Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Department of Mathematics
Prof. Dr. T. H. Willwacher

Semester Project in Pure Mathematics

Spring Semester 2019

Simon Gruening

A Categorical Formalism for Encryption and Quantum Teleportation

Supervisor: Andres Fontalvo Orozco

July 2019

Overview

We aim to examine and implement the Deutsch-Jozsa algorithm. This algorithm marked the first time that the advantages of quantum computing were realized theoretically. We will inspect this algorithm from two perspectives using the strengths of each to complement the other. In section 1 we cover the category theoretical preliminaries necessary for our first approach. In section 2 we describe the target problem of the algorithm, continuing to define it in category theory, and proving its correctness in the same setting. We switch to the quantum computation circuit setting for a more concrete description and implementation.

Contents

1. Categorical Formalism for Quantum Computing

We summarize the salient points from Vicary [1] by enumerating the definitions we require later, the relevant motivations for such be it mathematical or from physics, and including short notes on the graphical calculus. We have interspersed some relevant and fundamental statements about these definitions, although the detailed proofs are left out and can be reviewed in the original source. We assume the reader knows some basic category theory.

The table of examples in appendix A may provide helpful when reading the current section.

1.1. **Monoidal Categories.**

1.1.1. *Planar Isotopy.* Categories can be interpreted as defininig possible processes between states. We would like to add a temporal property to this structure. Since time in this context only makes sense in relation to some other thing, we shall do so by allowing things to happen in parallel. This can be done with the familiar properties of the tensor product.

Definition 1.1.1. *A **monoidal category** is a category $\mathbf{C}$ equipped with the following data:*

- *a **tensor product** functor $\otimes : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$*
- *a **unit object** $I \in Ob(\mathbf{C})$*
- *an **associator natural isomorphism** $(A \otimes B) \otimes C \xrightarrow{\alpha_{A,B,C}} A \otimes (B \otimes C)$*
- *a **left unitor** natural isomorphism $I \otimes A \xrightarrow{\lambda_A} A$*
- *a **right unitor** natural isomorphism $A \otimes I \xrightarrow{\rho_A} A$*

*such that the following **triangle** and **pentagon** equations hold for all $A, B, C, D \in Ob(\mathbf{C})$:*

$$\begin{tikzcd}
(A \otimes I) \otimes B \arrow[rr, "\alpha_{A,I,B}"] \arrow[dr, "\rho_A \otimes id_B"'] & & A \otimes (I \otimes B) \arrow[dl, "id_A \otimes \lambda_B"] \\
& A \otimes B &
\end{tikzcd}$$

and respectively,

$$\begin{tikzcd}
& (A \otimes (B \otimes C)) \otimes D \arrow[rr, "\alpha_{A,B \otimes C,D}"] & & A \otimes ((B \otimes C) \otimes D) \arrow[dr, "id_A \otimes \alpha_{B,C,D}"] & \\
((A \otimes B) \otimes C(\otimes D \arrow[ur, "\alpha_{A,B,C} \otimes id_D"] \arrow[drr, "\alpha_{A \otimes B,C,D}"'] & & & & A \otimes (B \otimes ((C \otimes D)) \\
& & (A \otimes B) \otimes (C \otimes D) \arrow[urr, "\alpha_{A,B,C \otimes D}"'] & &
\end{tikzcd}$$

Remark 1.1.2. *Note that naturality of the isomorphisms means that for $A, B, C, A', B', C' \in Ob(\mathbf{C})$ the following two diagrams commute:*

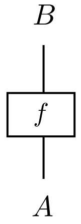
$$\begin{tikzcd}
(A \otimes B) \otimes C \arrow[r, "\alpha_{A,B,C}"] \arrow[d, "(f \otimes g) \otimes h"'] & A \otimes (B \otimes C) \arrow[d, "f \otimes (g \otimes h)"] \\
(A' \otimes B') \otimes C' \arrow[r, "\alpha_{A',B',C'}"] & A' \otimes (B' \otimes C')
\end{tikzcd}$$

$$\begin{tikzcd}
I \otimes A \arrow[r, "\lambda_A"] \arrow[d, "I \otimes f"] & A \arrow[r, "\rho_A"] \arrow[d, "f"] & A \otimes I \arrow[d, "f \otimes I"] \\
I \otimes B \arrow[r, "\lambda_B"] & B & B \otimes I \arrow[l, "\rho_B"']
\end{tikzcd}$$

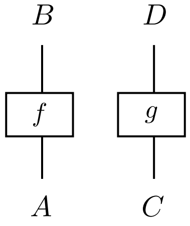
Intuitively I stands for the empty system, from which we will later create things.

THEOREM 1.1.3. *(Coherence for monoidal categories) Given the data of a monoidal category, provided that the required pentagon and triangle equations hold, then any well-type equation built from α, λ, ρ and their inverses holds.*

We shall now introduce the graphical calculus which shall guide us throughout entire. A morphism $f : A \to B$ shall be depicted simply as:



Here we read from the bottom to the top, time occuring across the vertical axis. A morphism $A \otimes C \xrightarrow{f \otimes g} B \otimes D$ constructed of two seperate morhpsisms $A \xrightarrow{f} B, C \xrightarrow{g} D$ is depicted like such:



We read this as the two processes f and g happening at the same time. The identity morphism id_I is the empty diagram:

It is omitted from now on.

We now introduce the first type of manipualation we shall allow in our graphical calculus: planar isotopy.

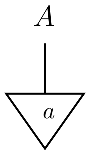
THEOREM 1.1.4. *(Corectness of the graphical calculus for monoidal categories) A well-typed eqaution between morphisms in a monoidal category follows from the axioms if and only if it holds in the graphical language up to planar isotopy.*

Planar isotopy allows for us to continuously deform the contents of a fixed rectangle inside a graphical calculus without causing any intersections nor changing the side at which a wire terminates or initiates.

We would now like to close examine the morphisms in our category. For this we are inspired the representation of set theoretic elements in categories, as can be seen fundamentally in Lawvere's works [2].

DEFINITION 1.1.5. *In a monoidal category, a* ***state*** *of an object A is a morphism $I \to A$.*

States create something from nothing, and so we use a memorable depiction. A state $a : I \to A$ is drawn as:



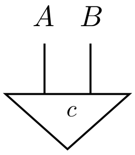
Notice the empty diagram beneath the state. Finally, we introduce a definition ensuring the descriptive strength of states.

DEFINITION 1.1.6. *A monoidal category is* ***well-pointed*** *if for all parllel pairs of morphisms $A \xrightarrow{f,g} B$, we have $f = g$ precisely when $f \circ a = g \circ a$ for all states $a : I \to A$. Furthermore, it is* ***monoidaly well-pointed*** *if for all parallel pairs of morphisms $A_1 \otimes \cdots \otimes A_n \xrightarrow{f,g} B$ we have $f = g$ when $f \circ (a_1 \otimes \cdots \otimes a_n) = g \circ (a_1 \otimes \cdots \otimes a_n)$ for all states $a_1 : I \to A, \cdots a_n : I \to A$.*

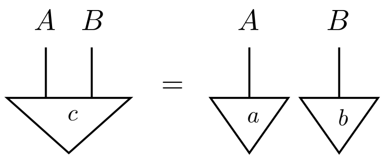
We can now begin to examine parallel states.

DEFINITION 1.1.7. *A morphism $c : I \to A \otimes B$ is a* ***joint state*** *of A and B.*

Graphically:



DEFINITION 1.1.8. *A joint state $c : I \to A \otimes B$ is a* ***product state*** *when it is of the form $I \xrightarrow{\lambda_I^{-1}} I \otimes I \xrightarrow{A} \otimes B$ for $a : I \to A$ and $b : I \to B$. Depicting this equation:*

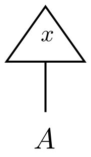


Definition 1.1.9. *A joint state is **entangled** when it is not a product state.*

Intuitively, entangled states cannot be separated cleanly into their component states. This paves the way for quantum physics. After examining creation we can also look at the flip side of the coin: destruction. In our case this may denote measurement of a state and can impose a requirement on our history.

Definition 1.1.10. *In a monoidal category, an **effect** of **costate** for an object A is a morphism $x : A \to I$.*

Again we keep the empty diagram in mind:



1.1.2. *Spatial Isotopy.* The order of our tensored objects very much matters, let us create further structure from this, as well as an ability to remove any dependency on this order.

Definition 1.1.11. *A **braided monoidal category** is a monoidal category equipped with a natural isomorphism*

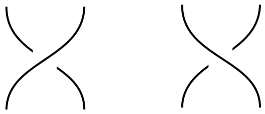
$$A \otimes B \xrightarrow{\sigma_{A,B}} B \otimes A$$

such that the hexagon equations hold:

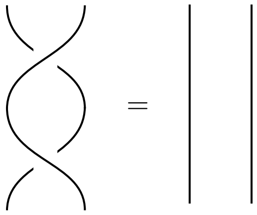
$$\begin{tikzcd}
 & A \otimes (B \otimes C) \arrow[r, "\sigma_{A,B\otimes C}"] \arrow[dl, "\alpha^{-1}_{A,B,C}"] & (B \otimes C) \otimes A \arrow[dr, "\alpha^{-1}_{B,C,A}"] & \\
(A \otimes B) \otimes C \arrow[dr, "\sigma_{A,B}\otimes id_C"] & & & B \otimes (C \otimes A) \\
 & (B \otimes A) \otimes C \arrow[r, "\alpha_{B,A,C}"] & B \otimes (A \otimes C) \arrow[ur, "id_B\otimes\sigma_{A,C}"] &
\end{tikzcd}$$

$$\begin{tikzcd}
 & (A \otimes B) \otimes C \arrow[r, "\sigma_{A,B\otimes C}"] \arrow[dl, "\alpha_{A,B,C}"] & C \otimes (A \otimes B) \arrow[dr, "\alpha_{C,A,B}"] & \\
A \otimes (B \otimes C) \arrow[dr, "id_A\otimes\sigma_{B,C}"] & & & (C \otimes A) \otimes B \\
 & A \otimes (C \otimes B) \arrow[r, "\alpha^{-1}_{A,C,B}"] & (A \otimes C) \otimes B \arrow[ur, "\sigma_{A.C}\otimes id_B"] &
\end{tikzcd}$$

We depict the braiding σ and σ^{-1} respectively as follows:



Notice that then $\sigma^{-1}\sigma$ results elegantly in:



We get an analogous image for the other inverse relation.

