Models of Multipartite Entanglement

Thesis submitted for the degree of
Master of Science

Michael Herrmann

September 2010

Abstract

While multipartite entanglement is a crucial resource for many quantum algorithms and protocols, the abstract, structural rules governing it are still not fully understood. In their recent paper [14], Coecke and Kissinger propose a new language called *GHZ/W-calculus* that aims to give a structural axiomatization of multipartite entanglement in terms of interacting "special" and "anti-special" commutative Frobenius algebras (S/A-CFAs). These algebraic structures correspond precisely to the two different types of genuine entanglement between three qubits, as given by inter-convertability by stochastic local (quantum) operations and classical communication (SLOCC). An important feature of the language is that it lives in the framework of categorical quantum mechanics and thus comes with a Penrose-Joyal-Street graphical calulus.

This thesis explores the expressive power of the GHZ/W-calculus. We prove a new result that demonstrates the canonicity of the recent concept of anti-speciality for CFAs and use it to give a new classification of Frobenius Algebras on $\mathbb{C}^2$ in **FdHilb**, the category of finite-dimensional Hilbert spaces and bounded linear maps. Next, we look at the non-standard model of categorical quantum mechanics given by **FRel**, the category of finite sets and binary relations. We classify all special CFAs in this category, prove some properties of anti-special ones and identify precisely which SCFAs/ACFAs are captured by the GHZ/W-language. Finally, we define a quantum AND gate in the graphical language and prove (or disprove) some of its properties. We come to the conclusion that the GHZ/W-calculus is unlikely to be the final answer to the structural entanglement problem in dimensions $D > 2$ and identify intermediate ranks $1 < rank < D$ as the notion it fails to explain.

Acknowledgements

I would particularly like to thank my supervisor, Dr. Bob Coecke, for giving me the oportunity to take part in his research and treating me like an equal member of his group. He made writing this dissertation a very rewarding experience and it was moreover nice to see him around the Wolfson College bar for the odd drink (or two). I am also indebted to Aleks Kissinger for many explanations and giving me the LaTeX code of his transfer report [29] that was used to typeset many of the diagrams in this thesis. Finally, I would like to thank Jacob Biamonte for his feedback on parts of a first draft of this report.

This dissertation marks the end of my taught studies in the UK. I am extremely grateful to my parents, Elisabeth and Kurt, and my brother Andreas for their continuous support throughout the last four years. I had a great time here, but now I look forward to coming back home.

Contents

Chapter 1

Introduction

In quantum informatics, computations are performed using qubits instead of bits as in classical computation. An important distinction between the two is that bits have only two possible values, 0 and 1, while each qubit can be in an arbitrary (complex) superposition $\alpha|0\rangle + \beta|1\rangle$ of the two ground states $|0\rangle$ and $|1\rangle$. Mathematically, these states are described as vectors in the complex Hilbert space $\mathbb{C}^2$, while state transitions are expressed as particular kinds of linear maps. Composition of systems is given by the linear tensor product $\otimes$ and composition of operators is normal function composition $\circ$. The formalism that prescribes these rules is called the *Hilbert space formalism* of quantum mechanics and was proposed by von Neumann in 1932 [42].

While the Hilbert space formalism has proven extremely successful in predicting the outcomes of experiments, using it to prove equational statements is often rather involved. Consider for example the derivation shown in Figure 1.1. At first glance, all we can make out from the series of equations is that the

$$\begin{aligned}
&\mathrm{Id} \otimes (|00\rangle + |11\rangle)(\langle 00| + \langle 11|) \otimes \mathrm{Id} \left[\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\right] &(1)\\
&= \frac{1}{2}(\mathrm{Id} \otimes (|00\rangle + |11\rangle)(\langle 00| + \langle 11|) \otimes \mathrm{Id})\left[(|00\rangle + |11\rangle) \otimes (|00\rangle + |11\rangle)\right]\\
&= \frac{1}{2}(\mathrm{Id} \otimes (|00\rangle + |11\rangle)(\langle 00| + \langle 11|) \otimes \mathrm{Id})\left[|0000\rangle + |0011\rangle + |1100\rangle + |1111\rangle\right]\\
&= \frac{1}{2}\left(|0\rangle(|00\rangle + |11\rangle)|0\rangle + |1\rangle(|00\rangle + |11\rangle)|1\rangle\right)\\
&= \frac{1}{\sqrt{2}}|0\rangle\left(\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\right)|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\left(\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\right)|1\rangle. &(2)
\end{aligned}$$

Figure 1.1: It doesn't have to be like that.

application of an operator to a state as in Equation (1) yields a new state given by Equation (2). We do not gain any structural insight into what is going on.

Let us rewrite Equation (1) graphically, as follows: We represent the entangled state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ by a cap,

 $:= \dfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$

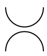
The two lines coming out of the cap stand for the two qubits that make up the state. We write tensor products by placing the corresponding graphical entities horizontally next to each other (juxtaposition). Thus for instance

$\dfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \otimes \dfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle) =$ .

Applications and compositions of linear operators are written vertically, from top to bottom, in the order in which they are applied. If, in the spirit of Dirac notation, we interpret kets $|\ .\ \rangle$ as linear maps from $\mathbb{C}$ to $\mathbb{C}^2$ and bras $\langle\ .\ |$ as maps from $\mathbb{C}^2$ to $\mathbb{C}$ and set

 $:= \sqrt{2}(\langle 00| + \langle 11|),$

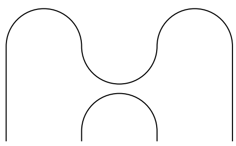
then the graphical rule for composition yields

 $= (|00\rangle + |11\rangle)(\langle 00| + \langle 11|).$

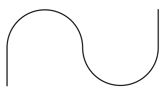
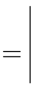
As a final step, we set

 $:= \mathrm{Id}.$

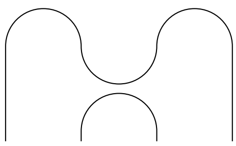
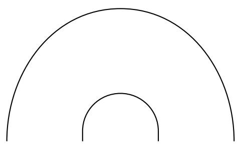
Using the graphical notation we have just introduced, Equation (1) can be rewritten as

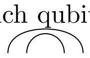
.

A simple calculation now shows that

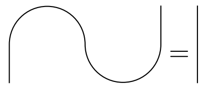
 $= \Big(\mathrm{Id} \otimes \sqrt{2}(\langle 00| + \langle 11|)\Big) \circ \Big(\dfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \otimes \mathrm{Id}\Big) = \mathrm{Id} =$ .

By a few more technical properties of the graphical calculus that essentially allow us to "stretch" wires, we then have

 $=$ .

If we recall that  represents an entangled state, then it is now easy to see that the application of the operator in Equation (1) transformed the system from the state , in which qubits one and two as well as three and four were entangled, into the state , in which qubits one and four as well as two and three are entangled. This is one of four cases of the entanglement swapping protocol [26].

The simple graphical calculus we have just seen has been generalised and given a rigorous mathematical foundation as part of the development of the field of *categorical quantum mechanics* [2, 18, 13, 11]. Using category theory, this area of active research abstracts away from the Hilbert space formalism and identifies the key structural principles that underly quantum mechanics. One of the most important of these principles is the identity



from above.

The graphical *Z/X-calculus* arose from categorical quantum mechanics and gives an abstract axiomatisation of (the interaction of) complementary quantum observables [11]. With a certain continuous extension that allows for arbitrary phases, it is universal for quantum computation. The Z/X-calculus can be seen as one of the cornerstones of the development of categorical quantum mechanics, however there are things it does not explain.

One of the long-standing open problems in the field of quantum information science is providing a generic description of entanglement in N-qubit systems [29, 15]. A common classification of entanglement in this field is given by considering states up to inter-convertability by stochastic local operations and classical communication (SLOCC) [15, 14, 24, 33, 34]. While there is only one SLOCC-class of genuinely entangled two-qubit systems, there are already two for the tripartite case [24], witnessed respectively by the GHZ- and W-states

$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle), \qquad |W\rangle = \frac{1}{\sqrt{3}}(|001\rangle + |010\rangle + |100\rangle).$$

A limitation of the Z/X-language is that it cannot easily express the W-state. As a consequence, the Z/X-calculus often fails to intuitively explain the interactions between the GHZ- and the W-class and generally between different types of entanglement.

In the recent paper [15], Coecke and Kissinger propose a new graphical language for categorical quantum mechanics called *GHZ/W-calculus*. This calculus tries to overcome the limitations of the Z/X-calculus by giving an axiomatisation of the interactions between the GHZ- and the W-class. Technically, the states in the GHZ -class are identified in the categorical framework as special commutative Frobenius algebras (SCFAs) while members of the W-class are identified as anti-special ones (ACFAs). For the model of qubits in the Hilbert space formalism, the authors prove that there is a bijective correspondence between SCFAs and ACFAs and that the GHZ/W-calculus refines the Z/X-calculus.

One of the advantages of the categorical approach to quantum mechanics is that it allows one to explore quantum mechanical structures in settings other than Hilbert spaces. Two important such non-standard models are Spekkens' toy quantum theory [41] and the category **FRel** of finite sets and binary relations. By previous work of several authors [13, 39, 25], the meaning of the Z/X-calculus in these models is well understood. The GHZ/W-calculus on the other hand has not yet been studied in these contexts.

This thesis evaluates the expressive power of the recent GHZ/W-calculus, with a particular emphasis on the non-standard model **FRel**. Its main contributions can be summarized as follows:

- We prove a new result that, in a highly abstract manner, demonstrates the canonicity of the recent concept of anti-speciality for Frobenius algebras (Chapter 3).

- We use the result just mentioned to give a new classification of Frobenius algebras on $\mathbb{C}^2$ in the standard model of categorical quantum mechanics given by the category **FdHilb** of finite dimensional Hilbert spaces and bounded linear maps (Chapter 3). We argue that this classification can to some extent be seen as giving a new, abstract explanation of the existence of two classes of genuine tripartite entanglement for qubits.

- We give a Haskell implementation that can produce all commutative Frobenius algebras on the two- and three-element sets in **FRel** and use it to show that analogues of several results known to hold for the GHZ/W-calculus on $\mathbb{C}^2$ in **FdHilb** do not hold in **FRel** (Section 4.2).

- We strengthen a result by Pavlovic [39] to give a new characterisation of all SCFAs in **Rel**, the category of arbitrary sets and binary relations (Section 4.3).

- We prove several properties of ACFAs in **Rel** (Section 4.4) and use them together with the new classification of SCFAs to identify precisely which SCFAs/ACFAs are captured by the GHZ/W-language in this category (Section 4.5).

- We use the GHZ/W-calculus to define a quantum analogue of the Boolean AND gate and identify several additional graphical axioms that can be used to prove some of its properties (Chapter 5).

Chapter 2

Background

2.1 Assumed knowledge

We will assume that the reader is familiar with basic notions of category theory and quantum computer science. For the former, a good general introduction is [3], while a treatise that is more directed towards our needs is [9]. For the latter, accessible first introductions are [36, 30]; a more comprehensive standard reference is [37].

2.2 LOCC and SLOCC-equivalence

Entanglement is one of the most powerful and yet one of the least understood features of quantum mechanics. This section lists some of its known structural properties; a slightly more detailed overview is for instance given in [29].

In quantum information theory, it is often useful to regard quantum states to be equivalent as computational resources iff they are *LOCC-equivalent*

Definition 1. Two states $|\psi\rangle$ and $|\phi\rangle$ of an n-qubit system are *LOCC-equivalent* iff they deterministically inter-converted with only local (i.e. one-qubit) physical operations and classical communication.

LOCC-equivalence has a clear mathematical meaning in terms of local unitary maps:

Theorem 2 ([4])**.** *Two states $|\psi\rangle$ and $|\phi\rangle$ of an n-qubit system are LOCC-equivalent iff there exist unitary transformations $U_1, \ldots, U_n : \mathbb{C}^2 \to \mathbb{C}^2$ such that*

$$|\psi\rangle = (U_1 \otimes \cdots \otimes U_n)|\phi\rangle.$$

A natural generalisation of LOCC-equivalence is that of SLOCC-equivalence:

Definition 3. Two states $|\psi\rangle$ and $|\phi\rangle$ of an n-qubit system are *SLOCC-equivalent* iff they can be made LOCC-equivalent with some non-zero probability.

This yields an even nicer mathematical interpretation:

Theorem 4 ([24]). *Two states* $|\psi\rangle$ *and* $|\phi\rangle$ *of an* n*-qubit system are* SLOCC-equivalent *iff there exist invertible operators* $L_1, \ldots, L_n : \mathbb{C}^2 \to \mathbb{C}^2$ *such that*

$$|\psi\rangle = (L_1 \otimes \cdots \otimes L_n)|\phi\rangle.$$

On $N = 2$ qubits, there are only two SLOCC classes, namely product states, which are SLOCC-equivalent to $|00\rangle$ and entangled states, which are SLOCC-equivalent to the Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

2.2.1 GHZ- and W-states

Recall the following key definition:

Definition 5. Let $\mathcal{H}_1, \mathcal{H}_\in$ be Hilbert spaces. A state $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ is called *(genuinely) entangled* iff it cannot be written as a product state, i.e. there does not exist $|\phi_1\rangle \in \mathcal{H}_1, |\phi_2\rangle \in \mathcal{H}_2$ such that

$$|\psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle.$$

An important point to make is that we do not call a state entangled when just one of its subparts is entangled – e.g. $|0\rangle \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is not entangled even though $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is.

Up to SLOCC, there are only two genuinely entangled tripartite states [24]: The *Greenberger-Horne-Zeilinger (GHZ) state*

$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

and the *W-state*

$$|W\rangle = \frac{1}{\sqrt{3}}(|001\rangle + |010\rangle + |100\rangle).$$

The GHZ and W states represent qualitatively very different kinds of entanglement. The GHZ state can (up to SLOCC) be seen as the state that maximises true three party entanglement, as measured by a metric called the three tangle [20]. However, when we trace out one of its qubits, we get an unentangled state. The W state is the exact opposite: It minimises the three tangle (up to SLOCC) amongst entangled tripartite states, but retains maximally bipartite entangled under tracing-out one qubit [24]. Intuitively, this is because all of the entanglement present in the W state is due to pairwise correlations between the three qubits while the entanglement of the GHZ state is solely due to correlations involving all three qubits [29]. Another interesting distinction between the two states is that certain classical computational problems are only solvable with one but not the other [21].

2.2.2 SLOCC super-classes

While there are only finitely many SLOCC classes for $N = 3$ qubits, there are necessarily infinitely many for the cases where $N \geq 4$ [24]. To nevertheless be able to obtain finitary classification results, Lamata et al. introduced so-called SLOCC *super-classes* [33, 34]. This inductive scheme regards an N-partite state as a map M from $\otimes^{N-1}\mathbb{C}^2$ to $\mathbb{C}^2$. Performing a singular value decomposition on M yields a one- or two-dimensional subspace spanned by two vectors $|\phi\rangle$ and $|\psi\rangle$ in $\otimes^{N-1}\mathbb{C}^2$. The SLOCC-superclass of M is then described by the SLOCC super-classes of $|\phi\rangle$ and $|\psi\rangle$. The base case of this scheme is $\mathbb{C}^2 \otimes \mathbb{C}^2$, where a state is regarded as either a product state or the Bell state [15].

2.3 Categorical quantum mechanics

The field of categorical quantum mechanics was initiated by Abramsky and Coecke in [2]. It uses the fact that a lot of the structure of the von Neumann formalism can be recast at an abstract level. Good first tutorials that assume little or no prior knowledge of category theory are [8, 9, 10]; more advanced cornerstones of the field are [11, 15].

2.3.1 Vectors and scalars as morphisms

The formal realm in which most of categorical quantum mechanics takes place is that of *symmetric monoidal categories (SMCs)*:

Definition 6. A *monoidal category* $(\mathbf{C}, \otimes, I)$ is a category which comes with a bifunctor

$$- \otimes - : \mathbf{C} \times \mathbf{C} \to \mathbf{C},$$

a unit object I, natural left and right unit isomorphisms

$$\lambda_A : A \simeq I \otimes A \quad \text{and} \quad \rho_A : A \simeq A \otimes I,$$

and a natural associativity isomorphism

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

which are subject to certain coherence conditions spelled out for instance in [35]. A monoidal category is called *symmetric* if it moreover comes with a natural symmetry isomorphism

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

again subject to certain coherence conditions [35]. The four morphisms are called the *structure maps* of the SMC.

Symmetric monoidal categories are a very natural setting for describing physical systems [9]: Consider a system of type A (e.g. a billiard ball, or a qubit, or two qubits). We can perform an operation f on it (e.g. change the momentum of the tennis ball or discard one of the two qubits), obtaining a system of possibly different type B:

$$A \xrightarrow{f} B$$

Given an operation of the form $B \xrightarrow{g} C$, we can first do f and then g, obtaining

$$A \xrightarrow{g \circ f} C\ .$$

Clearly, we have $h \circ (g \circ f) = (h \circ g) \circ f$ since the brackets merely indicate which of the operations we conceive as one. We also have identity operations

$$A \xrightarrow{1_A} A$$

that represent “doing nothing” to a system and hence satisfy

$$1_B \circ f = f = f \circ 1_A.$$

These observations yield a category $\mathcal{C}$. If we then want to be able to conceive compound systems (e.g. two billiard balls), we need the product $A \otimes B$ and compound operations

$$A \otimes B \xrightarrow{f \otimes g} C \otimes D\ .$$

In the usual case where the order in which physical systems are composed does not matter, we also have $A \otimes B \simeq B \otimes A$. This puts us in the setting of symmetric monoidal categories.

The prototypical category in categorical quantum mechanics is **FdHilb**, the category of finite dimensional Hilbert spaces and bounded linear maps. If we take $\otimes$ to be the tensor product and the tensor unit I to be $\mathbb{C}$, then it is not difficult to see that $(\mathbf{FdHilb}, \otimes, \mathbb{C})$ is a symmetric monoidal category.

How would we define vectors and scalars in the language of symmetric monoidal categories? When we think about **FdHilb**, we can identify each vector $|\phi\rangle$ in a Hilbert space $\mathcal{H}$ with the linear map

$$\begin{aligned} f : \mathbb{C} &\longrightarrow \mathcal{H} \\ \lambda &\longmapsto \lambda|\phi\rangle. \end{aligned}$$

Since the scalars in **FdHilb** can be seen as vectors in the Hilbert space $\mathbb{C}$, we can use the same construction to represent scalars as linear maps of type $\mathbb{C} \to \mathbb{C}$. Generalizing these observations yields

Definition 7. Let $(\mathbf{C}, \otimes, I)$ be a symmetric monoidal category. The *vectors* or *states* of an object X of $\mathbf{C}$ are the morphisms of type $I \to X$. The *scalars* are the morphisms of type $I \to I$. For $c : I \to I$ a scalar and $f : A \to B$ a morphism, the *scalar multiplication* $c \cdot f : A \to B$ *of* f *with* c is defined as

$$c \cdot f := \lambda_B^{-1} \circ (c \otimes f) \circ \lambda_A.$$

In **FdHilb**, for example, $c \cdot f$ is the function that sends $|\phi\rangle \mapsto cf(|\phi\rangle)$.

Already at the abstract level of symmetric monoidal categories, it is possible to prove that scalars distribute through function composition and products:

$$(c \cdot f) \circ (d \cdot g) = (c \circ d) \cdot (f \circ g) \quad \text{and} \quad (c \cdot f) \otimes (d \cdot g) = (c \circ d) \cdot (f \otimes g). \quad (3)$$

Moreover, it can be shown that the scalars form a commutative monoid [28].

2.3.2 Adjoints

A piece of structure that plays an important role for quantum mechanics but is not captured by the SMC language is the existence of adjoints: Every linear map $f : \mathcal{H}_1 \to \mathcal{H}_2$ between Hilbert spaces has a unique adjoint $f^\dagger : \mathcal{H}_2 \to \mathcal{H}_1$. Technically, this makes **FdHilb** a †*-symmetric monoidal category* [11]:

Definition 8. A †*-symmetric monoidal category (†-SMC)* is a symmetric monoidal category equipped with an involutary identity-on-objects contravariant endofunctor

$$(-)^\dagger : \mathbf{C}^{op} \longrightarrow \mathbf{C}$$

which coherently preserves the monoidal structure, that is,

$$f^{\dagger\dagger} = f \qquad (f \circ g)^{\dagger} = g^{\dagger} \circ f^{\dagger} \qquad 1_A = 1_A^{\dagger} \qquad (f \otimes g)^{\dagger} = f^{\dagger} \otimes g^{\dagger},$$

together with the fact that each natural isomorphism θ of the symmetric monoidal structure is *unitary*, that is, $\theta^{-1} = \theta^{\dagger}$.

2.3.3 Non-standard models

Our abstract approach makes it possible to reinterpret key notions of quantum mechanics such as scalars and vectors in settings other than the Hilbert space formalism. Technically, this is achieved by looking at †-SMCs other than **FdHilb**. The two most important examples of such †-SMCs for categorical quantum mechanics are Robert Spekkens' toy quantum theory [41, 13] and **FRel**, the category which has finite sets as objects and binary relations as morphisms.

FRel

In the SMC **FRel** of finite sets and binary relations, the bifunctor $- \otimes -$ is the cartesian product while the unit I is the one-element set $\{*\}$. For R a binary relation, $R^{\dagger}$ is the relational converse $\{(b, a) \mid (a, b) \in R\}$. It is not difficult to see that **FRel** is a sub-†-SMC of **Rel**, the category of sets and binary relations with the same symmetric monoidal structure. Many of the results proved in later chapters are proved in the generality of **Rel** and then follow for **FRel** by this observation.

The vectors of an object $X \in \mathbf{FRel}$ precisely correspond to its subsets (cf. Definition 7). Since scalars are the vectors of the tensor unit, it follows that there are only two: the singleton relation $\{(*, *)\}$, which we write as 1, and the empty relation $\{\}$, which we write as 0. This allows us to view each vector as a linear combination of the elements of X, with coefficient 1 ("in the subset") or 0 ("not in the subset").

