : X \rightarrow I$ such that $X, \Delta^\dagger, \epsilon^\dagger, \Delta, \epsilon$ forms a special Frobenius algebra.

As above, this allows us to define \mathbf{D}_c , the category whose objects are the classical structures of \mathbf{D} with maps between classical structures being the maps in \mathbf{D} between the objects of the classical structure.

Note that a classical structure will induce a quantum structure, setting η_X to be $\epsilon_X^\dagger \Delta_X$.

4.4.5 Complete positivity

Given a \dagger -compact closed category, it is possible to construct its category of completely positive maps.

Definition 4.4.11 (Positive map). A map $f : A \rightarrow A$ in a dagger category is called *positive* if there is an object B and a map $g : A \rightarrow B$ with $f = gg^\dagger$.

Definition 4.4.12 (Trace). For $f : A \rightarrow A$ in a compact closed category, its *trace* is defined as $tr f : I \rightarrow I = \eta_A; c_{A^*, A}; (f \otimes A^*); \epsilon$.

The following lemma gives some properties of positive maps:

Lemma 4.4.13. *In any biproduct dagger compact closed category, the following hold:*

1. f positive $\implies hfh^\dagger$ is positive for all maps h .
2. id_A is positive.
3. If $f : A \rightarrow A$ and $g : B \rightarrow B$ are positive, so are $f \otimes g$ and $f \oplus g$.
4. $0_{A, A}$ is positive. If $f, g : A \rightarrow A$ is positive, so is $f + g$.
5. f positive $\implies f^\dagger = f$.
6. f positive $\implies f^*$ and $tr f$ are positive.
7. $f, g : A \rightarrow A$ positive $\implies tr(gf)$ is positive.

Proof. The first six items follow immediately from the definitions and how structure is preserved for $(-)^{\dagger}$. For item 6, note that $g = h h^\dagger$ and $tr(gf) = tr(h^\dagger f h)$ which is positive by points 1 and 5. □

Definition 4.4.14. In a compact closed category, the *name* of a map $f : A \rightarrow B$ is the map $\lceil f \rceil : I \rightarrow A^* \otimes B$ defined as $\eta_A; (1 \otimes f)$. This is also called the *matrix* of f .

In the case of a positive map f , $\lceil f \rceil$ is referred to as a *positive matrix*.

Definition 4.4.15. In a dagger compact closed category, a map $f : A^* \otimes A \rightarrow B^* \otimes B$ is *completely positive* if for all objects C and all positive matrices $f : I \rightarrow C^* \otimes A^* \otimes A \otimes C$ the morphism $g; (1 \otimes f \otimes 1) : I \rightarrow C^* \otimes B^* \otimes B \otimes C$ is a positive matrix.

This now allows us to define the CPM construction.

Definition 4.4.16. Given a dagger compact closed category \mathbf{D} , define $\text{CPM}(\mathbf{D})$ as the category with the same objects as \mathbf{D} , and a map $f : A \rightarrow B$ in $\text{CPM}(\mathbf{D})$ is a completely positive map $f : A^* \otimes A \rightarrow B^* \otimes B$ in \mathbf{D} .

$\text{CPM}(\mathbf{D})$ is also a dagger compact closed structure, inheriting its tensor from \mathbf{D} . There is a functor $F : \mathbf{D} \rightarrow \text{CPM}(\mathbf{D})$ defined as $F(A) = A$ on objects and $F(f) = f_* \otimes f$ on maps. The image of the structure maps under F are structure maps for $\text{CPM}(\mathbf{D})$. The dagger of a map f is the same as its dagger in \mathbf{D} .

Biproduct completion

When the CPM construction is applied to a biproduct dagger compact closed category, it will not in general retain biproducts. However, it will be monoid enriched by lemma 4.4.13. This allows us to create the biproduct completion.

The biproduct completion of a category \mathbf{D} , which is enriched in commutative monoids is the category \mathbf{D}^\oplus which has as objects finite sequences $\langle A_1, \dots, A_n \rangle$ where $n \geq 0$. The morphisms of \mathbf{D}^\oplus are matrices of the morphisms of \mathbf{D} . Application and composition of morphisms is via matrix multiplication. The functor $F(A) = \langle A \rangle$, $F(f) = [f]$ is an embedding of \mathbf{D} in \mathbf{D}^\oplus . If \mathbf{D} is compact closed and the tensor is linear (i.e., interacts with the enrichment in a linear fashion), then \mathbf{D}^\oplus is also compact closed.

Furthermore, if \mathbf{D} is a dagger category and the dagger is linear, then \mathbf{D}^\oplus will be a dagger category. The dagger of a map $(f_{i,j})$ in \mathbf{D}^\oplus is $((f_{j,i})^\dagger)$.

This gives us the following theorem:

Theorem 4.4.17. *Given \mathbf{D} , a biproduct dagger compact closed category, $\mathbf{CPM}(\mathbf{D})$ is enriched in commutative monoids as a dagger compact closed category. Therefore, it is possible to construct its biproduct completion, $\mathbf{CPM}(\mathbf{D})^\oplus$.*

Note that the canonical embedding from above, F , while it preserves the dagger compact closed structure, it does *not* preserve biproducts.

Chapter 5

Frobenius Algebras and Quantum Computation

5.1 The category of Commutative Frobenius Algebras

Chapter 6

$D[\omega]$ based \dagger -categories

6.1 Toy quantum semantics

6.2 Introduction to synthesis

An important problem in quantum information theory is the decomposition of arbitrary unitary operators into gates from some fixed universal set [25]. Depending on the operator to be decomposed, this may either be done exactly or to within some given accuracy ϵ ; the former problem is known as *exact synthesis* and the latter as *approximate synthesis* [19].

6.3 Algebraic background

We first introduce some notation and terminology, primarily following [19]. Recall that \mathbb{N} is the set of natural numbers including 0, and \mathbb{Z} is the ring of integers. We write $\mathbb{Z}_2 = \mathbb{Z}/2\mathbb{Z}$ for the ring of integers modulo 2. Let \mathbb{D} be the ring of *dyadic fractions*, defined as $\mathbb{D} = \mathbb{Z}[\frac{1}{2}] = \{\frac{a}{2^n} \mid a \in \mathbb{Z}, n \in \mathbb{N}\}$.

Let $\omega = e^{i\pi/4} = (1+i)/\sqrt{2}$. Note that ω is an 8th root of unity satisfying $\omega^2 = i$ and $\omega^4 = -1$. We will consider three different rings related to ω :

Definition 6.3.1. Consider the following rings. Note that the first two are subrings of the complex numbers, and the third one is not:

- $\mathbb{D}[\omega] = \{a\omega^3 + b\omega^2 + c\omega + d \mid a, b, c, d \in \mathbb{D}\}$.
- $\mathbb{Z}[\omega] = \{a\omega^3 + b\omega^2 + c\omega + d \mid a, b, c, d \in \mathbb{Z}\}$.
- $\mathbb{Z}_2[\omega] = \{p\omega^3 + q\omega^2 + r\omega + s \mid p, q, r, s \in \mathbb{Z}_2\}$.

Note that the ring $\mathbb{Z}_2[\omega]$ only has 16 elements. The laws of addition and multiplication are uniquely determined by the ring axioms and the property $\omega^4 = 1 \pmod{2}$. We call the elements of $\mathbb{Z}_2[\omega]$ *residues* (more precisely, residue classes of $\mathbb{Z}[\omega]$ modulo 2).

Remark 6.3.2. The ring $\mathbb{D}[\omega]$ is the same as the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$. However, as already pointed out in [19], the formulation in terms of ω is far more convenient algebraically.

Remark 6.3.3. The ring $\mathbb{Z}[\omega]$ is also called the *ring of algebraic integers* of $\mathbb{D}[\omega]$. It has an intrinsic definition, i.e., one that is independent of the particular presentation of $\mathbb{D}[\omega]$. Namely, a complex number is called an *algebraic integer* if it is the root of some polynomial with integer coefficients and leading coefficient 1. It follows that ω , i , and $\sqrt{2}$ are algebraic integers, whereas, for example, $1/\sqrt{2}$ is not. The ring $\mathbb{Z}[\omega]$ then consists of precisely those elements of $\mathbb{D}[\omega]$ that are algebraic integers.

6.3.1 Conjugate and norm

Remark 6.3.4 (Complex conjugate and norm). Since $\mathbb{D}[\omega]$ and $\mathbb{Z}[\omega]$ are subrings of the complex numbers, they inherit the usual notion of complex conjugation. We note that $\omega^\dagger = -\omega^3$. This yields the following formula:

$$(a\omega^3 + b\omega^2 + c\omega + d)^\dagger = -c\omega^3 - b\omega^2 - a\omega + d. \quad (6.1)$$

Similarly, the sets $\mathbb{D}[\omega]$ and $\mathbb{Z}[\omega]$ inherit the usual norm from the complex numbers. It is given by the following explicit formula, for $t = a\omega^3 + b\omega^2 + c\omega + d$:

$$\|t\|^2 = t^\dagger t = (a^2 + b^2 + c^2 + d^2) + (cd + bc + ab - da)\sqrt{2}. \quad (6.2)$$

Definition 6.3.5 (Weight). For $t \in \mathbb{D}[\omega]$ or $t \in \mathbb{Z}[\omega]$, the *weight* of t is denoted $\|t\|_{\text{weight}}$, and is given by:

$$\|t\|_{\text{weight}}^2 = a^2 + b^2 + c^2 + d^2. \quad (6.3)$$

Note that the square of the norm is valued in $\mathbb{D}[\sqrt{2}]$, whereas the square of the weight is valued in \mathbb{D} . We also extend the definition of norm and weight to vectors in the obvious way: For $u = (u_j)_j$, we define

$$\|u\|^2 = \sum_j \|u_j\|^2 \quad \text{and} \quad \|u\|_{\text{weight}}^2 = \sum_j \|u_j\|_{\text{weight}}^2.$$

Lemma 6.3.6. *Consider a vector $u \in \mathbb{D}[\omega]^n$. If $\|u\|^2$ is an integer, then $\|u\|_{\text{weight}}^2 = \|u\|^2$.*

Proof. Any $t \in \mathbb{D}[\sqrt{2}]$ can be uniquely written as $t = a + b\sqrt{2}$, where $a, b \in \mathbb{D}$. We can call a the *dyadic part* of t . Now the claim is obvious, because $\|u\|_{\text{weight}}^2$ is exactly the dyadic part of $\|u\|^2$. \square

6.3.2 Denominator exponents

Definition 6.3.7. Let $t \in \mathbb{D}[\omega]$. A natural number $k \in \mathbb{N}$ is called a *denominator exponent* for t if $\sqrt{2}^k t \in \mathbb{Z}[\omega]$. It is obvious that such k always exists. The least such k is called the *least denominator exponent* of t .

More generally, we say that k is a denominator exponent for a vector or matrix if it is a denominator exponent for all of its entries. The least denominator exponent for a vector or matrix is therefore the least k that is a denominator exponent for all of its entries.

Remark 6.3.8. Our notion of least denominator exponent is almost the same as the “smallest denominator exponent” of [19], except that we do not permit $k < 0$.