We now introducte our next form of isotopy, spatial isotopy. We now may deform a cube as previously, respecting the order of depth of the wires.

THEOREM 1.1.12. *(Corectness of graphical calculus for braided monoidal categories) A well-typed equation between morphisms in a braided monoidal category follows from the axioms if and only if it holds in the graphical language up to spatial isotopy.*

1.1.3. *Four-dimensional Isotopy.* Sometimes we do not care about the horizontal order, only that two processes occur simultaneously.

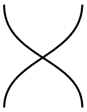
DEFINITION 1.1.13. *A braided monoidal category is **symmetric** when for all objects A and B it holds that*

$$\sigma_{B,A} \circ \sigma A, B = id_{A\otimes B}.$$

*We then call the braid σ a **symmetry**.*

LEMMA 1.1.14. *In a symmetric monoidal category $\sigma_{A,B} = \sigma^{-1}_{B,A}$.*

We thus may denote σ flatly.



In this case we may introduce four-dimensional isotopy, in which wires can pass through one another.

THEOREM 1.1.15. *(Corectness of the graphical calculus for symmetric monoidal categories) A well-typed equation between morphisms in a symmetric monoidal category follows from the axioms if and only if it holds in the graphical language up to four-dimensional isotopy.*

We may translate one monoidal structure into another, and as we are used to in category theory, ascend one further level from functors fo natural transformations. We briefly define.

DEFINITION. *A **Monoidal functor** $F : \mathbf{C} \to \mathbf{D}$ between two monoidal categories is a funtor equipped with natural isomorphisms*

$$(F_2)_{A,B} : F(A) \otimes F(B) \to F(A \otimes B)$$

$$F_0 : I \to F(I)$$

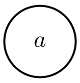
such that monoidal structure is preserved. A ***monoidal equivalence*** *is a monoidal functor that is an equivalence as a functor. A* ***monoidal natural transformation*** $\mu : F \implies G$ *between two monoidal functors* $F, G : \mathbf{C} \to \mathbf{D}$ *is a natural isomorphism compatible with the monoidal structure.*

1.2. **Linear structure.** Examining the morphisms $I \to I$ in **FHilb** we realize that they have familiar and useful properties, notably they correspond readily to the complex numbers. We generalize.

Definition 1.2.1. *In a monoidal category, the* ***scalars*** *are the morphisms* $I \to I$*.*

Lemma 1.2.2. *In a monoidal category, the scalars form a commutative monoid.*

Scalars shall be denoted as circles, vanishing above and below into the empty diagram. Notice that commutativity follows easily in this context.



We define scalar multiplication.

Definition 1.2.3. *In a monoidal category, for a scalar* $a : I \to I$ *and a morphism* $f : A \to B$*, the* ***left scalar multiplication*** $a \bullet f : A \to B$ *is defined through the following diagram:*

$$\begin{tikzcd}
A \arrow[r, "a\bullet f"] \arrow[d, "\lambda_A^{-1}"] & B \arrow[d, "\lambda_B"] \\
I \otimes A \arrow[r, "a\otimes f"] & I \otimes B
\end{tikzcd}$$

With some graphical manipulation we can show that scalar multiplication enjoys many of the properties we are used to. Superposition of qubits in quantum physics is a linear combination of two states. We strive now to achieve a similar structure. First we discover the unit of addition.

Definition 1.2.4. *An object* 1 *is* ***terminal*** *if for any object* A *there is a unique morphism* $A \to 1$*. An object* 0 *is* ***initial*** *if for any object* A *there is a unique morphism* $0 \to A$*.*

Definition 1.2.5. *An object* 0 *is a* ***zero object*** *when it is both initial and terminal. We can then define the* ***zero morphism*** $0_{A,B} : A \to B$ *as the unique morphism* $A \to 0 \to B$*.*

Lemma 1.2.6. *Composition with a zero morphism always gives a zero morphism.*

It is possible to define vector addition solely in the categorical sense, abstracting the properties we are familiar with from linear algebra.

Definition 1.2.7. *An operation* $(f, g) \mapsto f+g$ *which is defined for morphisms* $f, g : A \to B$ *is a* ***superposition rule*** *if it has the following properties:*

- *Commutativity:* $f + g = g + f$
- *Associativity:* $(f + g) + h = f + (g + h)$
- *Units:* $\forall A, B \in Ob(\mathbf{C}) \exists A \xrightarrow{u_{A,B}} B : f + u_{A,B} = f$

- *Composition distributes over addition:*

$$(g + g') \circ f = (g \circ f) + (g' \circ f)$$
$$g \circ (f + f') = (g \circ f) + (g \circ f')$$

- *Units are compatible with composition: for all $f : A \to B$*

$$f \circ u_{A,A} = u_{A,B} = u_{B,B} \circ f$$

Indeed the zero object plays the role of the unit.

Lemma 1.2.8. *In a category with a zero object and a superposition rule, $u_{A,B} = u_{0,B} \circ u_{A,0}$ for any objects A and B.*

Inspired by the direct sum, we may introduce a generalized concept of such.

Definition 1.2.9. *In a category with a zero object and a superposition rule, the* ***biproduct*** *of two objects A_1, A_2 is an object $A_1 \oplus A_2$ and a choice of injection morphisms $i_n : A_n \to A_1 \oplus A_2$, and projection morphisms $p_n : A_1 \oplus A_2 \to$ for $n = 1, 2$ such that:*

$$id_{A_n} = p_n \circ i_n$$
$$0_{A_m,A_n} = p_m \circ i_n$$
$$id_{A_1} \oplus A_2 = i_1 \circ p_1 + i_2 \circ p_2$$

Remark 1.2.10. [1, section 2.2.3] *The biproduct is a categorical product and coproduct. Furthermore, in a category with biproducts, every morphism $\bigoplus_{m=1}^{M} A_m \xrightarrow{f} \bigoplus_{n=1}^{N} B_n$ has a familiar matrix representation.*

1.2.1. *The way of the Dagger.* Motivated by the inner product in **Hilb**, we begin by generalizing the adjoint. The dagger functor is the first step on what Vicary refers to as "the way of the dagger".

Definition 1.2.11. *A* ***dagger functor*** *on a category* **C** *is an involutive contravariant functor $\dagger : \mathbf{C} \to \mathbf{C}$ that is the identity on objects. A* ***dagger category*** *is a category equipped with a choice of dagger functor.*

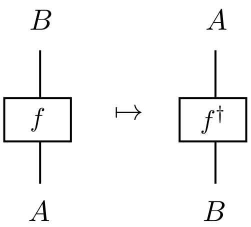
Unitarity plays a very strong role in quantum mechanics and we may now define it.

Definition 1.2.12. *A morphism $f : A \to B$ in a dagger category is* ***unitary*** *when $f^\dagger \circ f = id_A$ and $f \circ f^\dagger = id_B$.*

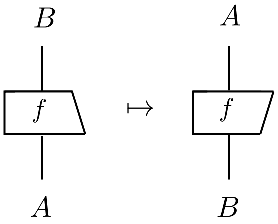
We wish to guide interaction between the dagger and monoidal structure by our intuition in **FHilb**, where we may think of the Kronecker product and adjoints.

Definition 1.2.13. *A* ***monoidal dagger category*** *is a dagger category that is also monoidal, such that $(f \otimes g)^\dagger = f^\dagger \otimes g^\dagger$ for all morphisms f, g. We also require that all components of the natural isomorphisms α, λ, ρ are unitary. A* ***braided monoidal dagger category*** *is a monoidal dagger category equipped with a unitary braiding. A* ***symmetric monoidal dagger category*** *is a braided monoidal dagger category in which the braiding is symmetric.*

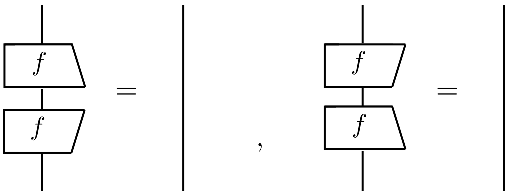
Taking the dagger can be elegantly drawn in our graphical calculus as flipping our morphisms along the horizontal axis:



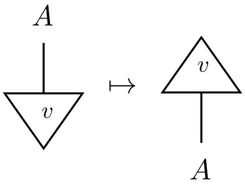
To help us differentiate morphisms, we adapt our depiction from rectangles to wedges, the orientation giving us the daggered state, as follows:



Unitarity can then be defined graphically in a visually pleasing way:

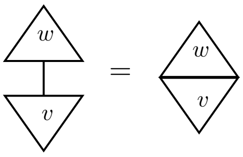


The dagger functor furthermore gives us a relation between preparation of a system and modelling its effect. In particular, observe a state $v : I \to A$ being daggered:



We now have an important correspondence. Finally we receive our inner product.

$< v|w >$ =



Simply rotating the diagram gives us a handy mnemonic.

Definition 1.2.14. *In a dagger category with a fixed zero object and superposition rule, a* ***dagger biproduct*** *of objects A, B is a biproduct $A \oplus B$ whose injections and projections satisfy $i_A^\dagger = p_A$ and $i_B^\dagger = p_B$.*

In this case, taking the adjoint of a function is the same as taking the adjoint of its matrix representation.

Inspired by the fundamental Born rule in quantum mechanics, we define an abstract notion of probability such that in **Hilb** it resolves as the usual definition in quantum mechanics.

Definition 1.2.15. *For a state $v : I \to A$ and an effect $x : A \to I$ in a monoidal dagger category, we define:*

$$Prob(x, v) := v^{\dagger} \circ x^{\dagger} \circ x \circ v$$

This definition necessarily gives us a scalar.

1.2.2. *Dual Objects.* Exploring dual objects will add a orientation to our graphical calculus as well as lend strength to our definition of entangled states.

