Category Theory for Quantum Computing

A Self-Contained Tutorial

Prerequisites for ZX-Calculus and Diagrammatic Reasoning

March 25, 2026

Contents

Chapter 1

Why Category Theory for Quantum Computing?

Quantum computing sits at the intersection of physics, computer science, and mathematics. At its core, quantum computation manipulates states in Hilbert spaces via linear maps, and composes these operations into circuits. Category theory provides a unifying language for all of these structures.

1.1 The Unifying Perspective

Consider the different ways we describe quantum computation:

- **Hilbert space picture**: states are vectors in $\mathbb{C}^{2^n}$, operations are unitary matrices.
- **Circuit picture**: qubits are wires, gates are boxes composed in sequence and parallel.
- **Algebraic picture**: observables form C*-algebras, states are positive linear functionals.

These are not competing descriptions—they are *different views of the same structure*. Category theory makes this precise: each view corresponds to a *category*, and the translations between views are *functors*.

1.2 Why Graphical Calculi Need Categories

The ZX-calculus and its relatives replace matrix algebra with graphical rewriting rules applied to string diagrams. These diagrams look like circuit diagrams, but with a crucial difference: they obey topological rules (you can bend wires, slide boxes) that arise from the axioms of *monoidal categories*.

Without the categorical foundation, the graphical rules are merely heuristic pictures. With it, every diagram manipulation corresponds to a rigorous algebraic equation, and we can prove *soundness* (every diagrammatic rule corresponds to a true equation) and *completeness* (every true equation can be derived diagrammatically).

1.3 Roadmap

This document introduces the category theory needed for quantum diagrammatic reasoning:

1. **Categories** (Chapter 2): composing quantum processes.
2. **Functors** (Chapter 3): translating between formalisms.

3. **Monoidal categories** (Chapter 4): parallel quantum systems.
4. **String diagrams** (Chapter 5): the graphical language.
5. **Symmetric monoidal categories** (Chapter 6): swapping systems.
6. **Dagger categories** (Chapter 7): adjoints and unitaries.
7. **Compact closed categories** (Chapter 8): entanglement and duality.
8. **Monoids and comonoids** (Chapter 9): copying and measurement.
9. **Frobenius algebras** (Chapter 10): the heart of ZX-calculus.
10. **Bialgebras and Hopf algebras** (Chapter 11): complementary bases.
11. **Props and graphical theories** (Chapter 12): the big picture.

Each chapter begins with a quantum computing motivation before presenting formal definitions.

Chapter 2

Composing Quantum Processes → Categories

2.1 Motivation: Sequential Composition of Gates

A quantum computation is built by composing gates in sequence. If gate $g\colon \mathbb{C}^2 \to \mathbb{C}^2$ is followed by gate $h\colon \mathbb{C}^2 \to \mathbb{C}^2$, the composite operation is the matrix product $h \circ g$. This sequential composition is:

- *Associative*: $(f \circ g) \circ h = f \circ (g \circ h)$.
- *Has identities*: the "do nothing" gate I satisfies $g \circ I = g = I \circ g$.

These are exactly the axioms of a *category*.

2.2 The Definition

Definition 2.1 (Category)**.** A *category* $\mathcal{C}$ consists of:

1. A collection $\mathrm{ob}(\mathcal{C})$ of *objects*.
2. For each pair of objects A, B, a collection $\mathrm{Hom}_{\mathcal{C}}(A, B)$ of *morphisms* (or *arrows*) from A to B. We write $f\colon A \to B$ for $f \in \mathrm{Hom}_{\mathcal{C}}(A, B)$.
3. For each triple of objects A, B, C, a *composition* function

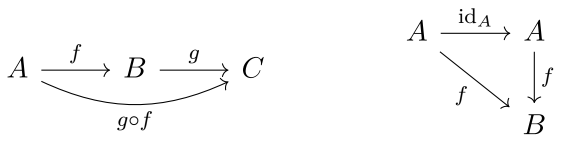
$$\circ\colon \mathrm{Hom}(B, C) \times \mathrm{Hom}(A, B) \to \mathrm{Hom}(A, C), \qquad (g, f) \mapsto g \circ f.$$

4. For each object A, an *identity morphism* $\mathrm{id}_A\colon A \to A$.

Subject to:

- *Associativity*: $h \circ (g \circ f) = (h \circ g) \circ f$ for all composable f, g, h.
- *Identity*: $f \circ \mathrm{id}_A = f = \mathrm{id}_B \circ f$ for all $f\colon A \to B$.

The associativity and identity axioms can be expressed as commutative diagrams:



2.3 Key Examples

Example 2.2 (**Set**)**.** The category **Set** has sets as objects and functions as morphisms. Composition is function composition; identities are identity functions.

Example 2.3 (**Vect**)**.** The category **Vect** (over $\mathbb{C}$) has complex vector spaces as objects and linear maps as morphisms. This is the ambient category for quantum mechanics.