6.3.3 Residues

Remark 6.3.9. The ring $\mathbb{Z}_2[\omega]$ is not a subring of the complex numbers; rather, it is a quotient of the ring $\mathbb{Z}[\omega]$. Indeed, consider the *parity function* $\overline{(\cdot)} : \mathbb{Z} \rightarrow \mathbb{Z}_2$, which is the unique ring homomorphism. It satisfies $\bar{a} = 0$ if a is even and $\bar{a} = 1$ if a is odd. The parity map induces a surjective ring homomorphism $\rho : \mathbb{Z}[\omega] \rightarrow \mathbb{Z}_2[\omega]$, defined by

$$\rho(a\omega^3 + b\omega^2 + c\omega + d) = \bar{a}\omega^3 + \bar{b}\omega^2 + \bar{c}\omega + \bar{d}.$$

$\rho(t)$	$\rho(\sqrt{2}t)$	$\rho(t^\dagger t)$	$\rho(t)$	$\rho(\sqrt{2}t)$	$\rho(t^\dagger t)$
0000	0000	0000	1000	0101	0001
0001	1010	0001	1001	1111	1010
0010	0101	0001	1010	0000	0000
0011	1111	1010	1011	1010	0001
0100	1010	0001	1100	1111	1010
0101	0000	0000	1101	0101	0001
0110	1111	1010	1110	1010	0001
0111	0101	0001	1111	0000	0000

Table 6.1: Some operations on residues

We call ρ the *residue map*, and we call $\rho(t)$ the *residue* of t .

Convention 6.3.10. Since residues will be important for the constructions of this thesis, we introduce a shortcut notation, writing each residue $p\omega^3 + q\omega^2 + r\omega + s$ as a string of binary digits $pqrs$.

What makes residues useful for our purposes is that many important operations on $\mathbb{Z}[\omega]$ are well-defined on residues. Here, we say that an operation $f : \mathbb{Z}[\omega] \rightarrow \mathbb{Z}[\omega]$ is *well-defined on residues* if for all t, s , $\rho(t) = \rho(s)$ implies $\rho(f(t)) = \rho(f(s))$.

For example, two operations that are obviously well-defined on residues are complex conjugation, which takes the form $(pqrs)^\dagger = rqp s$ by ([equation \(6.1\) on page 163](#)), and multiplication by ω , which is just a cyclic shift $\omega(pqrs) = qrsp$. Table [table 6.1](#) shows two other important operations on residues, namely multiplication by $\sqrt{2}$ and the squared norm.

Definition 6.3.11 (k -Residue). Let $t \in \mathbb{D}[\omega]$ and let k be a (not necessarily least) denominator exponent for t . The k -residue of t , in symbols $\rho_k(t)$, is defined to be

$$\rho_k(t) = \rho(\sqrt{2}^k t).$$

Definition 6.3.12 (Reducibility). We say that a residue $x \in \mathbb{Z}_2[\omega]$ is *reducible* if it is of the form $\sqrt{2}y$, for some $y \in \mathbb{Z}_2[\omega]$. Moreover, we say that $x \in \mathbb{Z}_2[\omega]$ is *twice reducible* if it is of the form $2y$, for some $y \in \mathbb{Z}_2[\omega]$.

Lemma 6.3.13. *For a residue x , the following are equivalent:*

- (a) x is reducible;
- (b) $x \in \{0000, 0101, 1010, 1111\}$;
- (c) $\sqrt{2}x = 0000$;
- (d) $x^\dagger x = 0000$.

Moreover, x is twice reducible iff $x = 0000$.

Proof. By inspection of Table [table 6.1 on the preceding page](#). □

Lemma 6.3.14. *Let $t \in \mathbb{Z}[\omega]$. Then $t/2 \in \mathbb{Z}[\omega]$ if and only if $\rho(t)$ is twice reducible, and $t/\sqrt{2} \in \mathbb{Z}[\omega]$ if and only if $\rho(t)$ is reducible.*

Proof. The first claim is trivial, as $\rho(t) = 0000$ if and only if all components of t are even. For the second claim, the left-to-right implication is also trivial: assume $t' = t/\sqrt{2} \in \mathbb{Z}[\omega]$. Then $\rho(t) = \rho(\sqrt{2}t')$, which is reducible by definition. Conversely, let $t \in \mathbb{Z}[\omega]$ and assume that $\rho(t)$ is reducible. Then $\rho(t) \in \{0000, 0101, 1010, 1111\}$, and it can be seen from Table [table 6.1 on the previous page](#) that $\rho(\sqrt{2}t) = 0000$. Therefore, $\sqrt{2}t$ is twice reducible by the first claim; hence t is reducible. □

Corollary 6.3.15. *Let $t \in \mathbb{D}[\omega]$ and let $k > 0$ be a denominator exponent for t . Then k is the least denominator exponent for t if and only if $\rho_k(t)$ is irreducible.*

Proof. Since k is a denominator exponent for t , we have $\sqrt{2}^k t \in \mathbb{Z}[\omega]$. Moreover, k is least if and only if $\sqrt{2}^{k-1} t \notin \mathbb{Z}[\omega]$. By Lemma [6.3.14](#), this is the case if and only if $\rho(\sqrt{2}^k t) = \rho_k(t)$ is irreducible. □

Lemma 6.3.16. *For all a in $\mathbb{Z}[\omega]$, $a + a^t$ is divisible by $\sqrt{2}$ in $\mathbb{Z}[\omega]$.*

Proof. It is sufficient to show this on the generators $\{1, \omega, \omega^2, \omega^3\}$.

1. $1 + 1 = 2$ and $\frac{2}{\sqrt{2}} = \sqrt{2} = \omega - \omega^3$.

2. $\omega + \omega^t = \omega - \omega^3 = \sqrt{2}$.
3. $\omega^2 + (\omega^2)^t = \omega^2 - \omega^2 = 0$.
4. $\omega^3 + (\omega^3)^t = \omega^3 - \omega = -\sqrt{2}$.

□

Definition 6.3.17. The notions of residue, k -residue, reducibility, and twice-reducibility all extend in an obvious componentwise way to vectors and matrices. Thus, the residue $\rho(u)$ of a vector or matrix u is obtained by taking the residue of each of its entries, and similar for k -residues. Also, we say that a vector or matrix is reducible if each of its entries is reducible, and similarly for twice-reducibility.

Example 6.3.18. Consider the matrix

$$U = \frac{1}{\sqrt{2}^3} \begin{pmatrix} -\omega^3 + \omega - 1 & \omega^2 + \omega + 1 & \omega^2 & -\omega \\ \omega^2 + \omega & -\omega^3 + \omega^2 & -\omega^2 - 1 & \omega^3 + \omega \\ \omega^3 + \omega^2 & -\omega^3 - 1 & 2\omega^2 & 0 \\ -1 & \omega & 1 & -\omega^3 + 2\omega \end{pmatrix}.$$

It has least denominator exponent 3. Its 3-, 4-, and 5-residues are:

$$\begin{aligned} \rho_3(U) &= \begin{pmatrix} 1011 & 0111 & 0100 & 0010 \\ 0110 & 1100 & 0101 & 1010 \\ 1100 & 1001 & 0000 & 0000 \\ 0001 & 0010 & 0001 & 1000 \end{pmatrix}, \\ \rho_4(U) &= \begin{pmatrix} 1010 & 0101 & 1010 & 0101 \\ 1111 & 1111 & 0000 & 0000 \\ 1111 & 1111 & 0000 & 0000 \\ 1010 & 0101 & 1010 & 0101 \end{pmatrix}, \quad \rho_5(U) = 0. \end{aligned}$$

6.4 Exact synthesis of single qubit operators

Matsumoto and Amano [23] showed that every single-qubit Clifford+ T operator can be uniquely written in the following form, which we call the *Matsumoto-Amano normal form*:

$$(T \mid \varepsilon) (HT \mid SHT)^* \mathcal{C}. \tag{6.4}$$

Here, we have used the syntax of regular expressions [17] to denote a set of sequences of operators. The symbol ε denotes the empty sequence (more precisely, the singleton set containing just the empty sequence); if \mathcal{L} and \mathcal{K} are two sets of sequences, then $\mathcal{L} \mid \mathcal{K}$ denotes their union; $\mathcal{L}\mathcal{K}$ denotes the set $\{st \mid s \in \mathcal{L}, t \in \mathcal{K}\}$; \mathcal{L}^* denotes the set $\{s_1 \dots s_n \mid n \geq 0; s_1, \dots, s_n \in \mathcal{L}\}$; and \mathcal{C} denotes any Clifford operator. In words, the Matsumoto-Amano representation of an operator consists of a Clifford operator, followed by any number of *syllables* of the form HT or SHT , followed by an optional syllable T . (We follow the usual convention of multiplying operators right-to-left, so when we say one operator “follows” another, we mean that it appears to its left).

The most important properties of the Matsumoto-Amano decomposition are:

- Existence: all single-qubit Clifford+ T operators can be written in Matsumoto-Amano normal form (moreover, there is an efficient algorithm for converting the operator to normal form);
- Uniqueness: no operator can be written in Matsumoto-Amano normal form in more than one way;
- T -optimality: of all the possible exact decompositions of a given operator into the Clifford+ T set of gates, the Matsumoto-Amano normal form contains the smallest possible number of T -gates.

It is perhaps less well-known that the uniqueness proof given by Matsumoto and Amano yields an efficient *algorithm* for T -optimal exact single-qubit synthesis. One may contrast this, for example, with the recent algorithm by Kliuchnikov et al. [19], which is efficient, but only asymptotically T -optimal. The purpose of this note is to give a detailed presentation of the algorithmic content of Matsumoto and Amano’s result. Along the way, we also simplify Matsumoto and Amano’s proofs, and we give an intrinsic characterization of the Clifford+ T subgroup of $SO(3)$.

6.4.1 Existence

In the following, we will often speak of sequences of operators. For our purposes, a sequence is just an n -tuple. We write st for the operation of concatenating two sequences, and we write ε for the empty sequence. We identify a 1-tuple (A) with the operator A itself. If $s = (A_1, \dots, A_n)$ is a sequence of operators, we write $\llbracket s \rrbracket = A_1 \cdots A_n$ for the product of the operators in the sequence; naturally, $\llbracket \varepsilon \rrbracket = I$. Note that the notation is ambiguous; for example, depending on the context, SHT may denote either the sequence (S, H, T) of 3 operators, or their product, which is a single operator. To alleviate the ambiguity, we assume that everything is a sequence by default, and we write $s \equiv t$ if two sequences are equal as tuples, and $s = t$ if they are equal as operators, i.e., if $\llbracket s \rrbracket = \llbracket t \rrbracket$.

Remark 6.4.1. Consider any operator A in Matsumoto-Amano normal form. If λ is any unit scalar, then λA can clearly also be written in Matsumoto-Amano normal form with the same T -count, namely by multiplying λ into the rightmost Clifford operator. Moreover, if A can be *uniquely* written in Matsumoto-Amano normal form, then the same is true for λA . Therefore, nothing is added or lost to the Matsumoto-Amano normal form whether one allows arbitrary global phases, a suitable discrete set of global phases (for example, powers of $e^{i\pi/4}$), or whether one works modulo global phase. Since it is convenient to work modulo global phase, we do so in the remainder; however, this does not restrict the generality of the results.

Definition 6.4.2. Let \mathcal{C} denote the Clifford group on one qubit, modulo global phases. This group has 24 elements. Let \mathcal{S} be the 8-element subgroup spanned by S and X . Let $\mathcal{C}' = \mathcal{C} \setminus \mathcal{S}$. Let $\mathcal{H} = \{I, H, SH\}$ and $\mathcal{H}' = \{H, SH\}$.