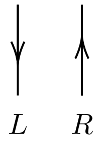
Definition 1.2.16. *In a monoidal category, an object L is a* ***left-dual*** *to the* ***right-dual****, an object R, when there exist a unit morphism $I \xrightarrow{\eta} R \otimes L$ and a counit morphism $L \otimes R \xrightarrow{\varepsilon} I$ such that the following two diagrams commute:*

$$\begin{tikzcd}
L \arrow[r, "\rho_L^{-1}"] \arrow[d, "id_L"] & L \otimes I \arrow[r, "id_L \otimes \eta"] & L \otimes (R \otimes L) \arrow[d, "\alpha^{-1}_{L,R,L}"] \\
L & I \otimes L \arrow[l, "\lambda_L"'] & (L \otimes R) \otimes L \arrow[l, "\varepsilon \otimes id_L"']
\end{tikzcd}$$

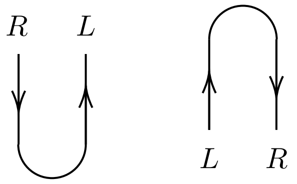
$$\begin{tikzcd}
R \arrow[r, "\lambda_R^{-1}"] \arrow[d, "id_R"] & I \otimes R \arrow[r, "\eta \otimes id_r"] & (R \otimes L) \otimes R \arrow[d, "\alpha_{R,L,R}"] \\
R & R \otimes I \arrow[l, "\rho_R"'] & R \otimes (L \otimes R) \arrow[l, "id_R \otimes \varepsilon"']
\end{tikzcd}$$

We denote this relationship as $L \dashv R$.

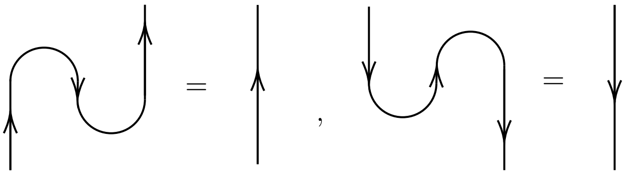
We denote this relationship as orientation in our graphical calculus.



The unit η and counit ε follow:



Translating the duality equation into our notation we see why this choice has been made.



Duals are well-defined up to canonical isomorphism [1]. They also enjoy other expected basic properties and work together well with our previously defined structures.

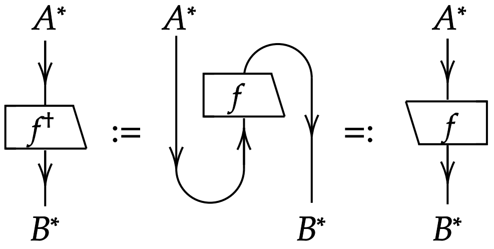
LEMMA 1.2.17. *In a monoidal category, $L \dashv R$ and $L' \dashv R'$ implies $L \otimes L' \dashv R \otimes R'$. $I \dashv I$.*

LEMMA 1.2.18. *In a braided monoidal category, $L \dashv R \implies R \dashv L$.*

THEOREM 1.2.19. *Monoidal functors preserve duals.*

We can extend the definition of duals from objects to morphisms functiorially.

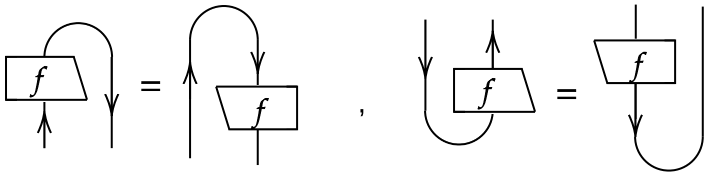
DEFINITION 1.2.20. *Fix a morphism $f : A \to B$, two dualities $A \dashv A^*, B \dashv B^*$. The* ***right dual*** *is defined as:*



LEMMA 1.2.21. *Let $\mathbf{C}$ be a monoidal category in which every object X has a chosen right dual X^*. The* ***right duals functor*** *$(-)^* : \mathbf{C}^{op} \to \mathbf{C}$ defined by $(X)^* = X^*$ and $(f)^* = f^*$ exists. The right duals functor is monoidal.*

Graphically we may manipulate the right dual by sliding the morphism along the twists in our wires.

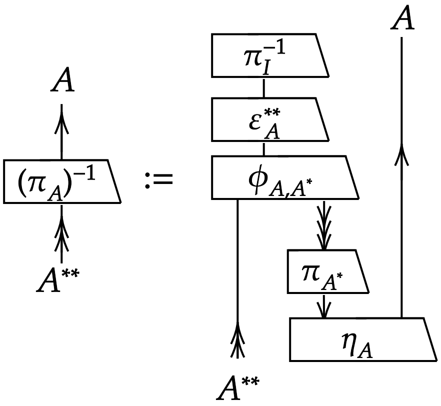
LEMMA 1.2.22. *In a monoidal category, fix dualities $A \dashv A^*, B \dashv B^*$. Then the following holds for all $f : A \to B$:*



1.2.3. *Planar Oriented Isotopy.* In finite dimensional vector space, taking the dual twice leads to the existence of an isomorphism $V \simeq V$. This lends strong additional structure.

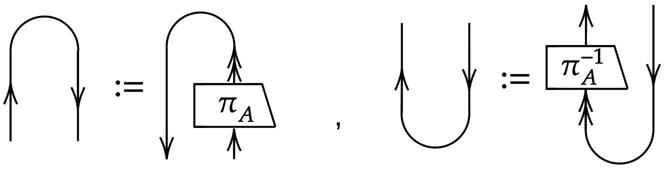
DEFINITION 1.2.23. *A monoidal category with right duals is* ***pivotal*** *when there exists a choice for a monoidal natural transformation $\pi_A : A \to A^{**}$.*

LEMMA 1.2.24. *In a pivotal category, the morphisms $\pi_A : A \to A^{**}$ are invertible. Inverses are constructable:*



From this we may further construct left duals with a guarantee of existing.

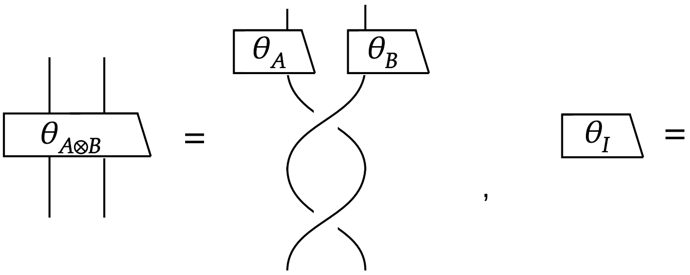
Proposition 1.2.25. *A pivotal monoidal category has left duals for al objects.*



Theorem 1.2.26. *(Corectness of the graphical calculus for pivotal categories) A well-formed equation between morphisms in a pivotal category follows from the axioms if and only if it holds in the graphical language up to planar oriented isotopy.*

Oriented here means that we must now also preserve the direction, as indicated by arrows, of our wires when deforming isotopically. The next definition comes naturally.

Definition 1.2.27. *A braided monoidal category is* ***balanced*** *when it is equipped with a natural isomorphism $\theta_A : A \to A$ called a* ***twist****, if the following holds:*



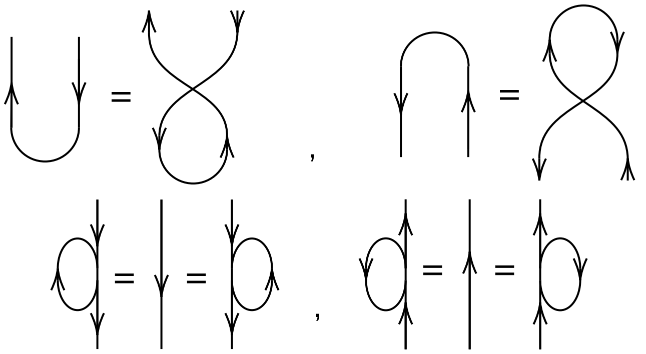
The second equation makes use of the empty diagram.

Theorem 1.2.28. *For a braided monoidal category with duals, any pivotal structure uniquely induces a twist structure, and vice versa.*

When our braided monoidal category is symmetric, there is a canonical choice.

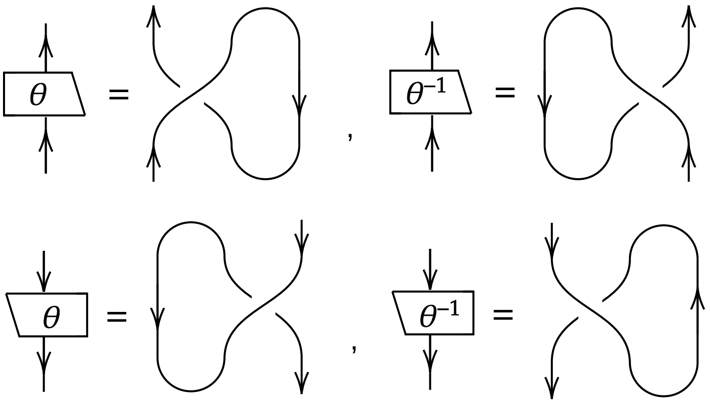
Definition 1.2.29. *A* ***compact category*** *is a pivotal symmetric monoidal category equipped with the identity balancing $\theta_A = id_A$.*

Lemma 1.2.30. *In a compact category, the following hold:*



On the other hand if we wish to remove slightly less structure, we cannot fully ignore the twists.

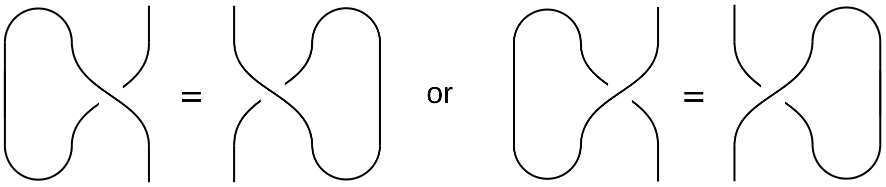
Lemma 1.2.31. *In a braided pivotal category, the following hold:*



1.2.4. *Framed Isotopy in Three Dimensions.* We are now ready to hint at a further level of isotopy.