Example 2.4 (**FdHilb** — the quantum category)**.** The category **FdHilb** has:

- *Objects*: finite-dimensional Hilbert spaces ($\mathbb{C}^n$ with the standard inner product).
- *Morphisms*: linear maps $f\colon H \to K$.
- *Composition*: composition of linear maps (matrix multiplication).
- *Identity*: the identity matrix I_n on each $\mathbb{C}^n$.

This is the category where quantum computing "lives." Every quantum gate, measurement operator, and state preparation is a morphism in **FdHilb**.

Example 2.5 (**Mat**)**.** The category **Mat** has natural numbers $0, 1, 2, \ldots$ as objects. A morphism from n to m is an $m \times n$ complex matrix. Composition is matrix multiplication. This is a "skeletalized" version of **FdHilb**—it remembers only dimensions.

2.4 Commutative Diagrams

A *commutative diagram* is a diagram of objects and morphisms where any two paths with the same start and endpoint give the same composite morphism. For example, the diagram

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ {\scriptstyle h}\big\downarrow & & \big\downarrow{\scriptstyle g} \\ C & \xrightarrow[k]{} & D \end{array}$$

commutes if $g \circ f = k \circ h$. Commutative diagrams are the primary tool for stating equations in category theory.

2.5 Isomorphisms

Definition 2.6 (Isomorphism)**.** A morphism $f\colon A \to B$ is an *isomorphism* if there exists $g\colon B \to A$ such that $g \circ f = \mathrm{id}_A$ and $f \circ g = \mathrm{id}_B$. We write $A \cong B$.

In **FdHilb**, an isomorphism is an invertible linear map. In quantum computing, unitary gates $U\colon \mathbb{C}^n \to \mathbb{C}^n$ satisfy $U^\dagger U = I$, so they are isomorphisms (in fact, they are special isomorphisms—see Chapter 7).

Chapter 3

Translating Between Formalisms → Functors

3.1 Motivation: Changing Representations

In quantum computing, we constantly switch between representations:

- A circuit diagram describes *which gates* are applied.
- The matrix representation computes *what happens* numerically.
- The Hilbert space picture identifies *the physical spaces* involved.

Each of these translations preserves composition (a sequence of gates translates to a product of matrices) and identities (the "do nothing" circuit translates to the identity matrix). A structure-preserving map between categories is called a *functor*.

3.2 The Definition

Definition 3.1 (Functor)**.** A *functor* $F\colon \mathcal{C} \to \mathcal{D}$ between categories consists of:

1. A mapping on objects: $A \mapsto F(A)$ for each $A \in \mathrm{ob}(\mathcal{C})$.
2. A mapping on morphisms: $f \mapsto F(f)$ for each $f\colon A \to B$, with $F(f)\colon F(A) \to F(B)$.

Subject to:

- *Preservation of composition*: $F(g \circ f) = F(g) \circ F(f)$.
- *Preservation of identities*: $F(\mathrm{id}_A) = \mathrm{id}_{F(A)}$.

This is expressed by the commutative diagram:

$$\begin{array}{ccccc} A & \xrightarrow{f} & B & \xrightarrow{g} & C \\ {\scriptstyle F}\downarrow & & {\scriptstyle F}\downarrow & & {\scriptstyle F}\downarrow \\ F(A) & \xrightarrow[F(f)]{} & F(B) & \xrightarrow[F(g)]{} & F(C) \end{array}$$

3.3 Key Examples

Example 3.2 (Forgetful functor **FdHilb** → **Vect**)**.** Every Hilbert space is a vector space (just "forget" the inner product). This defines a functor $U\colon \mathbf{FdHilb} \to \mathbf{Vect}$ that sends each Hilbert space to its underlying vector space, and each linear map to itself.

Example 3.3 (Dimension functor)**.** The functor $\dim\colon \mathbf{FdHilb} \to \mathbf{Mat}$ sends each Hilbert space H to its dimension $\dim(H) \in \mathbb{N}$, and each linear map to its matrix representation. This functor is an *equivalence of categories*: **FdHilb** and **Mat** are essentially the same category up to isomorphism of objects.

Example 3.4 (The state functor)**.** The functor $\mathbb{C}\text{-Map}\colon \mathbf{Set} \to \mathbf{Vect}$ sends each set S to the free vector space $\mathbb{C}[S]$ with basis S. Quantum computing uses this when embedding classical data into quantum states: a classical bit $b \in \{0, 1\}$ maps to the basis state $|b\rangle \in \mathbb{C}^2$.

3.4 Natural Transformations (Brief)

Definition 3.5 (Natural transformation)**.** Given functors $F, G\colon \mathcal{C} \to \mathcal{D}$, a *natural transformation* $\alpha\colon F \Rightarrow G$ is a family of morphisms $\alpha_A\colon F(A) \to G(A)$ (one for each object A in $\mathcal{C}$) such that for every $f\colon A \to B$:

$$\begin{array}{ccc} F(A) & \xrightarrow{\alpha_A} & G(A) \\ {\scriptstyle F(f)}\downarrow & & \downarrow{\scriptstyle G(f)} \\ F(B) & \xrightarrow[\alpha_B]{} & G(B) \end{array}$$

commutes.