Similarly to **FdHilb**, the Dirac notation can be used for **FRel**[1]: Let X be an object and $\psi \subseteq X$. We write

$$|\psi\rangle := \{*\} \times \psi \qquad \text{and} \qquad \langle\psi| := |\psi\rangle^{\dagger}.$$

Given another subset $\phi \subseteq X$, we define

$$\begin{aligned} |\psi\rangle + |\phi\rangle &:= |\psi\rangle \cup |\phi\rangle \\ |\psi\rangle\langle\phi| &:= |\psi\rangle \circ \langle\phi| = \psi \times \phi \\ \langle\psi|\phi\rangle &:= \langle\phi| \circ |\psi\rangle = \begin{cases} \{(*, *)\} = 1 & \text{if } \phi \cap \psi \neq \{\} \\ \{\} = 0 & \text{otherwise.} \end{cases} \end{aligned}$$

If we identify $\{i\}$ with i, then these rules suffice to represent any morphism $f \subseteq X \otimes Y$ as

$$f = \sum_{(i,j) \in f} |j\rangle\langle i|.$$

The induced calculus behaves exactly like the Dirac notation for Hilbert spaces, with the Boolean semiring $(\{0, 1\}, \vee, \wedge)$ instead of $(\mathbb{C}, +, \times)$ and the n-element set $\mathbf{n} := \{0, \ldots, n-1\}$ instead of $\mathbb{C}^n$ [13].

[1] For a similar argument involving matrices, see for instance [13].

The above observations show that **FRel** is similar in structure but considerably simpler than **FdHilb**. At the same time, it is powerful enough to simulate the quantum teleportation and dense coding protocols [13]. This is why it is the main model under study in this thesis.

Spek

Spekkens constructs his toy quantum theory [41] by analysis of the simple principle that, for a quantum system, the amount of information we have about the state of the system cannot exceed the amount we lack. In line with this emphasis on knowledge, the states in Spekkens' theory always express (necessarily incomplete) knowledge rather than physical reality. Spekkens' theory is finitary in that all its systems are built from a system with four possible states, however, it is powerful enough to simulate the teleportation and dense coding protocols. Spekkens uses this fact to argue that the main driving feature of his theory, viewing quantum states as states of incomplete knowledge, is more natural than viewing them as states of physical reality.

In their paper [13], Coecke and Edwards show that it is possible to give a quantum categorical model of Spekkens' toy theory called **Spek**, a sub-†-SMC of **FRel**. The construction reveals the natural necessity of relations rather than functions to model measurements in the theory, a case not considered in [41]. More importantly, however, the fact that categorical quantum mechanics has an independent quantum theory as a model substantiates its claim to structural generality.

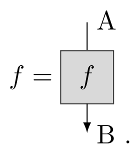
2.3.4 Graphical calculi

The interplay of the two key structural ingredients $\circ$ and $\otimes$ of SMCs makes the associated theory very two-dimensional: The tensor product acts like a spatial dimension while the composition of morphisms provides a causal, or temporal dimension [29]. This follows from the bifunctoriality of the tensor product – for any morphisms a, b, c and d of appropriate types,

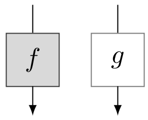
$$(c \otimes d) \circ (a \otimes b) = (c \circ a) \otimes (d \circ b). \tag{4}$$

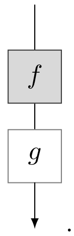
We will now see how the two-dimensional character can be exploited to define a graphical calculus.

We represent objects as edges and morphisms as nodes that connect edges, with the inputs at the top and the outputs at the bottom. For example, a morphism $f : A \to B$ is written as

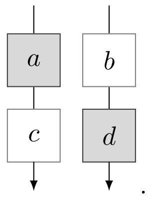


When the types of the edges are clear or not important, we omit the corresponding annotations. Tensor products are expressed by juxtaposition, function composition is obtained by connecting outputs and inputs of the corresponding graphs:

$f \otimes g =$ 

$g \circ f =$ 

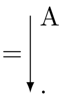
Note how Equation (4) is implicit in this graphical notation: Both of its sides are equal to



The ingredients of the monoidal structure are represented in particularly simple ways. By Mac Lane's strictification theorem [35, p. 257], we can assume without loss of generality that our SMC is *strict*, that is,

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C \quad \text{and} \quad A \otimes I = A = I \otimes A.$$

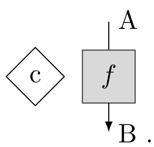
This implies that the structure maps λ_A, ρ_A and $\alpha_{A,B,C}$ become (tensor products of) identities, which are represented as edges:

$1_A =$ 

The only exception to this rule is when A is the tensor unit – the strict equality $A = A \otimes I$ makes it natural to represent I by an empty space instead of an edge,

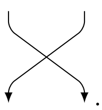
$$1_I = \quad .$$

Putting everything together, we for example have that the scalar multiplication $c \cdot f$ of $f : A \to B$ with $c : I \to I$ (cf. Definition 7) is

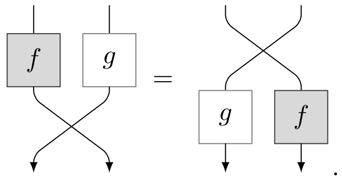
$\lambda_B^{-1} \circ (c \otimes f) \circ \lambda_A =$ 

Note how the graphical representation of c has no incoming or outgoing lines – this is because both are of type I.

The symmetry map σ is represented by a crossing of wires:

$\sigma =$ 

Its naturality for instance is captured by the identity



We end this section with a powerful completeness result for the above graphical language. It was first proved by Joyal and Street in [27, Thm 2.3] and given its succinct phrasing by Selinger in [40]:

Theorem 9 (Coherence for symmetric monoidal categories)**.** *A well-formed equation between morphisms in the language of symmetric monoidal categories follows from the axioms of symmetric monoidal categories if and only if it holds, up to isomorphism of diagrams, in the graphical language.*

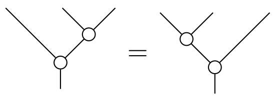
2.3.5 Observables and bases

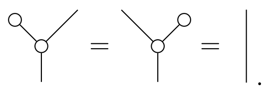
Just as scalars, vectors and adjoints can be axiomatized in the language of †-SMCs, so can quantum observables and bases. Following the presentation given in [11] to some extent, we will now see how.

Definition 10. An *internal monoid* in a monoidal category is a triple

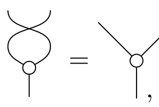
$$(A, \ : A \otimes A \to A, \ : I \to A)$$

for which the *multiplication* is *associative* and has as its *unit*, that is, respectively,





If we furthermore have



then $(A, \ , \ )$ is called *commutative*.

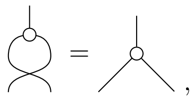
In the SMC **Set** of sets and functions with the cartesian product as tensor, internal (commutative) monoids precisely correspond to the usual notion of (commutative) monoids. The category-theoretic viewpoint however also lets us consider the dual notion, internal (*co*commutative) *comonoids*:

Definition 11. An *internal comonoid* in a monoidal category is a triple

$$(A, \ : A \to A \otimes A, \ : A \to I)$$

for which the *comultiplication* is *coassociative* and has as its *counit*, that is, respectively,

If we furthermore have



then $(A,$ , $)$ is called *cocommutative*.

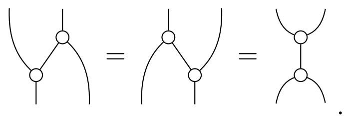
It is not difficult to see that, for each internal commutative monoid $(A,$ , $)$ in a †-SMC,

$\Big(A, ($  $)^\dagger, ($  $)^\dagger\Big)$

is an internal cocommutative comonoid and vice versa.

In [18], Coecke and Pavlovic showed that particular kinds of *commutative Frobenius algebras* (CFAs) can be used to represent the classical interfaces to the quantum universe given by the †-SMC:

Definition 12. A *Frobenius algebra* consists of an internal monoid $(A,$ , $)$ and an internal comonoid $(A,$ , $)$ on the same carrier such that the *Frobenius condition*

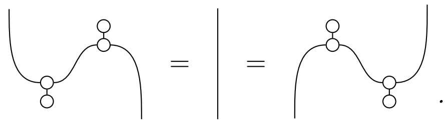


(5)

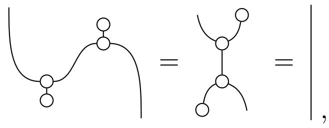
is satisfied. A Frobenius algebra is called *(co)commutative* iff its (co)monoid part is (co)commutative.

Historically, the Frobenius condition (5) first appeared in Carboni and Walters' paper [6]. This paper exhibits a relationship between commutative Frobenius algebras and *compact closed categories* which played an important role in the initial development of categorical quantum mechanics [2]. The crucial property for establishing the relationship is given by

Proposition 13 (Compactness [6])**.** *For any Frobenius algebra* $\big(A,$ , , , $\big)$,

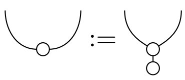
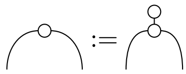


Proof. By the Frobenius condition and unitality,



and similarly for the vertical mirror image of these equations. □

Because they are so common and important, we write

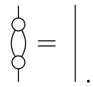
 and 

for the *cup* and *cap* of a Frobenius algebra, respectively.

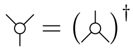
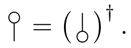
The following lemma allows us to drop the distinction between commutative and cocommutative Frobenius algebras in Definition 12:

Lemma 14 ([31, Lem 3.6.14])**.** *A Frobenius algebra in a symmetric monoidal category is commutative iff it is cocommutative.*

Definition 15. A Frobenius algebra is called *special* iff the *loop*



It is called *dagger* (written †) iff

 and 

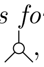
A commutative Frobenius algebra that satisfies both of these conditions is called an *observable* or *classical structure*[2].

Example 16. The notion of classical structures is intrinsic in a †-SMC: The unit object I always comes with observable structure

 $:= \lambda_I : I \simeq I \otimes I$ and  $:= 1_I$.

Theorem 17 ([18, 19])**.** *The classical structures in* ***FdHilb*** *precisely correspond to orthonormal bases* $\{|\psi_i\rangle\}_i$ *via the construction*

 $: |\psi_i\rangle \mapsto |\psi_i\rangle \otimes |\psi_i\rangle$ *and*  $: |\psi_i\rangle \mapsto 1.$ (6)

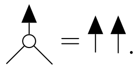
In other words, if $\{|\psi_i\rangle\}_i$ *is an orthonormal basis for a Hilbert space* $\mathcal{H}$ *then Equation (6) defines a classical structure and if* ($\mathcal{H}$, , ) *is a classical structure then it is of the form (6) for some orthonormal basis* $\{|\psi_i\rangle\}_i$ *of* $\mathcal{H}$*.*

Theorem 17 provides evidence for the above claim that commutative Frobenius algebras represent classical interfaces to the quantum universe: In quantum mechanics, each orthonormal basis gives rise to a measurable and performing the measurement yields a scalar that identifies one of the basis vectors. Orthogonal bases can thus be seen as the "classical data" that can be obtained about the quantum system. Theorem 17 tells us that this classical data corresponds to the copyable and deletable points of observable structures. This is in line with an important distinction between classical and quantum data in general, namely that the former can be copied and deleted while the latter, by the no-cloning and no-deleting theorems [43, 38], cannot. Further evidence in support of the description as classical interfaces is that (†-) commutative Frobenius algebras enable us to specify projector spectra and quantum measurements [18].

The notion of *copyable points* is easily captured in the †-SMC language:

[2] Classical structures were called *classical objects* in the original paper [18]. In recent work on the subject, the two names from Definition 15 appear to be more common.

Definition 18. We call a morphism $: I \to A$ a *copyable point* of a Frobenius algebra $\left(A, \;, \;, \;, \;\right)$ iff



Similarly to the correspondence between †-special commutative Frobenius algebras (i.e. observable structures) and orthonormal bases, more general types of CFAs correspond to more general types of bases. This is shown in Table 2.1. In each case, the basis is determined by the copyable points of the respective Frobenius algebra [19].

Type of basis	**Algebraic structure**
Arbitrary	Special commutative Frobenius algebra (SCFA)
Orthogonal	†-commutative Frobenius algebra (†-CFA)
Orthonormal	†-special commutative Frobenius algebra (†-SCFA)

Table 2.1: Correspondence bases – CFAs in **FdHilb** ([19])

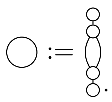
The following is a final very nice result that we will refer to later:

Lemma 19 ([7, Lem 2])**.** *In a symmetric monoidal category* $(\mathcal{C}, \otimes, I)$*, there is at most one special commutative Frobenius algebra* $\left(A, \;, \;, \;, \;\right)$ *for each cocommutative comultiplication* $: A \to A \otimes A$*.*

From the proof of Lemma 19 [7], it is not difficult to see that the corresponding statement also holds for commutative multiplications $: A \otimes A \to A$.

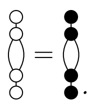
Dimension

A scalar that turns out to be of particular importance is the *dimension*



It satisfies

Lemma 20 ([11])**.** *Any two commutative Frobenius algebras* $\left(A, \;, \;, \;, \;\right)$ *and* $\left(A, \;, \;, \;, \;\right)$ *on the same carrier induce the same dimension, that is,*



The choice of the word "dimension" is of course not arbitrary:

Proposition 21. *In **FdHilb**, the circle induced by a CFA corresponds to the dimension of the underlying Hilbert space* $\mathcal{H}$*.*

Proof. Let $\mathcal{H}$ be a Hilbert space of finite dimension D. By Lemma 20, it suffices to find one CFA on $\mathcal{H}$ whose circle corresponds to D. Let $\{|\psi_i\rangle\}_{i=1}^{D}$ be an orthonormal basis for $\mathcal{H}$. By Theorem 17, this induces a †-SCFA

$$: |\psi_i\rangle \mapsto |\psi_i\rangle \otimes |\psi_i\rangle \qquad\qquad : |\psi_i\rangle \mapsto 1.$$

We get that

$$= \sum_{i=1}^{D} |\psi_i\rangle \otimes |\psi_i\rangle$$

and thus, by orthonormality

$$\bigcirc = \sum_{i=1}^{D} \langle\psi_i|\langle\psi_i|\psi_i\rangle|\psi_i\rangle = D.$$

$\square$

Observables in FRel

The following theorem due to Pavlovic classifies all observable structures in **Rel** (and thus in **FRel**):

Theorem 22 ([39])**.** *The observable structures in* **Rel** *are precisely*

$$= \bigcup_{j \in J} M_j \qquad\qquad = \{e_j\}_{j \in J}$$

where J is any set, $G_j = (X_j, *_j, e_j)$ are abelian groups with disjoint carriers and $M_j \subseteq (X_j \times X_j) \times X_j$ are the graphs of the respective group multiplications $*_j$.

Definition 23. For an observable structure $\circ = \left(A, \ , \ \right)$ in **Rel**, we call the set $\{G_j \mid j \in J\}$ from Theorem 22 the *group decomposition* or simply the *groups* of $\circ$ and write

$$\circ = \sum_{j \in J} G_j.$$

Example 24. By Theorem 22, there is only one observable structure on the one-element set in **Rel**, given by the singleton group $\mathbb{Z}_1$. For the two-element set $\{0, 1\}$, since 2 can be written as $1 + 1$ or as 2, we get two observable structures (up to swapping 0s and 1s): The first corresponds to the disjoint union $\mathbb{Z}_1 + \mathbb{Z}_1$,

$$_{1+1} = \{((0,0),0),((1,1),1)\} \qquad\qquad _{1+1} = \{(*,0),(*,1)\},$$

while the second corresponds to the two-element group $\mathbb{Z}_2$,

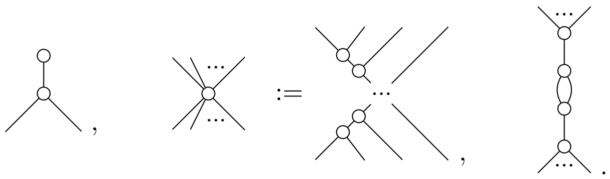
$$_{2} = \{((0,0),0),((0,1),1),((1,0),1),((1,1),0)\} \qquad\qquad _{2} = \{(*,0)\}.$$

We will encounter these observable structures again in the computational classification of SCFAs on the two-element set in **FRel** (Section 4.2).

Normal forms

Definition 25 ([29])**.** We say that a morphism f *is constructed from a Frobenius algebra* $(A, \ , \ , \ , \ )$ iff it can be written as a combination of the algebra maps , , , and the (symmetric) monoidal structure maps $\lambda, \rho, \sigma, \alpha$ only. We say such a morphism is *connected* iff its graphical representation is connected.

Example 26. The following are examples of connected morphisms constructed from a Frobenius algebra $(A, \ , \ , \ , \ )$:

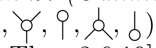


Connected morphisms constructed from a commutative Frobenius algebra have particularly nice normal forms, as shown by the following folk theorem (stated eg. in [29, 15]):

Theorem 27. *A connected morphism constructed from a commutative Frobenius algebra is completely determined by the numbers of its inputs, outputs and loops where*

- *"the numbers of inputs/outputs" are the numbers of incoming/outgoing wires in the graphical representation (so copies of the tensor unit I do not count) and*
- *"the number of loops" is the number of "closed compartments" in the graphical representation, or equivalently, the number of holes in the corresponding cobordism [31].*

The proof of Theorem 27 exploits the fact that the symmetric monoidal category generated by morphisms constructed from a CFA is canonically isomorphic to **2Cob**, the category of 1-dimensional manifolds and 2-dimensional cobordisms [31]:

Proof of Theorem 27 (Outline). Let f and g be connected morphisms constructed from a CFA (A, ), with the same numbers of inputs, outputs and loops. By [31, Thm 3.6.19], there is a unique symmetric monoidal functor $F : \mathbf{2Cob} \to \mathbf{C}$ such that $F\mathbf{1} = A$, where $\mathbf{1}$ is the unique closed connected 1-manifold. By the definition of F [31, Thm 3.3.2], each of the basic morphisms listed in Definition 25 has a singleton preimage under F, hence there exist unique cobordisms f' and g' such that

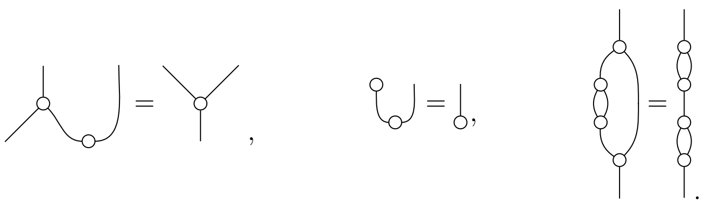
$$f = Ff' \quad \text{and} \quad g = Fg'.$$

Since f and g are connected and have the same number of loops, f' and g' must be connected and have the same genus. Similarly, because f and g have the same number of inputs and the same number of outputs, f' and g' must agree on the numbers of in-boundaries and out-boundaries. This implies by a fundamentel result of cobordisms (cf. [31, p. 64]) that $f' = g'$ and thus

$$f = Ff' = Fg' = g,$$

as required. □

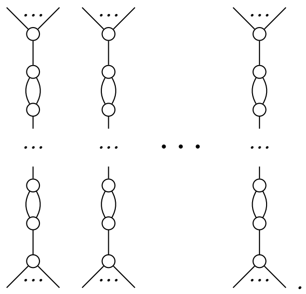
Example 28. By Theorem 27, we have



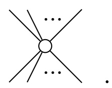
The interested reader should not find it too difficult to prove these equalities from first principles.

For non-connected morphisms constructed from a CFA, we have

Corollary 29. *Any morphism constructed from a commutative Frobenius algebra can be written as*

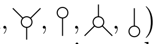


The restriction of Theorem 27 to (†-)SCFAs is sometimes called "spider theorem" in literature (eg. [11, 15, 17]) and has been proved for †-SCFAs in [16] and for arbitrary SCFAs in [32, Example 5.4]. The terminology stems from the fact that connected morphisms constructed from *special* commutative Frobenius algebras can be written as *spiders*



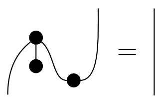
This is because the speciality condition allows us to remove all loops.

Frobenius states

The very recent paper [14] shows that some of the data $\big(A,$  $\big)$ is redundant for commutative Frobenius algebras. The authors give a crisper definition using the (new) notion of *Frobenius states*, a particular kind of symmetric states:

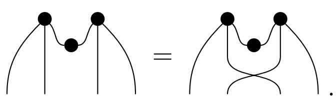
Definition 30. A state $\psi : I \to A \otimes \cdots \otimes A$ is called *symmetric* iff it remains constant under swapping of outcoming wires in its graphical representation.

Definition 31. A symmetric state  $: I \to A \otimes A \otimes A$ is said to be a *Frobenius state* iff there exist morphisms  $: A \otimes A \to I$ and  $: A \to I$ such that



(7)

and

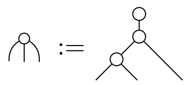


(8)

In **FdHilb**, Equation (7) already has an interesting connection with entanglement: Any symmetric tripartite state $: I \to A \otimes A \otimes A$ that satisfies it is SLOCC-maximal, in that any other state in the same SLOCC class can be obtained from by stochastic local operations and classical communication.

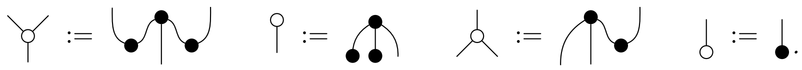
The following two theorems establish the correspondence between CFAs and Frobenius states:

Theorem 32. *For every commutative Frobenius algebra* $(A, , , ,)$, *the* induced tripartite state



forms a Frobenius state under the cup and the counit .

Theorem 33. *Any Frobenius state induces at least one commutative Frobenius algebra, given by*



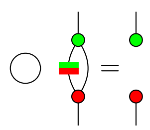
Furthermore, each cup of a Frobenius state uniquely determines a counit and vice versa.

Theorems 32 and 33 imply that every commutative Frobenius algebra is determined by its induced tripartite state and one of $\{ , \}$.

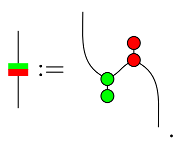
2.3.6 Complementarity – The Z/X-calculus

Given the axiomatisation of quantum observables as †-special commutative Frobenius algebras, we can ask how different quantum observables relate to each other. The most important such relationship for quantum mechanics is that of *complementarity*, which has been axiomatised by Coecke and Duncan in [11], as follows:

Definition 34. Two observable structures $(A, ,), (A, ,)$ are called *complementary* iff



where the *dualiser*



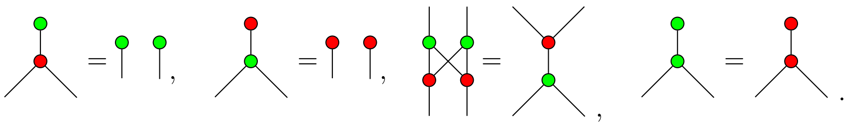
The standard example of complementary observables are the *Z*- and *X*-*observables* given by, respectively,

$$: \begin{cases} |0\rangle \mapsto |00\rangle \\ |1\rangle \mapsto |11\rangle \end{cases} \qquad\qquad : |0\rangle + |1\rangle \mapsto 1$$

and

 $: \begin{cases} |+\rangle \mapsto |++\rangle \\ |-\rangle \mapsto |--\rangle \end{cases}$  $: \sqrt{2}|0\rangle \mapsto 1.$

Direct calculation shows that these two observables moreover constitute a *scaled bialgebra with trivial antipode*, that is, they satisfy, up to scalars,



(9)

Because of their importance as a motivating example, the graphical calculus of complementary observables is called the *Z/X-calculus*.