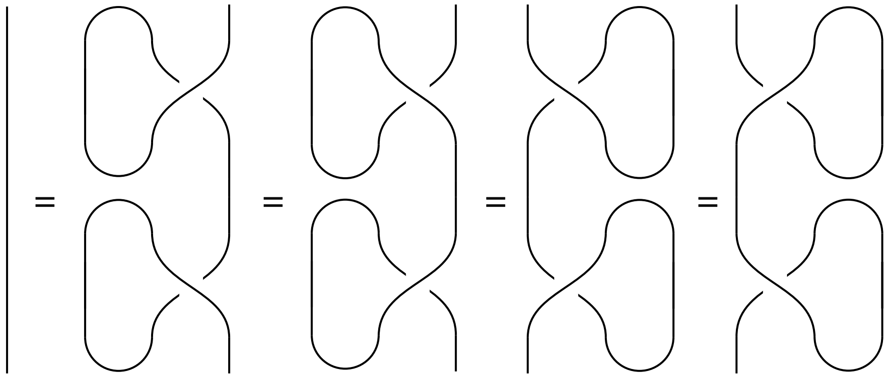
Definition 1.2.32. *A* ***ribbon*** *or* ***tortile*** *category is a balanced monoidal category with duals such that* $(\theta_A)^* = \theta_{A^*}$.

Lemma 1.2.33. *A balanced monoidal category with duals us a ribbon category if and only if either of these equivalent equations are satisfied:*



Corollary 1.2.34. *A compact category is a ribbon category.*

Lemma 1.2.35. *In a ribbon category we have:*



The wires here act as ribbons would, inspiring framed isotopy. Here the wires are thought of as flat ribbons, for example thin strips of paper. Their material property cannot be violated nor can an internal twist be allowed.

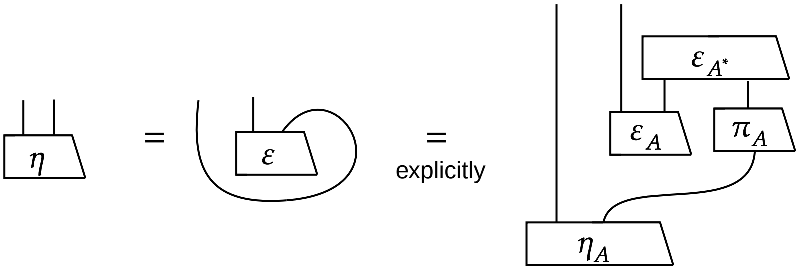
Theorem 1.2.36. *(Corecteness of the graphical calculus for ribbon categories) A well-formed equation between morphisms in a ribbon category follows from the axioms if and only if it holds in the graphical language up to framed isotopy in three dimensions.*

Lemma 1.2.37. *In a symmetric ribbon category, $\theta^2 = id$.*

Let us examine how dagger duals interact with the pivotal structure.

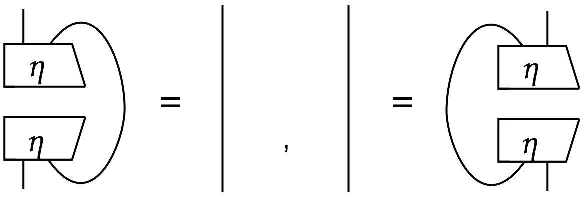
Lemma 1.2.38. *In a monoidal dagger category, $L \dashv R \Leftrightarrow R \dashv L$.*

Definition 1.2.39. *In a dagger category with a fixed pivotal structure, a* ***dagger dual*** *is a duality $A \dashv A^*$ such that there exist two morphisms $\eta : I \to A^* \otimes A, \varepsilon : A \otimes A^* \to I$ with:*



Finally we can again take motivation from physics.

Definition 1.2.40. *In a dagger category with a fixed pivotal structure, a* ***maximally entangled state*** *is a bipartite state such that*

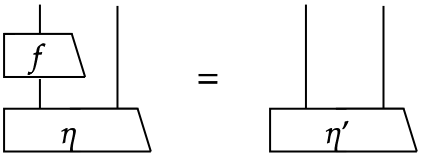


Lemma 1.2.41. *In a dagger category with a fixed pivotal structure, a bipartite state is maximally entangled if and only if it is part of a dagger duality.*

Lemma 1.2.42. *In a dagger category with a fixed pivotal structure, dagger duals are unique up to unique unitary isomorphism.*

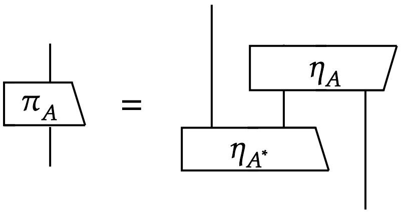
Combining the previous two results yields:

Theorem 1.2.43. *In a dagger category with a fixed pivotal structure, any two maximally entangled states $\eta, \eta' : I \to A \otimes B$ give a unique unitary $f : A \to A$ such that*

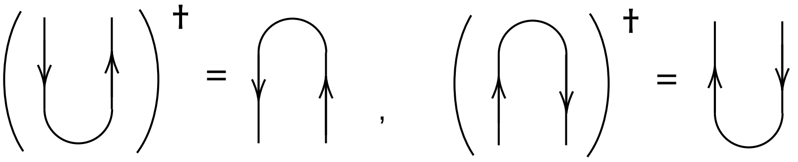


Definition 1.2.44. *A* ***pivotal dagger category*** *is a dagger category with a pivotal structure, such that the chosen right duals are all dagger duals.*

Proposition 1.2.45. *In a pivotal dagger category, the pivotal structure is given by*



Lemma 1.2.46. *In a dagger category we have*

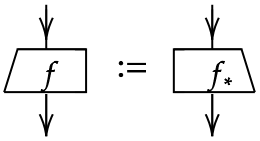


Lemma 1.2.47. *In a pivotal dagger category, for every morphism f we have $(f^*)^\dagger = (f^\dagger)^*$.*

This motivates the following natural notation.

Definition 1.2.48. *In a pivotal dagger category, we define the* ***conjugation*** $(-)_* := (-^*)^\dagger = (-^\dagger)^*$.

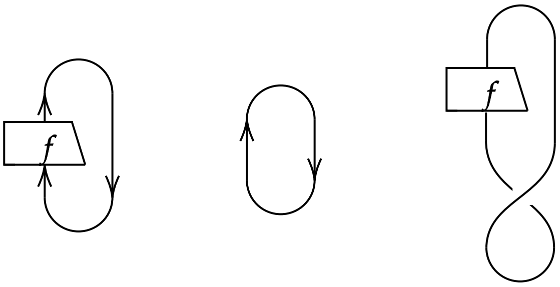
Since both its components are contravariant, the conjugation functor is covariant. Graphically we may interpret this functor as flipping our diagram vertically:



Definition 1.2.49. *A* ***ribbon dagger category*** *is a braided pivotal dagger category with unitary braiding and twist. A* ***compact dagger category*** *is a symmetric pivotal dagger category with unitary symmetry and $\theta = id$.*

Again we strongly care about unitarity in lieu of operators in quantum mechanics. Finally, we have enough structure to introduce the category theoretic equivalent of the trace.

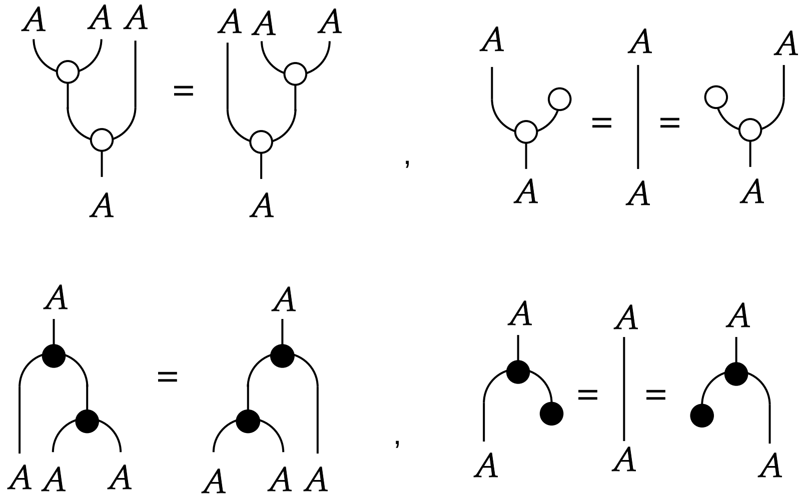
Definition 1.2.50. *In a pivotal category, the* ***trace*** *and* ***dimension*** *of a morphism $f : A \to A$ for an object A is defined by the first and second diagram respectively. In a monoidal braided category we also define the* ***braided trace*** *as shown in the third diagram.*



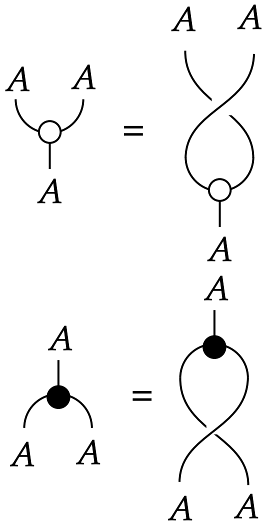
We notice in particualar that the cyclic property of the trace follows directly from chasing the morphisms around the cycle in graphical notation. Through diagramatic manipulation, we may gather other familiar properties of the trace and dimension.

1.3. **Monoids and Comonoids.** Where information is concerned, there is a strong distinction between calssical and quantum information. Whereas the former can be copied and deleted, the later cannot. We define structures which deal with this notion, having prepared the requirements through monoidal categories, and further show when these structures collapse our categories and when not.