Natural transformations appear implicitly in our source material whenever a construction is "basis-independent" or "natural in the input." We will not need their full theory, but the concept underlies many of the coherence conditions in later chapters.

Chapter 4

Parallel Quantum Systems → Monoidal Categories

4.1 Motivation: Tensor Products and Parallel Wires

A single qubit lives in $\mathbb{C}^2$. Two qubits live in $\mathbb{C}^2 \otimes \mathbb{C}^2 \cong \mathbb{C}^4$. The tensor product $\otimes$ combines systems in parallel:

- *States*: $|\psi\rangle \otimes |\phi\rangle$ is the joint state.
- *Gates*: $U \otimes V$ applies U to the first qubit and V to the second.
- *Wires*: parallel wires in a circuit diagram represent the tensor product.

A category equipped with a "tensor product" operation satisfying natural coherence conditions is called a *monoidal category*.

4.2 The Definition

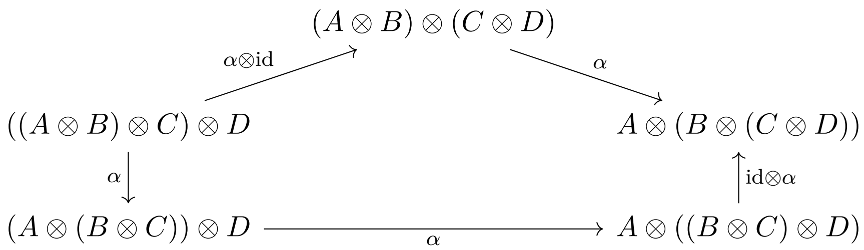
Definition 4.1 (Monoidal category)**.** A *monoidal category* $(\mathcal{C}, \otimes, I, \alpha, \lambda, \rho)$ consists of:

1. A category $\mathcal{C}$.
2. A bifunctor $\otimes \colon \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ (the *tensor product*).
3. A distinguished object I (the *monoidal unit* or *tensor unit*).
4. Natural isomorphisms:

$$\alpha_{A,B,C} \colon (A \otimes B) \otimes C \xrightarrow{\sim} A \otimes (B \otimes C) \qquad \text{(associator)}$$
$$\lambda_A \colon I \otimes A \xrightarrow{\sim} A \qquad \text{(left unitor)}$$
$$\rho_A \colon A \otimes I \xrightarrow{\sim} A \qquad \text{(right unitor)}$$

Subject to the *pentagon axiom* and the *triangle axiom*.

The **pentagon axiom** states that the following diagram commutes for all objects A, B, C, D:



The **triangle axiom** states that for all A, B:

$$\begin{array}{ccc} (A \otimes I) \otimes B & \xrightarrow{\quad\alpha\quad} & A \otimes (I \otimes B) \\ & {\scriptstyle \rho \otimes \mathrm{id}} \searrow \quad \swarrow {\scriptstyle \mathrm{id} \otimes \lambda} & \\ & A \otimes B & \end{array}$$

4.3 Strict Monoidal Categories and Coherence

Definition 4.2 (Strict monoidal category)**.** A monoidal category is *strict* if α, λ, and ρ are all identity morphisms—i.e., $(A \otimes B) \otimes C = A \otimes (B \otimes C)$ and $I \otimes A = A = A \otimes I$ on the nose.

Theorem 4.3 (Mac Lane's Coherence Theorem)**.** *Every monoidal category is monoidally equivalent to a strict one.*

The practical consequence: we may suppress parentheses and unitors in calculations, writing $A \otimes B \otimes C$ without ambiguity. This is essential for string diagrams (Chapter 5), where associativity is built into the graphical notation.

4.4 Key Examples

Example 4.4 (($\mathbf{FdHilb}, \otimes, \mathbb{C}$))**.** The category **FdHilb** is monoidal with:

- Tensor product: $H \otimes K$ is the Hilbert space tensor product.
- Unit: $I = \mathbb{C}$ (the one-dimensional Hilbert space).
- On morphisms: $(f \otimes g)(v \otimes w) = f(v) \otimes g(w)$.

This is the monoidal structure underlying multi-qubit systems: $(\mathbb{C}^2)^{\otimes n} = \underbrace{\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{n}$.

Example 4.5 (($\mathbf{Set}, \times, \{*\}$))**.** The category **Set** is monoidal with cartesian product $\times$ and the one-element set $\{*\}$.

Example 4.6 (($\mathbf{Mat}, \otimes, 1$))**.** In **Mat**, the tensor product on objects is multiplication of natural numbers ($m \otimes n = mn$), and on morphisms it is the Kronecker product of matrices. The unit is 1.