Coecke and Duncan showed that, with an extension that allows for arbitrary phase shifts, the Z/X-calculus is universal for quantum computation on qubits [11]. Meanwhile, the Z/X-calculus has found applications in the foundations of quantum mechanics [12] and measurement-based quantum computation [23]. Moreover, there exists software called `quantomatic` which semi-automates reasoning within it [22].

Complementarity in FRel

The complementary observables in **Rel** were classified by Evans et al. in [25]. Interestingly, the group structure exhibited by Pavlovic's classification of single observables [39] (see also Section 2.3.5) does not play a role: The only thing that influences whether two observable structures are complementary are the partitions given by the respective biproducts of abelian groups:

Theorem 35 ([25])**.** *An observable structure $\sum_{j\in J} G_j$ in **FRel** has a complementary observable structure iff its partition is uniform, that is, all its constituent groups G_j have the same size. Further, two observable structures $\sum_{j\in J} G_j$ and $\sum_{k\in K} H_k$ are complementary iff the respective partitions are complementary, that is, $|G_j \cap H_k| = 1$ for all $j \in J, k \in K$.*

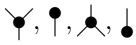
Example 36. The two observable structures $\mathbb{Z}_1 + \mathbb{Z}_1$ and $\mathbb{Z}_2$ from Example 24 above are complementary. More generally, the observable structures $\sum_{1\leq i\leq n} \mathbb{Z}_1$ and $\mathbb{Z}_n$ are complementary. The observable structure $\mathbb{Z}_1 + \mathbb{Z}_2$ does not have a complementary observable because its partition is not uniform.

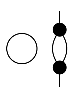
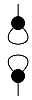
2.3.7 Multipartite entanglement

In their recent papers [29, 15, 14], Coecke and Kissinger show that special commutative Frobenius algebras precisely correspond to the members of the SLOCC equivalence class of the GHZ state:

Theorem 37. *In* **FdHilb**, *the induced tripartite state*  *of a special commutative Frobenius algebra on $\mathbb{C}^2$ is always SLOCC-equivalent to $|GHZ\rangle$. Conversely, any symmetric state that is SLOCC-equivalent to $|GHZ\rangle$ arises as the induced tripartite state of a SCFA on $\mathbb{C}^2$.*

If we recall that the Z/X-calculus only involves (interacting) †-SCFAs, then Theorem 37 suggests that it will not be able to give an abstract explanation of the interactions between the GHZ and W class, nor those between more general kinds of entanglement. For this reason, Coecke and Kissinger propose the new *GHZ/W-calculus* whose primitives are (arbitrary) SCFAs and so-called *anti-special* CFAs:

Definition 38. A Frobenius algebra (A, ) is called *anti-special* iff

 $=$  , (10)

where the *anti-unit* and *anti-counit* are, respectively,

 $:=$  and  $:=$  .

The new notion of anti-speciality yields the analogue of Theorem 37 for the SLOCC class of the W state:

Theorem 39. *The induced tripartite state*  *of an anti-special commutative Frobenius algebra on* $\mathbb{C}^2$ *in* **FdHilb** *is always SLOCC-equivalent to* $|W\rangle$*. Conversely, any symmetric state that is SLOCC-equivalent to* $|W\rangle$ *arises as the induced tripartite state of an ACFA.*

Example 40. The special and anti-special commutative Frobenius algebras that respectively give rise to (scalar multiples of) the GHZ and W states are

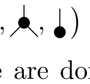
 $= |0\rangle\langle 00| + |1\rangle\langle 11|$  $= |0\rangle + |1\rangle$
 $= |00\rangle\langle 0| + |11\rangle\langle 1|$  $= \langle 1| + \langle 1|$ (11)

and

 $= |1\rangle\langle 11| + |0\rangle\langle 01| + |0\rangle\langle 10|$  $= |1\rangle$
 $= |00\rangle\langle 0| + |01\rangle\langle 1| + |10\rangle\langle 1|$  $= \langle 0|.$ (12)

Because of their importance as a motivating example, these two CFAs are called the *standard GHZ/W-pair.*

An interesting difference between SCFAs and ACFAs in **FdHilb** is that, while there is an abundance of †-SCFAs (one for each orthonormal basis), there are no non-trivial †-ACFAs:

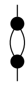
Proposition 41. [3] *If* $\bullet = \big(A,$ $\big)$ *is a †-ACFA in* **FdHilb** *then* $\dim A \leq 1$.

Proof. If $\dim A = 0$, then we are done. Suppose $\dim A > 0$. Since $\bullet$ is a †-CFA, Corollary 4.5 of [19] (see also Table 2.1 above) implies that there is an orthogonal basis $\{\psi_i\}_i$ for A such that

 $$= \sum_i |\psi_i\psi_i\rangle\langle\psi_i|.$$

[3]This fact was first observed by Bob Coecke and Aleks Kissinger and has been transmitted by personal communication.

Since ● is dagger, we then have that



$$= \sum_i ||\psi_i||^4 |\psi_i\rangle\langle\psi_i| \tag{13}$$

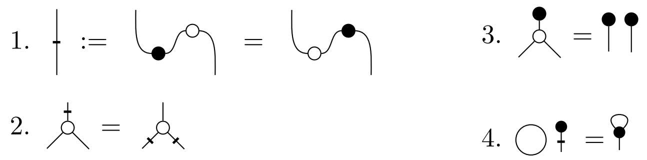
has (full) rank $\dim A$. But now, recalling that in **FdHilb** the circle ○ corresponds to $\dim A$ and is thus > 0, the anti-speciality condition (10) implies that the rank of the loop (13) is less than or equal to 1. Hence, as required,

$$\dim A \leq 1.$$

□

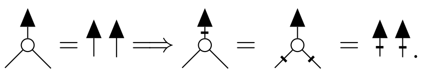
Similarly to how the Z/X-calculus axiomatises the interactions between complementary observable structures, Coecke and Kissinger propose the following axiomatisation of the relationship between the GHZ- and the W-class:

Definition 42 ([15])**.** A *GHZ/W-pair* consists of a SCFA $\big(A, \quad, \quad, \quad, \quad\big)$ and an ACFA $\big(A, \quad, \quad, \quad, \quad\big)$ on the same carrier such that the following four conditions are satisfied:



The standard example of a GHZ/W-pair in **FdHilb** are the GHZ SCFA and the W ACFA from Example 40 above. By interpreting the bra-ket notation as described in Section 2.3.3, we furthermore have that their defining equations (11) and (12) also define a GHZ/W-pair in (**F**)**Rel**.

The four conditions of Definition 42 have a clear interpretation [15]: By compactness, the first condition implies that a tick on a wire is self-inverse, which together with condition 2. implies that it is a permutation of the copyable points of the SCFA:



(14)

The third condition means that the black unit is a copyable point and the fourth that so is (up to a scalar).

The remaining three results in this (sub-)section demonstrate the power of the GHZ/W-calculus in **FdHilb**:

Theorem 43. *The equations in Definition 42 suffice to establish (up to permutation of basis vectors) a bijective correspondence between SCFAs on* $\mathbb{C}^2$ *and ACFAs on* $\mathbb{C}^2$ *in* **FdHilb**. *That is to say, fixing one Frobenius algebra uniquely determines the other.*

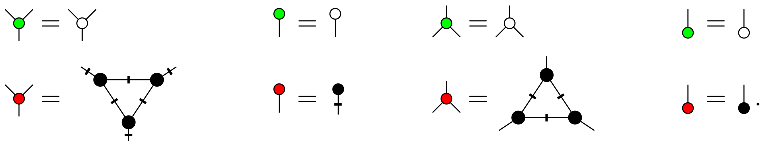
Theorem 44. *If we allow single-qubit states of the form*

$$|z(\alpha)\rangle := |0\rangle + e^{2i\alpha}|1\rangle,$$

then the GHZ/W-calculus is universal for quantum computation. That is, it can express any N*-qubit entangled state and any linear map* $L : \otimes^M \mathbb{C}^2 \to \otimes^N \mathbb{C}^2$.

Recall from Equation (9) above that *scaled bialgebras* are particular kinds of complementary quantum observables. If we generalise the notion of a scaled bialgebra to arbitrary SCFAs by requiring that the identities given by Equation (9) also hold when written upside-down, we get that the GHZ/W-calculus is at least as “fine-grained” as the Z/X-calculus on $\mathbb{C}^2$:

Theorem 45. *Each GHZ/W-pair on* $\mathbb{C}^2$ *in* **FdHilb** *induces a scaled bialgebra, given up to scalars by*



As we shall see in Chapter 4, Theorems 43 and 45 do not hold in general in **FRel**.

Graphical Lemmas

This subsection lists some lemmas, either obvious or taken from [29, 15], that we will use for the proofs in later sections.

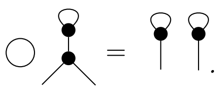
Lemma 46. *For any GHZ/W-pair, the conditions* 1.−4. *also hold when written upside-down.*

Proof. Using the normal form theorem 27 similarly to Example 28. ☐

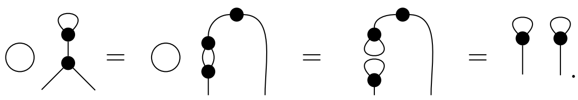
Since the same holds for the Frobenius algebra axioms, we get

Corollary 47. *Any equational statement that involves only morphisms constructed from a GHZ/W-pair and can be proved in the GHZ/W-language also holds when written upside-down.*

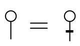
Lemma 48 (Loop copy)**.** *For any ACFA, we have*



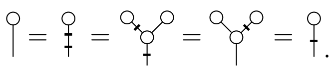
Proof. The lemma follows from Theorem 27 and anti-speciality (10):



☐

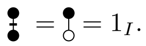
Lemma 49. *For any GHZ/W-pair,*  .

Proof. Using the fact that the tick is self-inverse and axiom 2. upside-down (cf. Lemma 46),

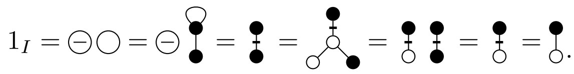


☐

Lemma 50. *For any GHZ/W-pair, if the dimension* $\bigcirc : I \to I$ *has an inverse* $\ominus$*, then*



Proof. Using the fact that $\bigcirc$ has an inverse and Lemma 49,



□

2.3.8 Classical computation

An important component in classical computation is the so-called *multiplexer*. It has three inputs ($in1, in2$ and *ctrl*) and one output (*out*) and computes

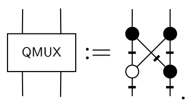
$$\begin{cases} in1 & \text{if } ctrl = 0 \\ in2 & \text{if } ctrl = 1. \end{cases}$$

In other words, it acts as a switch. The GHZ/W-calculus from the previous section can be used to define the following quantum analogue of the classical multiplexer:

Definition 51 ([15])**.** Given a GHZ/W-pair, the *quantum multiplexer*

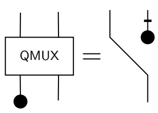
$$QMUX : A \otimes A \to A \otimes A$$

is defined as

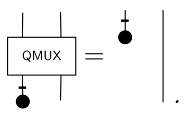


The defining property of the quantum multiplexer is given by

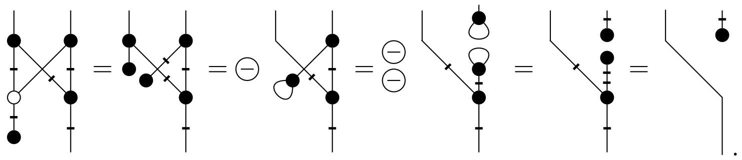
Theorem 52 ([15])**.** *If the scalar* $\bigcirc : I \to I$ *has an inverse* $\ominus$ *then the quantum multiplexer QMUX acts like a classical multiplexer when the first ("control") output is projected onto*  *resp.*  *. That is,*



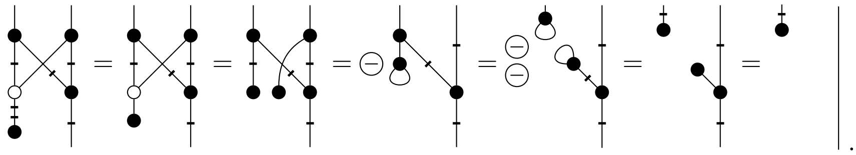
and



Proof. By the GHZ/W-axioms, Lemma 46 and Lemma 48,



Similarly,



□

The quantum multiplexer can be used to realise the inductive step of the classification of states into SLOCC super-classes (recall Section 2.2.2) [29, 15]. We will see in Chapter 5 how its computational capabilities can furthermore be exploited to define a quantum AND gate.

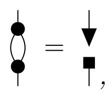
Chapter 3

A new classification of Frobenius algebras

We saw in the previous chapter that special and anti-special commutative Frobenius algebras enjoy certain canonical properties in the category **FdHilb**. The present chapter exhibits several new results that set out special and anti-special Frobenius algebras, in any symmetric monoidal category. These results then give rise to a new classification of Frobenius algebras on $\mathbb{C}^2$ in **FdHilb**, in terms of the rank of the induced loop.

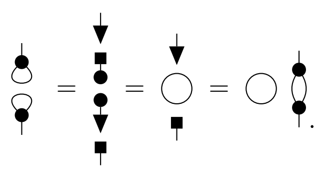
The following new result establishes the canonicity of the recent notion of anti-speciality, in any SMC:

Theorem 53. *If the loop of a Frobenius algebra* $\bullet = (A, \ , \ , \ , \ )$ *in a symmetric monoidal category* $(\mathcal{C}, \otimes, I)$ *is "disconnected", that is, there are* $: A \to I$ *and* $: I \to A$ *such that*



then $\bullet$ *is anti-special.*

Proof. Under the assumption of the proposition, and using the fact that scalars move freely around diagrams (recall Equation (3)), we have

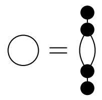


$\square$

In **FdHilb**, Theorem 53 means that any Frobenius algebra whose loop has rank ≤ 1 is anti-special. Since conversely the anti-speciality condition (10) implies that the loop has rank ≤ 1 (recall that $\bigcirc$ is equal to the dimension of the underlying Hilbert space and thus 0 iff the dimension is zero), we have

Corollary 54. *In* **FdHilb**, *the Frobenius algebras whose loop has rank* ≤ 1 *are precisely the anti-special ones.*

Remark. If the rank of the loop is 0, then also the dimension

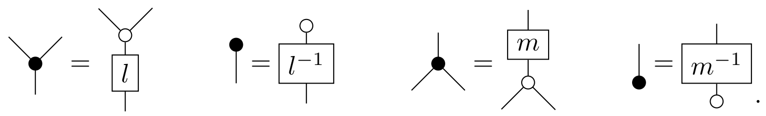


is equal to 0. Since the only Frobenius algebra on the 0-dimensional space is both special and anti-special, this means that we moreover have

Proposition 55. *If the loop of a Frobenius algebra* $\bullet = (\mathcal{H}, , , ,)$ *in* **FdHilb** *has rank* 0*, then* $\dim \mathcal{H} = 0$ *and* $\bullet$ *is both special and anti-special.*

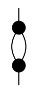
The following definition will allow us to obtain analogous results to Theorem 53 and Corollary 54 for special Frobenius algebras:

Definition 56. We call two Frobenius algebras $(A, , , ,)$ and $(A, , , ,)$ in an SMC $(\mathcal{C}, \otimes, I)$ *locally equivalent* iff there are invertible morphisms $l : A \to A$ and $m : A \to A$ such that



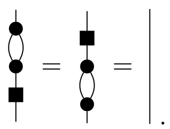
It should not be difficult to see that local equivalence of Frobenius algebras is an equivalence relation. Using Definition 56, we get the following, to our knowledge also new, result. Its proof was inspired by that of Theorem 102 in Appendix A.

Theorem 57. *Let* $\bullet = (A, , , ,)$ *be a (commutative) Frobenius algebra in an SMC* $(\mathcal{C}, \otimes, I)$*. Then the loop*



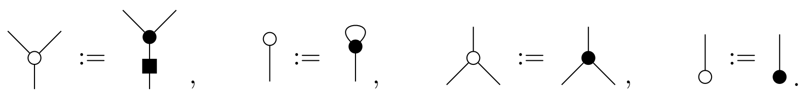
is invertible iff $\bullet$ *is locally equivalent to a special (commutative) Frobenius algebra* $\circ$.

Proof. **"⇒":** Suppose the loop is invertible, that is, there exists $: A \to A$ such that



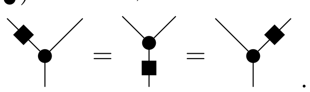
(15)

Define

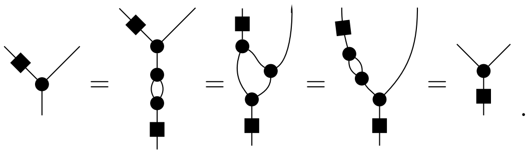


We want to show that $\circ$ is a special (commutative) Frobenius algebra. By definition, we clearly have that $(A, ,)$ is a (cocommutative) comonoid because $(A, ,)$ is. Next, we need the following

Claim.

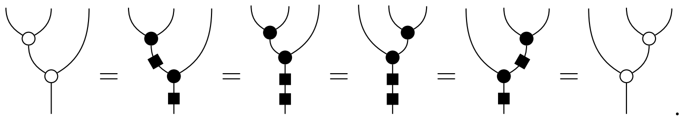


Proof. By Equation (15), the Frobenius condition and associativity,

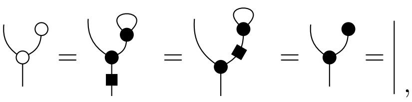


The same equalities hold for the vertical mirror images of the diagrams. □

Using the claim and associativity of ●, we have

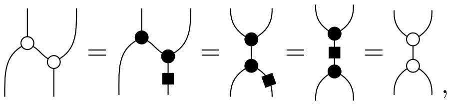


Also by the claim and unitality of ○,

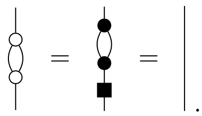


and similarly for the vertical mirror images of these equations. This shows that $(A, \;, \;)$ is an internal monoid.

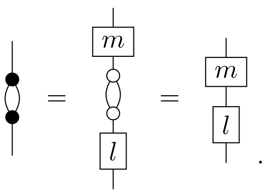
For the Frobenius condition, we have



and again similarly for the vertical mirror images. This shows that ○ is a (commutative) Frobenius algebra. To see that it is moreover special, observe that



"⇐": Suppose ● is locally equivalent to a special (commutative) Frobenius algebra ○, via invertible $l, m : A \to A$ as in Definition 56. Then



This is invertible because l and m are. □

Corollary 58. *Suppose* $\bullet = (\mathcal{H}, \;, \;, \;, \;)$ *is a commutative Frobenius algebra in* **FdHilb** *whose loop is invertible. Then the copyable points of* ● *determine a basis for* $\mathcal{H}$.

Proof. By the proof of Theorem 57, $\bullet$ induces a SCFA with the same comultiplication and thus the same copyable points[1]. The result follows from the fact that the copyable points of any SCFA on $\mathcal{H}$ determines a basis (see Table 2.1). $\square$

The above observations yield the following new classification of Frobenius algebras on $\mathbb{C}^2$ in **FdHilb** in terms of the loop rank:

Theorem 59. *Each Frobenius algebra on* $\mathbb{C}^2$ *in* **FdHilb** *is either*

- *anti-special, if the rank of its loop is* 1*, or*
- *locally equivalent to a special one, otherwise.*

Proof. By Proposition 55, the rank of the loop must be greater than 0. The result then follows from Corollary 54, Theorem 57 and the fact that a morphism $f : \mathbb{C}^2 \to \mathbb{C}^2$ in **FdHilb** is invertible iff it has rank 2. $\square$

Remark. Under assumption of a correspondence between Frobenius algebras and genuinely entangled tripartite qubit states along the lines of Theorems 32, 33, 37 and 39, Theorem 59 may be seen as giving a new, abstract explanation of the existence of two different types of genuine tripartite entanglement. This is mentioned again in Chapter 6.

[1] By Lemma 19, this SCFA is actually unique.

Chapter 4

GHZ/W-Pairs in (F)Rel

This chapter covers the meaning of the GHZ/W-calculus and its ingredients in the non-standard model **FRel** of categorical quantum mechanics. All results are proved for the category **Rel** of arbitrary sets and binary relations and then follow for **FRel**. We begin by extending the graphical language of symmetric monoidal categories and proving a few preliminary results about CFAs. Then, we describe a Haskell program that can be used to produce all commutative Frobenius algebras on the two- and three-element sets and discuss some of its (partly surprising) findings. Next, we use Pavlovic's classification of †-SCFAs [39] to give a new classification of arbitrary SCFAs and prove some properties of the (recent concept of) anti-special CFAs. Finally, we use these results to give a new classification of GHZ/W-pairs.

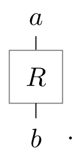
4.1 Preliminaries

4.1.1 Notation

Since we are working in **Rel**, we will be making many statements of the form

$$(a, b) \in R$$

for some binary relation $R \subseteq A \times B$. Whenever it makes sense, we will instead write this graphically, as



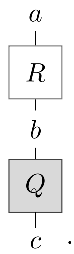
For the particular case of vectors/subsets $\uparrow : I \to A$, we will write the statement

$$(*, a) \in \uparrow \tag{16}$$

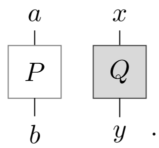
as just

(17)

When $Q \subseteq B \times C$ is another binary relation and $(a, c) \in Q \circ R$ with witnesses $(a, b) \in R$, $(b, c) \in Q$, we will write



Similarly, we will sometimes write $((a, b), (x, y)) \in P \times Q$ as



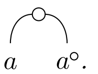
In writing, we will often identify A with the canonically isomorphic sets $\{*\} \times A$ and $A \times \{*\}$. When we use this notation, the precise type should always be clear from the context. Similarly, we will sometimes write just x for the singleton sets $\{(*, x)\}$ and $\{(x, *)\}$. In this way, for instance, Equations (16) and (17) from above could also be written as

$a \in$  .