Lemma 6.4.3. *The following hold:*

$$\mathcal{C} = \mathcal{H}\mathcal{S}, \quad (6.5)$$

$$\mathcal{C}' = \mathcal{H}'\mathcal{S}, \quad (6.6)$$

$$\mathcal{S}\mathcal{H}' \subseteq \mathcal{H}'\mathcal{S}, \quad (6.7)$$

$$\mathcal{S}T = T\mathcal{S}, \quad (6.8)$$

$$T\mathcal{S}T = \mathcal{S}. \quad (6.9)$$

Proof. Since \mathcal{S} is an 8-element subgroup of \mathcal{C} , it has three left cosets. They are \mathcal{S} , $H\mathcal{S}$, and $SH\mathcal{S}$. Since \mathcal{C} is the disjoint union of these cosets, (equation (6.5)) and (equation (6.6)) immediately follow. For (equation (6.7)), first notice that $\mathcal{S}\mathcal{S} = \mathcal{S}$, and therefore $\mathcal{S}\mathcal{H}' = \mathcal{S}H \cup \mathcal{S}SH = \mathcal{S}H$. Since $\mathcal{S}H$ is a non-trivial right coset of \mathcal{S} , it follows that $\mathcal{S}H \subseteq \mathcal{C} \setminus \mathcal{S} = \mathcal{C}'$. Combining these facts with (equation (6.6)), we have (equation (6.7)). Finally, the equations (equation (6.8)) and (equation (6.9)) are trivial consequences of the equations $ST = TS$, $XT = TXS$, and $TT = S$. \square

Proposition 6.4.4 (Matsumoto and Amano [23]). *Every single-qubit Clifford+ T operator can be written in Matsumoto-Amano normal form.*

Proof. Let M be a single-qubit Clifford+ T operator. Clearly, M can be written as

$$M = C_n T C_{n-1} \cdots C_1 T C_0, \quad (6.10)$$

for some $n \geq 0$, where $C_0, \dots, C_n \in \mathcal{C}$. First note that if $C_i \in \mathcal{S}$ for any $i \in \{1, \dots, n-1\}$, then we can immediately use (equation (6.9)) to replace TC_iT by a single Clifford operator. This yields a shorter expression of the form (equation (6.10)) for M . We may therefore assume without loss of generality that $C_i \notin \mathcal{S}$ for $i = 1, \dots, n-1$. If $n = 0$, then M is a

Clifford operator, and there is nothing to show. Otherwise, we have

$$M \in \mathcal{C} T \mathcal{C}' \dots \mathcal{C}' T \mathcal{C} \quad \text{by (equation (6.10))} \quad (6.11)$$

$$= \mathcal{H} \mathcal{S} T \mathcal{H}' \mathcal{S} \dots \mathcal{H}' \mathcal{S} T \mathcal{C} \quad \text{by (equation (6.5)) and (equation (6.6))} \quad (6.12)$$

$$\subseteq \mathcal{H} T \mathcal{H}' \dots \mathcal{H}' T \mathcal{C} \quad \text{by (equation (6.7)) and (equation (6.8)).} \quad (6.13)$$

Note how, in the last step, the relations (equation (6.7) on the preceding page) and (equation (6.8) on the previous page) were used to move all occurrences of \mathcal{S} to the right, where they were absorbed into the final \mathcal{C} . It is now trivial to see that every element of (equation (6.13)) can be written in Matsumoto-Amano normal form, finishing the proof. \square

Corollary 6.4.5. *There exists a linear-time algorithm for symbolically reducing any sequence of Clifford+ T operators to Matsumoto-Amano normal form. More precisely, this algorithm runs in time at most $O(n)$, where n is the length of the input sequence.*

Proof. The proof of Proposition 6.4.4 on the previous page already contains an algorithm for reducing any sequence of Clifford+ T operators to Matsumoto-Amano normal form. However, in the stated form, it is perhaps not obvious that the algorithm runs in linear time. Indeed, a naive implementation of the first step would require up to n searches of the entire sequence for a term of the form TST , which can take time $O(n^2)$.

One obtains a linear time algorithm from the following observation: if M is already in Matsumoto-Amano normal form, and A is either a Clifford operator or T , then MA can be reduced to Matsumoto-Amano normal form in constant time. This is trivial when A is a Clifford operator, because it will simply be absorbed into the rightmost Clifford operator of M . In the case where $A = T$, a simple case distinction shows that at most the rightmost 5 elements of MA need to be updated. The normal form of a sequence of operators $A_1 A_2 \dots A_n$ can now be computed by starting with $M = I$ and repeatedly right-multiplying by A_1, \dots, A_n , reducing to normal form after each step. \square

6.4.2 T -Optimality

Corollary 6.4.6. *Let M be an operator in the Clifford+ T group, and assume that M can be written with T -count n . Then there exists a Matsumoto-Amano normal form for M with T -count at most n .*

Proof. This is an immediate consequence of the proof of Proposition 6.4.4 on page 170, because the reduction from (equation (6.10) on page 170) to (equation (6.13) on the preceding page) does not increase the T -count. \square

6.4.3 Uniqueness

Theorem 6.4.7 (Matsumoto and Amano [23]). *If M and N are two different Matsumoto-Amano normal forms, then they describe different operators.*

Recall that each single-qubit unitary operator (modulo global phase) can be uniquely represented as a rotation on the Bloch sphere, or equivalently, as an element of $SO(3)$, the real orthogonal 3×3 matrices with determinant 1. The relationship between an operator $U \in U(2)$ and its Bloch sphere representation $\hat{U} \in SO(3)$ is given by

$$\hat{U} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} \iff U(xX + yY + zZ)U^\dagger = x'X + y'Y + z'Z, \quad (6.14)$$

where X , Y , and Z are the Pauli operators. The Bloch sphere representations of the operators H , S , and T are:

$$\hat{H} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \hat{S} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \hat{T} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & \sqrt{2} \end{pmatrix}. \quad (6.15)$$

Remark 6.4.8. The Bloch sphere representation of any scalar is the identity matrix. The Bloch sphere representations of the 24 Clifford operators (modulo phase) are precisely those elements of $SO(3)$ that can be written with matrix entries in $\{-1, 0, 1\}$; these are exactly the 24 symmetries of the cube $\{(x, y, z) \mid -1 \leq x, y, z \leq 1\}$.

Definition 6.4.9. Recall that \mathbb{N} denotes the natural numbers including 0; \mathbb{Z} denotes the integers; and \mathbb{Z}_2 denotes the integers modulo 2. We define three subrings of the real numbers:

- $\mathbb{D} = \mathbb{Z}[\frac{1}{2}] = \{\frac{a}{2^n} \mid a \in \mathbb{Z}, n \in \mathbb{N}\}$. This is the ring of *dyadic fractions*.
- $\mathbb{Z}[\sqrt{2}] = \{a + b\sqrt{2} \mid a, b \in \mathbb{Z}\}$. This is the ring of *quadratic integers* with radicand 2.
- $\mathbb{D}[\sqrt{2}] = \mathbb{Z}[\frac{1}{\sqrt{2}}] = \{r + s\sqrt{2} \mid r, s \in \mathbb{D}\}$.

We will also need a fourth ring, which is not a subring of the real numbers.

- $\mathbb{Z}_2[\sqrt{2}] = \{a + b\sqrt{2} \mid a, b \in \mathbb{Z}_2\}$.

Note that the ring $\mathbb{Z}_2[\sqrt{2}]$ has only 4 elements; they are residue classes modulo 2 of the ring $\mathbb{Z}[\sqrt{2}]$. For brevity, we refer to the elements of $\mathbb{Z}_2[\sqrt{2}]$ as *residues*.

Definition 6.4.10 (Residue¹ and parity). Consider the unique ring homomorphism $\mathbb{Z} \rightarrow \mathbb{Z}_2$, mapping $a \in \mathbb{Z}$ to $\bar{a} \in \mathbb{Z}_2$, where $\bar{a} = 0$ if a is even and $\bar{a} = 1$ if a is odd. This induces a surjective ring homomorphism $\rho : \mathbb{Z}[\sqrt{2}] \rightarrow \mathbb{Z}_2[\sqrt{2}]$, defined by $\rho(a + b\sqrt{2}) = \bar{a} + \bar{b}\sqrt{2}$. For any given $x \in \mathbb{Z}[\sqrt{2}]$, we refer to $\rho(x)$ as the *residue of x* .

Moreover, consider the ring homomorphism $p : \mathbb{Z}[\sqrt{2}] \rightarrow \mathbb{Z}_2$ given by $p(a + b\sqrt{2}) = \bar{a}$. We refer to $p(x)$ as the *parity of x* .

Definition 6.4.11 (Denominator exponent). For every element $q \in \mathbb{D}[\sqrt{2}]$, there exists some natural number $k \geq 0$ such that $\sqrt{2}^k q \in \mathbb{Z}[\sqrt{2}]$, or equivalently, such that q can be written as $\frac{x}{\sqrt{2}^k}$, for some quadratic integer x . Such k is called a *denominator exponent* for q . The least such k is called the *least denominator exponent* of q .

More generally, we say that k is a denominator exponent for a vector or matrix if it is a denominator exponent for all of its entries. The least denominator exponent for a vector or matrix is therefore the least k that is a denominator exponent for all of its entries.

¹I don't think we need residue here, which is good as it will then be used only in section 7, the $U(2)$ case

$$\begin{array}{ccc}
\text{Start: } \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} & \xrightarrow{\mathfrak{C}} & \\
\downarrow \begin{matrix} k++ \\ T \end{matrix} & & \\
\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \xleftarrow[k++]{T} & \\
\begin{matrix} H \downarrow \\ \uparrow T_{k++} \end{matrix} & & \\
\begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} & \xrightarrow{S} & \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}
\end{array}$$

Figure 6.1: The action of Matsumoto-Amano normal forms on k -parities. All matrices are written modulo the right action of the Clifford group, i.e., modulo a permutation of the columns.

Definition 6.4.12 (k -parity). Let k be a denominator exponent for $q \in \mathbb{D}[\sqrt{2}]$. We define the k -residue of q , in symbols $\rho_k(q) \in \mathbb{Z}_2[\sqrt{2}]$, and the k -parity of q , in symbols $p_k(q) \in \mathbb{Z}_2$, by

$$\rho_k(q) = \rho(\sqrt{2}^k q), \quad p_k(q) = p(\sqrt{2}^k q).$$

The k -residue and k -parity of a vector or matrix are defined componentwise.

Remark 6.4.13. Let C be any Clifford operator, and \hat{C} its Bloch sphere representation. As noted above, the matrix entries of \hat{C} are in $\{-1, 0, 1\}$; it follows that \hat{C} has denominator exponent 0. In particular, it follows that multiplication by \hat{C} is a well-defined operation on parity matrices: for any 3×3 -matrix U with entries in $\mathbb{Z}_2[\sqrt{2}]$, we define $U \bullet C := U \cdot p(\hat{C})$. This defines a right action of the Clifford group on the set of parity matrices.

Definition 6.4.14. If G is any subgroup of the Clifford group, we define \sim_G to be the equivalence relation induced by this right action, i.e., for parity matrices U, V , we write $U \sim_G V$ if there exists some $C \in G$ such that $V = U \bullet C$. In case $G = \mathfrak{C}$ is the entire Clifford group, $U \sim_{\mathfrak{C}} V$ holds if and only if U and V differ by a permutation of columns.

Lemma 6.4.15. *Let M be a Matsumoto-Amano normal form, and $\hat{M} \in SO(3)$ the Bloch sphere operator of M . Let k be the least denominator exponent of \hat{M} . Then exactly one of the following holds:*

- $k = 0$, and M is a Clifford operator.
- $k > 0$, $p_k(\hat{M}) \sim_{\mathbb{C}} \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$, and the leftmost syllable in M is T .
- $k > 0$, $p_k(\hat{M}) \sim_{\mathbb{C}} \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$, and the leftmost syllable in M is HT .
- $k > 0$, $p_k(\hat{M}) \sim_{\mathbb{C}} \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$, and the leftmost syllable in M is SHT .

Moreover, the T -count of M is equal to k .