Chapter 5

String Diagrams — The Graphical Language

5.1 Motivation: Quantum Circuits Are String Diagrams

Every physicist knows quantum circuit notation: qubits are horizontal wires, gates are boxes on wires, and time flows left to right. String diagrams are a *generalization* of circuit notation to arbitrary monoidal categories, with a rigorous mathematical foundation.

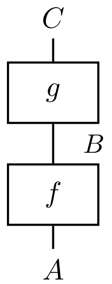
5.2 The Basic Dictionary

In a string diagram for a monoidal category:

Category theory		**String diagram**
Object A	$\longleftrightarrow$	Wire labeled A
Morphism $f\colon A \to B$	$\longleftrightarrow$	Box labeled f with input wire A, output wire B
Identity id_A	$\longleftrightarrow$	Plain wire (no box)
Composition $g \circ f$	$\longleftrightarrow$	Connect output of f to input of g (vertically)
Tensor $f \otimes g$	$\longleftrightarrow$	Place f and g side by side (horizontally)
Unit I	$\longleftrightarrow$	Empty region (no wire)

5.3 Sequential Composition

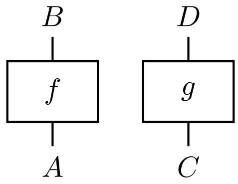
A morphism $f\colon A \to B$ followed by $g\colon B \to C$ is drawn vertically (we adopt the convention that diagrams are read *bottom to top*):



$$= \quad g \circ f\colon A \to C$$

5.4 Parallel Composition (Tensor Product)

Two morphisms $f\colon A \to B$ and $g\colon C \to D$ placed side by side represent $f \otimes g\colon A \otimes C \to B \otimes D$:



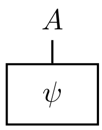
$$= \quad f \otimes g$$

5.5 The Identity and the Unit

The identity morphism id_A is a plain wire—no box:

$$= \quad \mathrm{id}_A$$

The monoidal unit I corresponds to the *absence* of a wire. A morphism $\psi\colon I \to A$ (a "state") is drawn as a wire with no input:



$$= \quad \psi\colon I \to A$$

In quantum computing, states $|\psi\rangle \in \mathbb{C}^n$ are morphisms $\mathbb{C} \to \mathbb{C}^n$, so they appear as wires emerging from a box at the bottom of the diagram.

5.6 Soundness and Completeness

Theorem 5.1 (Joyal–Street)**.** *String diagrams are* sound and complete *for monoidal categories: two morphisms in a monoidal category are equal if and only if their string diagrams are related by planar isotopy (continuous deformation preserving connectivity).*

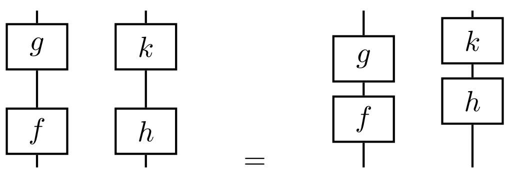
This theorem is what gives graphical calculi their power: *only the topology matters*, not the precise geometry. You can stretch, bend, and slide boxes along wires freely. This is the mathematical foundation underlying the ZX-calculus, abstract wiring diagrams, and every other graphical language for quantum computation.

5.7 Interchange Law

A key consequence of functoriality of $\otimes$ is the *interchange law*: for $f\colon A \to B$, $g\colon B \to C$, $h\colon A' \to B'$, $k\colon B' \to C'$,

$$(g \otimes k) \circ (f \otimes h) = (g \circ f) \otimes (k \circ h).$$

In string diagrams, this is automatic: sliding boxes vertically on separate wires doesn't change the topology.



5.8 Connection to Abstract Wiring Diagrams

The "abstract wiring diagrams" used in Wakeham's SIQP lectures are a variant of string diagrams adapted to the C*-algebraic framework. The key ideas are the same: boxes represent processes, wires represent systems, and composition is by connecting compatible ports. The categorical structure ensures that all manipulations of these diagrams are algebraically sound.

Chapter 6

Swapping Systems → Symmetric Monoidal Categories

6.1 Motivation: The SWAP Gate

In a quantum circuit, we often need to reorder qubits. The SWAP gate exchanges two qubits:

$$\mathrm{SWAP}\colon \mathbb{C}^2 \otimes \mathbb{C}^2 \to \mathbb{C}^2 \otimes \mathbb{C}^2, \qquad |a\rangle \otimes |b\rangle \mapsto |b\rangle \otimes |a\rangle.$$

This operation is natural (it doesn't depend on the choice of basis) and satisfies $\mathrm{SWAP} \circ \mathrm{SWAP} = \mathrm{id}$. These properties generalize to *braidings* and *symmetries* in monoidal categories.

6.2 The Definition

Definition 6.1 (Braided monoidal category)**.** A *braided monoidal category* is a monoidal category $(\mathcal{C}, \otimes, I)$ equipped with a natural isomorphism

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

called the *braiding*, satisfying coherence conditions (hexagon axioms).