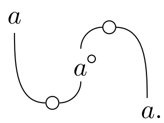
4.1.2 General results

This section proves several results that hold for all commutative Frobenius algebras in **Rel** and will be used throughout the remainder of this chapter.

Proposition 60. *For any CFA* $\circ = (A, \ldots)$ *in* **Rel**, *the cap is a permutation. That is, for all $a \in A$ there is a unique a° such that*

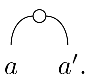


Proof. Let $a \in A$. We have $(a, a) \in 1_A$ so by compactness, there is a° such that

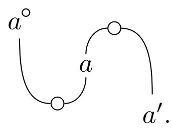


(18)

This shows existence. For uniqueness, suppose $a' \in A$ is such that



Since $\circ$ is commutative, we then have by Equation (18) that

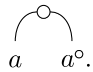


This implies by compactness that $a' = a^\circ$. Hence is a permutation. $\square$

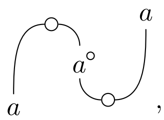
Definition 61. Given a commutative Frobenius algebra $\circ = (A, , , ,)$ and an element $a \in A$, we will keep writing $(-)^\circ$ for the unique induced element from Proposition 60.

Proposition 62. *For any CFA* $\circ$ *in* **Rel***, we have* $= ( )^\dagger$.

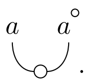
Proof. Suppose



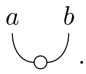
By Proposition 60, (a, a°) is the only such tuple that involves a on the left side. Therefore, by compactness,



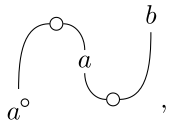
and thus, by commutativity,



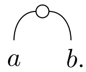
This shows $\subseteq ( )^\dagger$. Suppose now that



We must have



hence $b = a^\circ$. This implies again by commutativity that

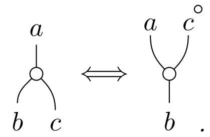


Hence $( )^\dagger \subseteq$. $\square$

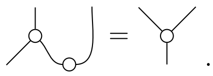
Corollary 63. *The function* $(-)^\circ$ *is an involution, that is,* $(-)^{\circ\circ} = (-)$.

Proof. Immediate from Proposition 60, Proposition 62 and compactness. $\square$

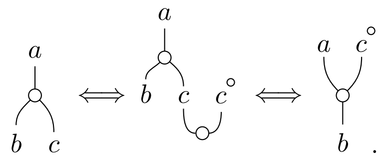
Corollary 64. *We can use* $(-)^\circ$ *to bend wires around: For all* a, b, c,



Proof. Recall from Example 28 that



Together with the above properties of $(-)^\circ$, this implies



$\square$

It is not too difficult to see that, in conjunction with commutativity, Corollary 64 allows us to bend wires of not just multiplication and comultiplication, but of any connected morphism constructed from a CFA.

Proposition 65. *For any CFA on a non-empty carrier in* **Rel**, *we have*

$$\bigcirc = 1_I.$$

Proof. Since the carrier is non-empty, Propositions 60 and 62 imply that $1_I \subseteq \bigcirc$. The result follows from the fact the only scalars are $\emptyset$ and 1_I. $\square$

4.2 Computational results

In order to get anunderstanding of the meaning of commutative Frobenius algebras in **FRel** on top of the above results, a Haskell program was used to exhaustively search for and thus classify all CFAs on the two- and three-element sets. Once this was in place, the program was expanded to generate CFAs in higher dimensions (i.e. sets with more elements) and query their properties. This section gives an overview of the implementation and the results obtained. The list of all CFAs on the two- and three-element sets **2** and **3** can be found in Appendix B. A bare-bones version of the Haskell code that can be used to reproduce all computational results is given in Appendix C.

4.2.1 Method

The main aim of this subsection is to provide a handle on seeing that the implementation and thus the computational results given in this thesis are correct. We will make references to the code given in Appendix C, however, these references will not be crucial for gaining an understanding of the general structure of the approach. For this reason, no prior knowledge of Haskell is required.

The reason Haskell was used is that it is a mature programming language whose syntax and primitives make it very easy to describe sets and binary relations. This is exemplified by the fact that many of the functions of the implementation are near-literal translations of the corresponding mathematical definitions. Furthermore, interactive command-line interpreters such as `GHCi` [1] make it very easy to test and then “play around with” an implementation.

The basic ingredients of the †-SMC **FRel** are modelled as follows: Objects are represented as lists, binary relations as lists of tuples (Haskell type `BinRel`). Composition and tensoring are near-literal translations of the mathematical definitions for **FRel** (Haskell functions `o`, `x`). Similarly identity maps and the symmetric monoidal structure maps are direct translations of the mathematical definitions (`idBinRel`, `alBinRel`, `lamBinRel`, `rhoBinRel`, `sigBinRel`). Finally, a dedicated Haskell function (`eqBinRel`) determines when two binary relations are equal. These building blocks make it possible to define Boolean predicates that capture when a tuple consisting of an object and binary relations of the right type define a (commutative, special or anti-special) Frobenius algebra (`isFrobeniusAlgebra`, `isCFA`, `isSCFA`, `isACFA`).

Finding CFAs, Attempt 1

Recall that a Frobenius algebra on a given carrier A consists of four morphisms , , and . In **FRel**, these are all binary relations and thus subsets of $A^{\otimes n}$ for particular values of n. The first version of the Haskell implementation simply went through each possible combination of subsets of the right type and checked whether it was a commutative Frobenius algebra (`cfasOn`). Not surprisingly, this was extremely inefficient and too slow to obtain results for the three-element set[1].

Finding CFAs, Attempt 2

The second (and final) attempt at finding all commutative Frobenius algebras on the two- and three-element sets in **FRel** relies on the fact that each commutative Frobenius algebra is uniquely determined by its induced tripartite state and counit (Theorems 33 and 32). The implementation generates all possible combinations of symmetric tripartite states and counits and filters out those for which the induced tuple (, , ,) is a CFA (`cfasOn’`). This approach is much faster than the previous one, resulting in a reduction of time from over one minute to less than one second to find all commutative Frobenius algebras on the two-element set and making it possible to find all CFAs on the three-element set in less than three minutes[2].

Querying CFA properties

Once the approach described above had found all CFAs on the two- and three-element sets, the Haskell implementation was extended by Boolean predicates to query their properties. The most important examples of such predicates are

[1] Running the command `cfasOn [0, 1]` took over 1 minute on an Intel T2400 Dual Core machine with 2x1.83 GHZ, 1.5 GB of RAM, Windows 7 and version 6.12.3 of the interactive environment of the Glasgow Haskell compiler. The command `cfasOn [0, 1, 2]` failed to produce a single CFA in over four hours.

[2] Commands `cfasOn [0, 1]`, `cfasOn’ [0, 1]` and `cfasOn’ [0, 1, 2]` in the environment described in Footnote 1.

`eqFrobenius- AlgebraUpToPermutation`, that determines whether to Frobenius algebras are equal up to a bijection between carriers, and `isGHZWPair`, that identifies when two commutative Frobenius algebras form a GHZ/W-pair.

Generating (canonical) examples

In order to get a grip on CFAs on carriers with more than three elements, the Haskell implementation contains functions that generate examples of (canonical) CFAs for any dimension. The function `discreteSCFAProduct` can be used to generate SCFAs with group decompositions of the form $\mathbb{Z}_{n_1} + \cdots + \mathbb{Z}_{n_k}$ (see Theorem 66). The function `minACFA` can be used to generate minimal ACFAs, as per Definition 81 below.

4.2.2 Results

SCFAs

The SCFAs exhibited by the implementation are precisely those predicted by Pavlovic's classification of *dagger*-SCFAs [39] (see also Section 2.3.5). This sparked the proof of Theorem 66 below, which states that every SCFA in **Rel** is dagger.

ACFAs

The Haskell implementation found 4 ACFAs on **2** (two up to a permutation of carrier elements) and 66 on **3** (13 up to permutation). Since there are only 10 SCFAs on **3**, this means that there are more ACFAs than SCFAs. An intuitive explanation for this phenomenon is that the speciality condition imposes an "absolute" constraint (the loop of the FA has to be equal to the identity, a constant) while anti-speciality is only a relative statement between FA morphisms.

Several of the ACFAs found show that some of the theorems from Chapter 2 do not hold in greater generality. Unlike in **FdHilb** (Proposition 41), there are †-ACFAs on non-trivial carriers ($\bullet_b$, $\bullet_6 - \bullet_{12}$). Unlike for SCFAs (Lemma 19), there can be several different ACFAs for the same commutative multiplication ($\bullet_3$, $\bullet_6$).

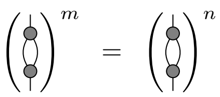
A final interesting observation is that all ACFAs exhibited by the Haskell implementation have singleton units and counits. This turns out to be true for all ACFAs in **Rel**, see Proposition 76 below.

GHZ/W-pairs

Interestingly, there is only one GHZ/W-pair each on the two- and three-element sets (up to permutation of carrier elements). On **2**, this pair is $(\circ_{1+1}, \bullet_a)$, while on **3**, it is $(\circ_{1+1+1}, \bullet_1)$. Already on the two-element set, this leaves out a SCFA ($\circ_2$) and an ACFA ($\bullet_b$), that are not part of any GHZ/W-pair. This is in contrast to **FdHilb**, where every SCFA on $\mathbb{C}^2$ uniquely determines (up to a permutation of basis vectors) an ACFA under the GHZ/W-axioms, and vice versa (Theorem 43).

Non-special or -anti-special CFAs

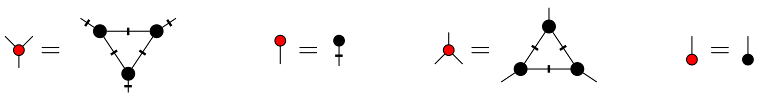
The Haskell implementation revealed (up to permutation) one CFA that is neither special nor anti-special on the two-element set, and nine such CFAs on the three-element set. In terms of their loops, all of them satisfy



for some (distinct) m and n.

Non-generality of induced Z/X-calculus

Recall from Theorem 45 above that each GHZ/W-pair on $\mathbb{C}^2$ in **FdHilb** induces a scaled bialgebra and thus gives rise to a Z/X-calculus. Unfortunately, the construction of the corresponding proof does not work in general: The X-observable

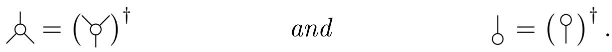


induced by the GHZ/W-pair $(\circ_{1+1+1}, \bullet_1)$ is not even a SCFA[3].

4.3 Classification of SCFAs

The main aim of this section is to prove the following (new)

Theorem 66. *Every special commutative Frobenius algebra* $(A, \ldots)$ *in* **Rel** *is dagger, that is,*



Theorem 66 allows us to strengthen Pavlovic's classification of †-SCFAs in **Rel** ([39], see also Section 2.3.5):

Corollary 67. *The special commutative Frobenius algebras in* **Rel** *are precisely the quintuples* $\left(A, \ldots, (\ldots)^\dagger, (\ldots)^\dagger\right)$ *such that*

$$A = \bigcup_{j\in J} X_j \qquad = \bigcup_{j\in J} M_j \qquad = \{e_j\}_{j\in J},$$

where J is any set, $G_j = (X_j, *_j, e_j)$ are abelian groups with disjoint carriers and $M_j \subseteq (X_j \times X_j) \times X_j$ are the graphs of the respective group multiplications $*_j$.

As is natural, we extend Definition 23:

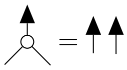
Definition 68. For a SCFA $\circ$ in **Rel**, we call the set $\{G_j \mid j \in J\}$ from Corollary 67 the *group decomposition* or simply the *groups* of $\circ$ and write

$$\circ \;=\; \sum_{j\in J} G_j.$$

[3]Haskell command `(uncurry5 isSCFA)(inducedXObservable (discreteSCFAProduct [1,1,1])(minACFA [0..2] [(0, 1)]))`.

Corollary 67 also allows us to give a more general form of a result due to Evans et al., that we will use later:

Lemma 69 ([25, Thm 3.2])**.** *The copyable points of a SCFA* $\circ$ *in* **Rel** *are precisely the group carriers* X_j *from Corollary 67. That is,*

 $\iff$  $= \{*\} \times X_j,$

for some $j \in J$.

Example 70. The copyable points of the SCFA $\mathbb{Z}_1 + \mathbb{Z}_1$, given by

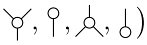
 $_{1+1} = \{((0,0),0),((1,1),1)\}$ and  $_{1+1} = \{(*,0),(*,1)\}$,

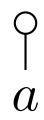
are precisely the sets $\{(*,0)\}$ and $\{(*,1)\}$. The SCFA $\mathbb{Z}_2$, for which

 $_2 = \{((0,0),0),((0,1),1),((1,0),1),((1,1),0)\}$ and  $_2 = \{(*,0)\}$,

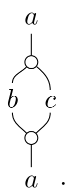
has only one copyable point, namely $\{(*,0),(*,1)\}$.

4.3.1 Proof

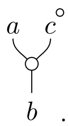
Let $\circ = ($A, ) be a special commutative Frobenius algebra in **Rel**.

Lemma 71. *For all* $a \in A$,  *implies* $a = a^\circ$.

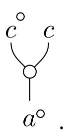
Proof. Suppose . By speciality, there are b and c such that



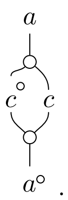
By Corollary 64, we have



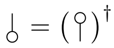
Since $a \in$ , this implies $b = c^\circ$. By repeated applications of Corollary 64, commutativity and Corollary 63, we then also get



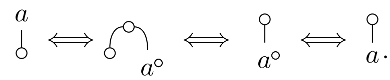
But then,



This implies by speciality that $a = a^\circ$. $\square$

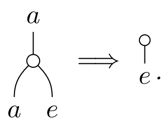
Corollary 72. 

Proof. Using Lemma 71, we have

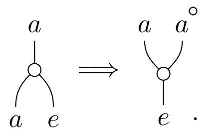


$\square$

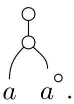
Lemma 73.  *contains all coidentity elements, i.e.*



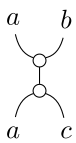
Proof. We have



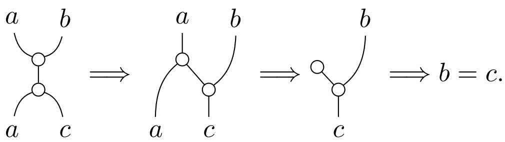
By the definition of $(-)^\circ$, we furthermore have



The result follows by speciality. $\square$

Corollary 74.  *implies $b = c$.*

Proof. By the Frobenius condition, Lemma 73 and unitality,



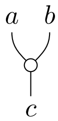
$\square$

Proof of Theorem 66.  $=$ ()† is Corollary 72. We now show

 $=$ ()† ,

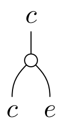
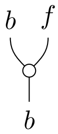
the result then follows from the fact that $(-)^\dagger$ is an involution.

Suppose

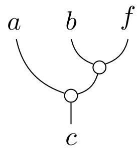


.

By unitality and Corollary 72, there are $e, f \in$  such that

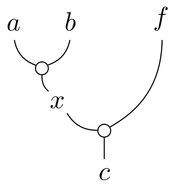
 and  .

Gluing together diagrams, we get

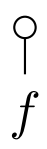


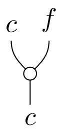
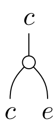
.

By associativity, we get the existence of some x such that

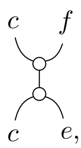


.

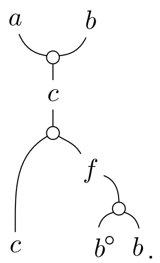
Since  , we must have $x = c$. Hence we have

 and  .

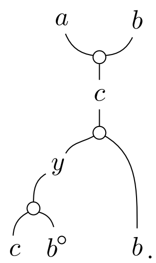
Gluing these two diagrams together yields



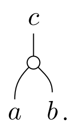
so, by Corollary 74, $e = f$. Bending wires, this implies

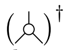


By coassociativity, there now exists y such that

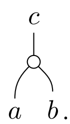


By commutativity and Corollary 74, $a = y$. Hence, finally,

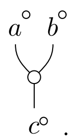


This shows  $\subseteq$ .

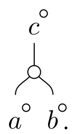
Suppose now that



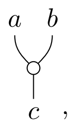
Bending wires, we get,



We can apply the above reasoning to obtain



Bending the wires back, we then get



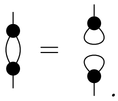
as required. This shows $(\;)^\dagger \subseteq$ and thus concludes the proof. $\square$

Remark. It may be possible to obtain a nicer proof of Theorem 66 using Lemma 19.

4.4 Properties of ACFAs

This section exhibits several (new) results about ACFAs in **Rel** that will be used throughout the remainder of this chapter. As a first observation, note that the (only) CFA on the empty carrier $\emptyset$ is (both special and) anti-special. Since by Proposition 65 the circle induced by all other CFAs is equal to 1_I, this means that we have

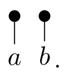
Proposition 75. *A commutative Frobenius algebra* $\bullet$ *in* **Rel** *is anti-special iff*



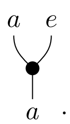
The following property was already observed to hold for the ACFAs exhibited by the Haskell implementation. We now prove it for the general case:

Proposition 76. *If the carrier of an ACFA* $\bullet$ *in* **Rel** *is non-empty, then its unit and counit are singleton sets. That is,* $|\;| = |\;| = 1$.

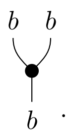
Proof. By unitality and the fact that the carrier is non-empty, we have $|\;| \geq 1$. Suppose



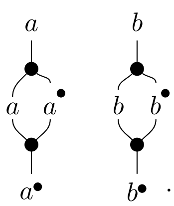
We will show that $a = b$. By unitality, there is $e \in$ such that



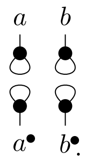
However, since $a \in$, we have $e = a$. By a similar argument, it can be shown that



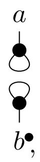
Bending wires and using unitality, we then have



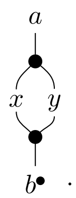
Hence, by anti-speciality,



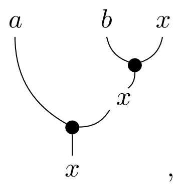
This implies



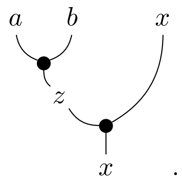
and so, by anti-speciality, there are x and y such that



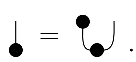
Since $a \in$, we have $y = x^{\bullet}$. Then, bending wires,



so by associativity, there is a z such that



Since $a, b \in$, we then have that $a = z = b$. This implies that $\left| \ \right| = 1$. That also $\left| \ \right| = 1$ follows from the fact that is a permutation (Propositions 60 and 62) and the equation



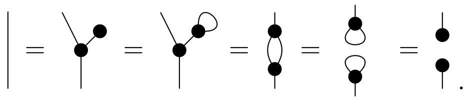
$\square$

The next result represents a certain analogue of Proposition 41, which stated that there are no non-trivial †-ACFAs in **FdHilb**:

Proposition 77. *For* $(A, \ldots)$ *an ACFA in* **Rel**, *we have*

$$= \quad \Longleftrightarrow \quad |A| \leq 1.$$

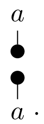
Proof. **"⇒":** Suppose $=$. Then by unitality, the normal form theorem for CFA morphisms (Theorem 27) and anti-speciality in **Rel** (Proposition 75), we have



(19)

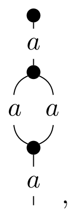
Let $a \in A$. Then by the definition of 1_A,

and thus, by Equation (19),



This implies that $a \in$ and thus that $A \subseteq$. Since by Proposition 76 the size $|\;| \leq 1$, we then have $|A| \leq 1$, as required.

"⇐": If $|A| = 0$, then trivially $= \emptyset =$. Suppose $|A| = 1$ and write $A = \{a\}$. By unitality and counitality, we must have $= \{(*, a)\}$ and $= \{(a, *)\}$. This implies



and thus $\{(*, a)\} \subseteq$. Since by definition $\subseteq I \times A$, this shows $=$. ☐

Finally, we have

Lemma 78. *If* $|\;| = 1$, *then* $= (\;)^\dagger$.