Proof. By induction on the length of the Matsumoto-Amano normal form M . Figure [figure 6.1 on the previous page](#) shows the action of Matsumoto-Amano operators on parity matrices. Each vertex represents a $\sim_{\mathbb{C}}$ -equivalence class of k -parities. The vertex labelled “Start” represents the empty Matsumoto-Amano normal form, i.e., the identity operator. Each arrow represents left multiplication by the relevant operator, i.e., a Clifford operator, T , H , or S . Thus, each Matsumoto-Amano normal form, read from right to left, gives rise to a unique path in the graph of Figure [figure 6.1 on the preceding page](#). The label $k++$ on an arrow indicates that the least denominator exponent increases by 1. The claims of the lemma then immediately follow from Figure [figure 6.1 on the previous page](#). \square

Proof of Theorem 6.4.7 on page 172. This is an immediate consequence of Lemma 6.4.15. Indeed, suppose that M and N are two Matsumoto-Amano normal forms describing the same Bloch sphere operator U . We show that $M = N$ by induction on the length of M . Let k be the least denominator exponent of U . If $k = 0$, then by Lemma 6.4.15, both M and N

are Clifford operators; since they describe the same Bloch sphere operator, they differ only by a phase.² If $k > 0$, then by Lemma 6.4.15, the Matsumoto-Amano normal forms M and N have the same leftmost operator (either T , H , or S), and the claim follows by induction hypothesis. \square

6.4.4 The Matsumoto-Amano decomposition algorithm

As an immediate consequence of Lemma 6.4.15 on page 174, we can an efficient algorithm for calculating the Matsumoto-Amano normal form of any Clifford+ T operator, given as a matrix.

Theorem 6.4.16. *Let $U \in SO(3)$ be the Bloch sphere representation of some Clifford+ T operator. Let k be the least denominator exponent of U . Then the Matsumoto-Amano normal form M of U can be efficiently computed with $O(k)$ arithmetic operations.*

Proof. By assumption, U is the Bloch sphere representation of some Clifford+ T operator. Let M be the unique Matsumoto-Amano normal form of this operator³. Note that, by Lemma 6.4.15 on page 174, the T -count of M is k . We compute M recursively. If $k = 0$, then by Lemma 6.4.15 on page 174, M is a Clifford operator; it can be determined from the matrix U in constant time. If $k > 0$, we compute $p_k(U)$, which must be one of the three cases listed in Lemma 6.4.15 on page 174. This determines whether the leftmost syllable of M is T , HT , or SHT . Let N be this syllable, so that $M = NM'$, for some Matsumoto-Amano normal form M' . Then M' can be recursively computed as the Matsumoto-Amano normal form of $U' = \hat{N}^{-1}U$; moreover, since M' has T -count $k - 1$, the recursion terminates after k steps. Since each induction step only requires a constant number of arithmetic operations, the total number of operations is $O(k)$. \square

²Todo: fix the treatment of phases

³Todo: treat phase correctly

6.4.5 A characterization of Clifford+ T on the Bloch sphere

Lemma 6.4.17. *Let $U \in SO(3)$ be an orthogonal matrix with entries in $\mathbb{D}[\sqrt{2}]$. Let k be a denominator exponent of U , and let v_1, v_2, v_3 be the columns of U , with*

$$v_j = \frac{1}{\sqrt{2}^k} \begin{pmatrix} a_j + b_j\sqrt{2} \\ c_j + d_j\sqrt{2} \\ e_j + f_j\sqrt{2} \end{pmatrix},$$

for $a_j, \dots, f_j \in \mathbb{Z}$. Then for all $j, \ell \in \{1, 2, 3\}$,

$$a_j b_\ell + b_j a_\ell + c_j d_\ell + d_j c_\ell + e_j f_\ell + f_j e_\ell = 0 \quad (6.16)$$

and

$$a_j a_\ell + c_j c_\ell + e_j e_\ell + 2(b_j b_\ell + d_j d_\ell + f_j f_\ell) = 2^k \langle v_j, v_\ell \rangle. \quad (6.17)$$

In particular, we have, for all $j \in \{1, 2, 3\}$,

$$a_j b_j + c_j d_j + e_j f_j = 0 \quad (6.18)$$

and

$$a_j^2 + c_j^2 + e_j^2 + 2(b_j^2 + d_j^2 + f_j^2) = 2^k. \quad (6.19)$$

Proof. Computing the inner product, we have

$$\begin{aligned} \langle v_j, v_\ell \rangle = & \frac{1}{2^k} (a_j a_\ell + c_j c_\ell + e_j e_\ell + 2(b_j b_\ell + d_j d_\ell + f_j f_\ell) \\ & + \sqrt{2}(a_j b_\ell + b_j a_\ell + c_j d_\ell + d_j c_\ell + e_j f_\ell + f_j e_\ell)). \end{aligned} \quad (6.20)$$

Since $U^\dagger U = I$, we have $\langle v_j, v_j \rangle = 1$, and $\langle v_j, v_\ell \rangle = 0$ when $\ell \neq j$. Therefore, the coefficient of $\sqrt{2}$ in equation (6.20) must be zero, proving (6.16) and (6.17).

Equations (6.18) and (6.19) immediately follow by letting $j = \ell$. \square

Remark 6.4.18. In Lemma 6.4.17 on the preceding page, we have worked with columns v_j of the matrix U . But since U is orthogonal, the analogous properties also hold for the rows of U .

Lemma 6.4.19. *Let $U \in SO(3)$ be an orthogonal matrix with entries in $\mathbb{D}[\sqrt{2}]$, and with least denominator exponent $k = 0$. Then U is the Bloch sphere representation of some Clifford operator.*

Proof. Let v_j be any column of U , with the notation of Lemma 6.4.17 on the previous page. By equation (6.19) on the preceding page, we have $a_j^2 + c_j^2 + e_j^2 + 2(b_j^2 + d_j^2 + f_j^2) = 1$. Since each summand is a positive integer, we must have $b_j, d_j, f_j = 0$, and exactly one of a_j, c_j or $e_j = \pm 1$, for each $j = 1, 2, 3$. Therefore, all the matrix entries are in $\{-1, 0, 1\}$, and the claim follows by Remark 6.4.8 on page 172. \square

Lemma 6.4.20. *Let $U \in SO(3)$ be an orthogonal matrix with entries in $\mathbb{D}[\sqrt{2}]$, and let k be the least denominator exponent of U . If $k = 0$, then $p_k(U) \sim_{\mathcal{C}} M_1$. If $k > 0$, then $p_k(U) \sim_{\mathcal{C}} M$ for some $M \in \{M_T, M_H, M_S\}$, where*

$$M_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad M_T = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad M_H = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \quad M_S = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}.$$

Proof. First consider the case $k = 0$. Let v_j be any column of U , with the notation of Lemma 6.4.17 on the preceding page. By equation (6.19) on the previous page, we have $a_j^2 + c_j^2 + e_j^2 + 2(b_j^2 + d_j^2 + f_j^2) = 1$. Since each summand is a positive integer, we must have $b_j, d_j, f_j = 0$, and exactly one of a_j, c_j or $e_j = \pm 1$, for each $j = 1, 2, 3$. Noting that the columns of U are orthogonal, we see that $p_k(U) \sim_{\mathcal{C}} M_1$.

Now consider the case $k > 0$. Let v_j be any row or column of U , with the notation of Lemma 6.4.17 on the preceding page. By equation (6.19) on the previous page, it follows that $a_j^2 + c_j^2 + e_j^2$ is even, and therefore an even number of a_j, c_j , and e_j have parity 1. Therefore, each row or column of $p_k(U)$ has an even number of 1's. Moreover, since k is the least

denominator exponent of U , $p_k(U)$ has at least one non-zero entry. Modulo a permutation of columns, this leaves exactly four possibilities for $p_k(U)$:

$$(a) \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (b) \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \quad (c) \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \quad (d) \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

In cases (a)–(c), we are done. Case (d) is impossible because it implies that $a_1a_2 + c_1c_2 + e_1e_2$ is odd, contradicting the fact that it is even by [equation \(6.17\) on page 177](#). \square

Lemma 6.4.21. *Let $U \in SO(3)$ be an orthogonal matrix with entries in $\mathbb{D}[\sqrt{2}]$, and with least denominator exponent $k > 0$. Then there exists $N \in \{T, HT, SHT\}$ such that the least denominator exponent of $\hat{N}^{-1}U$ is $k - 1$.*

Proof. By [Lemma 6.4.20 on the previous page](#), we know that $p_k(U) \sim_{\mathbb{C}} M$, for some $M \in \{M_T, M_H, M_S\}$. We consider each of these cases.

1. $p_k(U) \sim_{\mathbb{C}} M_T$. By assumption, U has two columns v with $p_k(v) = (1, 1, 0)^T$.

Let

$$v = \frac{1}{\sqrt{2}^k} \begin{pmatrix} a + b\sqrt{2} \\ c + d\sqrt{2} \\ e + f\sqrt{2} \end{pmatrix}$$

be any such column. By [equation \(6.18\) on page 177](#), we have $ab + cd + ef = 0$.

Since $\bar{e} = 0$, we have $\bar{a}\bar{b} + \bar{c}\bar{d} = 0$. Since $\bar{a} = \bar{c} = 1$, we can conclude $\bar{b} + \bar{d} = 0$.

Applying \hat{T}^{-1} to v , we compute:

$$\begin{aligned} \hat{T}^{-1}v &= \frac{1}{\sqrt{2}^{k+1}} \begin{bmatrix} c + a & + & (d + b)\sqrt{2} \\ c - a & + & (d - b)\sqrt{2} \\ e\sqrt{2} & + & 2f \end{bmatrix} = \frac{1}{\sqrt{2}^{k-1}} \begin{bmatrix} \frac{c+a}{2} & + & \frac{d+b}{\sqrt{2}} \\ \frac{c-a}{2} & + & \frac{d-b}{\sqrt{2}} \\ \frac{e}{\sqrt{2}} & + & f \end{bmatrix} \\ &= \frac{1}{\sqrt{2}^{k-1}} \begin{bmatrix} a' & + & b'\sqrt{2} \\ c' & + & d'\sqrt{2} \\ f & + & e'\sqrt{2} \end{bmatrix} \end{aligned}$$

where $a' = \frac{c+a}{2}, b' = \frac{d+b}{2}, c' = \frac{c-a}{2}, d' = \frac{d-b}{2}$ and $e' = \frac{e}{2}$ are all integers. Hence, $k-1$ is a denominator exponent of $\hat{T}^{-1}v$. Moreover, since $a' + c' = c$ is odd, one of a' and c' is odd, proving that $k-1$ is the least denominator exponent of $\hat{T}^{-1}v$.

Now consider the third column w of U , where $p_k(w) = (0, 0, 0)^T$. Then $k-1$ is a denominator exponent of w , so that k is a denominator exponent for $\hat{T}^{-1}w$.

Let

$$p_k(\hat{T}^{-1}w) = \begin{pmatrix} x \\ y \\ z \end{pmatrix}.$$

As the least denominator exponent of the other two column of $p_k(\hat{T}^{-1}U)$ is $k-1$, we have

$$p_k(\hat{T}^{-1}U) \sim_{\mathbb{C}} \begin{bmatrix} 0 & 0 & x \\ 0 & 0 & y \\ 0 & 0 & z \end{bmatrix}.$$

But $\hat{T}^{-1}U$ is orthogonal, so by [equation \(6.19\) on page 177](#), applied to each row of $\hat{T}^{-1}U$, we conclude that $x = y = z = 0$. It follows that the least denominator exponent of $\hat{T}^{-1}U$ is $k-1$.