Definition 1.3.1. *In a monoidal category a* ***comonoid*** *is a triple $(A, ⋎, ⫯)$ where $A \in ob(\mathbf{C})$, $⋎ : A \to A \otimes A$, $⫯ : A \to I$ such that the following equations hold:*

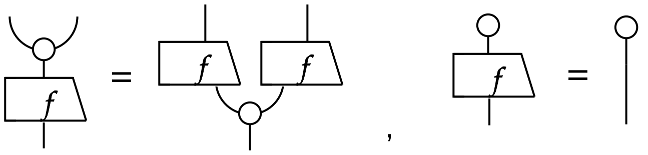


A ***monoid*** *in the same setting is a triple $(A, ⋏, ⫰)$ where $⋏ : A \otimes A \to A$, $⫰ : I \to A$ satisfy the relevant equations above. We call $⋎$, $⋏$ the* ***comultiplication*** *and* ***multiplication*** *respectively,* ⫯,⫰ *the* ***counit*** *and* ***unit****. The comonoid and monoid are* ***cocommutative*** *and* ***commutative*** *respectively when the relevant diagram below holds:*



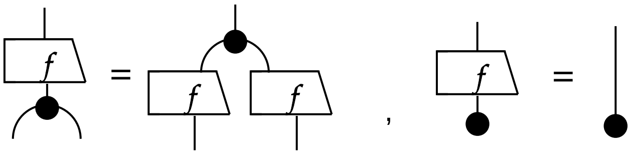
We may think of a comonoid as an object with a notion of copying (coproduct) and deleting (counit) information. It should be irrelevant which of two copied states we remove through application of deletion. We may build our collection of comonoids in a monoidal category into its own category.

Definition 1.3.2. *A* ***comonoid homomorphism*** *from a comonoid* (A, d, e) *to a comonoid* (A', d', e') *is a morphism* $f : A \to A'$ *such that* $(f \otimes f) \circ d = d' \circ f$ *and* $e' \circ f = e$*, or in other words:*



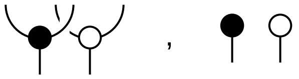
It does not matter whether the process of f is applied before copying or two both copies individually, nor if it is applied to a system which is deleted. Again we define the dual with analogue intuition.

Definition 1.3.3. *A* ***monoid homomorphism*** *from a monoid* (A, m, u) *to a monoid* (A', m', u') *is a morphism* $f : A \to A'$ *such that* $f \circ m = m' \circ (f \otimes f)$ *and* $u' = f \circ u$*, or in other words:*



We may also take the tensor product. The axioms are readily verified by drawing them out. This also works for monoids.

Lemma 1.3.4. *In a braided monoidal category, two comonoids can generate a new comonoid:*



We can state the duality between comonoids and monoids concretely.

Lemma 1.3.5. *In a monoidal dagger category, if (A, d, e) is a comonoid, then $(A, d^\dagger, e^\dagger)$ is a monoid.*

1.3.1. *No Deleting.* We would like to impart a stronger method of deletion; we want to say that it should not matter if a system has passed through our structures and been processed apropriately and is then deleted, or is simply deleted from the beginning.

Definition 1.3.6. *A monoidal category has the property of* ***uniform deleting*** *if there exists a natural transformation $e_A : A \to I$ such that $e_I = id_I$.*

Proposition 1.3.7. *A monoidal category has uniform deleting if and only if the choice of tensor unit I is terminal.*

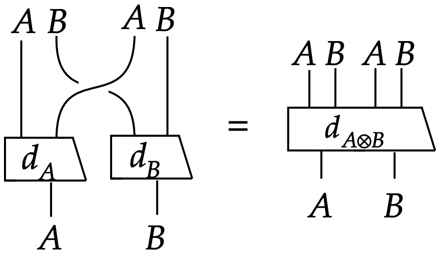
Definition 1.3.8. *A* ***preorder*** *is a category that has at most one morphism between any pair of objects.*

Theorem 1.3.9. *(No deleting) If a compact category has uniform deleting, then it is a preorder.*

Preorders are uninteresting from our perspective as there is no choice in how a system can evolve. There is only one history, if at all. We are thus strongly encouraged to make the appropriate choice of tensor.

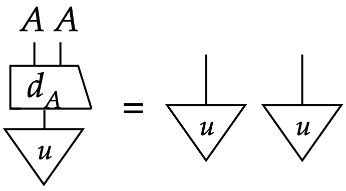
1.3.2. *No Cloning.* We shall now examine the analogue for copying.

Definition 1.3.10. *A braided monoidal category has* ***uniform copying*** *if there exists a natural transformation $d_A : A \to A \otimes A$ such that $d_I = \rho_I^{-1}$, satisfying the coassociativity and cocommutitivity equation of comonoids, and such that the following holds:*



Or in other words, we do not care if something is processed and then copied, or cloned first with both copies processed equally and parallely.

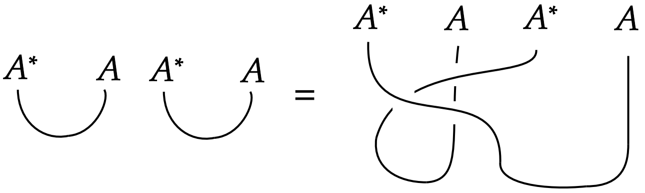
Definition 1.3.11. *A state $u : I \to A$ in a braided monoidal category with a copying map d_A is* ***copyable*** *when*



PROPOSITION 1.3.12. *In a braided monoidal category with copying maps d_A for each object A and uniform copying, every state is copyable. The converse holds when the category is monoidally well-pointed.*

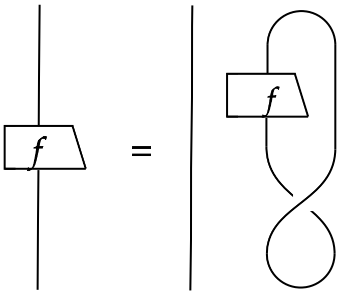
We have thus restricted back to a more set-theoretic notion of copying. The following lemma displays a paramount problem, equating two morphisms with different connectivity, morally turning two disjoint systems into one entangled system.

LEMMA 1.3.13. *In a braided monoidal category with duals and uniform copying, we have:*



Capitalizing on this, we receive the no cloning theorem.

THEOREM 1.3.14. *(No cloning) In a braided monoidal category with duals and uniform copying, every endomorphism is a scalar multiple of the identity:*



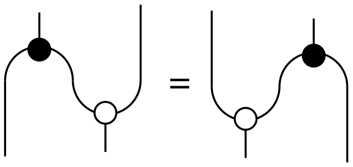
Once again the category degenerates, something we also would like to avoid. Ultimately, we may create a link between the tensor product and the categorical product.

DEFINITION 1.3.15. *A category with a terminal and object and all products is called* ***cartesian****.*

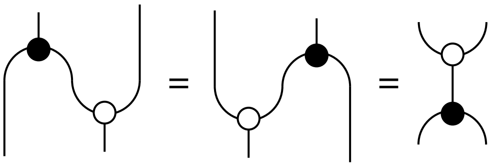
THEOREM 1.3.16. *A symmetric monoidal category is cartesian if and only if it has uniform copying and deleting and the counit equations hold.*

1.4. **Frobenius structures.** We now look at interactions between monoids and comonoids in the most natural way.

DEFINITION 1.4.1. *In a monoidal category, a* ***Frobenius structure*** *is a comonoid and a monoid such that the following relation holds:*

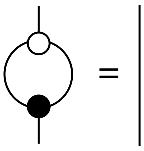


Lemma 1.4.2. *Any Frobenius structure satisfies:*



In some cases this interaction does not allow for much complexity in itself.

Definition 1.4.3. *In a monoidal category, a choice of comonoid and monoid is* ***special*** *when*



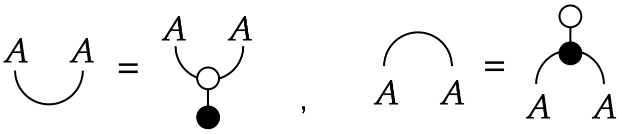
In the Frobenius structures relevant to us, the daggered multiplication is the comultiplication. We now ensure this compatibility.

Definition 1.4.4. *A Frobenius structure in a monoidal dagger category is a* ***dagger Frobenius structure*** *when* $=$ *and* $=$.

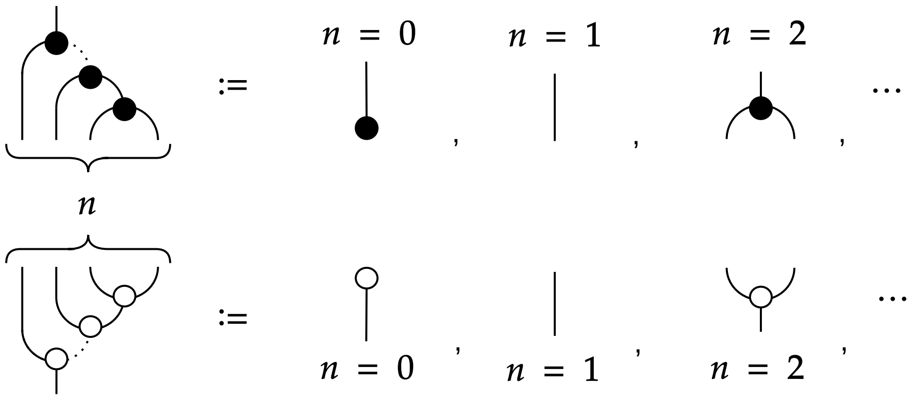
Definition 1.4.5. *A* ***classical structure*** *is a dagger Frobenius structure in a braided monoidal dagger category that is special and commutative.*

The name derives from classical information theory where it is possible to copy and delete systems.

Theorem 1.4.6. *In a monoidal category, if an object A has a Frobenius structure then A is self-dual. The cap and cup can be constructed:*

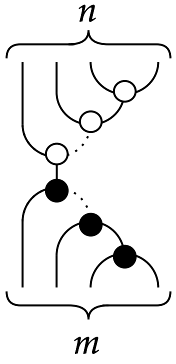


1.4.1. *Spider Theorems.* We shall now shortly harness the power of graph theory, switching perspectives. Here endpoints of wires and (co)monoidal operations are seen as vertices, and the wires themselves as edges. This allows us to define a powerful normal form as in logical proofs, and define the spider theorems which allow us to simplify a lot of structure in our graphical calculus. First, let us define a convenience in notation:



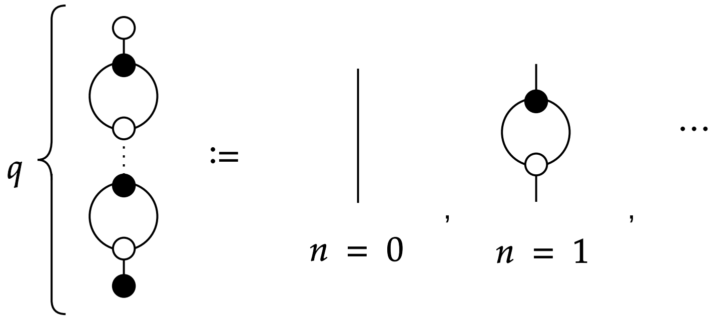
Then, by inductive shuffling of our diagrams, naturality and unitality we receive the following result.