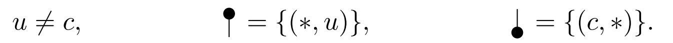
Proof. Since $|\;| = 1$, we have that the carrier of • is non-empty. This implies by Proposition 65 that

$$1_I = \bigcirc = \;.$$

Since by Proposition 76 also $|\;| = 1$, this implies that $= (\;)^\dagger$. ☐

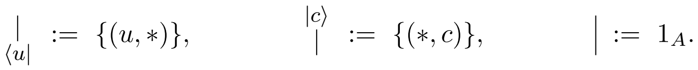
4.4.1 Minimal ACFAs

Suppose $\bullet = (A,$, , , $)$ is a CFA in **Rel** with distinct singleton unit and counit, i.e. such that there exist u and $c \in A$ which satisfy



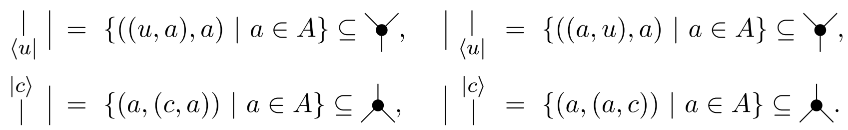
(20)

Write

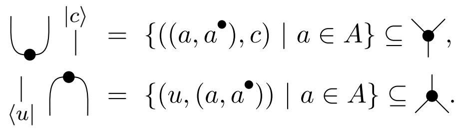


(21)

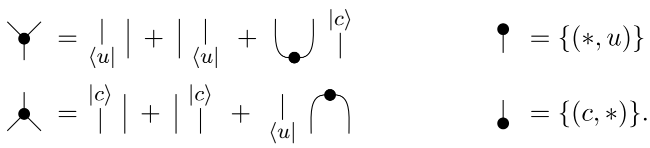
Then, using the graphical notation for the †-SMC **Rel** and assuming strictness of the symmetric monoidal structure (recall Section 2.3.4), unitality and counitality of $\bullet$ imply that



Furthermore, by Propositions 60 and 62, we know that the cup and cap induce an involutary bijection $(-)^\bullet : A \to A$. By these results and the definition of $(-)^\bullet$, we must also have



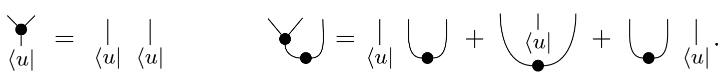
Suppose now that , , and contain nothing but the elements described above. That is, writing $+$ instead of $\cup$ in order not to confuse cups,



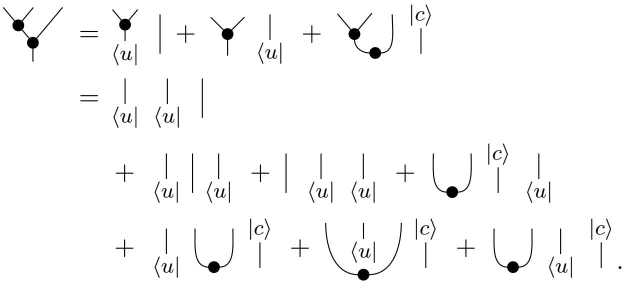
(22)

(23)

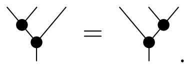
Does this give a CFA? By construction, we have unitality and counitality. For associativity, observe that, since $u \neq c$, we have



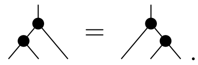
Hence



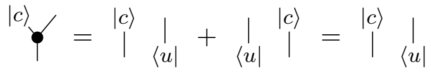
This is invariant with respect to any swapping of incoming wires. Therefore,



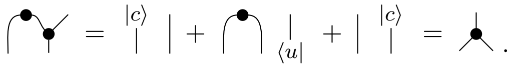
A similar proof shows that likewise



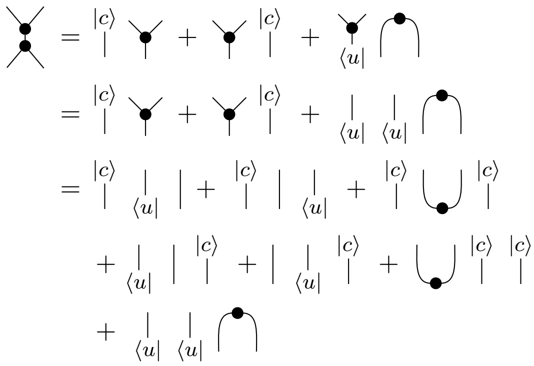
For the Frobenius condition, note that, again since $u \neq c$, we have



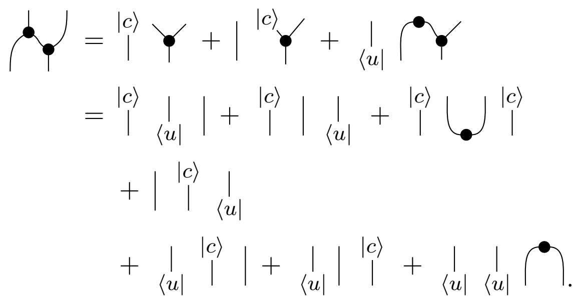
and



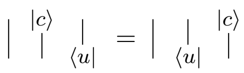
This implies



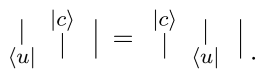
and



These are equal since

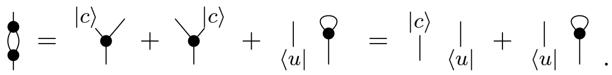


and

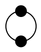


Since  is an involutary bijection, we have that  is commutative. Therefore, $\bullet$ is a commutative Frobenius algebra.

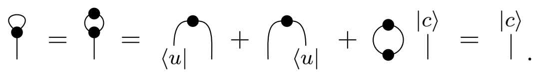
We have one more interesting property: Observe that



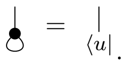
Since $u^\bullet = c$ and because $(-)^\bullet$ being an involutory bijection implies

 $= 1_I$,

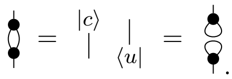
(note $A \neq \emptyset$ as $u \in A$), we have



Similarly, it can be shown that

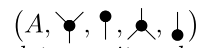


This finally implies



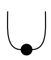
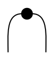
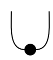
By Proposition 75, this means that $\bullet$ is an ACFA.

From the above discussion, we have

Theorem 79. *Suppose* $\bullet$ $=$  *is a commutative Frobenius algebra in* **Rel** *with distinct singleton unit and counit, i.e. such that there are* u *and* $c \in A$ *which satisfy Equation (20). Then* $\bullet$ *contains an ACFA, given by Equations (22) and (23).*

We also have the converse:

Theorem 80. *Given a set* A*, two distinct elements* $u, c \in A$ *and an involutory bijection* $(-)^\bullet : A \to A$ *such that* $u^\bullet = c$*, Equations (21)–(23) define an anti-special commutative Frobenius algebra if we set*

 $:= \{((a, a^\bullet), *) \mid a \in A\}$ *and*  $:=$ †.

Definition 81 (Minimal ACFAs)**.** We call the induced ACFA from Theorem 80 the *minimal ACFA* for the respective parameters.

Minimal ACFAs will be used as a convenient source of counterexamples in Chapter 5.

4.5 Classification of GHZ/W-pairs

Building on the results from the previous two sections, we can now explain the meaning of the GHZ/W-axioms in **Rel**. The main result we are going to prove is

Theorem 82. *The GHZ/W-pairs on a carrier* A *with at least two elements in* **Rel** *are precisely those SCFA/ACFA-pairs* $(\circ, \bullet)$ *such that* *i.* *there are at least two copies of the singleton group $\mathbb{Z}_1$ in the group decomposition of* $\circ$*, given by* *and*

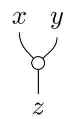
ii. $(-)^{\bullet}$ *is an isomorphism of the groups of* $\circ$*. That is, if x and y are in the same group $G_j = (X_j, *_j, e_j)$ of* $\circ$*, then $x^{\bullet}$ and $y^{\bullet}$ are in the same group G_k of* $\circ$ *and*

$$x^{\bullet} *_k y^{\bullet} = (x *_j y)^{\bullet}.$$

Since it is not difficult to see that there is precisely one GHZ/W-pair on the empty and one-element sets (recall Example 16), this gives a complete classification of all GHZ/W-pairs in **Rel**.

Before the proof, we need two small auxiliary results. The first follows immediately from Corollary 67:

Lemma 83. *For a SCFA* $\circ$ *in* **Rel** *with group decomposition $\sum_j G_j$, where $G_j = (X_j, *_j, e_j)$, we have*

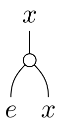
 $\iff$ *x, y, z are in the same group G_j and $x *_j y = z$.*

The second explains the meaning of the function $(-)^{\circ}$ for a SCFAs:

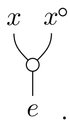
Proposition 84. *For a SCFA* $\circ$ *in* **Rel***, the function $(-)^{\circ}$ identifies group inverses. That is, for an element x of a group $(X_j, *_j, e_j)$ of* $\circ$*, x° is the unique element such that*

$$x *_j x^{\circ} = e_j. \tag{24}$$

Proof. Let x be an element of a group $G_j = (X_j, *_j, e_j)$ of $\circ$. By counitality, we have



for some $e \in$. Bending wires, this implies



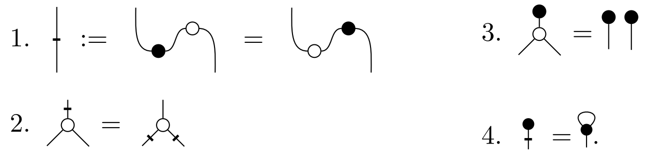
By Lemma 83, we must now have that x° and e are in G_j and that

$$x *_j x^{\circ} = e.$$

By Theorem 66, $\circ$ must be dagger so we must have $e \in$. By the classification of SCFAs (Corollary 67), this then implies that $e = e_j$.

That x° is the unique element satisfying Equation (24) follows from the well-known fact that group inverses are unique. $\square$

Proof of Theorem 82. By Proposition 65, the GHZ/W-axioms become

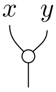


"⇒": Suppose $(\circ, \bullet)$ is a GHZ/W-pair on a carrier A with at least two elements. First observe that, by the fact that cups and caps are permutations (Propositions 60 and 62), compactness and Axiom 1., the tick is an involutary function.

By Axiom 3., is a copyable point of $\circ$. This implies by Lemma 69 that is one of the groups that make up $\circ$. Since by Proposition 76 the size $|\;|$ is equal to 1, this group is (isomorphic to) $\mathbb{Z}_1$. By Axiom 2., we have that identifies another group in the decomposition of $\circ$ (recall Equation (14)). By Axiom 4., this group is given by , and, since the tick is a function, it also has size 1. If $=$, then by Proposition 77 the size $|A| \leq 1$, a contradiction. This shows that and identify two distinct copies of $\mathbb{Z}_1$ in the group decomposition of $\circ$.

To see that $(-)^{\bullet}$ is an isomorphism of the groups of $\circ$, let $G_j = (X_j, *_j, e_j)$ be one of these groups and $x, y \in G_j$. By Lemma 83, we have

$x \quad y$

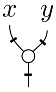


$x *_j y.$

Now by Axiom 2. upside-down (Lemma 46) and the fact that the tick is self-inverse, we have

and thus

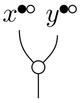
$x \quad y$



$x *_j y.$

By the definition of the tick, this implies

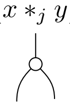
$x^{\bullet\circ} \quad y^{\bullet\circ}$



$(x *_j y)^{\bullet\circ}$

and hence, bending wires,

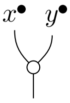
$(x *_j y)^{\bullet}$



$x^{\bullet} \quad y^{\bullet}.$

Since $\circ$ must be dagger, we then have

$x^\bullet \quad y^\bullet$



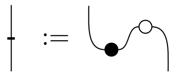
$(x *_j y)^\bullet$.

By Lemma 83, this implies that $x^\bullet, y^\bullet$ and $(x *_j y)^\bullet$ are in the same group G_k of $\circ$ and that

$$x^\bullet *_k y^\bullet = (x *_j y)^\bullet.$$

This shows that $(-)^\bullet$ is a group homomorphism; that it is moreover an isomorphism follows from compactness.

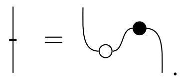
"$\Leftarrow$": Suppose $(\circ, \bullet)$ is a SCFA/ACFA pair on a carrier A with at least two elements such that conditions i. and ii. are satisfied. Again by the fact that cups and caps are permutations (Propositions 60 and 62), we have that the tick



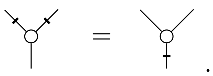
is a function. By Proposition 84, condition ii. and a well-known fact of group homomorphisms, we then have that, for $x \in A$,

$$x^{\circ\bullet} = (x^{-1})^\bullet = (x^\bullet)^{-1} = x^{\bullet\circ}.$$

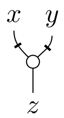
This implies



Axiom 1. (which we have just shown to hold) implies that the tick is self-adjoint. Together with the fact that $\circ$ is dagger, this implies that Axiom 2. is equivalent to



By Lemma 83 and Axiom 1., we have



$\iff$ $x^{\circ\bullet}, y^{\circ\bullet}, z$ are in the same group G_j and $x^{\circ\bullet} *_j y^{\circ\bullet} = z$.

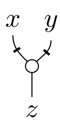
Now, since $(-)^\bullet$ is an involutary group isomorphism, for some group G_k of $\circ$,

$$x^{\circ\bullet} *_j y^{\circ\bullet} = (x^\circ *_k y^\circ)^\bullet.$$

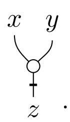
Then, by Proposition 84 and the fact that G_k is abelian,

$$(x^\circ *_k y^\circ)^\bullet = (x^{-1} *_k y^{-1})^\bullet = \left((x *_k y)^{-1}\right)^\bullet = (x *_k y)^{\circ\bullet}.$$

Hence, by Lemma 83,



$\iff (x *_k y)^{\circ\bullet} = z \iff x *_k y = z^{\bullet\circ} \iff$



From this, it can be seen that Axiom 2. holds.

By Lemma 69, Axiom 3. is the first half of condition i. Write e for the unique element of . By Proposition 84, identifying singleton sets $\{(*, x)\}$ with x,

$$ = e^{\circ\bullet} = \left(e^{-1}\right)^{\bullet} = e^{\bullet} = \left(\;\right)^{\dagger} = \left(\;\right)^{\dagger}.$$

The fact that identifies a copy of $\mathbb{Z}_1$ in the group decomposition of $\circ$ implies that $\left|\;\right| = 1$. Then, by the above and Lemma 78, we have

$$ = .$$

This is Axiom 4. $\square$

Example 85. The two GHZ/W-pairs $(\circ_{1+1}, \bullet_a)$ and $(\circ_{1+1+1}, \bullet_1)$ exhibited by the Haskell implementation (cf. Section 4.2) satisfy the conditions given by Theorem 82. More generally, let $\circ$ be a SCFA with group decomposition of the form $\mathcal{D} = \mathbb{Z}_1 + \cdots + \mathbb{Z}_1$ and $\bullet$ be any ACFA on the same carrier such that $|\;| = 1$. By Propositions 76 and 77, we have $\left|\;\right| = 1$ and $\neq$. This implies that condition i. is satisfied. Since moreover any bijection from $\mathcal{D}$ to $\mathcal{D}$ is an isomorphism, we then have by Theorem 82 that $(\circ, \bullet)$ forms a GHZ/W-pair.

The fact that SCFAs of the form $\mathbb{Z}_1 + \cdots + \mathbb{Z}_1$ form a GHZ/W-pair with any ACFA on the same carrier satisfying the (weak) condition $|\;| = 1$ indicates that the GHZ/W-axioms do not "scale" as we look at sets with more elements. This can also be seen from the fact that, on the two-element set, the SCFA is completely determined by the constraint that it must contain two distinct copies of $\mathbb{Z}_1$, while on carriers with more than 3 elements, this is no longer the case. We will get back to this in Chapter 6.

Chapter 5

A quantum AND gate

As a final, more practical evaluation of the expressive power of the GHZ/W-calculus, we use it to define a quantum analogue of the Boolean AND gate.[1] We identify one new and several known graphical conditions that hold for the standard GHZ/W-pair from Example 40 and use these conditions in conjunction with the GHZ/W-axioms to prove some properties of our gate. We also exhibit some properties it does not satisfy and argue that there are probably no additional simple graphical axioms that could rectify this situation. In order not to blur the line of argument, almost all proofs are given in Appendix A.

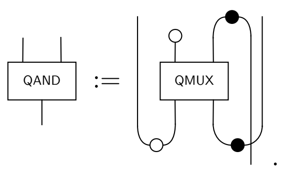
By Propositions 21 and 65, the following assumption holds for all non-trivial CFAs in **FdHilb** and (**F**)**Rel** and will be used throughout this chapter:

Assumption 86. *The dimension* $\bigcirc : I \to I$ *has an inverse* $\ominus$.

Definition 87. Given a GHZ/W-pair, the *quantum and gate*

$$QAND : A \otimes A \to A$$

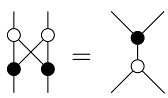
is defined as



5.1 Correctness and commutativity

The following condition already played a role in the Z/X-calculus [11] (see also Equation (9) above):

Definition 88. We say a GHZ/W-pair satisfies *bialgebraic commutation* or *the bialgebra law* iff

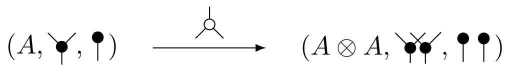
 (25)

and the same equality also holds when written upside-down.

[1]Another quantum AND gate was recently proposed in [5].

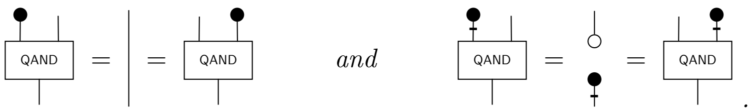
The fact that the standard GHZ/W-pair satisfies bialgebraic commutation was noted in [29]. As a small aside, this paper also makes the following interesting observation:

Proposition 89 ([29])**.** *A GHZ/W-pair satisfies Equation (25) if and only if the GHZ comonoid is a homomorphism of W monoids:*

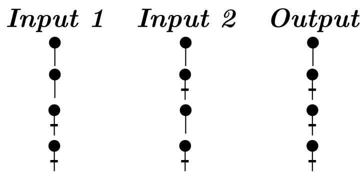


Bialgebraic commutation allows us to prove the following crucial defining property of $QAND$:

Theorem 90 (Correctness of $QAND$)**.** *Under assumption of the bialgebra law, we have*



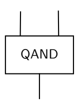
This implies that the quantum AND gate behaves like Boolean $\wedge$ with respect to the inputs (“true”) and (“false”). That is, it has the following truth table:



Note how the tick , by virtue of being self-inverse, acts like a logical NOT gate. This is exemplified by the standard GHZ/W-pair in both **FdHilb** and (**F**)**Rel**, where

$$= |0\rangle\langle 1| + |1\rangle\langle 0|.$$

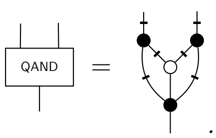
Moreover, for the standard GHZ/W-pairs, since the morphisms and act like a basis for $\mathbb{C}^2$ resp. **2**, Theorem 90 implies



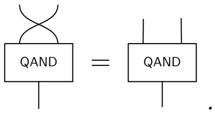
$$= |0\rangle\langle 00| + |0\rangle\langle 01| + |0\rangle\langle 10| + |1\rangle\langle 11|.$$

Using the spider notation for connected CFA morphisms without loops (recall Example 26), we also get that $QAND$ is commutative for all inputs:

Theorem 91 (Commutativity of $QAND$)**.** *If a GHZ/W-pair satisfies bialgebraic commutation, then its induced quantum AND gate has a symmetric graphical representation, given by*

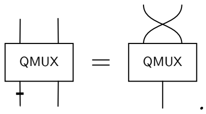


This implies that it is commutative, i.e.



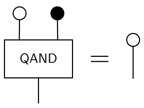
A key ingredient of the proof of Theorem 91 is that a tick on the control output of the quantum multiplexer effects a swapping of inputs.

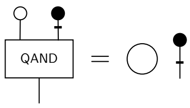
Proposition 92. *For any GHZ/W-pair, we have*

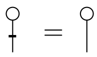


5.2 A third truth value

An interesting observation to make at this point is that, by Theorem 90 and Lemma 49, the white unit nearly behaves like a third truth value "unknown" (up to a scalar):

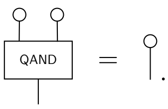




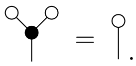


In words, $unknown \wedge true = unknown$ since replacing $unknown$ by $true$ or $false$ could result in either $true$ or $false$, while $unknown \wedge false = false$ because both $true \wedge false$ and $false \wedge false$ are equal to $false$. Furthermore, $NOT\ unknown = unknown$.

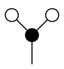
The only equality that does not follow from Theorem 90 is $unknown \wedge unknown = unknown$, or



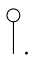
Definition 93. [2] We say a GHZ/W-pair satisfies *eta idempotence* iff



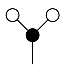
The standard GHZ/W-pair from Example 40 does *not* satisfy eta idempotence in **FdHilb**:



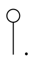
$$= 2|0\rangle + |1\rangle \neq |0\rangle + |1\rangle =$$



However, in (**F**)**Rel**, since there are only two scalars and thus intuitively "2 = 1", we do have



$$= |0\rangle + |1\rangle =$$



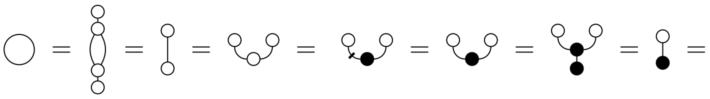
The fact that the equality holds in (**F**)**Rel** but not in **FdHilb** can be explained graphically by

[2]The use of eta idempotence to prove Proposition 95 was suggested by Aleks Kissinger. The presentation given here and Lemma 94 are my own work.

Lemma 94. *If a GHZ/W-pair satisfies eta itempotence, then*

$$= 1_I.$$

Proof. By speciality, eta idempotence, Lemmas 49, 50 and Corollary 47,

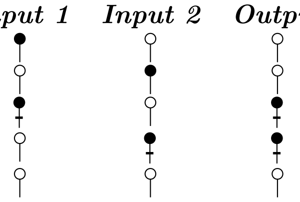


$$= 1_I.$$

□

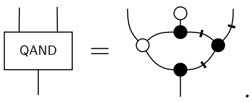
Proposition 95. *If a GHZ/W-pair satisfies bialgebraic commutation and eta idempotence, then the quantum AND gate behaves like ternary $\wedge$ with respect to the inputs (“true”), (“false”) and (“unknown”). That is, it has the following truth table in addition to that given by Theorem 90:*

Input 1 ***Input 2*** ***Output***



The proof of Proposition 95 uses yet another, simpler graphical representation of $QAND$.

Lemma 96. *Under assumption of the bialgebra law, we have*

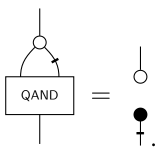


5.3 Complements

Similarly to the general proof of commutativity above, we would like to prove other axioms of Boolean algebra without reference to particular truth values. In order to be able to do this, we can use the fact that and are both copied by the white comultiplication . In this way for instance, the law of complements

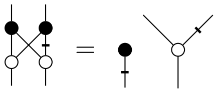
$$p \wedge \neg p = \perp$$

can be expressed as



(26)

Definition 97. We say a GHZ/W-pair satisfies *ticked bialgebraic commutation* iff



(27)

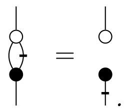
and the same equality also holds when written upside-down.

Direct calculation shows that the standard GHZ/W-pair satisfies ticked bialgebraic commutation in both **FdHilb** and (**F**)**Rel**.[3]

Theorem 98 (Complements for $QAND$)**.** *If a GHZ/W-pair satisfies ticked and non-ticked bialgebraic commutation, then it satisfies Equation (26).*

The proof of Theorem 98 uses the following result, which in a similar form already appeared in the Z/X-calculus:

Lemma 99 ([11, Thm 9])**.** *If a GHZ/W-pair satisfies bialgebraic commutation, then it satisfies the* Hopf law



5.4 Absorption

Since the tick behaves like logical NOT, we can use De Morgan's law

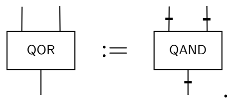
$$\neg(p \vee q) = (\neg p) \wedge (\neg q)$$

to define a quantum or gate:

Definition 100. Given a GHZ/W-pair, the *quantum or gate*

$$QOR : A \otimes A \to A$$

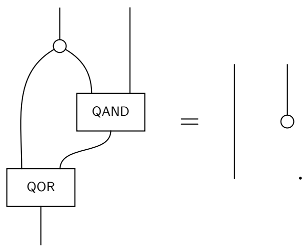
is defined as



By using  and  similarly to above as "placeholders" for truth values, this allows us to express the absorption law,

$$p \vee (p \wedge q) = p,$$

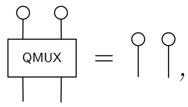
graphically as



(28)

[3] The paper [29] states that the standard GHZ/W-pair satisfies Equation (27) with  instead of . In my calculations,  seems to be correct.