2. $p_k(U) \sim_{\mathbb{C}} M_H$. In this case, $p_k(\hat{H}^{-1}U) \sim_{\mathbb{C}} p(\hat{H}^{-1}M_H) = M_T$. We then continue as in [case 1 on the preceding page](#).
3. $p_k(U) \sim_{\mathbb{C}} M_S$. In this case, $p_k(\hat{H}^{-1}\hat{S}^{-1}U) \sim_{\mathbb{C}} p(\hat{H}^{-1}\hat{S}^{-1}M_S) = M_T$. We then continue as in [case 1 on the previous page](#). □

Combining [Lemmas 6.4.19 on page 178](#) and [6.4.20 on page 178](#), we easily get the following result:

Theorem 6.4.22. *Let $U \in SO(3)$ be an orthogonal matrix. Then U is the Bloch sphere representation of some Clifford+T operator M if and only if the entries of U are in the ring $\mathbb{D}[\sqrt{2}]$.*

Proof. The “only if” direction is trivial, since all the generators of the Clifford+ T group have this property (see [equation \(6.15\)](#)). To prove the “if” direction, let k be the least denominator exponent of U . We proceed by induction on k . If $k = 0$, by [Lemma 6.4.19](#), U is the Bloch sphere representation of some Clifford operator, and therefore a Clifford+ T operator. If $k > 0$, then by [Lemma 6.4.20](#), we can write $U = \hat{N}U'$, where $N \in \{T, HT, SHT\}$ and U' has least denominator exponent $k - 1$. By induction hypothesis, U' is a Clifford+ T operator, and therefore so is U . \square

Remark 6.4.23. Combining this result with the algorithm of [Theorem 6.4.16 on page 176](#), we have a linear-time algorithm for computing the Matsumoto-Amano normal form of any unitary operator $U \in SO(3)$ with entries in $\mathbb{D}[\sqrt{2}]$.

Corollary 6.4.24 (Kliuchnikov et al. [\[19\]](#)). *Let $U \in U(2)$ be a unitary matrix. Then U is a Clifford+ T operator if and only if the matrix entries of U are in the ring $\mathbb{D}[\sqrt{2}, i]$.*

Proof. Again, the “only if” direction is trivial, as it is true for the generators. For the “if” direction, it suffices to note that, by [equation \(6.14\) on page 172](#), whenever U takes its entries in $\mathbb{D}[\sqrt{2}, i]$, then \hat{U} takes its entries in $\mathbb{D}[\sqrt{2}]$.⁴ \square

[Corollary 6.4.24](#) was first proved by Kliuchnikov et al. [\[19\]](#), using a direct method (i.e., not going via the Bloch sphere representation). It is interesting to note that [Theorem 6.4.22 on the previous page](#) is stronger than [Corollary 6.4.24](#), in the sense that the Theorem obviously implies the Corollary, whereas it is not a priori obvious that the Corollary implies the Theorem.

6.4.6 Alternative normal forms

With the exception of the left-most and right-most gates, the Matsumoto-Amano normal form uses syllables of the form HT and SHT . It is of course possible to use different sets of syllables instead.

⁴Todo: treat phase correctly; also introduce the ring $\mathbb{D}[\sqrt{2}, i]$ at some appropriate time

E - T normal form

Consider the Clifford operator

$$E = HS^3\omega^3 = \frac{1}{2} \begin{pmatrix} -1+i & 1+i \\ -1+i & -1-i \end{pmatrix}.$$

It has the following properties:

$$E^3 = I, \quad EXE^{-1} = Y, \quad EYE^{-1} = Z, \quad EZE^{-1} = X.$$

The operator E serves as a convenient operator for switching between the X -, Y -, and Z -bases. On the Bloch sphere, it represents a rotation by 120 degrees about the axis $(1, 1, 1)^T$:

$$\hat{E} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

The operators E and E^2 have properties analogous to H and SH . Specifically, if we let $\mathcal{H} = \{I, E, E^2\}$ and $\mathcal{H}' = \{E, E^2\}$, then the properties of Lemma 6.4.3 are satisfied. The proofs of Proposition 6.4.4 and Corollary 6.4.5 only depend on these properties, and the uniqueness proof (Theorem 6.4.7 on page 172) also goes through without significant changes. We therefore have:

Proposition 6.4.25 (E - T normal form). *Every single-qubit Clifford+ T operator can be uniquely written in the form*

$$(T \mid \varepsilon) (ET \mid E^2T)^* \mathcal{C}. \quad (6.21)$$

Moreover, this normal form has minimal T -count, and there exists a linear-time algorithm for symbolically reducing any sequence of Clifford+ T operators to this normal form.

T_x - T_y - T_z normal form

It is plain to see that every syllable of the E - T normal form (except perhaps the first or last one) consists of a 45 degree z -rotation, followed by a basis change that rotates either the x - or y -axis into the z -position. Abstracting away from these basis changes, the entire

normal form can therefore be regarded as a sequence of 45-degree rotations about the x -, y -, and z -axes. More precisely, let us define variants of the T -gate that rotate about the three different axes:

$$T_x = ETE^2,$$

$$T_y = E^2TE,$$

$$T_z = T.$$

Using the commutativities $ET_x = T_yE$, $ET_y = T_zE$, and $ET_z = T_xE$, it is then clear that every expression of the form ([equation \(6.21\) on the preceding page](#)) can be uniquely rewritten as a sequence of T_x , T_y , and T_z rotations, with no repeated symbol, followed by a Clifford operator. This can be easily proved by induction, but is best seen in an example:

$$\begin{aligned} TETETE^2TEC &= T_zET_zET_zE^2T_zEC \\ &\rightarrow T_zT_xE^2T_zE^2T_zEC \\ &\rightarrow T_zT_xT_yE^4T_zEC \\ &\rightarrow T_zT_xT_yET_zEC \\ &\rightarrow T_zT_xT_yT_xE^2C \\ &\rightarrow T_zT_xT_yT_xC'. \end{aligned}$$

We have:

Proposition 6.4.26 (T_x - T_y - T_z normal form). *Every single-qubit Clifford+ T operator can be uniquely written in the form*

$$T_{r_1}T_{r_2}\dots T_{r_n}C,$$

where $n \geq 0$, $r_1, \dots, r_n \in \{x, y, z\}$, and $r_i \neq r_{i+1}$ for all $i \leq n-1$. We define the T -count of such an expression to be n ; then this normal form has minimal T -count. Moreover, there exists a linear-time algorithm for symbolically reducing any sequence of Clifford+ T operators to this normal form.

The T_x - T_y - T_z normal form is, in a sense, the most “canonical” one of the normal forms considered here; it also explains why T -count is an appropriate measure of the size of a

Clifford+ T operator. In a physical quantum computer with error correction, there is in general no reason to expect the T_z gate to be more privileged than the T_x or T_y gates; one may imagine that it would be efficient for a quantum computer to provide all three T -gates as primitive logical operations.

Bocharov-Svore normal form

Bocharov and Svore [10, Proposition 1] consider the following normal form for single-qubit Clifford+ T circuits:

$$(H \mid \varepsilon)(TH \mid SHTH)^* \mathcal{C}. \quad (6.22)$$

This normal form is not unique; for example, $H.H$ and I are two different normal forms denoting the same operator, as are $SHTH.Z$ and $H.SHTH$. (Here we have used a dot to delimit syllables; this is for readability only). Recall that two regular expressions are *equivalent* if they define the same set of strings. Using laws of regular expressions, we can equivalently rewrite (equation (6.22)) as

$$((\epsilon \mid T \mid SHT)(HT \mid HSHT)^* HC) \mid \mathcal{C}. \quad (6.23)$$

Since HC is just a redundant way to write a Clifford operator, we can simplify it to \mathcal{C} ; moreover, in this case, $\epsilon\mathcal{C}$ and \mathcal{C} are the same, so (equation (6.23)) simplifies to

$$(\epsilon \mid T \mid SHT)(HT \mid HSHT)^* \mathcal{C}. \quad (6.24)$$

Moreover, since $SHT = HSHT.X$, any expression starting with SHT can be rewritten as one starting with $HSHT$, so the SHT syllable is redundant and we can eliminate it:

$$(\epsilon \mid T)(HT \mid HSHT)^* \mathcal{C}. \quad (6.25)$$

Let us say that an operator is in *Bocharov-Svore normal form* if it is written in the form (equation (6.25)). This version of the Bocharov-Svore normal form is indeed unique; note that it is almost the same as the Matsumoto-Amano normal form, except that the syllable

SHT has been replaced by $HSHT$. Since the set $\mathcal{H} = \{I, H, HSH\}$ satisfies Lemma 6.4.3 on page 169, existence, uniqueness, T -optimality, and efficiency are proved in the same way as for the Matsumoto-Amano and E - T normal forms.

Bocharov and Svore [10, Prop.2] also consider a second normal form, which has Clifford operators on both sides, but the first four interior syllables restricted to TH :

$$\mathcal{C}(\epsilon \mid TH \mid (TH)^2 \mid (TH)^3 \mid (TH)^4(TH \mid SHTH)^*)\mathcal{C} \quad (6.26)$$

However, this normal form is not at all unique; for instance, $Z.TH$ and $TH.X$ denote the same operator, as do $YS.TH.TH$ and $TH.TH.X\omega$.

6.4.7 Matsumoto-Amano normal forms and $U(2)$

Example 6.4.27. Consider the matrix

$$U = \frac{1}{\sqrt{2}^3} \begin{pmatrix} \omega^2 + \omega & -2\omega^3 + \omega^2 + \omega \\ \omega^3 - 2\omega^2 - 1 & -\omega^3 + 1 \end{pmatrix}.$$

It has least denominator exponent 3. Its 3-, 4-, and 5-residues are:

$$\rho_3(U) = \begin{pmatrix} 0110 & 0110 \\ 1001 & 1001 \end{pmatrix}, \quad \rho_4(U) = \begin{pmatrix} 1111 & 1111 \\ 1111 & 1111 \end{pmatrix}, \quad \rho_5(U) = 0.$$

Definition 6.4.28. Let \mathcal{S}^ω be the 64-element subgroup of the Clifford group in $U(2)$ spanned by S, X and ω . Then $\sim_{\mathcal{S}^\omega}$ defined by right multiplication by \mathcal{S}^ω is an equivalence relation on the residue matrices of operators in the Clifford+ T group.

Remark 6.4.29. The equivalence relation $\sim_{\mathcal{S}^\omega}$ is characterized by the following operations:

1. “Rotating” all of the entries in the matrix by 1,2 or 3 positions. This corresponds to multiplication by a power of ω .
2. Swapping the two columns. This corresponds to the right action of X .
3. “Rotating” the entries of the second column by two positions. This corresponds to the right action of S .