Lemma 1.4.7. *(Special noncuommutative spider theorem) Fix a special Frobenius structure* $(A,$ ●, •, ○, ∘$)$ *in a monoidal category. Any (graph theoretically) connected morphism* $A^{\otimes m} \to A^{\otimes n}$ *created in finited steps from pieces* ●, •, ○, ∘, *id using only composition and the tensor product equals:*

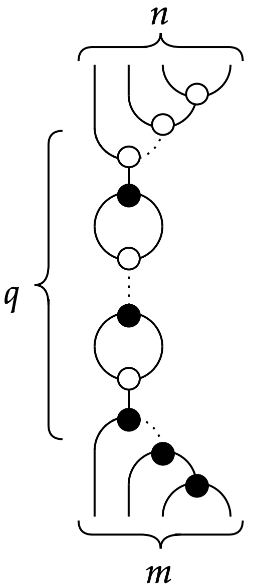


This is called the ***normal form****.*

Notice that the normal form looks a bit like an $(m+n)$-legged spider. We introduce further notation for the non-special case, where slightly more structure remains in the diagram.

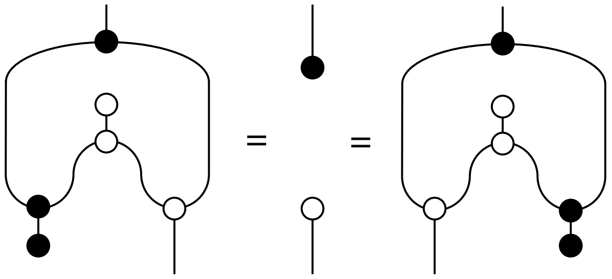


Theorem 1.4.8. *(Noncommutative spider theorem) Fix a Frobenius structure* $(A,$ ⅄●, ●, ⅄○, ○$)$ *in a monoidal category. Any (graph theoretically) connected morphism* $A^{\otimes m} \to A^{\otimes n}$ *created in finited steps from pieces* ⅄●, ●, ⅄○, ○, *id using only composition and the tensor product equals:*

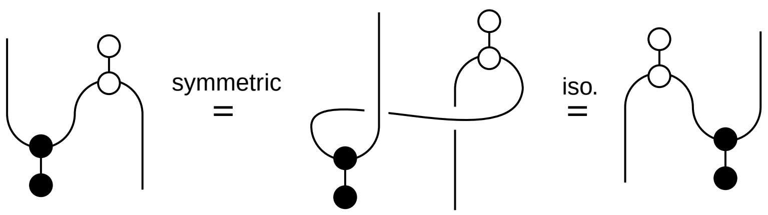


1.4.2. *Final Preparations.* We wish to describe Heisenberg's uncertainty principle by way of one measurement being independent of another measurement.

Definition 1.4.9. *In a bradided monoidal dagger category, two symmetric dagger Frobenius structures* ⅄○, ⅄● *defined on the same object are* ***complementary*** *when:*



The disconectedness here represents the independence of measurement results. By symmetry, we find an equivalence in this definition:



Definition 1.4.10. *Two bases $\{d_i\}, \{e_j\}$ in **FHilb** are **mutually unbiased** when $| < d_i, e_j > |^2$ is independent of i, j.*

This coincides with our notion of complementarity. Conversely:

Proposition 1.4.11. *Two commutative dagger Frobenius structures on an object H in **FHilb** are complementary if and only if their orthogonal bases of nonzero copyable states are mutually unbiased, the nonzero copyable states of each basis have the same length, and the product of these two lengths is $\sqrt{dim(H)}$.*

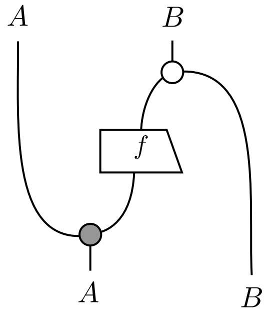
2. Deutsch-Jozsa Algorithm

The Deutsch-Jozsa algorithm is of great historic importance, it marks the first time that the advantages of quantum computing where realized and is still a great illustrative tool for such. Named afer David Deutsch and Richard Jozsa [3], it solves the problem of whether a function $f : \{0,1\}^n \longrightarrow \{0,1\}$ is constant or balanced, where balanced means that precisely half the entries are mapped to 0. Two different perspectives of the algorithm follow. We shall specifically prove corectenss in the category theoretic approach while the practical implementation shall stem from the quantum computation circuit approach.

2.1. **The Quantum Category Theory Perspective.** In this section we strictly follow the script of Vicary [1]. For a quick summary of such refer to section A, however further background is expected for this section. On the other hand, the next section stands almost independently and may be read first in order to obtain a concrete idea before the obtaining an abstract one.

2.1.1. *The Algorithm.* We would like to provide our algorithm with a translation from the classical object of f to a quantum theoretical object to be usable in our quantum computations. This presents itself in the form of a unitary map since we must be careful to preserve the norm of our qubits and would like our quantum gate to be reversible. This motivates the following definition:

Definition 2.1.1. [1] *An oracle in a monoidal dagger category is a morphism $f : A \to B$ equipped with a dagger Frobenius structure ◉ on A and ○ on B such that the following morphism is unitary:*



It is possible to further solidify this definition by showing that any oracle in **FHilb** must extend a function between bases:

Proposition 2.1.2. [1] *Fix $(A,$◉$)$, $(B,$●$)$, $(C,$○$)$ to be symmetric dagger Frobenius structures in a braided monoidal dagger category. A self-conjugate comonoid homomorphism $f : (A,$◉$) \to (B,$●$)$ is an oracle $(A,$◉$) \to (C,$○$)$ if and only if ● is complementary to ○.*

Remaining in **FHilb** in order to adapt to quantum circuit theory, we now fix a setting for our algorithm to work in.

Definition 2.1.3. *The* **Pauli bases** *for the Hilbert space* $\mathbb{C}^2$ *are as follows:*

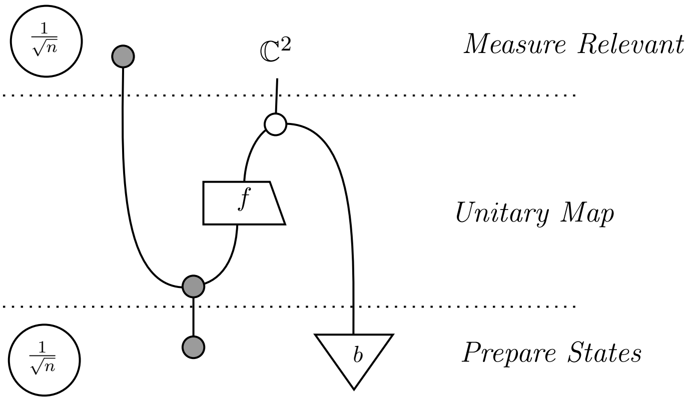
$$X\ basis := \left\{ \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix} \right\}$$
$$Y\ basis := \left\{ \frac{1}{\sqrt{2}}\begin{bmatrix}1\\i\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-i\end{bmatrix} \right\}$$
$$Z\ basis := \left\{ \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix} \right\}$$

They are mutually complementary. The Z-basis is also called the computational basis, and we shall use it when implementing our algorithm.

Fix A with $|A| = n$ and a function $f : A \to \{0, 1\}$. Create from f an oracle by extending, as in Def. 2.1.1. The two complementary bases on $\mathbb{C}^2$ are then the Z-basis and the X-basis scaled by a factor $\sqrt{2}$. Let $b := \begin{bmatrix}\frac{1}{\sqrt{2}}\\-\frac{1}{\sqrt{2}}\end{bmatrix}$ for the initial state of $\mathbb{C}^2$.

We may now define the Deutsch-Jozsa algorithm in our prepared setting.

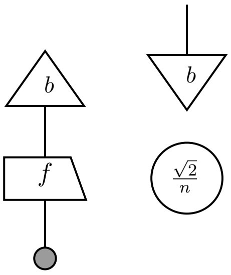
Definition 2.1.4. [1] *The* **Deutsch-Jozsa algorithm** *is the following morphism in* ***FHilb****:*



The algorithm can be broken into three parts analogously to the implementation in 2.2 as we shall soon see; preparation of a system in maximally entangled state and one in state b, application of a unitary map later denoted as the oracle U_f, and finally measurement of the first system ignoring the second. To obtain the probability of this history occuring, we apply $\| - \|^2$ to this entire state. The idea being that the history must occur with probability 1 if f is constant, and may never occur if f is balanced.

Furthermore, with an application of the noncommutative Spider theorem after movement of the copyable state $\sqrt{2}b$ through ⅄ we further can simplify the algorithm:

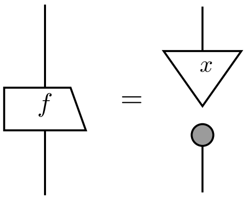
Lemma 2.1.5. [1] *The Deutsch-Jozsa algorithm simplifies to:*



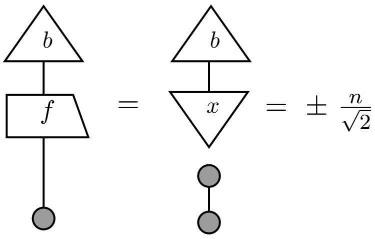
2.1.2. *Correctness.* It is possible to show correctness of the algorithm in this setting. Unlike in the quantum computation setting, we do not need to wade through sums and tensor products. We examine the posibility of the history depicted by the algorithm in each case of f being constant or balanced.