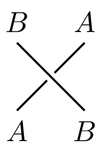
Definition 6.2 (Symmetric monoidal category)**.** A braided monoidal category is *symmetric* if the braiding satisfies

$$\sigma_{B,A} \circ \sigma_{A,B} = \mathrm{id}_{A \otimes B}$$

for all objects A, B—i.e., swapping twice is the identity.

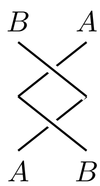
6.3 String Diagrams: Crossing Wires

In a symmetric monoidal category, the braiding $\sigma_{A,B}$ is drawn as a crossing of wires:

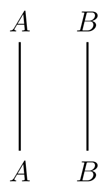


$$= \qquad \sigma_{A,B}\colon A \otimes B \to B \otimes A$$

The symmetry condition $\sigma_{B,A} \circ \sigma_{A,B} = \mathrm{id}$ means that two crossings cancel—they can be "pulled straight":



$=$



In a *braided* (but not symmetric) category, overcrossings and undercrossings are distinguished, much like in knot theory. For quantum computing, we almost always work in the symmetric setting.

6.4 Examples

Example 6.3 (**FdHilb** is symmetric monoidal)**.** In **FdHilb**, the braiding is the SWAP map: $\sigma_{H,K}(v \otimes w) = w \otimes v$. This is the mathematical content of the SWAP gate. Since $\sigma_{K,H} \circ \sigma_{H,K} = \mathrm{id}$, we have a symmetric monoidal category.

Example 6.4 (Naturality of SWAP)**.** For any linear maps $f\colon H \to H'$ and $g\colon K \to K'$, the diagram

$$\begin{array}{ccc} H \otimes K & \xrightarrow{\sigma} & K \otimes H \\ {\scriptstyle f\otimes g}\downarrow & & \downarrow{\scriptstyle g\otimes f} \\ H' \otimes K' & \xrightarrow[\sigma]{} & K' \otimes H' \end{array}$$

commutes. In string diagrams, this means we can "slide" boxes through crossings.

Chapter 7

Adjoints and Unitaries → Dagger Categories

7.1 Motivation: The Adjoint in Quantum Mechanics

In quantum mechanics, every linear map $f\colon H \to K$ between Hilbert spaces has an *adjoint* $f^\dagger\colon K \to H$, defined by $\langle f^\dagger(w), v\rangle_H = \langle w, f(v)\rangle_K$. The adjoint satisfies:

- $(g \circ f)^\dagger = f^\dagger \circ g^\dagger$ (reverses composition order).
- $(f^\dagger)^\dagger = f$ (involutive).
- $\mathrm{id}_H^\dagger = \mathrm{id}_H$.

Unitaries ($U^\dagger U = I$) and self-adjoints ($H^\dagger = H$) are fundamental.

7.2 The Definition

Definition 7.1 (Dagger category)**.** A *dagger category* (or †-category) is a category $\mathcal{C}$ equipped with an *identity-on-objects* contravariant functor $\dagger\colon \mathcal{C}^{\mathrm{op}} \to \mathcal{C}$ that is involutive: $(f^\dagger)^\dagger = f$ for all morphisms f.

Explicitly, for each morphism $f\colon A \to B$, there is a morphism $f^\dagger\colon B \to A$ such that:

1. $\mathrm{id}_A^\dagger = \mathrm{id}_A$.
2. $(g \circ f)^\dagger = f^\dagger \circ g^\dagger$.
3. $(f^\dagger)^\dagger = f$.

Definition 7.2 (Unitary and self-adjoint morphisms)**.** In a dagger category:

- A morphism $f\colon A \to B$ is *unitary* if $f^\dagger \circ f = \mathrm{id}_A$ and $f \circ f^\dagger = \mathrm{id}_B$.
- An endomorphism $f\colon A \to A$ is *self-adjoint* (or *Hermitian*) if $f^\dagger = f$.

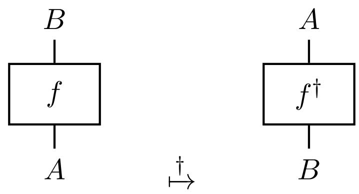
7.3 Dagger Monoidal Categories

Definition 7.3 (Dagger monoidal category)**.** A *dagger monoidal category* is a monoidal category $(\mathcal{C}, \otimes, I)$ that is also a dagger category, such that:

- $(f \otimes g)^\dagger = f^\dagger \otimes g^\dagger$.
- The associator, unitors, and (if present) braiding are all unitary.

7.4 String Diagram Notation

In string diagrams, the dagger $f^\dagger$ is depicted by *reflecting* the diagram for f across a horizontal axis and reversing the direction of all wires:



The condition $(g \circ f)^\dagger = f^\dagger \circ g^\dagger$ is automatic in this notation: reflecting a vertical stack of boxes reverses their order.