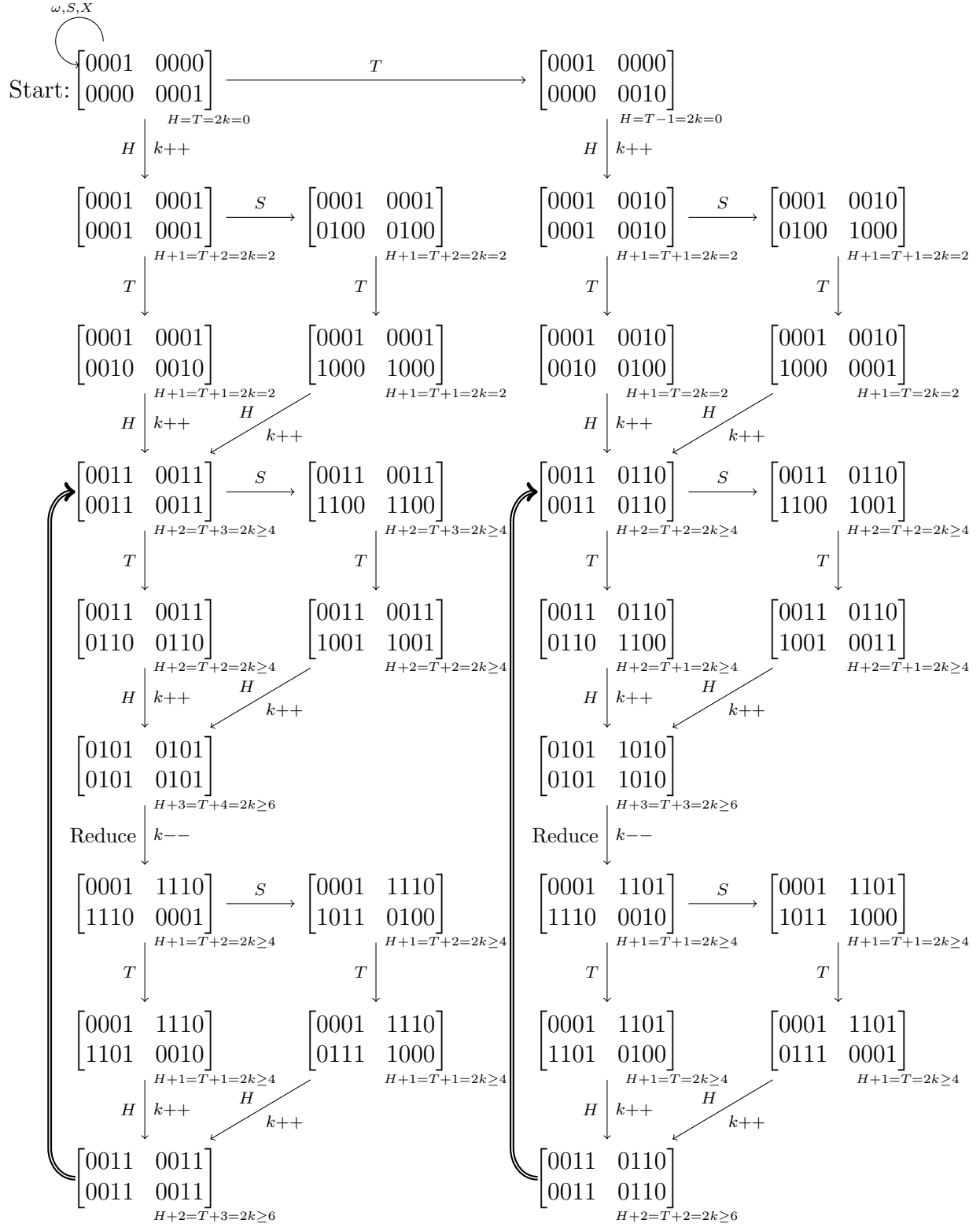


Figure 6.2: Transitions of residue matrices in U2 when applying the Matsumoto-Amano algorithm

Remark 6.4.30. The residue matrices in figure [figure 6.2 on the previous page](#) are modulo $\sim_{S\omega}$.

Lemma 6.4.31. *Let $k \geq 2$ and $b, d \in \mathbb{Z}[\omega]$. Then $\frac{1}{\sqrt{2}^k} \begin{pmatrix} 1 + 2b \\ 1 + 2d \end{pmatrix}$ is not a unit vector.*

Proof. Suppose otherwise, then

$$2^k = 1 + 2(b + b^t) + 4bb^t + 1 + 2(d + d^t) + 4dd^t,$$

so $2 = 2^k - 2(b + b^t) - 2(d + d^t) - 4(bb^t + dd^t)$. By lemma [6.3.16 on page 166](#), the right hand side of this is divisible in $\mathbb{Z}[\omega]$ by $2\sqrt{2}$, while the left hand side is not. Thus we have a contradiction. \square

Lemma 6.4.32. *Given a unitary matrix $U \in \mathbb{D}[\omega]^{2 \times 2}$ with least denominator exponent $k \geq 2$, such that:*

$$\begin{aligned} 1. \quad \rho_{k+1}(U) &= \begin{bmatrix} 0101 & 0101 \\ 0101 & 0101 \end{bmatrix} \text{ and} \\ 2. \quad \rho_k(HU) &= \begin{bmatrix} 0011 & 0011 \\ 0110 & 0110 \end{bmatrix}, \end{aligned}$$

$$\text{then, } \rho_k(U) = \begin{bmatrix} 0010 & 1101 \\ 1101 & 0010 \end{bmatrix}$$

Proof. Referencing Table [table 6.1 on page 165](#), we see the first condition limits the possible choices for the entries of $\rho_k(U)$ to the set $\{0010, 0111, 1000, 1101\}$. The second condition implies that $\rho_{k+1}(HU)$ is reducible and in fact that each entry is 1111. This means each column of U must be either $[0010, 1101]^t$, $[1101, 0010]^t$, $[0111, 1000]^t$ or $[1000, 0111]^t$. As we are considering equivalence classes, we can assume without loss of generality, that the columns

are in $\{[0010, 1101]^t, [1101, 0010]^t\}$. But by Lemma 6.4.31 on the preceding page, we can not have a row like $[0010, 0010]$, therefore $U = \begin{bmatrix} 0010 & 1101 \\ 1101 & 0010 \end{bmatrix}$. \square

Corollary 6.4.33. *In Figure figure 6.2 on page 186, the transitions labelled with “Reduce” are correct.*

Proof. For the left most reduce, we directly apply Lemma 6.4.32 on the preceding page and then note that

$$\begin{bmatrix} 0001 & 1110 \\ 1110 & 0001 \end{bmatrix} \sim_{S^\omega} \begin{bmatrix} 0010 & 1101 \\ 1101 & 0010 \end{bmatrix}.$$

For the rightmost reduce, the same argument as shown in Lemma 6.4.32 on the previous page can be applied to the preconditions of this reduce, giving us a similar result. \square

6.5 Exact synthesis of multi-qubit operators

Here, we focus on the problem of exact synthesis for n -qubit operators, using the Clifford+ T universal gate set. Recall that the Clifford group on n qubits is generated by the Hadamard gate H , the phase gate S , the controlled-not gate, and the scalar $\omega = e^{i\pi/4}$ (one may allow arbitrary unit scalars, but it is not convenient for our purposes to do so). It is well-known that one obtains a universal gate set by adding the non-Clifford operator T [25].

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

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}. \quad (6.27)$$

In addition to the Clifford+ T group on n qubits, as defined above, we also consider the slightly larger group of Clifford+ T operators “with ancillas”. We say that an n -qubit

operator U is a Clifford+ T operator *with ancillas* if there exists $m \geq 0$ and a Clifford+ T operator U' on $n + m$ qubits, such that $U'(|\phi\rangle \otimes |0\rangle) = (U|\phi\rangle) \otimes |0\rangle$ for all n -qubit states $|\phi\rangle$.

Kliuchnikov, Maslov, and Mosca [19] showed that a single-qubit operator U is in the Clifford+ T group if and only if all of its matrix entries belong to the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$. They also showed that the Clifford+ T groups “with ancillas” and “without ancillas” coincide for $n = 1$, but not for $n \geq 2$. Moreover, Kliuchnikov et al. conjectured that for all n , an n -qubit operator U is in the Clifford+ T group with ancillas if and only if its matrix entries belong to $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$. They also conjectured that a single ancilla qubit is always sufficient in the representation of a Clifford+ T operator with ancillas. This section of the thesis will prove these conjectures. In particular, this yields an algorithm for exact Clifford+ T synthesis of n -qubit operators. We also obtain a characterization of the Clifford+ T group on n qubits without ancillas.

It is important to note that, unlike in the single-qubit case, the circuit synthesized here are not in any sense canonical, and very far from optimal. Thus, the question of *efficient* synthesis is not addressed here.

6.5.1 Decomposition into two-level matrices

Recall that a *two-level matrix* is an $n \times n$ -matrix that acts non-trivially on at most two vector components [25]. If

$$U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

is a 2×2 -matrix and $j \neq \ell$, we write $U_{[j,\ell]}$ for the two-level $n \times n$ -matrix defined by

$$U_{[j,\ell]} = \begin{array}{c} \vdots \\ j \\ \vdots \\ \ell \\ \vdots \end{array} \begin{array}{ccccc} \cdots & j & \cdots & \ell & \cdots \\ \hline & I & & & \\ \hline & & a & & b \\ \hline & & & I & \\ \hline & & c & & d \\ \hline & & & & I \end{array},$$

and we say that $U_{[j,\ell]}$ is a two-level matrix *of type U* . Similarly, if a is a scalar, we write $a_{[j]}$ for the one-level matrix

$$a_{[j]} = \begin{array}{c} \vdots \\ j \\ \vdots \end{array} \begin{array}{ccc} \cdots & j & \cdots \\ \hline & I & \\ \hline & & a \\ \hline & & I \end{array},$$

and we say that $a_{[j]}$ is a one-level matrix *of type a* .

Lemma 6.5.1 (Row operation). *Let $u = (u_1, u_2)^T \in \mathbb{D}[\omega]^2$ be a vector with denominator exponent $k > 0$ and k -residue $\rho_k(u) = (x_1, x_2)$, such that $x_1^\dagger x_1 = x_2^\dagger x_2$. Then there exists a sequence of matrices U_1, \dots, U_h , each of which is H or T , such that $v = U_1 \cdots U_h u$ has denominator exponent $k - 1$, or equivalently, $\rho_k(v)$ is defined and reducible.*

Proof. It can be seen from Table [table 6.1 on page 165](#) that $x_1^\dagger x_1$ is either 0000, 1010, or 0001.

- Case 1: $x_1^\dagger x_1 = x_2^\dagger x_2 = 0000$. In this case, $\rho_k(u)$ is already reducible, and there is nothing to show.
- Case 2: $x_1^\dagger x_1 = x_2^\dagger x_2 = 1010$. In this case, we know from Table [table 6.1 on page 165](#) that $x_1, x_2 \in \{0011, 0110, 1100, 1001\}$. In particular, x_1 is a cyclic

permutation of x_2 , say, $x_1 = \omega^m x_2$. Let $v = HT^m u$. Then

$$\begin{aligned}\rho_k(\sqrt{2}v) &= \rho_k\left(\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & \omega^m \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}\right) \\ &= \rho_k\begin{pmatrix} u_1 + \omega^m u_2 \\ u_1 - \omega^m u_2 \end{pmatrix} \\ &= \begin{pmatrix} x_1 + \omega^m x_2 \\ x_1 - \omega^m x_2 \end{pmatrix} = \begin{pmatrix} 0000 \\ 0000 \end{pmatrix}.\end{aligned}$$

This shows that $\rho_k(\sqrt{2}v)$ is twice reducible; therefore, $\rho_k(v)$ is defined and reducible as claimed.