Theorem 101 (Absorption for $QAND$)**.** *Under assumption of ticked and non-ticked bialgebraic commutation and the identity*

 (29)

we have that Equation (28) holds.

Direct calculation again verifies that the standard GHZ/W-pair satisfies Equation (29) in **FdHilb** and (**F**)**Rel**. Interestingly, Equation (28) is independent of the GHZ/W-axioms, eta idempotence, ticked and non-ticked bialgebraic commutation: The SCFA $\circ = \mathbb{Z}_1 + \mathbb{Z}_1 + \mathbb{Z}_2 + \mathbb{Z}_2$ with respective group carriers $\{0\}$, $\{1\}$, $\{2,3\}$, $\{4,5\}$ and ⚲ $= \{0,1,2,4\}$ and the minimal ACFA on $\{0,\dots 5\}$ with ⚲ $= \{(*,0)\}$ and (in cycle notation)

$$(-)^{\bullet} = (01)(24)(35)$$

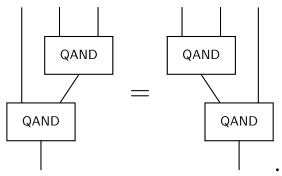
satisfies all these constraints but not absorption as per Equation (28). [4]

5.5 Violated properties

There are two more properties that we would like our quantum AND gate to satisfy. The first is associativity,

$$a \wedge (b \wedge c) = (a \wedge b) \wedge c.$$

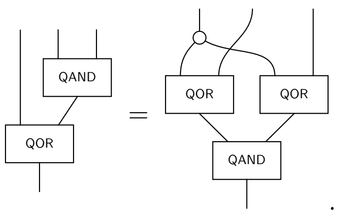
This can be expressed graphically, as follows:

 (30)

The second is distributivity,

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c),$$

graphically,

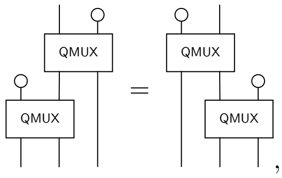


The GHZ/W-pair described at the end of the previous section does not satisfy these conditions.[5] This implies that they are not provable from the

[4]Haskell commands `isGHZWPair s a`, `satisfiesEtaIdempotence s a`, `satisfies-TickedBialge s a`, `satisfiesBialge s a` and `inducedAndSatisfiesAbsorption s a` where `s = discreteSCFAProduct [1,1,2,2]` and `a = minACFA (carrier s)[(0,1),(2,4),(3,5)]`.

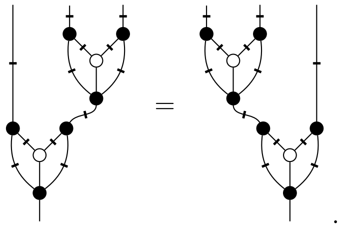
[5]Haskell commands `isAssociative (carrier s)(logicalAnd s a)` and `induced-AndSatisfiesDistributivity s a` for `s` and `a` as in Footnote 4.

GHZ/W-axioms, eta idempotence, ticked and non-ticked bialgebraic commutation. Moreover, it can be shown that associativity is equivalent to



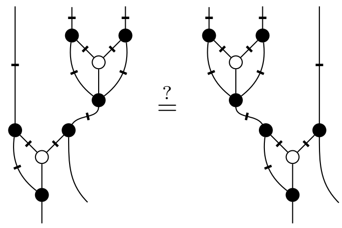
which is at least not easily provable by additionally assuming Equation (29) from the previous section.

In order to find another graphical axiom from which associativity could be derived, Equation (30) was rewritten using Theorem 91, obtaining the condition



(31)

In this equation, connecting wires were then “cut” in turn to obtain subgraphs that could serve as new graphical axioms. For instance, the equality



would imply Equation (31) and could maybe in turn be explained by an equation between subgraphs that were obtained by cutting wires. Unfortunately, cutting any of the wires in Equation (31) like this actually destroys equality for the standard GHZ/W-pair. This suggests that there may not be a simpler, equational graphical axiom from which associativity of $QAND$ can be proved.

Chapter 6

Conclusions and further work

We have roughly followed three strands in this thesis. In the first, we abstractly established the canonicity of the recent notion of anti-speciality for Frobenius algebras and showed how speciality and anti-speciality in **FdHilb** correspond to maximality resp. minimality of the rank of the induced loop. Using this result, we obtained a new classification of FAs on $\mathbb{C}^2$, which we interpreted as to some extent giving a new, abstract explanation of the existence of two types of genuine tripartite entanglement for qubits.

In the second strand, we looked at the non-standard model **FRel** of categorical quantum mechanics. We first used a Haskell implementation to find all commutative Frobenius algebras on the two- and three-element sets and pointed out several counterexamples that show that analogues of certain results known to hold for the GHZ/W-calculus in **FdHilb** do not generalise. Next, we gave a new classification of special commutative Frobenius algebras and proved that every CFA with distinct, singleton unit and counit contains an ACFA. Finally, we used the theory developed to classify GHZ/W-pairs in terms of the group decomposition of the constituent SCFA and the anti-unit and the cup of the constituent ACFA.

In the third and final, we used the GHZ/W-language to define an analogue of the Boolean AND gate. We identified several graphical axioms that can be used on top of those of the GHZ/W-calculus to prove some properties this gate and showed that it fails to satisfy several properties that would normally be expected to hold. We came to the conclusion that introducing additional (equational) axioms is unlikely to rectify this situation.

One of the main conclusions to be drawn from this thesis is that anti-speciality, just like speciality, seems to be a crucial notion in the study of Frobenius algebras. This stems from its connection to multipartite entanglement in **FdHilb** ([15] and Chapter 3 above), the canonicity of (minimal) ACFAs in **Rel** (Section 4.4.1) but also abstractly from the fact that a Frobenius algebra has a disconnected loop iff it is anti-special (Theorem 53). For these reasons, while the GHZ/W-axioms may change (see below), I expect anti-special Frobenius algebras to become an even more important player in categorical quantum mechanics.

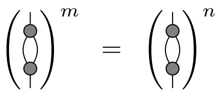
Another conclusion to be drawn is that the current form of the GHZ/W-calculus may not be the final answer to the problem of obtaining an abstract, structural understanding of multipartite entanglement. This is suggested by several results in this thesis:

- The counterexamples exhibited by the Haskell implementation (cf. Section 4.2) show that the proof that the GHZ/W-calculus refines the Z/X-calculus does not work for the general case and that similarly the current axiomatisation does not in general establish a bijective correspondence between SCFAs and ACFAs.

- Theorem 82 shows that in (**F**)**Rel**, the GHZ/W-axioms only constrain two of the groups of the SCFA and the anti-unit and cup of the ACFA. As the size of the carrier is increased, these constraints fail to fully relate the structure of the two constituent Frobenius algebras, a fact exemplified by the existence of SCFAs which form a GHZ/W-pair with many different ACFAs (Example 85).

- Theorems 57 and 53 establish that the speciality and anti-speciality conditions for Frobenius algebras in **FdHilb** correspond precisely to maximality resp. minimality of the rank of the induced loop. While the GHZ/W-calculus works well for qubits, where every loop rank is either maximal or minimal, this seems to indicate that it may not be as successful in higher dimensions, where intermediate ranks exist.

Following the above discussion, an obvious next step is to try to classify the Frobenius algebras with intermediate loop ranks in **FdHilb**. It appears that this will be more difficult to formulate in the language of †-SMCs, as intermediate ranks are unlikely to enjoy similarly canonical properties (disconnected/invertible) as minimal/maximal ones. A first starting point might be the observation from Section 4.2 that the non-special or -anti-special CFAs on the two- and three-element sets in **FRel** satisfy



for some distinct values m and n.

Another interesting next step would be to extend the theory developed in the first strand of this thesis in such a way that it can give an autonomous, structural explanation of the existence of two different types of genuine tripartite entanglement. This would involve establishing a correspondence between tripartite qubit states and Frobenius algebras without using the SLOCC classification [24] by Dür et al.

Bibliography

[1] GHC: The Glasgow Haskell compiler, August 2010. Available online at `http://www.haskell.org/ghc`.

[2] 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 2004*, pages 415–425. IEEE Computer Society, 2004. `arXiv:quant-ph/0402130v5`.

[3] S. Abramsky and N. Tzevelekos. Introduction to categories and categorical logic. In Bob Coecke, editor, *New Structures for Physics*. Springer-Verlag, 2010. Available online at `http://web.comlab.ox.ac.uk/people/Bob.Coecke/AbrNikos.pdf`.

[4] C. H. Bennett, S. Popescu, D. Rohrlich, J. A. Smolin, and A. V. Thapliyal. Exact and asymptotic measures of multipartite pure-state entanglement. *Phys. Rev. A*, 63(1):012307, Dec 2000.

[5] J. Biamonte, S. Clark, M. Williamson, and V. Vedral. The Quantum Theory of Information and Computation. *Oxford Graduate Course*, TT2010. Course contents available at `www.comlab.ox.ac.uk/activities/quantum/course/`.

[6] A. Carboni and R. F. C. Walters. Cartesian bicategories I. *Journal of Pure and Applied Algebra*, 49:11–32, 1987.

[7] Aurelio Carboni. Matrices, relations, and group representations. *Journal of Algebra*, 136(2):497 – 529, 1991.

[8] B. Coecke. Kindergarten quantum mechanics – lecture notes. In *Quantum Theory: Reconsiderations of the Foundations III*, pages 81–98. AIP Press, 2005.

[9] B. Coecke. Introducing categories to the practising physicist. In G. Sica, editor, *What is Category Theory?*, volume 3 of *Advanced Studies in Mathematics and Logic*, pages 45–74. Polimetrica Publisher, 2006.

[10] B. Coecke. Quantum picturalism. *Contemporary Physics*, 51:59–83, January 2010. `arXiv:0908.1787v1 [quant-ph]`.

[11] B. Coecke and R. W. Duncan. Interacting quantum observables: Categorical algebra and diagrammatics. In *Automata, Languages and Programming, ICALP'08,* number 5126 in Lecture Notes in Computer Science, pages

298–310. Springer-Verlag, 2008. Extended version: `arXiv:0906.4725v1 [quant-ph]`.

[12] B. Coecke, B. Edwards, and R. W. Spekkens. Phase groups and the origin of non-locality for qubits. `arXiv:1003.5005v1 [quant-ph]`, March 2010.

[13] B. Coecke and W. Edwards. Toy quantum categories. In B. Coecke and P. Panangaden, editors, *Proceedings of the 2008 QPL-DCM Workshop*, pages 25–35. Springer-Verlag, 2008. `arXiv:0808.1037v1 [quant-ph]`.

[14] B. Coecke and A. Kissinger. The compositional structure of multipartite quantum entanglement. `arXiv:1002.2540v2 [quant-ph]`, August 2010.

[15] B. Coecke and A. Kissinger. The compositional structure of multipartite quantum entanglement. `arXiv:1002.2540v1 [quant-ph]`, February 2010.

[16] B. Coecke and E. O. Paquette. POVMs and Naimark's theorem without sums. In *Proc. of the 4th International Workshop on Quantum Programming Languages*, 2006.

[17] B. Coecke, E. O. Paquette, and D. Pavlovic. Classical and quantum structuralism. In I. Mackie and S. Gay, editors, *Semantic Techniques in Quantum Computation*, pages 29–69. Cambridge University Press, February 2010. `arXiv:0904.1997v2 [quant-ph]`.

[18] B. Coecke and D. Pavlovic. Quantum measurements without sums. In G. Chen, L. Kauffmann, and S. Lomonaco, editors, *The Mathematics of Quantum Computation and Quantum Technology*, pages 567–604. Chapman and Hall/CRC, 2007. `arXiv:quant-ph/0608035v2`.

[19] B. Coecke, D. Pavlovic, and J. Vicary. A new description of orthogonal bases. *Mathematical Structures in Computer Science*, 13:08100812, 2008.

[20] V. Coffman, J. Kundu, and W. K. Wootters. Distributed entanglement. *Phys. Rev. A*, 61(5):052306, April 2000.

[21] E. D'Hondt and P. Panangaden. The computational power of the W and GHZ states. *Journ. Quantum Inf. and Comp.*, 6(2):173–183, 2005.

[22] L. Dixon, R. Duncan, and A. Kissinger. `quantomatic`. `http://dream.inf.ed.ac.uk/projects/quantomatic`.

[23] R. Duncan and S. Perdrix. *Automata, Languages and Programming*, volume 6199 of *Lecture Notes in Computer Science*, chapter Rewriting Measurement-Based Quantum Computations with Generalised Flow, pages 285–296. Springer Berlin / Heidelberg, 2010.

[24] W. Dür, G. Vidal, and J. I. Cirac. Three qubits can be entangled in two inequivalent ways. *Phys. Rev. A*, 62(6):062314, November 2000.

[25] J. Evans, R. W. Duncan, A. Lang, and P. Panangaden. Classifying all mutually unbiased bases in **Rel**. `arXiv:0909.4453v2 [quant-ph]`, September 2009.

[26] M. Żukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert. "Event-ready-detectors" Bell experiment via entanglement swapping. *Phys. Rev. Lett.*, 71(26):4287–4290, Dec 1993.

[27] A. Joyal and R. Street. The geometry of tensor calculus I. *Advances in Mathematics*, 88:55–113, 1991.

[28] G. M. Kelly and M. L. Laplaza. Coherence for compact closed categories. *Journal of pure and applied algebra*, 19:193–219, December 1980.

[29] A. Kissinger. Entanglement, monoidal categories and Frobenius algebras. Transfer report from the status of probationary research student to DPhil student at the Computing Laboratory, University of Oxford, January 2010.

[30] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, 2001.

[31] J. Kock. *Frobenius Algebras and 2D Topological Quantum Field Theories*, volume 59 of *London Mathematical Society Student Texts*. Cambridge University Press, 2003.

[32] S. Lack. Composing PROPs. *Theory and Applications of Categories*, 13(9):147–163, 2004.

[33] L. Lamata, J. León, D. Salgado, and E. Solano. Inductive classification of multipartite entanglement under stochastic local operations and classical communication. *Phys. Rev. A*, 74(5):052336, Nov 2006.

[34] L Lamata, J Leon, D Salgado, and E Solano. Inductive entanglement classification of four qubits under SLOCC. *Physical Review A*, 75(2):022318, 2007.

[35] S. Mac Lane. *Categories for the Working Mathematician (Graduate Texts in Mathematics)*. Springer, 2nd edition, September 1998.

[36] N. D. Mermin. *Quantum Computer Science*. Cambr, 2007.

[37] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[38] A. K. Pati and S. L. Braunstein. Impossibility of deleting an unknown quantum state. *Nature*, 404(6774):164–165, March 2000.

[39] D. Pavlovic. Quantum and classical structures in nondeterministic computation. In *Proceedings of the 3rd International Symposium on Quantum Interaction, Lecture Notes In Artificial Intelligence* **5494**, pages 143–157. Springer-Verlag, 2009. `arXiv:0812.2266v3 [quant-ph]`.

[40] P. Selinger. A survey of graphical languages for monoidal categories. In B. Coecke, editor, *New Structures for Physics*, pages 275–337. Springer-Verlag, 2009.

[41] R. W. Spekkens. In defense of the epistemic view of quantum states: a toy theory. *Phys. Rev. A*, 75(3):032110, March 2007.

[42] J. von Neumann. *Mathematische Grundlagen der Quantenmechanik*. Springer-Verlag, 1932. English translation: *Mathematical Foundations of Quantum Mechanics*, Princeton University Press, 1996.

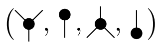
[43] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, October 1982.

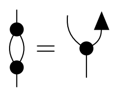
Appendix A

Auxiliary results and proofs

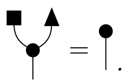
A.1 ... for Chapter 3

The following unpublished result is due to Bob Coecke and Aleks Kissinger and has been transmitted by personal communication. The reason it is included here is that its proof inspired Theorem 57 in Chapter 3.

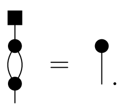
Theorem 102. *If the loop of a commutative Frobenius algebra*  *on* $\mathbb{C}^D$ *in* **FdHilb** *has (full) rank* D*, then it is equal to a* phase. *That is,*



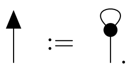
for some  $: \mathbb{C} \to \mathbb{C}^D$ *such that there exists*  $: \mathbb{C} \to \mathbb{C}^D$ *for which*



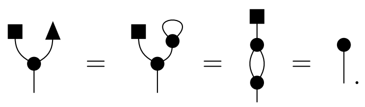
Proof. When the loop has full rank, it is invertible so there exists  $: \mathbb{C} \to \mathbb{C}^D$ such that



Set



Then, by the normal form theorem for CFA-morphisms (Theorem 27),

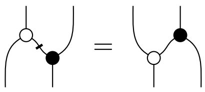


□

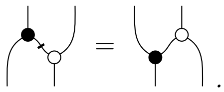
A.2 ... for Chapter 5

This section contains most of the graphical proofs for Chapter 5 and some associated lemmas. Throughout, we will often make (unmentioned) use of

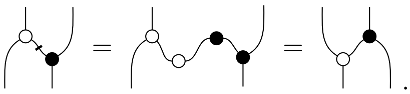
Lemma 103. *For any GHZ/W-pair, the tick ⟊ allows us to "slide" dots of different colours past each other. That is,*



and



Proof. Using Axiom 1. of the GHZ/W-axioms, we have



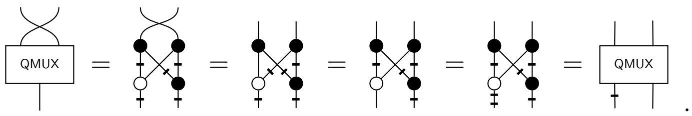
A similar proof can be used for the case where black and white nodes are swapped. ☐

Recall also that we are assuming that the dimension $\bigcirc : I \to I$ has an inverse $\ominus$ (Assumption 86).

A.2.1 Correctness and commutativity

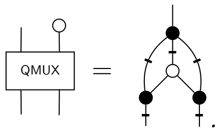
Unlike the presentation given in Chapter 5, we first prove the commutativity of $QAND$. This will simplify the proof of its correctness (Theorem 90).

Proof of Proposition 92. Using the fact that the tick is self-inverse, axiom 2. of the GHZ/W-axioms and Corollary 47,

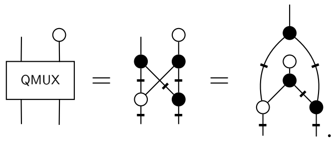


☐

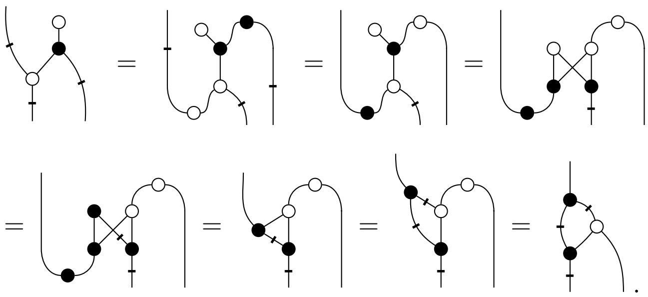
Lemma 104. *If a GHZ/W-pair satisfies bialgebraic commutation, then we have*



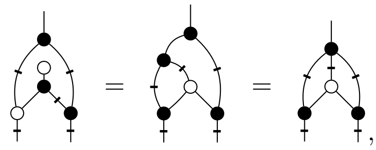
Proof. First observe that we can "stretch wires" to get



Next, using the bialgebra law,

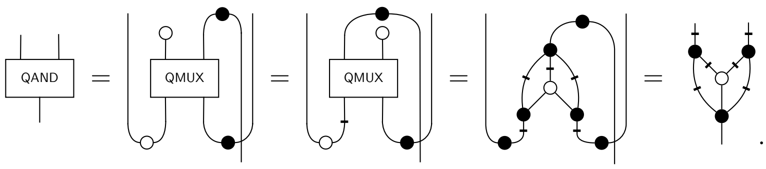


This implies



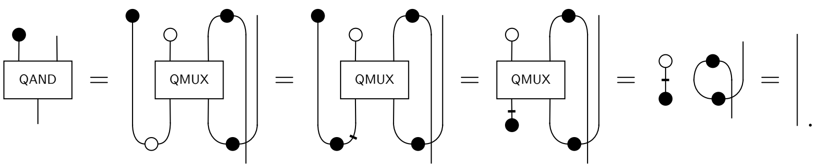
as required. ☐

Proof of Theorem 91. Using Proposition 92 and Lemma 104,

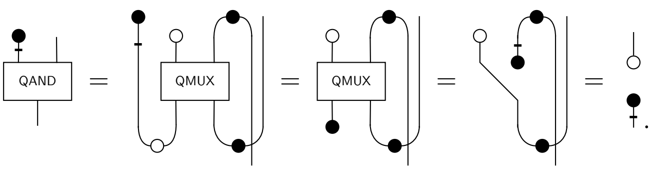


Since all involved Frobenius algebras are commutative, the symmetry of this graphical representation implies that $QAND$ is, too. ☐

Proof of Theorem 90. Using Theorem 52, the graphical lemmas from section 2.3.7, commutativity and compactness,



Similarly,

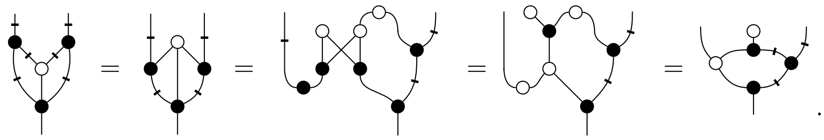


The remaining two equations follow from the commutativity of $QAND$ (Theorem 91).

The fact that $QAND$ behaves like Boolean $\wedge$ with respect to  and  follows from Assumption 86 and Lemma 50. ☐

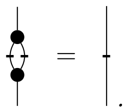
A.2.2 A third truth value

Proof of Lemma 96.