7.5 The Key Example

Example 7.4 (**FdHilb** as a dagger category)**. FdHilb** is a dagger compact category (combining all the structures from this chapter and Chapter 8). The dagger is the adjoint $f \mapsto f^\dagger$ defined by the Hilbert space inner product.

- *Unitary morphisms*: quantum gates ($U^\dagger U = I$).
- *Self-adjoint morphisms*: observables ($H^\dagger = H$).
- *Positive morphisms*: density operators ($\rho = A^\dagger A$ for some A).

Chapter 8

Entanglement and Duality → Compact Closed Categories

8.1 Motivation: Bell States and Teleportation

The Bell state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ creates a maximally entangled pair of qubits. Quantum teleportation uses a Bell state ("entanglement resource") and a Bell measurement ("consuming the resource") to transmit a qubit. Categorically, the Bell state is a *cup* (coevaluation) and the Bell measurement is a *cap* (evaluation), and teleportation follows from the *snake equations*.

8.2 Dual Objects

Definition 8.1 (Duality)**.** In a monoidal category $(\mathcal{C}, \otimes, I)$, an object A^* is a *(right) dual* of A if there exist morphisms

$$\eta_A \colon I \to A \otimes A^* \qquad \text{(coevaluation / unit / cup)}$$
$$\varepsilon_A \colon A^* \otimes A \to I \qquad \text{(evaluation / counit / cap)}$$

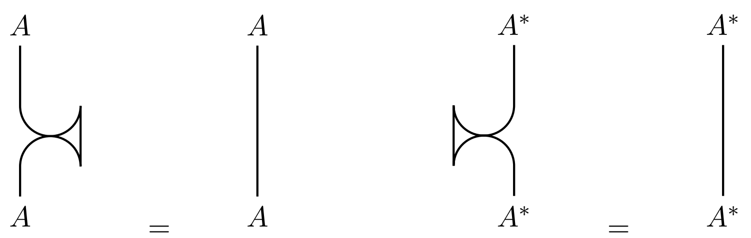
satisfying the *snake equations* (or *zig-zag identities*):

$$(\varepsilon_A \otimes \mathrm{id}_A) \circ (\mathrm{id}_A \otimes \eta_A) = \mathrm{id}_A \tag{8.1}$$
$$(\mathrm{id}_{A^*} \otimes \varepsilon_A) \circ (\eta_A \otimes \mathrm{id}_{A^*}) = \mathrm{id}_{A^*} \tag{8.2}$$

8.3 String Diagrams: Cups and Caps

The cup (coevaluation) and cap (evaluation) have characteristic string diagram shapes:

The snake equations (8.1) and (8.2) state that a bent wire can be pulled straight:



These "snake" diagrams motivate the name "zig-zag identities."

8.4 Compact Closed Categories

Definition 8.2 (Compact closed category)**.** A *compact closed category* is a symmetric monoidal category in which every object A has a dual A^*, with cups and caps satisfying the snake equations.

Definition 8.3 (Dagger compact category)**.** A *dagger compact category* is a compact closed category that is also a dagger category, with the additional requirement that $\varepsilon_A = \sigma_{A,A^*} \circ \eta_A^\dagger$ (the cap is the "daggered and swapped" cup).

Theorem 8.4 (Abramsky–Coecke)**.** **FdHilb** *is a dagger compact category with $H^* = \bar{H}$ (the conjugate Hilbert space),*

$$\eta_H \colon \mathbb{C} \to H \otimes \bar{H}, \qquad 1 \mapsto \sum_i |e_i\rangle \otimes \overline{|e_i\rangle}$$

where $\{|e_i\rangle\}$ is an orthonormal basis of H.

8.5 Quantum Teleportation as Snake Equations

In quantum computing terms:

- The cup $\eta \colon \mathbb{C} \to \mathbb{C}^2 \otimes \mathbb{C}^2$ creates a Bell state $|\Phi^+\rangle$.
- The cap $\varepsilon \colon \mathbb{C}^2 \otimes \mathbb{C}^2 \to \mathbb{C}$ performs a Bell measurement.
- The snake equation says: create a Bell pair, then measure one half with the qubit to teleport → the qubit emerges on the other end. This is teleportation!

The snake equation is not just a formal identity—it is the *proof that teleportation works*.

8.6 The Choi–Jamiołkowski Isomorphism

In a compact closed category, there is a canonical bijection

$$\mathrm{Hom}(A, B) \cong \mathrm{Hom}(I, B \otimes A^*)$$

sending each morphism $f \colon A \to B$ to the "Choi state" $(f \otimes \mathrm{id}_{A^*}) \circ \eta_A$. In quantum information, this is the *Choi–Jamiołkowski isomorphism*: every quantum channel is equivalent to a bipartite state (its Choi matrix). The compact closed structure makes this bijection natural and canonical.

Chapter 9

Copying and Measurement → Monoids and Comonoids

9.1 Motivation: Classical Data Can Be Copied

Classical bits can be freely copied ($b \mapsto (b, b)$) and deleted ($b \mapsto *$). Quantum data cannot—this is the *no-cloning theorem*. Categorically, copying and deleting are the operations of a *comonoid*, and the no-cloning theorem says that **FdHilb** does not have a "universal" comonoid on every object.