- Case 3: $x_1^\dagger x_1 = x_2^\dagger x_2 = 0001$. In this case, we know from Table [table 6.1 on page 165](#) that $x_1, x_2 \in \{0001, 0010, 0100, 1000\} \cup \{0111, 1110, 1101, 1011\}$. If both x_1, x_2 are in the first set, or both are in the second set, then x_1 and x_2 are cyclic permutations of each other, and we proceed as in case 2. The only remaining cases are that x_1 is a cyclic permutation of 0001 and x_2 is a cyclic permutation of 0111, or vice versa. But then there exists some m such that $x_1 + \omega^m x_2 = 1111$. Letting $u' = HT^m u$, we have

$$\begin{aligned}\rho_k(\sqrt{2}u') &= \rho_k\left(\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & \omega^m \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}\right) \\ &= \rho_k\begin{pmatrix} u_1 + \omega^m u_2 \\ u_1 - \omega^m u_2 \end{pmatrix} \\ &= \begin{pmatrix} x_1 + \omega^m x_2 \\ x_1 - \omega^m x_2 \end{pmatrix} = \begin{pmatrix} 1111 \\ 1111 \end{pmatrix}.\end{aligned}$$

Since this is reducible, u' has denominator exponent k . Let $\rho_k(u') = (y_1, y_2)$. Because $\sqrt{2}y_1 = \sqrt{2}y_2 = 1111$, we see from Table [table 6.1 on page 165](#) that $y_1, y_2 \in \{0011, 0110, 1100, 1001\}$ and $y_1^\dagger y_1 = y_2^\dagger y_2 = 1010$. Therefore, u' satisfies the condition of case 2 above. Proceeding as in case 2, we find m' such that $v = HT^{m'} u' = HT^{m'} HT^m u$ has denominator exponent $k - 1$. This finishes the proof. \square

Lemma 6.5.2 (Column lemma). *Consider a unit vector $u \in \mathbb{D}[\omega]^n$, i.e., an n -dimensional column vector of norm 1 with entries from the ring $\mathbb{D}[\omega]$. Then there exist a sequence U_1, \dots, U_h of one- and two-level unitary matrices of types X , H , T , and ω such that $U_1 \cdots U_h u = e_1$, the first standard basis vector.*

Proof. The proof is by induction on k , the least denominator exponent of u . Let $u = (u_1, \dots, u_n)^T$.

- Base case. Suppose $k = 0$. Then $u \in \mathbb{Z}[\omega]^n$. Since by assumption $\|u\|^2 = 1$, it follows by Lemma 6.3.6 on page 164 that $\|u\|_{\text{weight}}^2 = 1$. Since u_1, \dots, u_n are elements of $\mathbb{Z}[\omega]$, their weights are non-negative integers. It follows that there is precisely one j with $\|u_j\|_{\text{weight}} = 1$, and $\|u_\ell\|_{\text{weight}} = 0$ for all $\ell \neq j$. Let $u' = X_{[1,j]}u$ if $j \neq 1$, and $u' = u$ otherwise. Now u'_1 is of the form ω^{-m} , for some $m \in \{0, \dots, 7\}$, and $u'_\ell = 0$ for all $\ell \neq 1$. We have $\omega_{[1]}^m u' = e_1$, as desired.
- Induction step. Suppose $k > 0$. Let $v = \sqrt{2}^k u \in \mathbb{Z}[\omega]^n$, and let $x = \rho_k(u) = \rho(v)$. From $\|u\|^2 = 1$, it follows that $\|v\|^2 = v_1^\dagger v_1 + \dots + v_n^\dagger v_n = 2^k$. Taking residues of the last equation, we have

$$x_1^\dagger x_1 + \dots + x_n^\dagger x_n = 0000. \quad (6.28)$$

It can be seen from Table table 6.1 on page 165 that each summand $x_j^\dagger x_j$ is either 0000, 0001, or 1010. Since their sum is 0000, it follows that there is an even number of j such that $x_j^\dagger x_j = 0001$, and an even number of j such that $x_j^\dagger x_j = 1010$.

We do an inner induction on the number of irreducible components of x . If x is reducible, then u has denominator exponent $k - 1$ by Corollary 6.3.15 on page 166, and we can apply the outer induction hypothesis. Now suppose there is some j such that x_j is irreducible; then $x_j^\dagger x_j \neq 0000$ by Lemma 6.3.13 on page 166. Because of the evenness property noted above, there must exist

some $\ell \neq j$ such that $x_j^\dagger x_j = x_\ell^\dagger x_\ell$. Applying Lemma 6.5.1 on page 190 to $u' = (u_j, u_\ell)^T$, we find a sequence \vec{U} of row operations of types H and T , making $\rho_k(\vec{U}u')$ reducible. We can lift this to a two-level operation $\vec{U}_{[j,\ell]}$ acting on u ; thus $\rho_k(\vec{U}_{[j,\ell]}u)$ has fewer irreducible components than $x = \rho_k(u)$, and the inner induction hypothesis applies. \square

Lemma 6.5.3 (Matrix decomposition). *Let U be a unitary $n \times n$ -matrix with entries in $\mathbb{D}[\omega]$. Then there exists a sequence U_1, \dots, U_h of one- and two-level unitary matrices of types X, H, T , and ω such that $U = U_1 \cdots U_h$.*

Proof. Equivalently, it suffices to show that there exist one- and two-level unitary matrices V_1, \dots, V_h of types X, H, T , and ω such that $V_h \cdots V_1 U = I$. This is an easy consequence of the column lemma, exactly as in e.g. [25, Sec. 4.5.1]. Specifically, first use the column lemma to find suitable one- and two-level row operations V_1, \dots, V_{h_1} such that the leftmost column of $V_{h_1} \cdots V_1 U$ is e_1 . Because $V_{h_1} \cdots V_1 U$ is unitary, it is of the form

$$\left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & U' \end{array} \right).$$

Now recursively find row operations to reduce U' to the identity matrix. \square

Example 6.5.4. We will decompose the matrix U from Example 6.3.18. We start with the first column u of U :

$$u = \frac{1}{\sqrt{2}^3} \begin{pmatrix} -\omega^3 + \omega - 1 \\ \omega^2 + \omega \\ \omega^3 + \omega^2 \\ -1 \end{pmatrix},$$

$$\rho_3(u) = \begin{pmatrix} 1011 \\ 0110 \\ 1100 \\ 0001 \end{pmatrix}, \quad \rho_3(u_j^\dagger u_j) = \begin{pmatrix} 0001 \\ 1010 \\ 1010 \\ 0001 \end{pmatrix}.$$

Rows 2 and 3 satisfy case 2 of Lemma 6.5.1 on page 190. As they are not aligned, first apply $T_{[2,3]}^3$ and then $H_{[2,3]}$. Rows 1 and 4 satisfy case 3. Applying $H_{[1,4]}T_{[1,4]}^2$, the residues become $\rho_3(u'_1) = 0011$ and $\rho_3(u'_4) = 1001$, which requires applying $H_{[1,4]}T_{[1,4]}$. We now have

$$H_{[1,4]}T_{[1,4]}H_{[1,4]}T_{[1,4]}^2H_{[2,3]}T_{[2,3]}^3u = v = \frac{1}{\sqrt{2}^2} \begin{pmatrix} 0 \\ 0 \\ \omega^2 + \omega \\ -\omega + 1 \end{pmatrix},$$

$$\rho_2(v) = \begin{pmatrix} 0000 \\ 0000 \\ 0110 \\ 0011 \end{pmatrix}, \quad \rho_2(v_j^\dagger v_j) = \begin{pmatrix} 0000 \\ 0000 \\ 1010 \\ 1010 \end{pmatrix}.$$

Rows 3 and 4 satisfy case 2, while rows 1 and 2 are already reduced. We reduce rows 3 and 4 by applying $H_{[3,4]}T_{[3,4]}$. Continuing, the first column is completely reduced to e_1 by further applying $\omega_{[1]}^7 X_{[1,4]}H_{[3,4]}T_{[3,4]}^3$. The complete decomposition of u is therefore given by

$$W_1 = \omega_{[1]}^7 X_{[1,4]}H_{[3,4]}T_{[3,4]}^3H_{[3,4]}T_{[3,4]} \\ H_{[1,4]}T_{[1,4]}H_{[1,4]}T_{[1,4]}^2H_{[2,3]}T_{[2,3]}^3.$$

Applying this to the original matrix U , we have $W_1U =$

$$\frac{1}{\sqrt{2}^3} \begin{pmatrix} \sqrt{2}^3 & 0 & 0 & 0 \\ 0 & \omega^3 - \omega^2 + \omega + 1 & -\omega^2 - \omega - 1 & \omega^2 \\ 0 & 0 & \omega^3 + \omega^2 - \omega + 1 & \omega^3 + \omega^2 - \omega - 1 \\ 0 & \omega^3 + \omega^2 + \omega + 1 & \omega^2 & \omega^3 - \omega^2 + 1 \end{pmatrix}.$$

Continuing with the rest of the columns, we find $W_2 = \omega_{[2]}^6 H_{[2,4]}T_{[2,4]}^3H_{[2,4]}T_{[2,4]}$, $W_3 = \omega_{[3]}^4 H_{[3,4]}T_{[3,4]}^3H_{[3,4]}$, and $W_4 = \omega_{[4]}^5$. We then have $U = W_1^\dagger W_2^\dagger W_3^\dagger W_4^\dagger$, or explicitly:

$$U = T_{[2,3]}^5 H_{[2,3]}T_{[1,4]}^6 H_{[1,4]}T_{[1,4]}^7 H_{[1,4]} \\ T_{[3,4]}^7 H_{[3,4]}T_{[3,4]}^5 H_{[3,4]}X_{[1,4]}\omega_{[1]} \\ T_{[2,4]}^7 H_{[2,4]}T_{[2,4]}^5 H_{[2,4]}\omega_{[2]}^2 H_{[3,4]}T_{[3,4]}^5 H_{[3,4]}\omega_{[3]}^4 \omega_{[4]}^3.$$

6.5.2 Main result

Consider the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$, consisting of complex numbers of the form

$$\frac{1}{2^n}(a + bi + c\sqrt{2} + di\sqrt{2}),$$

where $n \in \mathbb{N}$ and $a, b, c, d \in \mathbb{Z}$. Our goal is to prove the following theorem, which was conjectured by Kliuchnikov et al. [19]:

Theorem 6.5.5. *Let U be a unitary $2^n \times 2^n$ matrix. Then the following are equivalent:*

- (a) *U can be exactly represented by a quantum circuit over the Clifford+ T gate set, possibly using some finite number of ancillas that are initialized and finalized in state $|0\rangle$.*
- (b) *The entries of U belong to the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$.*

Moreover, in (a), a single ancilla is always sufficient.

Proof. First note that, since all the elementary Clifford+ T gates, as shown in ([equation \(6.27\) on page 188](#)), take their matrix entries in $\mathbb{D}[\omega] = \mathbb{Z}[\frac{1}{\sqrt{2}}, i]$, the implication (a) \implies (b) is trivial. For the converse, let U be a unitary $2^n \times 2^n$ matrix with entries from $\mathbb{D}[\omega]$. By [Lemma 6.5.3 on page 193](#), U can be decomposed into one- and two-level matrices of types X , H , T , and ω . It is well-known that each such matrix can be further decomposed into controlled-not gates and multiply-controlled X , H , T , and ω -gates, for example using Gray codes [25, Sec. 4.5.2]. But all of these gates have well-known exact representations in Clifford+ T with ancillas, see e.g. [5, Fig. 4(a) and Fig. 9] (and noting that a controlled- ω gate is the same as a T -gate). This finishes the proof of (b) \implies (a).

The final claim that needs to be proved is that a circuit for U can always be found using at most one ancilla. It is already known that for $n > 1$, an ancilla is sometimes necessary [19]. To show that a single ancilla is sufficient, in light of the above decomposition, it is enough to show that the following can be implemented with one ancilla:

quantum computing architectures, ancillas are relatively cheap. Moreover, the use of additional ancillas can significantly reduce the size and depth of the generated circuits (see e.g. [33]).