Lemma 2.1.6. [1] *If $f : A \to \{0, 1\}$ is constant, then the history of Def. 2.1.4 must occur, or in other words, the respective probability is* 1*.*

Proof. It follows from $f(a) \equiv x$ that the oracle $f : H \to \mathbb{C}^2$ as constructed in Def. 2.1.1 degenerates to:



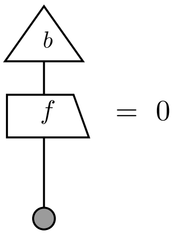
Calculating the amplitude of the system we measure, the first system, we find that:



It follows that the norm of the Deutsch-Jozsa algorithm is 1 and hence the history must occur. □

Lemma 2.1.7. [1] *If $f : A \to \{0, 1\}$ is balanced, then the history of Def. 2.1.4 can never occur, or in other words, the respective probability is* 0*.*

Proof. Assume f is balanced. We test for a balanced (normed) sum of $+1$ and -1 with $\sum_{a \in A} b^\dagger(f(a)) = 0$. Graphically:



It follows that the norm of the Deutsch-Jozsa algorithm is 0, and hence the history is impossible to occur. □

This illustrates the effect of (destructive) interference which gives the quantum implementation of this algorithm a strong advantage over the classical implementation.

Combining these two Lemmas gives us corectness of the algorithm:

THEOREM 2.1.8. [1] *The Deutsch-Jozsa algorithm must identify constant functions $f : A \to \{0, 1\}$.*

The following section gives an alternate view on the Deutsch-Jozsa algorithm from a more practical approach. We refrain however from once more showing corectness or identifying similar properties although this is very much possible through lengthy straight forward computations. The definitions align and a feeling of Déjà vu can be expected.

2.2. **The Quantum Computation Circuit Perspective.** For reference on the circuit we follow a paper on the efficient classical simulation of the Deutsch–Jozsa algorithm [4]. Unlike in category theory, the temporal axis of quantum circuit diagrams moves not from bottom to top but from left to right. As in category theory, parallel lines refer to a tensored product "occuring" at the same time. The tensor $\otimes$ is the kronecker product and we move in the realm of linear algebra. Fix $n \in \mathbb{N}_{>0}$, a function $f : \{0, 1\}^n \longrightarrow \{0, 1\}$ such that f is constant or balanced.

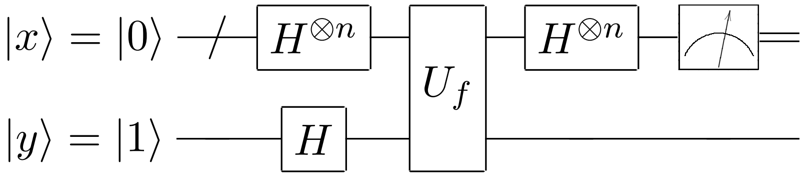


FIGURE 1. Deutsch-Jozsa Quantum Circuit

Here qubits are identified as $\mathbb{R}^2$, with

$$|0\rangle := \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle := \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The bra-ket notation allows for multiple qubits to be denoted as

$$|xy\rangle := |x\rangle\, |y\rangle := |x\rangle \otimes |y\rangle .$$

Quantum gates are represented as unitary matrices, hence being reversible and preserving norm. The Hadamard gate, mapping one qubit to another, takes the form of

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

The slash / denotes an n-tensoring of $|0\rangle$, hence above $|x\rangle$ denotes n qubits while $|y\rangle$ only stands for a single qubit. An application a quantum gate looks as follows:

$$H|0\rangle := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

with $H^{\otimes n}$ refering to the n-kronecker product of H and being applied analogously to n qubits via matrix multiplication.

U_f is the oracle of f:

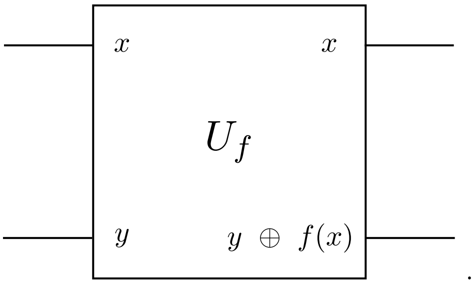


Figure 2. The oracle U_f

It maps $|x\rangle|y\rangle \mapsto |x\rangle\left|y \oplus f(x)\right\rangle$ where $\oplus$ denotes addition modulo 2. In this case we calculate the matrix for U_f following the generalized procedure of one found in [5] [1]. Let b_i denote the binary representation of integer $i \in \mathbb{N}_0$. Our square matrix will be of size $l := 2^{n+1}$.

$$\begin{aligned} U_f &= (U_f|b_1\rangle, \cdots, U_f|b_l\rangle) \\ &= (U_f|x_1\rangle|y_1\rangle, \cdots, U_f|x_l\rangle|y_l\rangle) \\ &= (|x_1\rangle\left|y_1 \oplus f(x_1)\right\rangle, \cdots, |x_l\rangle\left|y_l \oplus f(x_l)\right\rangle) \end{aligned}$$

Where for $|b_i\rangle = |x_i\rangle|y_i\rangle$, x_i once more represents all qubits save the final one, which is stored in y_i. The $\left|y_i \oplus f(x_i)\right\rangle$ can be computed by applying binary negation to $f(x_i)$ if $|y_i\rangle = |1\rangle$ and doing nothing otherwise. Finally computing the kronecker product and performing column-wise concatenation results in our sought matrix for U_f. It can be shown by generalizing similarly once more a proof in [5], that U_f is indeed unitary.

The dial concluding the circuit for x denotes measurement of the state. We shall measure the $n+1$ qubits $|x\rangle \otimes |y\rangle$ using the following projectors:

$$P00\cdots000 := \langle 00\cdots000| := (|0\rangle|0\rangle\cdots|0\rangle|0\rangle|0\rangle)^t$$

[1] We would like to thank Karan Khathuria for explaining the matrix notation of the oracle

$$P00\cdots 001 := \langle 00\cdots 001| := (|0\rangle\,|0\rangle \cdots |0\rangle\,|0\rangle\,|1\rangle)^t,$$

where $(-)^t$ is simply the transpose. Applying a projector to the final state gives us the probability of this state existing. Adding both probabilities we receive 1 if f is constant and 0 if f is balanced. We thus ignore the state of the y qubit and only ask whether $x = |00..00\rangle$ or not. To deduce the result of the algorithm, we only measure the x component, however the y component was crucial in allowing for an oracle to exist as unitarity would not necessarily be achievable otherwise.

Note that we apply here only three gates in total after initial state preparation, while in the classical implementation we need at least $2^{n-1}+1$ calls of f (section 2.4). Notice that in simulating the quantum effect with a classical computer, the computationally intensive part lies in creating the oracle. As we have to provide a binary computer with a function, we will have to provide the quantum computer with an oracle of said function.

Finally note that $|1\rangle$ can be produced from $|0\rangle$ via the Pauli-X gate:

$$X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

2.3. **Quantum Implementation.** While there are some great libraries for quantum computing in python, we've opted to implement the Deutsch-Jozsa algorithm without the help of one, in order to maximize visibility of what is occuring at every step. An example of output follows. The code is based on the quantum computation circuit detailed in 2.2 and should be self-explanatory[2]. Compare with the classical implementation 2.4. The Ψ_i in the print statements refer to the overal qubit state before and after applying hadamard, after applying the oracle, and the final state respectively. When running this simulation, be advised that increasing n also rapidly increases the size of our state, illustrating why simulating large quantum computers is so difficult. The code is also available on github: https://github.com/ZirconCode/quantum-deutsch-jozsa-public

```
## Deutsch-Jozsa Algorithm
# - Simon Gruening

# python v3.*
import numpy as np
import itertools
import random

# np.set_printoptions(suppress=True)


## Required gates
pauli_x_gate = [ [0,1],[1,0] ]
hadamard_gate = (1.0/(2**0.5)) * np.array([[1,1],[1,-1]])


```

[2] We were inspired in beginning parts by a fractional exposition of the material by Dawid Kopczyk [6]

```
## Prepare States
s0 = np.array( [[1.0],[0.0]] )
s1 = np.matmul(pauli_x_gate,s0)
# s1 = np.array( [[0.0],[1.0]] )


## Function f

# number of _bits_ f takes as input
n = 2
print("n =",n)

# function to be tested
# ex. for n=2:  f = {(0,0):0,(0,1):0,(1,0):0,(1,1):0} -> P=1

f_index = list(itertools.product([0, 1], repeat=n))
f = {k:0 for k in f_index}

if random.random()>0.5:
   # make balanced
  ones_i = np.random.choice(len(f_index),size=int(len(f_index)/2),replace=False)
  ones = [f_index[i] for i in ones_i]
  for o in ones:
    f[o] = 1

print("f:",f)


## Auxiliary functions

# tensor a list of arrays in given order
def kron(states):
  result = np.array([[1.0]])
  for s in states:
    result = np.kron(result,s)
  return result

# translate binary tuple to qubit array
def binary_to_qubit(binary_tuple):
  return kron([s0 if x == 0 else s1 for x in binary_tuple])


## Prepare state |00...1> = |x>|y> with y as the final qubit
```

```
prepared_state = kron((n*[s0])+[s1])
print("Psi_0: \n",prepared_state)

# apply hadamard individually to each qubit
tensored_h = kron([hadamard_gate]*(n+1))
print("Hadamard^(x)(n+1): \n",tensored_h)

pre_oracle_state = np.matmul(tensored_h,prepared_state)
print("Psi_1: \n",pre_oracle_state)

# print("norm:",np.linalg.norm(prepared_state))
# print("norm:",np.linalg.norm(pre_oracle_state))
# very slight loss due to accuracy of floats =)


## Create Oracle

# x,y -> x,y (+) f(x) where y is the last qubit
#   (+) is addition modulo 2
# column-wise creation
lst = list(itertools.product([0, 1], repeat=n+1)) # note order

oracle = np.array([],dtype=float)
for tmp in lst:
  x = tmp[0:-1]
  y = tmp[-1]
  f_x = f[x]

  qx = binary_to_qubit(x)
  qf_x = binary_to_qubit((f_x,))

  if y == 1:
    # addition modulo 2
    qf_x = np.mod(qf_x+1,2)

  column = kron([qx,qf_x])

  column = np.array([j[0] for j in column])

  oracle = np.concatenate((oracle,column)) #,axis=0)

qn = 2**(n+1)
oracle = np.transpose(oracle.reshape((qn,qn)))
print("Oracle: \n",oracle)
```

```


## Apply last two gates

# apply oracle
post_oracle_state = np.matmul(oracle,pre_oracle_state)
print("Psi_2: \n",post_oracle_state)

# apply hadamard to all but last (y) qubit
tensored_h_id = kron([hadamard_gate]*(n)+[np.identity(2)])
print("Hadamard^(x)n (x) id_2: \n", tensored_h_id)
end_state = np.matmul(tensored_h_id,post_oracle_state)

print("Psi_3: \n",end_state)


## Measure
# "ignore" last qubit (y)

print("Numerical error on norm: ",1-np.linalg.norm(end_state))

# projectors for |00..00y>, i.e. <y00...00| with y in {0,1}
P000 = np.transpose(kron([s0]*n+[s0]))
P001 = np.transpose(kron([s0]*n+[s1]))

# apply projectors
prob_000 =
 ↪  np.dot(end_state.reshape(2**(n+1)),P000.reshape(2**(n+1)))**2
prob_001 =
 ↪  np.dot(end_state.reshape(2**(n+1)),P001.reshape(2**(n+1)))**2
prob_total = prob_000+prob_001

print("Probability of |00...00>|0>: ",prob_000)
print("Probability of |00...00>|1>: ",prob_001)
print("Probability of |00...00>|y>: ",prob_total)

if round(prob_total) == 1:
  print("Thus the function is constant.")
else:
  print("Thus the function is balanced.")
```