9.2 Monoids

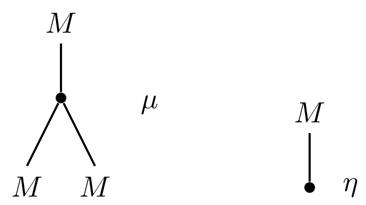
Definition 9.1 (Monoid in a monoidal category)**.** A *monoid* in a monoidal category $(\mathcal{C}, \otimes, I)$ is an object M equipped with:

- A *multiplication* morphism $\mu\colon M \otimes M \to M$.
- A *unit* morphism $\eta\colon I \to M$.

Satisfying associativity and unitality:

$$\begin{array}{ccc} M \otimes M \otimes M & \xrightarrow{\mu \otimes \mathrm{id}} & M \otimes M \\ {\scriptstyle \mathrm{id} \otimes \mu}\downarrow & & \downarrow{\scriptstyle \mu} \\ M \otimes M & \xrightarrow[\mu]{} & M \end{array} \qquad \begin{array}{ccccc} I \otimes M & \xrightarrow{\eta \otimes \mathrm{id}} & M \otimes M & \xleftarrow{\mathrm{id} \otimes \eta} & M \otimes I \\ & {\scriptstyle \lambda}\searrow & \downarrow{\scriptstyle \mu} & \swarrow{\scriptstyle \rho} & \\ & & M & & \end{array}$$

In string diagrams, we draw the multiplication as a "merging" node and the unit as a small node with no inputs:



9.3 Comonoids

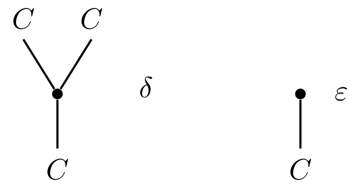
Definition 9.2 (Comonoid)**.** A *comonoid* in $(\mathcal{C}, \otimes, I)$ is an object C with:

- A *comultiplication* $\delta\colon C \to C \otimes C$ ("copying").

- A *counit* $\varepsilon \colon C \to I$ ("deleting").

Satisfying coassociativity and counitality (the axioms of a monoid, reversed).

String diagrams for comonoids mirror those for monoids, but upside down:



9.4 Classical Data: Copying and Deleting in Set

Example 9.3 (Comonoid in $(\mathbf{Set}, \times, \{*\})$)**.** Every set S carries a canonical comonoid:

$$\delta \colon S \to S \times S, \qquad s \mapsto (s, s) \qquad \text{(diagonal / copy)}$$
$$\varepsilon \colon S \to \{*\}, \qquad s \mapsto * \qquad \text{(terminal / delete)}$$

This is why classical information can be freely copied and discarded.

9.5 No-Cloning from a Categorical Perspective

Theorem 9.4 (Categorical no-cloning)**.** *In* **FdHilb***, there is no natural family of comultiplications* $\delta_H \colon H \to H \otimes H$ *that copies all states. Specifically, there is no morphism* $\delta \colon \mathbb{C}^2 \to \mathbb{C}^2 \otimes \mathbb{C}^2$ *such that* $\delta(|\psi\rangle) = |\psi\rangle \otimes |\psi\rangle$ *for all* $|\psi\rangle$*.*

The proof is straightforward: if $\delta(|0\rangle) = |00\rangle$ and $\delta(|1\rangle) = |11\rangle$, then by linearity $\delta(|+\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \neq |+\rangle \otimes |+\rangle$.

However, individual orthonormal bases *can* be copied. The comonoid $\delta(|i\rangle) = |i\rangle \otimes |i\rangle$ (extended linearly) defines a comultiplication that copies the basis $\{|i\rangle\}$ but not superpositions. This observation is crucial for Frobenius algebras (Chapter 10).

Chapter 10

The Heart of ZX-Calculus → Frobenius Algebras

10.1 Motivation: Spiders in the ZX-Calculus

In the ZX-calculus, the fundamental building blocks are *Z-spiders* (green nodes) and *X-spiders* (red nodes). A Z-spider with m inputs and n outputs represents the linear map

$$\underbrace{|0\cdots0\rangle}_{n}\underbrace{\langle0\cdots0|}_{m}+e^{i\alpha}\underbrace{|1\cdots1\rangle}_{n}\underbrace{\langle1\cdots1|}_{m}$$

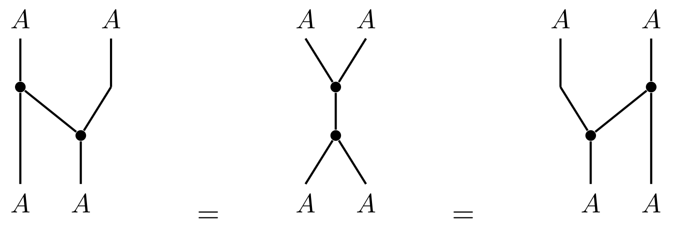
where α is the spider's phase. The crucial algebraic fact is that Z-spiders obey the *spider fusion rule*: any connected network of Z-spiders with the same total number of inputs and outputs, and the same total phase, is equal. This is precisely the *spider theorem* for *special commutative Frobenius algebras*.