6.5.3 The no-ancilla case

Lemma 6.5.7. *Under the hypotheses of Theorem 6.5.5 on page 195, assume that $\det U = 1$. Then U can be exactly represented by a Clifford+ T circuit with no ancillas.*

Proof. This requires only minor modifications to the proof of Theorem 6.5.5 on page 195. First observe that when an operator of the form HT^m was used in the proof of Lemma 6.5.1 on page 190, we can instead use $T^{-m}(iH)T^m$ without altering the rest of the argument. In the base case of Lemma 6.5.2 on page 192, the operator $X_{[1,j]}$ can be replaced by $iX_{[1,j]}$. Also, in the base case of Lemma 6.5.2 on page 192, whenever $n \geq 2$, the operator $\omega_{[1]}$ can be replaced by $W_{[1,2]}$, where

$$W = \begin{pmatrix} \omega & 0 \\ 0 & \omega^{-1} \end{pmatrix}.$$

Therefore, the decomposition of Lemma 6.5.3 on page 193 can be performed so as to yield only two-level matrices of types

$$iX, \quad T^{-m}(iH)T^m, \quad \text{and } W, \tag{6.29}$$

plus at most one one-level matrix of type ω^m . But since all two-level matrices of types (equation (6.29)), as well as U itself, have determinant 1, it follows that $\omega^m = 1$. We finish the proof by observing that the multiply-controlled operators of types (equation (6.29)) possess ancilla-free Clifford+ T representations, with the latter two given by

$$\begin{array}{c} \begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} = \begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} \\ \boxed{T^{-m}(iH)T^m} \quad \boxed{T^{-m}} \boxed{S} \boxed{H} \boxed{T} \boxed{iX} \boxed{T^\dagger} \boxed{H} \boxed{S^\dagger} \boxed{T^{-m}} \end{array}$$

$$\begin{array}{c} \begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} = \begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} \\ \boxed{W} \quad \boxed{iX} \boxed{T} \boxed{iX} \boxed{T^\dagger} \end{array}$$

□

As a corollary, we obtain a characterization of the n -qubit Clifford+ T group (with no ancillas) for all n :

Corollary 6.5.8. *Let U be a unitary $2^n \times 2^n$ matrix. Then the following are equivalent:*

(a) *U can be exactly represented by a quantum circuit over the Clifford+ T gate set on n qubits with no ancillas.*

(b) *The entries of U belong to the ring $\mathbb{Z}[\frac{1}{\sqrt{2}}, i]$, and:*

- $\det U = 1$, if $n \geq 4$;
- $\det U \in \{-1, 1\}$, if $n = 3$;
- $\det U \in \{i, -1, -i, 1\}$, if $n = 2$;
- $\det U \in \{\omega, i, \omega^3, -1, \omega^5, -i, \omega^7, 1\}$, if $n \leq 1$.

Proof. For (a) \implies (b), it suffices to note that each of the generators of the Clifford+ T group, regarded as an operation on n qubits, satisfies the conditions in (b). For (b) \implies (a), let us define for convenience $d_0 = d_1 = \omega$, $d_2 = i$, $d_3 = -1$, and $d_n = 1$ for $n \geq 4$. First note that for all n , the Clifford+ T group on n qubits (without ancillas) contains an element D_n whose determinant is d_n , namely $D_n = I$ for $n \geq 4$, $D_3 = T \otimes I \otimes I$, $D_2 = T \otimes I$, $D_1 = T$, and $D_0 = \omega$. Now consider some U satisfying (b). By assumption, $\det U = d_n^m$ for some m . Let $U' = U D_n^{-m}$, then $\det U' = 1$. By Lemma 6.5.7 on the previous page, U' , and therefore U , is in the Clifford+ T group with no ancillas. \square

Remark 6.5.9. Note that the last condition in Corollary 6.5.8, namely that $\det U$ is a power of ω for $n \leq 1$, is of course redundant, as this already follows from $\det U \in \mathbb{Z}[\frac{1}{\sqrt{2}}, i]$ and $|\det U| = 1$. We stated the condition for consistency with the case $n \geq 2$.

Remark 6.5.10. The situation of Theorem 6.5.5 on page 195 and Corollary 6.5.8 is analogous to the case of classical reversible circuits. It is well-known that the not-gate, controlled-not gate, and Toffoli gate generate all classical reversible functions on $n \leq 3$ bits. For $n \geq 4$

bits, they generate exactly those reversible boolean functions that define an *even permutation* of their inputs (or equivalently, those that have determinant 1 when viewed in matrix form) [24]; the addition of a single ancilla suffices to recover all boolean functions.

6.5.4 Complexity

The proof of Theorem 6.5.5 on page 195 immediately yields an algorithm, albeit not a very efficient one, for synthesizing a Clifford+ T circuit with ancillas from a given operator U . We estimate the size of the generated circuits.

We first estimate the number of (one- and two-level) operations generated by the matrix decomposition of Lemma 6.5.3 on page 193. The row operation from Lemma 6.5.1 on page 190 requires only a constant number of operations. Reducing a single n -dimensional column from denominator exponent k to $k - 1$, as in the induction step of Lemma 6.5.2 on page 192, requires $O(n)$ operations; therefore, the number of operations required to reduce the column completely is $O(nk)$.

Now consider applying Lemma 6.5.3 to an $n \times n$ -matrix with least denominator exponent k . Reducing the first column requires $O(nk)$ operations, but unfortunately, it may *increase* the least denominator exponent of the rest of the matrix, in the worst case, to $3k$. Namely, each row operation of Lemma 6.5.1 potentially increases the denominator exponent by 2, and any given row may be subject to up to k row operations, resulting in a worst-case increase of its denominator exponent from k to $3k$ during the reduction of the first column. It follows that reducing the second column requires up to $O(3(n - 1)k)$ operations, reducing the third column requires up to $O(9(n - 2)k)$ operations, and so on. Using the identity $\sum_{j=0}^{n-1} 3^j(n - j) = (3^{n+1} - 2n - 3)/4$, this results in a total of $O(3^n k)$ one- and two-level operations for Lemma 6.5.3.

In the context of Theorem 6.5.5 on page 195, we are dealing with n qubits, i.e., a $2^n \times 2^n$ -operator, which therefore decomposes into $O(3^{2^n} k)$ two-level operations. Using one ancilla, each two-level operation can be decomposed into $O(n)$ Clifford+ T gates, resulting in a total

gate count of $O(3^{2^n}nk)$ elementary Clifford+ T gates.

Chapter 7

Conclusions and future work

Bibliography

- [1] S. Abramsky. A structural approach to reversible computation. *Theoretical Computer Science*, 347(3):441–464, 2005.
- [2] S. Abramsky and B. Coecke. Physical traces: Quantum vs. classical information processing. *Electr. Notes Theor. Comput. Sci*, 69, 2002.
- [3] S. Abramsky and B. Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LiCS'04)*, IEEE Computer Science Press. (extended version at *arXiv:quant-ph/0402130*), pages 415–425, 2004.
- [4] S. Abramsky and B. Coecke. Abstract physical traces. *Theory and Applications of Categories*, 14:114–124, 2005.
- [5] Matthew Amy, Dmitri Maslov, Michele Mosca, and Martin Roetteler. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. Version 2, *arXiv:1206.0758v2*, August 2012.
- [6] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52:3457–3467, 1995. Available from *arXiv:quant-ph/9503016v1*.
- [7] Erik Barendsen, Inge Bethke, Jan Heering, Richard Kennaway, Paul Klint, Vincent van Oostrom, Femke van Raamsdonk, Fer-Jan de Vries, and Hans Zantema. Cambridge University Press, The Edinburgh Building, Cambridge, CB2 2RU, UK, 2003.
- [8] Michael Barr and Charles Wells. *Category Theory for Computing Science*. Prentice Hall, 2nd edition, 1995.
- [9] Charles H. Bennet. Logical reversibility of computation. *IBM Journal of Research and Development*, 6:525–532, 1973.
- [10] Alex Bocharov and Krysta M. Svore. Resource-optimal single-qubit quantum circuits. *Physical Review Letters*, 109:190501 (5 pages), 2012. Also available from *arXiv:1206.3223*.
- [11] J.R.B. Cockett and Brett G. Giles. Discrete inverse categories. In preparation, May 2010.
- [12] J.R.B. Cockett, Xiuzhan Guo, and Pieter Hofstra. Range categories ii: Towards regularity. Submitted for Publication, June 2012.
- [13] Bob Coecke, Eric O. Paquette, and Duško Pavlović. Classical and quantum structures. Technical Report RR-08-02, Oxford University Computing Laboratory, 2008.

- [14] Bob Coecke, Duško Pavlović, and Jamie Vicary. A new description of orthogonal bases. *Math. Structures in Comp. Sci.*, page 13, 2008. 13pp, to appear, arxiv.org/abs/0810.0812.
- [15] David Deutsch. Quantum computational networks. *Proceedings of the Royal Society of London Ser. A*, A425:73–90, 1989.
- [16] Xiuzhan Guo. *Products, Joins, Meets, and Ranges in Restriction Categories*. PhD thesis, University of Calgary, April 2012.
- [17] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to automata theory, languages, and computation*. Pearson/Addison Wesley, 3rd edition, 2007.
- [18] Landy Huet and Peter Selinger. Semantics of covariant quantum data types. Unpublished research internship report, Ecole Polytechnique, April 2007.
- [19] Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca. Fast and efficient exact synthesis of single qubit unitaries generated by Clifford and T gates. arXiv:1206.5236v2, June 2012.
- [20] Joachim Kock. *Frobenius Algebras and 2D Topological Quantum Field Theories*. Number 59 in London Mathematical Society Student Texts. Cambridge University Press, 2004.
- [21] Dexter Kozen. Semantics of probabilistic programs. *Journal of Computer and System Sciences*, 22(3):328–350, 1981.
- [22] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer Verlag, Berlin, Heidelberg, Germany, second edition, 1997. ISBN 0-387-98403-8. Dewey QA169.M33 1998.
- [23] Ken Matsumoto and Kazuyuki Amano. Representation of quantum circuits with clifford and $\pi/8$ gates. arXiv:0806.3834v1, June 2008.
- [24] Julien Musset. Générateurs et relations pour les circuits booléens réversibles. Technical Report 97-32, Institut de Mathématiques de Luminy, 1997. Available from <http://iml.univ-mrs.fr/editions/>.
- [25] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK, 2000. ISBN 0 521 63235 8.
- [26] Robin Cockett. Category theory for computer science. Available at <http://pages.cpsc.ualgary.ca/~robin/class/617/notes.pdf>, October 2009.
- [27] Robin Cockett and Stephen Lack. Restriction categories I: categories of partial maps. *Theoretical Computer Science*, 270:223–259, 2002.
- [28] Robin Cockett and Stephen Lack. Restriction categories II: Partial map classification. *Theoretical Computer Science*, 294:61–102, 2003.

- [29] Robin Cockett and Stephen Lack. Restriction categories III: colimits, partial limits, and extensivity. *Mathematical Structures in Computer Science*, 17(4):775–817, 2007. Available at <http://au.arxiv.org/abs/math/0610500v1>.
- [30] Peter Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14(4):527–586, 2004.
- [31] Peter Selinger. Towards a semantics for higher-order quantum computation. In *Proceedings of the 2rd International Workshop on Quantum Programming Languages, Turku, Finland*, pages 127–143. TUCS General Publication No. 33, June 2004.
- [32] Peter Selinger. Dagger compact closed categories and completely positive maps. In *Proceedings of the 3rd International Workshop on Quantum Programming Languages, Chicago*, 2005.
- [33] Peter Selinger. Quantum circuits of T -depth one. *Physical Review A*, 2013. To appear. Available from [arXiv:1210.0974](https://arxiv.org/abs/1210.0974).
- [34] Benoit Valiron. *Semantics for a Higher Order Functional Programming Language for Quantum Computation*. PhD thesis, University of Ottawa, 2008.
- [35] André van Tonder and Miquel Dorca. Quantum computation, categorical semantics and linear logic. *CoRR*, quant-ph/0312174v4:1–22, 2007.