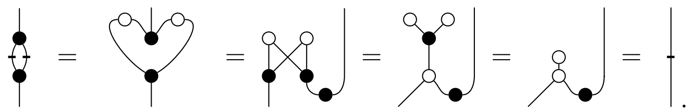


The result follows by Theorem 91. □

Lemma 105. *If a GHZ/W-pair satisfies bialgebraic commutation and eta idempotence, then*

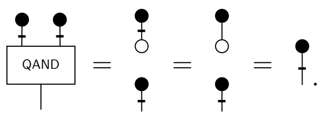


Proof.

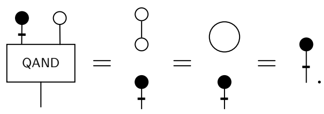


□

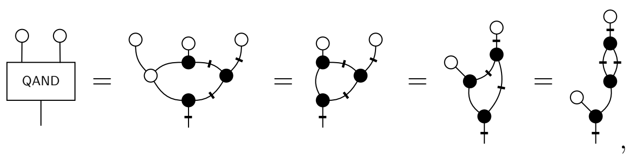
Proof of Proposition 95. Any case involving  follows immediately from Theorem 90. Next, by Lemmas 49, 50 and Corollary 47,



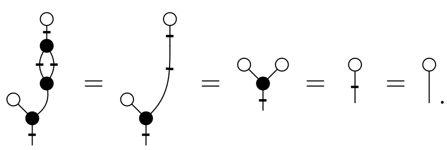
Similarly, by Lemma 94,



Finally, by Lemma 96,



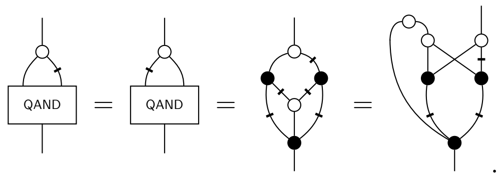
and, by Lemma 105,



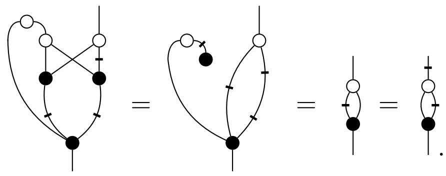
□

A.2.3 Complements

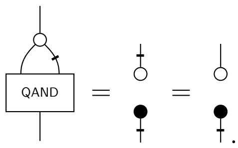
Proof of Theorem 98. By the commutativity of $QAND$ and then by "pulling up" the middle white dot,



Now, by the ticked bialgebra law,



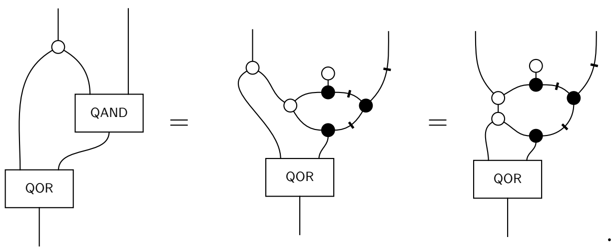
Hence, by Lemma 99, since the bialgebra law holds,



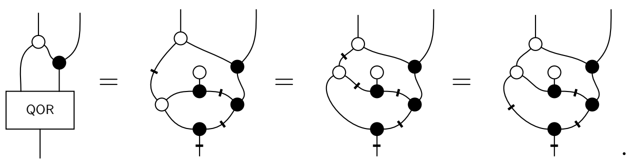
□

A.2.4 Absorption

Proof of Theorem 101. First, by Lemma 96 and the Frobenius condition,

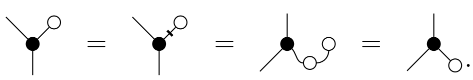


Now, by the definition of QOR, Lemmas 96 and 103,

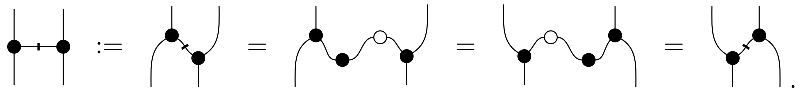


(32)

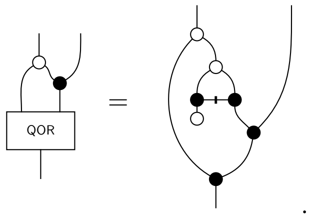
Next, by Lemma 49,



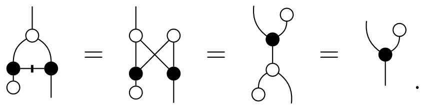
Also, it is not difficult to see that



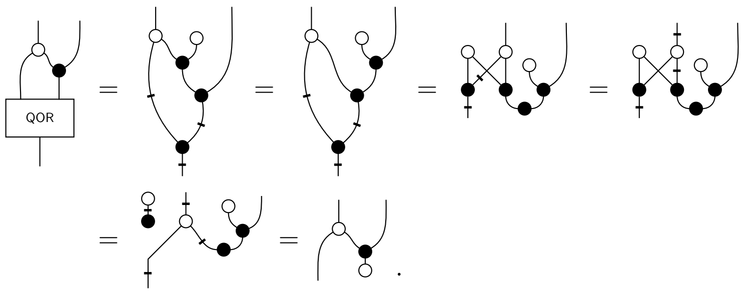
Together with (co-)associativity, these observations let us rewrite Equation (32) as



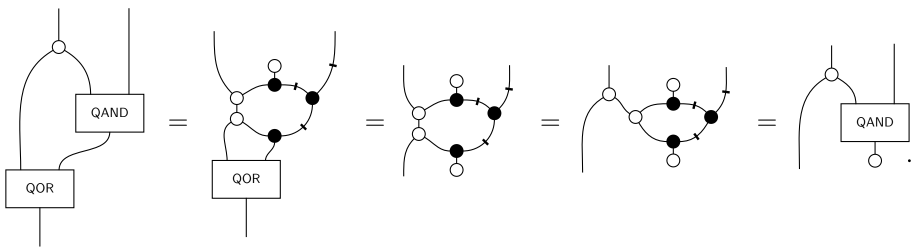
Now, by the bialgebra law and unitality,



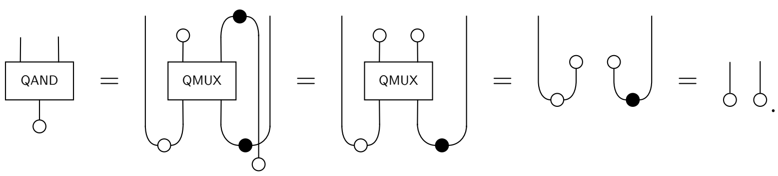
This implies by associativity, the ticked bialgebra law and Lemma 50 upside down (Corollary 47),



Hence, by putting the above together, the Frobenius condition and Lemma 96,



The result now follows from the derivation



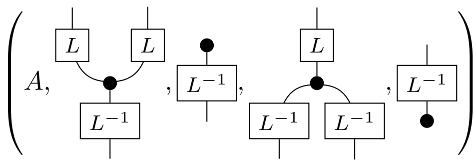
□

Appendix B

List of CFAs on 2 and 3 in (F)Rel

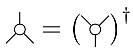
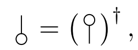
This appendix lists all commutative Frobenius Algebras on the two- and three-element sets $\mathbf{2} = \{0, 1\}$ and $\mathbf{3} = \{0, 1, 2\}$ in the category **FRel** of finite sets and binary relations. The results were obtained by using a Haskell program to exhaustively check all possibilities for multiplication, unit, comultiplication and counit, as described in Section 4.2. The Haskell code is listed in Appendix C.

Not surprisingly, if $\big(A, \ , \ , \ , \ \big)$ is a Frobenius algebra of some kind (commutative, dagger, special or anti-special) and $L : A \to A$ is a unitary morphism (i.e. invertible with $L^{-1} = L^\dagger$) , then



is a Frobenius algebra of the same kind. In **FRel**, unitary morphisms are precisely the (graphs of) invertible functions. For this reason, the Frobenius algebras in this appendix are only listed up to a permutation of the carrier elements.

Special commutative Frobenius algebras are displayed as white dots ○. Subscripts indicate the group partitioning given by Theorem 66. Sums of the form $\sum k_i$ suffice for these subscripts because on $k \leq 3$ elements, all abelian groups are isomorphic to the cyclic group $\mathbb{Z}_k$. For example, $\circ_{1+2}$ stands for the SCFA $\mathbb{Z}_1 + \mathbb{Z}_2$. Since by Theorem 66 all SCFAs in **Rel** are dagger, that is,

 and 

we furthermore only list the monoid parts of special commutative Frobenius algebras.

Anti-special CFAs are displayed as black dots ● and CFAs that are neither special nor anti-special are displayed as grey dots. Finally, we use the Dirac notation for **FRel** as described in Section 2.3.3.

B.1 Commutative Frobenius algebras on $\{0, 1\}$

There are nine commutative Frobenius Algebras on $\{0, 1\}$. These Frobenius Algebras can be divided into special commutative ones (SCFAs), anti-special commutative ones (ACFAs) and all others.

B.1.1 SCFAs

Up to swapping 0's and 1's, there are two SCFAs. Both are dagger so we only list the respective multiplications and units:

1.

$$_{1+1} = |0\rangle\langle 00| + |1\rangle\langle 11| \qquad\qquad _{1+1} = |0\rangle + |1\rangle.$$

2.

$$_{2} = |0\rangle\langle 00| + |1\rangle\langle 01| + |1\rangle\langle 10| + |0\rangle\langle 11| \qquad\qquad _{2} = |0\rangle.$$

B.1.2 ACFAs

Again up to swapping 0's and 1's, there are two ACFAs:

1.

$$_{a} = |0\rangle\langle 01| + |0\rangle\langle 10| + |1\rangle\langle 11| \qquad\qquad _{a} = |1\rangle$$
$$_{a} = |10\rangle\langle 1| + |01\rangle\langle 1| + |00\rangle\langle 0| \qquad\qquad _{a} = \langle 0|$$

2.

$$_{b} = |0\rangle\langle 01| + |0\rangle\langle 10| + |1\rangle\langle 11| + |0\rangle\langle 00| + |1\rangle\langle 00| \qquad\qquad _{b} = |1\rangle$$
$$_{b} = {}^{\dagger} \qquad\qquad _{b} = {}^{\dagger}.$$

B.1.3 Others

There is one other type of Frobenius Algebra on $\{0, 1\}$:

$$= |1\rangle\langle 00| + |0\rangle\langle 01| + |0\rangle\langle 10| + |1\rangle\langle 11| \qquad\qquad = |1\rangle$$
$$= |00\rangle\langle 0| + |01\rangle\langle 1| + |10\rangle\langle 1| + |11\rangle\langle 0| \qquad\qquad = \langle 0|.$$

B.2 Commutative Frobenius algebras on $\{0, 1, 2\}$

There are 130 commutative Frobenius Algebras on the three-element set.

B.2.1 SCFAs

There are 10 SCFAs, all of which are dagger. Up to a permutation of the elements of $\{0, 1, 2\}$, there are only three different types:

1.

$$
\begin{aligned}
{}_{1+2} &= |1\rangle\langle 22| + |2\rangle\langle 21| + |0\rangle\langle 00| + |2\rangle\langle 12| + |1\rangle\langle 11| \\
{}_{1+2} &= |0\rangle + |1\rangle.
\end{aligned}
$$

2.

$$
\begin{aligned}
{}_{1+1+1} &= |0\rangle\langle 00| + |1\rangle\langle 11| + |2\rangle\langle 22| \\
{}_{1+1+1} &= |0\rangle + |1\rangle + |2\rangle.
\end{aligned}
$$

3.

$$
\begin{aligned}
{}_{3} &= |0\rangle\langle 21| + |2\rangle\langle 20| + |1\rangle\langle 22| + |2\rangle\langle 11| + |0\rangle\langle 12| \\
&\quad + |1\rangle\langle 10| + |2\rangle\langle 02| + |1\rangle\langle 01| + |0\rangle\langle 00| \\
{}_{3} &= |0\rangle.
\end{aligned}
$$

B.2.2 ACFAs

There are 66 ACFAs, of which 30 are dagger. There are 13 different types:

ACFAs with unit $|0\rangle$ and counit $|1\rangle$:

1.

$$
\begin{aligned}
{}_{1} &= |0\rangle\langle 00| + |1\rangle\langle 01| + |2\rangle\langle 02| + |1\rangle\langle 22| + |2\rangle\langle 20| + |1\rangle\langle 10| \\
{}_{1} &= |01\rangle\langle 0| + |10\rangle\langle 0| + |22\rangle\langle 0| + |12\rangle\langle 2| + |21\rangle\langle 2| + |11\rangle\langle 1|.
\end{aligned}
$$

2.

$$
{}_{2} = {}_{1} + |2\rangle\langle 22| \qquad\qquad {}_{2} = {}_{1} + |22\rangle\langle 2|.
$$

3.

$$
\begin{aligned}
{}_{3} &= {}_{2} + |2\rangle\langle 12| + |0\rangle\langle 11| + |0\rangle\langle 22| + |2\rangle\langle 21| \\
{}_{3} &= {}_{2} + |22\rangle\langle 1| + |00\rangle\langle 1| + |02\rangle\langle 2| + |20\rangle\langle 2|.
\end{aligned}
$$

4.

$$
\begin{aligned}
{}_{4} &= {}_{2} + |2\rangle\langle 12| + |2\rangle\langle 11| + |0\rangle\langle 12| + |0\rangle\langle 22| + |2\rangle\langle 21| + |0\rangle\langle 21| \\
{}_{4} &= {}_{2} + |22\rangle\langle 1| + |20\rangle\langle 1| + |02\rangle\langle 1| + |02\rangle\langle 2| + |20\rangle\langle 2| + |00\rangle\langle 2|.
\end{aligned}
$$

5.

$$
{}_{5} = {}_{4} + |0\rangle\langle 11| \qquad\qquad {}_{5} = {}_{4} + |00\rangle\langle 1|.
$$

†-ACFAs

There are 7 types of †-ACFA. All their units are singleton, hence we can take $\top = |0\rangle$ throughout. Under these assumptions, each †-ACFA is completely determined by the multiplication $\curlyvee$.

1. $\curlyvee_6 = \curlyvee_3$.
2. $\curlyvee_7 = \curlyvee_6 + |1\rangle\langle 11|$.
3. $$\begin{aligned}\curlyvee_8 = {} & |2\rangle\langle 21| + |2\rangle\langle 20| + |0\rangle\langle 21| + |2\rangle\langle 22| + |1\rangle\langle 21| + |1\rangle\langle 01| + |2\rangle\langle 02| \\ & + |0\rangle\langle 00| + |1\rangle\langle 11| + |2\rangle\langle 12| + |1\rangle\langle 10| + |0\rangle\langle 12| + |1\rangle\langle 12|.\end{aligned}$$
4. $\curlyvee_9 = \curlyvee_1 + |0\rangle\langle 11| + |2\rangle\langle 11| + |1\rangle\langle 12| + |2\rangle\langle 12| + |1\rangle\langle 21| + |2\rangle\langle 21| + |0\rangle\langle 22|$.
5. $\curlyvee_{10} = \curlyvee_9 + |2\rangle\langle 22|$.
6. $$\begin{aligned}\curlyvee_{11} = \curlyvee_2 & + |0\rangle\langle 12| + |1\rangle\langle 12| + |1\rangle\langle 11| + |2\rangle\langle 12| \\ & + |2\rangle\langle 11| + |0\rangle\langle 21| + |1\rangle\langle 21| + |2\rangle\langle 21|.\end{aligned}$$
7. $\curlyvee_{12} = \curlyvee_3 + |1\rangle\langle 11| + |2\rangle\langle 11| + |1\rangle\langle 12| + |1\rangle\langle 21|$.

One More

$$\begin{aligned}\curlyvee_{13} &= \curlyvee_8 + |2\rangle\langle 11| & \top_{13} &= |0\rangle \\ \curlywedge_{13} &= \curlywedge_8 + |22\rangle\langle 1| & \perp_{13} &= \top_{13}^{\dagger}.\end{aligned}$$

B.2.3 Others

The remaining 54 CFAs fall into 9 different categories.

1. $$\begin{aligned}\curlyvee &= |0\rangle\langle 00| + |2\rangle\langle 02| + |1\rangle\langle 11| + |2\rangle\langle 20| & \top &= |0\rangle + |1\rangle \\ \curlywedge &= |02\rangle\langle 0| + |20\rangle\langle 0| + |11\rangle\langle 1| + |22\rangle\langle 2| & \perp &= \langle 1| + \langle 2|.\end{aligned}$$
2. $$\begin{aligned}\curlyvee &= |2\rangle\langle 20| + |0\rangle\langle 22| + |1\rangle\langle 11| + |0\rangle\langle 00| + |2\rangle\langle 02| & \top &= |0\rangle + |1\rangle \\ \curlywedge &= |22\rangle\langle 2| + |00\rangle\langle 2| + |11\rangle\langle 1| + |02\rangle\langle 0| + |20\rangle\langle 0| & \perp &= \langle 1| + \langle 2|.\end{aligned}$$
3. $$\begin{aligned}\curlyvee &= |1\rangle\langle 22| + |2\rangle\langle 20| + |0\rangle\langle 21| + |2\rangle\langle 02| + |0\rangle\langle 00| \\ &\quad + |1\rangle\langle 01| + |2\rangle\langle 11| + |0\rangle\langle 12| + |1\rangle\langle 10| \\ \top &= |0\rangle \\ \curlywedge &= |12\rangle\langle 2| + |21\rangle\langle 2| + |00\rangle\langle 2| + |22\rangle\langle 0| + |01\rangle\langle 0| \\ &\quad + |10\rangle\langle 0| + |20\rangle\langle 1| + |02\rangle\langle 1| + |11\rangle\langle 1| \\ \perp &= \langle 1|.\end{aligned}$$

4.

$$
\begin{aligned}
&= |0\rangle\langle 11| + |1\rangle\langle 10| + |2\rangle\langle 12| + |1\rangle\langle 22| + |2\rangle\langle 21| \\
&\quad + |0\rangle\langle 22| + |2\rangle\langle 20| + |1\rangle\langle 01| + |2\rangle\langle 02| + |0\rangle\langle 00| \\
&= |0\rangle \\
&= {}^{\dagger} \\
&= {}^{\dagger}.
\end{aligned}
$$

5.

$$
\begin{aligned}
&= |1\rangle\langle 10| + |2\rangle\langle 12| + |0\rangle\langle 11| + |1\rangle\langle 22| + |2\rangle\langle 20| \\
&\quad + |0\rangle\langle 22| + |2\rangle\langle 21| + |0\rangle\langle 00| + |1\rangle\langle 01| + |2\rangle\langle 02| \\
&= |0\rangle \\
&= |11\rangle\langle 1| + |22\rangle\langle 1| + |00\rangle\langle 1| + |12\rangle\langle 2| + |21\rangle\langle 2| \\
&\quad + |02\rangle\langle 2| + |20\rangle\langle 2| + |01\rangle\langle 0| + |10\rangle\langle 0| + |22\rangle\langle 0| \\
&= \langle 1|.
\end{aligned}
$$

6.

$$
\begin{aligned}
&= |1\rangle\langle 10| + |2\rangle\langle 12| + |2\rangle\langle 11| + |0\rangle\langle 12| + |0\rangle\langle 11| + |1\rangle\langle 22| + |2\rangle\langle 20| \\
&\quad + |0\rangle\langle 22| + |2\rangle\langle 21| + |0\rangle\langle 21| + |0\rangle\langle 00| + |1\rangle\langle 01| + |2\rangle\langle 02| \\
&= |0\rangle \\
&= |11\rangle\langle 1| + |22\rangle\langle 1| + |20\rangle\langle 1| + |02\rangle\langle 1| + |00\rangle\langle 1| + |12\rangle\langle 2| + |21\rangle\langle 2| \\
&\quad + |02\rangle\langle 2| + |20\rangle\langle 2| + |00\rangle\langle 2| + |01\rangle\langle 0| + |10\rangle\langle 0| + |22\rangle\langle 0| \\
&= \langle 1|.
\end{aligned}
$$

7.

$$
\begin{aligned}
&= |0\rangle\langle 11| + |1\rangle\langle 10| + |2\rangle\langle 11| + |1\rangle\langle 12| + |1\rangle\langle 01| + |0\rangle\langle 00| \\
&\quad + |2\rangle\langle 02| + |1\rangle\langle 21| + |0\rangle\langle 22| + |2\rangle\langle 20| + |2\rangle\langle 22| \\
&= |0\rangle \\
&= {}^{\dagger} \\
&= {}^{\dagger}.
\end{aligned}
$$

8.

$$
\begin{aligned}
&= |0\rangle\langle 00| + |1\rangle\langle 11| + |2\rangle\langle 12| + |1\rangle\langle 22| + |2\rangle\langle 21| + |2\rangle\langle 22| \\
&= |0\rangle + |1\rangle \\
&= {}^{\dagger} \\
&= {}^{\dagger}.
\end{aligned}
$$

9.

 $= |0\rangle\langle 00| + |2\rangle\langle 02| + |1\rangle\langle 01| + |0\rangle\langle 12| + |1\rangle\langle 10| + |1\rangle\langle 11|$
$+ |2\rangle\langle 12| + |2\rangle\langle 11| + |2\rangle\langle 20| + |0\rangle\langle 21| + |1\rangle\langle 22| + |2\rangle\langle 21|$

 $= |0\rangle$

 $= |00\rangle\langle 0| + |21\rangle\langle 0| + |12\rangle\langle 0| + |01\rangle\langle 1| + |10\rangle\langle 1| + |12\rangle\langle 1|$
$+ |21\rangle\langle 1| + |22\rangle\langle 1| + |20\rangle\langle 2| + |02\rangle\langle 2| + |11\rangle\langle 2| + |22\rangle\langle 2|$

 $=$  †.