See some sample output in the appendix B. Notice that there is introduced a very small numerical error on the norm which subsequently deviates ever so slightly from 1 as we perform operations.

2.4. **Classical Implementation.** We require $2^{n-1}+1$ steps inside of the loop in the worst case scenario.

```
# python v3.*
import numpy as np
import itertools
import random

n=4

## Create Function
f_index = list(itertools.product([0, 1], repeat=n))
f = {k:0 for k in f_index}

if random.random()>0.5:
   # make balanced
  ones_i =
↪   np.random.choice(len(f_index),size=int(len(f_index)/2),replace=False)
  ones = [f_index[i] for i in ones_i]
  for o in ones:
    f[o] = 1

print("f:",f)


## Check if Balanced
count = 0
state = -1
for i in f.keys():
  if state == -1:
    state = f[i]
  elif state == f[i]:
    count = count+1
    if count>(len(f.keys())/2): # maximum iterations
      print("Constant")
      break
  else:
    print("Balanced")
    break
```

Appendix A.
Table of Examples

Name	**Set**	**Rel**	**(F)Hilb**
Objects, Morphisms composition, identity	sets, functions function composition identity function	sets , $R \subseteq A \times B$ relation composition identity rel.	Hilbert spaces, bounded lin. maps function composition identity lin. maps
Isomorphisms	bijections	graphs of bijections	bijective morphisms
Product Coproduct	cartesian product disjoint union	disjoint union disjoint union	direct sum direct sum
Monoidal Category Tensor Product Tensor Unit	cartesian product $\{\bullet\}$	cartesian product $\{\bullet\}$	$\otimes$ of **Hilb** $I = \mathbb{C}$
State	$\{\bullet\} \to A$, "$\in A$"	$R : \{\bullet\} \to A$, "$\subseteq A$"	Lin. func. $\mathbb{C} \to H$ "$\in H$" by $im(1)$
Well Pointed	✓	✓	✓
Joint State Product State Entangled State	$\in A \times B$ All None	$U \subseteq A \times B$ $\exists V \subseteq A, W \subseteq B :$ $(v, w) \in U \iff$ $v \in V, w\ inW$ Others	$\in H \otimes K$ factorizable states non-fact. states
Braided Monoidal Cat.	$(a, b) \mapsto (b, a)$	$(a, b) \sim (b, a)$	Unique Lin. Map extending $a \otimes b \mapsto b \otimes a$
Scalars (monoid of)	$\{\bullet\} \mapsto \{\bullet\}$ trivial	$T = \emptyset, F = \{(\bullet, \bullet)\}$ Conjunction table	$\mathbb{C} \to \mathbb{C}$, consider Im(1) $\mathbb{C}$ under multipl.
(left) Scalar Multipl.	$id \bullet f = f$	$true \bullet R = R$, $false \bullet R = \emptyset$	$v \mapsto a \cdot f(v)$
Terminal Initial Zero (object) zero morph.	$\emptyset$ $\{\bullet\}$ No	$\emptyset$ $\emptyset$ $\emptyset$ empty relation	0-dim vec. space 0-dim vec. space 0-dim vec. space $x \mapsto$ zero vector
Superposition Rule	No	Union of subsets	element-wise addition of lin. maps
Biproducts	No	disjoint union of sets	direct sum of Hilb. spaces
Dagger Functor	No	relational converse	taking adjoints as bounded lin. maps
Braided Symmetric (mon. dag. cat.)		✓ ✓	✓ ✓
Dagger Biproduct		Every biproduct	orthogonal direct sums

Probability		effect X after state V is true if intersection is empty	$\| < x\|v > \|^2$
Dual (obj)	only for $\|S\| = 1$	Every object is self-dual	(F) take dual H^*
Right Dual		converse relation	(F) $f^* : W^* \to V^*$
Compact Cat.	✓	✓	✓
Ribbon Cat.		✓	✓
Compact Dag. Cat.		conjugation is identity	conj. is conjugate of lin. map
Trace		relation has fixpoint	(F) $\sum_i < i\|f\|i >$
Comonoid Comultipl. Counit	$A \to A \times A$ $A \to 1$	can be induced by a group	(F) $e_i \to e_i \otimes e_i$ $e_i \to 1$
Comonoid Homom.	any function	surj. homom. of groups	(F) any function between bases extends linearly
Uniform Deleting	✓		
Uniform Copying	✓ $a \mapsto (a, a)$		
Copyable States	All		(F) nonzero are orthogonal, length 1

Appendix B.
Sample Output

```
n = 2
f: {(0, 0): 0, (0, 1): 0, (1, 0): 1, (1, 1): 1}
Psi_0:
 [[0.]
 [1.]
 [0.]
 [0.]
 [0.]
 [0.]
 [0.]
 [0.]]
Hadamard^(x)(n+1):
 [[ 0.35355339  0.35355339  0.35355339  0.35355339  0.35355339
↪  0.35355339
   0.35355339  0.35355339]
 [ 0.35355339 -0.35355339  0.35355339 -0.35355339  0.35355339
↪  -0.35355339
```

```
   0.35355339 -0.35355339]
 [ 0.35355339  0.35355339 -0.35355339 -0.35355339  0.35355339
↪  0.35355339
  -0.35355339 -0.35355339]
 [ 0.35355339 -0.35355339 -0.35355339  0.35355339  0.35355339
↪  -0.35355339
  -0.35355339  0.35355339]
 [ 0.35355339  0.35355339  0.35355339  0.35355339 -0.35355339
↪  -0.35355339
  -0.35355339 -0.35355339]
 [ 0.35355339 -0.35355339  0.35355339 -0.35355339 -0.35355339
↪  0.35355339
  -0.35355339  0.35355339]
 [ 0.35355339  0.35355339 -0.35355339 -0.35355339 -0.35355339
↪  -0.35355339
   0.35355339  0.35355339]
 [ 0.35355339 -0.35355339 -0.35355339  0.35355339 -0.35355339
↪  0.35355339
   0.35355339 -0.35355339]]
Psi_1:
 [[ 0.35355339]
 [-0.35355339]
 [ 0.35355339]
 [-0.35355339]
 [ 0.35355339]
 [-0.35355339]
 [ 0.35355339]
 [-0.35355339]]
Oracle:
 [[1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0. 1. 0.]]
Psi_2:
 [[ 0.35355339]
 [-0.35355339]
 [ 0.35355339]
 [-0.35355339]
 [-0.35355339]
 [ 0.35355339]
 [-0.35355339]
 [ 0.35355339]]
Hadamard^(x)n (x) id_2:
[[ 0.5  0.   0.5  0.   0.5  0.   0.5  0. ]
 [ 0.   0.5  0.   0.5  0.   0.5  0.   0.5]
 [ 0.5  0.  -0.5 -0.   0.5  0.  -0.5 -0. ]
 [ 0.   0.5 -0.  -0.5  0.   0.5 -0.  -0.5]
 [ 0.5  0.   0.5  0.  -0.5 -0.  -0.5 -0. ]
 [ 0.   0.5  0.   0.5 -0.  -0.5 -0.  -0.5]
 [ 0.5  0.  -0.5 -0.  -0.5 -0.   0.5  0. ]
 [ 0.   0.5 -0.  -0.5 -0.  -0.5  0.   0.5]]
Psi_3:
 [[-2.29934717e-17]
 [ 2.29934717e-17]
 [ 0.00000000e+00]
 [ 0.00000000e+00]
 [ 7.07106781e-01]
 [-7.07106781e-01]
 [ 0.00000000e+00]
 [ 0.00000000e+00]]
Numerical error on norm:  4.440892098500626e-16
Probability of |00...00>|0>:  5.286997409534844e-34
Probability of |00...00>|1>:  5.286997409534844e-34
Probability of |00...00>|y>:  1.0573994819069687e-33
Thus the function is balanced.
```

References

[1] J. Vicary, “Categories, quantum computation and topology.” Summer 2016.

[2] F. W. LAWVERE, “An elementary theory of the category of sets (long version) with commentary,” *Reprints in Theory and Applications of Categories*, 2005.

[3] D. Deutsch and R. Jozsa, “Rapid solution of problems by quantum compputation,” *Proc. R. Soc. Lond. A*, 1992.

[4] N. J. Åke Larsson, “Efficient classical simulation of the deutsch–jozsaand simon’s algorithms,” *Quantum Inf Process*, 2017.

[5] M. Loceff, *A Course in Quantum Computing*. 2015.

[6] D. Kopczyk, “Quantum Computing in Python: Introduction.”