Appendix C

Code listings

This appendix lists a bare-bones version of the Haskell code that was used to obtain the computational results in this thesis. The implementation and most of its results are described in Section 4.2.

```
import Data.List
import qualified Data.Set as Set

data I = X deriving (Eq, Ord, Show)
type BinRel a b = [(a, b)]

o :: (Eq a, Eq b, Eq c) => BinRel b c -> BinRel a b ->
     BinRel a c
o q p = nub [(x, z) | (x, y) <- p, (y', z) <- q, y ==
   y']

x :: BinRel a b -> BinRel x y -> BinRel (a, x) (b, y)
x p q = [((i, k), (j, l)) | (i, j) <- p, (k, l) <- q]

eval :: (Eq a, Eq b) => BinRel a b -> [a] -> [b]
eval p x = nub (map snd (filter ((flip elem x) . fst)
   p))

idBinRel :: [a] -> BinRel a a
idBinRel = map (\x -> (x, x))

alBinRel' :: [a] -> [b] -> [c] -> BinRel ((a, b), c) (
   a, (b, c))
alBinRel' a b c = [(((x, y), z), (x, (y, z))) | x <- a
   , y <- b, z <- c]

alBinRel :: [a] -> [b] -> [c] -> BinRel (a, (b, c)) ((
   a, b), c)
alBinRel a b c = [((x, (y, z)), ((x, y), z)) | x <- a,
    y <- b, z <- c]
```

```
lamBinRel :: [a] -> BinRel a (I, a)
lamBinRel = map (\x -> (x, (X, x)))

lamBinRel' :: [a] -> BinRel (I, a) a
lamBinRel' = map (\x -> ((X, x), x))

rhoBinRel :: [a] -> BinRel a (a, I)
rhoBinRel = map (\x -> (x, (x, X)))

rhoBinRel' :: [a] -> BinRel (a, I) a
rhoBinRel' = map (\x -> ((x, X), x))

sigBinRel :: [a] -> [b] -> BinRel (a, b) (b, a)
sigBinRel a b = [((x, y), (y, x)) | x <- a, y <- b]

eqBinRel :: (Ord a, Ord b) => BinRel a b -> BinRel a b
     -> Bool
eqBinRel p q = (Set.fromList p) == (Set.fromList q)

isMonoid :: Ord a => [a] -> BinRel (a, a) a -> BinRel
   I a -> Bool
isMonoid a mu eta
  = eqBinRel (mu `o` (((idBinRel a) `x` mu) `o`
     alBinRel' a a a)) (mu `o` (mu `x` idBinRel a)) --
      assoc.
    && eqBinRel (mu `o` ((eta `x` (idBinRel a)) `o`
       lamBinRel a)) (idBinRel a) -- left unitality
    && eqBinRel (mu `o` (((idBinRel a) `x` eta) `o`
       rhoBinRel a)) (idBinRel a) -- right unitality

isComonoid :: Ord a => [a] -> BinRel a (a, a) ->
   BinRel a I -> Bool
isComonoid a delta epsilon
  = eqBinRel ((epsilon `x` idBinRel a) `o` delta) (
     lamBinRel a)
    && eqBinRel (((idBinRel a) `x` epsilon) `o` delta)
        (rhoBinRel a)
    && eqBinRel ((alBinRel' a a a) `o` (delta `x`
       idBinRel a) `o` (delta)) (((idBinRel a) `x`
       delta)`o` delta)

dag :: BinRel a b -> BinRel b a
dag = map (\(x, y) -> (y, x))

isFrobeniusAlgebra :: Ord a => [a] -> BinRel (a, a) a
   -> BinRel I a -> BinRel a (a, a) -> BinRel a I ->
   Bool
isFrobeniusAlgebra a mu eta delta epsilon
  = (isMonoid a mu eta)
     && isComonoid a delta epsilon
```

```
     && eqBinRel ((mu `x` idBinRel a) `o` ((alBinRel a
         a a) `o` ((idBinRel a) `x` delta))) (delta `o
        ` mu)
     && eqBinRel (((idBinRel a) `x` mu) `o` ((alBinRel
        ' a a a) `o` (delta `x` idBinRel a))) (delta `
        o` mu)

isCFA :: Ord a => [a] -> BinRel (a, a) a -> BinRel I a
    -> BinRel a (a, a) -> BinRel a I -> Bool
isCFA a mu eta delta epsilon
  = eqBinRel (mu `o` sigBinRel a a) mu
    && eqBinRel ((sigBinRel a a) `o` delta) delta
    && isFrobeniusAlgebra a mu eta delta epsilon

isSCFA :: Ord a => [a] -> BinRel (a, a) a -> BinRel I
   a -> BinRel a (a, a) -> BinRel a I -> Bool
isSCFA a mu eta delta epsilon
  = eqBinRel (mu `o` delta) (idBinRel a)
    && isCFA a mu eta delta epsilon

isACFA :: Ord a => [a] -> BinRel (a, a) a -> BinRel I
   a -> BinRel a (a, a) -> BinRel a I -> Bool
isACFA a mu eta delta epsilon
  = eqBinRel ((epsilon `x` idBinRel a) `o` ((l `x` l)
     `o` ((eta `x` idBinRel a) `o` lamBinRel a)))
     (((((lamBinRel a) `o` l) `o` eta) `o` epsilon) `o
     ` l)
    && isCFA a mu eta delta epsilon
  where
    l = mu `o` delta

uncurry5 f (a, b, c, d, e) = f a b c d e

powerset :: [a] -> [[a]]
powerset [] = [[]]
powerset (x:xs)
  = xss ++ map (x:) xss
  where
    xss = powerset xs

cartProd as bs = [(a, b) | a <- as, b <- bs]

binRels a b = powerset (cartProd a b)

choose _ 0 = [[]]
choose xs n = [y : ys | y:xs' <- tails xs, ys <-
   choose (y:xs') (n - 1)]

symmetrize :: [[a]] -> [[a]]
symmetrize as = concat (map permutations as)
```

```
faFrom :: Eq a => [a] -> BinRel I (a, (a, a)) ->
   BinRel a I -> FrobeniusAlgebra a
faFrom a tri epsilon
  = (a, mu, eta, delta, epsilon)
  where
    mu = (rhoBinRel' a) `o` (((idBinRel a) `x` cup) `o
       ` ((alBinRel' a a a) `o` (delta `x` idBinRel a)
       ))
    delta = (rhoBinRel' (cartProd a a)) `o` (((
       idBinRel (cartProd a a)) `x` cup) `o` (alBinRel
       ' (cartProd a a) a a) `o` (((alBinRel a a a) `x
       ` idBinRel a) `o` ((tri `x` idBinRel a) `o` (
       lamBinRel a))))
    eta = ((rhoBinRel' a) `o` ((idBinRel a) `x`
       epsilon)) `o` cap
    cap = (lamBinRel' (cartProd a a)) `o` ((epsilon `x
       ` ((idBinRel a) `x` (idBinRel a))) `o` tri)
    cup = dag cap

symmetrizeBinRel :: BinRel a a -> BinRel a a
symmetrizeBinRel p = p ++ dag p

isCommutative :: (Ord a, Ord b) => BinRel (a, a) b ->
   Bool
isCommutative p = eqBinRel p (map (\((x, y), z) -> ((y
   , x), z)) p)

isAssociative :: (Ord a) => [a] -> BinRel (a, a) a ->
   Bool
isAssociative c p = eqBinRel (p `o` ((idBinRel c) `x`
   p)) (p `o` ((p `x` idBinRel c) `o` (alBinRel c c c)
   ))

minACFA :: Eq a => [a] -> BinRel a a ->
   FrobeniusAlgebra a
-- Pre: length p > 0
minACFA a p
  = (a, mu, [(X, eta)], delta, [(epsilon, X)])
  where
    p' = (idBinRel (a \\ ((uncurry (++)) . unzip) p))
       ++ symmetrizeBinRel p
    eta = head a
    epsilon = head [x | (y, x) <- p', y == eta]
    mu = nub ([((x, eta), x) | x <- a] ++ [((eta, x),
       x) | x <- a] ++ [(t, epsilon) | t <- p'])
    delta = nub ([(x, (y, z)) | ((y, z), x) <- p' `o`
       (mu `o` (p' `x` p'))])

discreteSCFA :: Integer -> FrobeniusAlgebra Integer
```

```
-- Returns the SCFA corresponding to Z_n
discreteSCFA n
  = (a, [((x, y), mod (x + y) n) | (x, y) <- cartProd
      a a], [(X, 0)], [(mod (x + y) n, (x, y)) | (x, y)
       <- cartProd a a], [(0, X)])
  where
    a = [0..(n - 1)]

extendFA :: Eq a => FrobeniusAlgebra a -> BinRel (a, a
   ) a -> FrobeniusAlgebra a
-- Extend the multiplication of the given fa by xt.
   Updates the comultiplication accordingly.
extendFA f xt
  = (carrier f, mu f ++ xt, eta f, delta f ++ ((xt `x`
       i f) `o `((al f) `o` (((i f) `x` cap f) `o` (ru
      f)))), epsilon f)

extendCFA :: Eq a => FrobeniusAlgebra a -> BinRel (a,
   a) a -> FrobeniusAlgebra a
extendCFA f xt = extendFA f (xt ++ map (\((x, y), z)
   -> ((y, x), z)) xt)

relabelFA :: (a -> a) -> FrobeniusAlgebra a ->
   FrobeniusAlgebra a
relabelFA f (a, mu, eta, delta, epsilon)
  = (map f a, map (\((x, y), z) -> ((f x, f y), f z))
      mu, map (\(X, x) -> (X, f x)) eta, map (\(x, (y,
      z)) -> (f x, (f y, f z))) delta, map (\(x, X) ->
      (f x, X)) epsilon)

biproductFA :: Eq a => FrobeniusAlgebra a ->
   FrobeniusAlgebra a -> FrobeniusAlgebra a
biproductFA (a, mu, eta, delta, epsilon) (a', mu', eta
   ', delta', epsilon')
  = (nub (a ++ a'), nub (mu ++ mu'), nub (eta ++ eta')
      , nub (delta ++ delta'), nub (epsilon ++ epsilon
      '))

discreteSCFAProduct :: [Integer] -> FrobeniusAlgebra
   Integer
-- Returns the SCFA corresponding to Z_n_1 + ... +
   Z_n_k
discreteSCFAProduct (n:ns) = foldl (\fa n' ->
   biproductFA fa (relabelFA (+((maximum (carrier fa))
    + 1)) (discreteSCFA n'))) (discreteSCFA n) ns

candidates :: Eq a => [a] -> [FrobeniusAlgebra a]
candidates a = [(a, mu, eta, delta, epsilon) | mu <-
   binRels (cartProd a a) a, eta <- binRels [X] a,
   delta <- binRels a (cartProd a a), epsilon <-
```

```
   binRels a [X]]

commutativeCandidates :: Eq a => [a] -> [
   FrobeniusAlgebra a]
commutativeCandidates a = [faFrom a tri epsilon | tri
   <- map (map (\[a, b, c] -> (X, (a, (b, c))))) (map
   symmetrize (powerset (choose a 3))), epsilon <-
   powerset (map (\x -> (x, X)) a)]

cfasOn a = filter (uncurry5 isCFA) (candidates a)
cfasOn' a = filter (uncurry5 isCFA) (
   commutativeCandidates a)

fasOn a = filter (uncurry5 isFrobeniusAlgebra) (
   candidates a)
cfas a = filter (uncurry5 isCFA) a
scfas a = filter (uncurry5 isSCFA) a
acfas a = filter (uncurry5 isACFA) a
nonSCFAorACFAs a = filter (not . (\fa -> (uncurry5
   isSCFA) fa || (uncurry5 isACFA) fa)) a

type FrobeniusAlgebra a = ([a], BinRel (a, a) a,
   BinRel I a, BinRel a (a, a), BinRel a I)

eqFrobeniusAlgebra :: Ord a => FrobeniusAlgebra a ->
   FrobeniusAlgebra a -> Bool
eqFrobeniusAlgebra a b
  = (carrier a) == carrier b
   && eqBinRel (mu a) (mu b)
   && eqBinRel (eta a) (eta b)
   && eqBinRel (delta a) (delta b)
   && eqBinRel (epsilon a) (epsilon b)

bijections :: [a] -> [b] -> [BinRel a b]
bijections as bs = map (zipWith (\a b -> (a, b)) as) (
   permutations bs)

--eqBinRelUpToPermutation :: (Ord a, Ord b) => [a] ->
   [b] -> BinRel a b -> BinRel a b -> Bool
eqBinRelUpToPermutation as bs p q = not (null [(b, b')
    | b <- bijections as as, b' <- bijections bs bs,
   eqBinRel (b' `o` p `o` b) q])

eqFrobeniusAlgebraUpToPermutation :: (Ord a, Ord b) =>
    FrobeniusAlgebra a -> FrobeniusAlgebra b -> Bool
eqFrobeniusAlgebraUpToPermutation a b
  = length as == length bs
   && or [eqFrobeniusAlgebra (as, (dag p) `o` ((mu b)
      `o` (p `x` p)), (dag p) `o` (eta b), ((dag p) `x
      ` (dag p)) `o` ((delta b) `o` p), (epsilon b) `o
```

```
      ‘ p) a | p <- bijections as bs]
  where
    as = carrier a
    bs = carrier b

quotients :: (a -> a -> Bool) -> [a] -> [[a]]
quotients f as
  = quotients’ f as []
  where
    quotients’ _ [] qs = qs
    quotients’ f (a : as) [] = quotients’ f as [[a]]
    quotients’ f (a : as) (q : qs)
      | f a (head q) = quotients’ f as ((a : q) : qs)
      | otherwise = quotients’ f as (q : (quotients’ f
          [a] qs))

al :: FrobeniusAlgebra a -> BinRel (a, (a, a)) ((a, a)
   , a)
al (a, _, _, _, _) = alBinRel a a a

al’ :: FrobeniusAlgebra a -> BinRel ((a, a), a) (a, (a
   , a))
al’ (a, _, _, _, _) = alBinRel’ a a a

i :: FrobeniusAlgebra a -> BinRel a a
i (a, _, _, _, _) = idBinRel a

lu :: FrobeniusAlgebra a -> BinRel a (I, a)
lu (a, _, _, _, _) = lamBinRel a

lu’ :: FrobeniusAlgebra a -> BinRel (I, a) a
lu’ (a, _, _, _, _) = lamBinRel’ a

ru :: FrobeniusAlgebra a -> BinRel a (a, I)
ru (a, _, _, _, _) = rhoBinRel a

ru’ :: FrobeniusAlgebra a -> BinRel (a, I) a
ru’ (a, _, _, _, _) = rhoBinRel’ a

carrier :: FrobeniusAlgebra a -> [a]
carrier (c, _, _, _, _) = c

mu :: Eq a => FrobeniusAlgebra a -> BinRel (a, a) a
mu (_, m, _, _, _) = m

eta :: Eq a => FrobeniusAlgebra a -> BinRel I a
eta (_, _, e, _, _) = e

delta :: Eq a => FrobeniusAlgebra a -> BinRel a (a, a)
delta (_, _, _, d, _) = d
```

```
epsilon :: Eq a => FrobeniusAlgebra a -> BinRel a I
epsilon (_, _, _, _, e) = e

cup :: Eq a => FrobeniusAlgebra a -> BinRel (a, a) I
cup fa = (epsilon fa) `o` (mu fa)

cap :: Eq a => FrobeniusAlgebra a -> BinRel I (a, a)
cap fa = (delta fa) `o` (eta fa)

circ :: Eq a => FrobeniusAlgebra a -> BinRel I I
circ fa = (cup fa) `o` (cap fa)

loop :: Eq a => FrobeniusAlgebra a -> BinRel a a
loop fa = (mu fa) `o` (delta fa)

lollie :: Eq a => FrobeniusAlgebra a -> BinRel I a
lollie fa = (mu fa) `o` (cap fa)

tri :: Eq a => FrobeniusAlgebra a -> BinRel I (a, (a,
   a))
tri fa = ((i fa) `x` delta fa) `o` cap fa

isDag :: Ord a => FrobeniusAlgebra a -> Bool
isDag fa = (eqBinRel (dag (mu fa)) (delta fa)) && (
   eqBinRel (dag (eta fa)) (epsilon fa))

tick :: Eq a => FrobeniusAlgebra a -> FrobeniusAlgebra
    a -> BinRel a a
tick a b = ((((lu' a) `o` ((cup a) `x` i a)) `o` (al a
   )) `o` ((i a) `x` cap b)) `o` (ru a)

qmux :: Eq a => FrobeniusAlgebra a -> FrobeniusAlgebra
    a -> BinRel (a, a) (a, a)
qmux scfa acfa
  = (t `x` t) `o` (((mu scfa) `x` (mu acfa)) `o` (((t
     `x` i scfa) `x` (t `x` t)) `o` ((middleSwap a a a
     ) `o` ((delta acfa) `x` (delta acfa)))))
  where
    a = carrier scfa
    t = tick scfa acfa

logicalAnd :: Eq a => FrobeniusAlgebra a ->
   FrobeniusAlgebra a -> BinRel (a, a) a
logicalAnd s a
  = (mu a) `o` (((i s) `x` tick s a) `o` (((i s) `x`
     mu a) `o` (((i s) `x` ((tick s a) `x` i s)) `o`
     ((al' s) `o` ((((mu s) `x` i s) `x` i s) `o` (((
     al s ) `x` i s) `o` ((((i s) `x` delta a) `x` (i
     s)) `o` ((((i s) `x` (eta s)) `x` (tick s a)) `o`
```

```
      ((ru s) `x` (i s))))))))))

logicalOr :: Eq a => FrobeniusAlgebra a ->
   FrobeniusAlgebra a -> BinRel (a, a) a
logicalOr s a
  = t `o` ((logicalAnd s a) `o` (t `x` t))
  where
    t = tick s a

logicalXOr :: Eq a => FrobeniusAlgebra a ->
   FrobeniusAlgebra a -> BinRel (a, a) a
logicalXOr s a
  = (logicalOr s a) `o` (butterfly (((i s) `x` t) `o`
     delta s) ((t `x` i s) `o` delta s) (logicalAnd s
     a) (logicalAnd s a))
  where
    t = tick s a

blackX :: Eq a => FrobeniusAlgebra a ->
   FrobeniusAlgebra a -> BinRel (a, a) (a, a)
-- /   /
-- x-+-x
-- /   /
-- where x = black dot, + = tick
blackX s a = ((i a) `x` mu a) `o` ((al' a) `o` ((((i a
   ) `x` tick s a) `x` i a) `o` ((delta a) `x` i a)))

inducedXObservable :: Ord a => FrobeniusAlgebra a ->
   FrobeniusAlgebra a -> FrobeniusAlgebra a
inducedXObservable s a
  = (carrier s, inducedMu, lollie a, dag inducedMu,
     epsilon a)
  where
    t = tick s a
    inducedMu = t `o` ((mu a) `o` ((t `x` t) `o` ((
       blackX s a) `o` (t `x` t))))

isGHZWPair :: Ord a => FrobeniusAlgebra a ->
   FrobeniusAlgebra a -> Bool
isGHZWPair s a
 = eqBinRel (tick s a) (tick a s)
   && eqBinRel ((delta s) `o` t) ((t `x` t) `o` delta
      s)
   && eqBinRel ((delta s) `o` (eta a)) (((eta a) `x` (
      eta a)) `o` lamBinRel [X])
   && eqBinRel ((lu' a) `o` (((circ a) `x` (t `o` eta
      a))) `o` lamBinRel [X]) (lollie a)
 where
   t = tick s a
```

```
isGHZWPairUpToPermutation :: Ord a => FrobeniusAlgebra
    a -> FrobeniusAlgebra a -> Bool
isGHZWPairUpToPermutation s a
 = not  (null [p | p <- bijections (carrier s) (
    carrier s), isGHZWPair s (relabelFA (\x -> head (
    eval p [x])) a)])

butterfly :: Eq a => BinRel a (a, a) -> BinRel a (a, a
   ) -> BinRel (a, a) a -> BinRel (a, a) a -> BinRel (
   a, a) (a, a)
butterfly a b c d = (c `x` d) `o` (swapMiddle (a `x` b
   ))

satisfiesBialge :: Ord a => FrobeniusAlgebra a ->
   FrobeniusAlgebra a -> Bool
satisfiesBialge (_, _, _, ds, _) (_, ma, _, _, _)
 = eqBinRel (ds `o` ma) (butterfly ds ds ma ma)

satisfiesTickedBialge :: Ord a => FrobeniusAlgebra a
   -> FrobeniusAlgebra a -> Bool
satisfiesTickedBialge s a
 = eqBinRel (butterfly da da ms mst) (((lollie a) `x`
    mst) `o` ((alBinRel' [X] c c) `o` ((lu s) `x` i s)
    ))
 where
   (c, ms, _, _, _) = s
   da = delta a
   mst = ms `o` ((i s) `x` tick s a)

satisfiesEtaIdempotence :: Ord a => FrobeniusAlgebra a
    -> FrobeniusAlgebra a -> Bool
satisfiesEtaIdempotence s a
  = eqBinRel (eta s) ((mu a) `o` (((eta s) `x` eta s)
     `o` lamBinRel [X]))

inducedAndSatisfiesAbsorption :: Ord a =>
   FrobeniusAlgebra a -> FrobeniusAlgebra a -> Bool
inducedAndSatisfiesAbsorption s a
  = eqBinRel ((ru' s) `o` ((i s) `x` (epsilon s))) ((
     logicalOr s a) `o` (((i s) `x` logicalAnd s a) `o
     ` ((al' s) `o` ((delta s) `x` i s))))

inducedAndSatisfiesDistributivity :: Ord a =>
   FrobeniusAlgebra a -> FrobeniusAlgebra a -> Bool
inducedAndSatisfiesDistributivity s a
  = eqBinRel ((logicalOr s a) `o` ((i s) `x`
     logicalAnd s a)) ((logicalAnd s a) `o` (((
     logicalOr s a) `x` logicalOr s a) `o` (swapMiddle
      ((delta s) `x` ((i s) `x` i s)))))
```

```
findPairwiseIndices :: (a -> b -> Bool) -> [a] -> [b]
   -> [(Int, Int)]
findPairwiseIndices f as bs
 = map (\((i, a), (j, b)) -> (i, j)) (filter (\((i, a)
    , (j, b)) -> f a b) (cartProd (label as) (label bs
    )))

label = label' 0
 where
   label' _ [] = []
   label' n (x : xs) = (n, x) : label' (n + 1) xs

middleSwap :: (Eq a, Eq b, Eq c) => [a] -> [b] -> [c]
   -> BinRel ((a, b), (b, c)) ((a, b), (b, c))
middleSwap a b c = ((((alBinRel' (cartProd a b) b c) `
   o` ((alBinRel a b b) `x` idBinRel c)) `o` (((
   idBinRel a) `x` sigBinRel b b) `x` idBinRel c)) `o`
    ((alBinRel' a b b) `x` idBinRel c)) `o` alBinRel (
   cartProd a b) b c

--swapMiddle :: BinRel ((a, b), (c, d)) -> BinRel ((a,
    c), (b, d))
swapMiddle = map (\(x, ((a, b), (c, d))) -> (x, ((a, c
   ), (b, d))))
```