10.2 Frobenius Algebras

Definition 10.1 (Frobenius algebra)**.** A *Frobenius algebra* in a monoidal category $(\mathcal{C}, \otimes, I)$ is an object A that is simultaneously a monoid (A, μ, η) and a comonoid (A, δ, ε) satisfying the *Frobenius condition*:

$$(\mu \otimes \mathrm{id}_A) \circ (\mathrm{id}_A \otimes \delta) = \delta \circ \mu = (\mathrm{id}_A \otimes \mu) \circ (\delta \otimes \mathrm{id}_A).$$

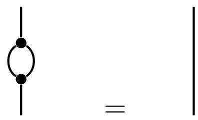
In string diagrams, the Frobenius condition says that the following three diagrams are equal:



The Frobenius condition is what allows wires to "bend" through nodes—it connects the monoid and comonoid structures in exactly the right way.

10.3 Special and Commutative Frobenius Algebras

Definition 10.2 (Special Frobenius algebra)**.** A Frobenius algebra is *special* if $\mu \circ \delta = \mathrm{id}_A$ (composing the comultiplication with the multiplication gives the identity):

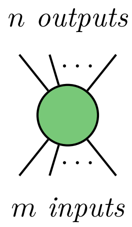


Definition 10.3 (Commutative Frobenius algebra)**.** A Frobenius algebra is *commutative* if $\mu \circ \sigma_{A,A} = \mu$ (multiplication doesn't depend on input order) and *cocommutative* if $\sigma_{A,A} \circ \delta = \delta$.

Definition 10.4 (SCFA)**.** A *special commutative Frobenius algebra* (SCFA) is a Frobenius algebra that is both special and commutative (and cocommutative).

10.4 The Spider Theorem

Theorem 10.5 (Spider theorem)**.** *In an SCFA, any connected diagram with m inputs and n outputs built entirely from the monoid and comonoid operations equals a single "spider":*



The spider depends only on m and n, not on the internal connectivity.

This is exactly why ZX-calculus works: any connected network of Z-spiders (which form an SCFA on $\mathbb{C}^2$ with respect to the $\{|0\rangle, |1\rangle\}$ basis) can be fused into a single Z-spider. Similarly for X-spiders (which form an SCFA with respect to the $\{|+\rangle, |-\rangle\}$ basis).

10.5 SCFAs and Orthonormal Bases

Theorem 10.6 (Classical structures theorem)**.** *In* **FdHilb**, *SCFAs on a Hilbert space H are in bijective correspondence with orthonormal bases of H.*

Given an orthonormal basis $\{|i\rangle\}_{i=1}^n$ of H, the corresponding SCFA has:

$\delta\colon H \to H \otimes H,$	$\|i\rangle \mapsto \|i\rangle \otimes \|i\rangle$	(copy in this basis)
$\varepsilon\colon H \to \mathbb{C},$	$\|i\rangle \mapsto 1$	(delete / accept all)
$\mu\colon H \otimes H \to H,$	$\|i\rangle \otimes \|j\rangle \mapsto \delta_{ij}\|i\rangle$	(merge if equal)
$\eta\colon \mathbb{C} \to H,$	$1 \mapsto \sum_i \|i\rangle$	(uniform superposition)

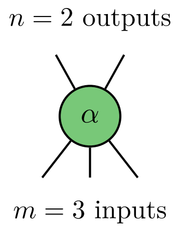
This is the bridge between category theory and quantum computing:

- The Z-spider corresponds to the SCFA for the computational basis $\{|0\rangle, |1\rangle\}$.
- The X-spider corresponds to the SCFA for the Hadamard basis $\{|+\rangle, |-\rangle\}$.

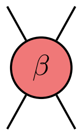
This directly connects to van de Wetering §7.5 and Genovese's slides on ZX-calculus.

10.6 ZX-Calculus Spiders as Frobenius Algebras

To make the connection explicit, a Z-spider with phase α, m inputs, and n outputs is:



When $\alpha = 0$, this is exactly the spider of Theorem 10.5 for the computational-basis SCFA. The phase α adds a phase gate $|0\rangle\langle 0| + e^{i\alpha}|1\rangle\langle 1|$ in the middle. Likewise, X-spiders (red) are built from the Hadamard-basis SCFA:



Chapter 11

Complementary Bases → Bialgebras and Hopf Algebras

11.1 Motivation: Z and X Are Complementary

The computational basis $\{|0\rangle, |1\rangle\}$ and the Hadamard basis $\{|+\rangle, |-\rangle\}$ are *mutually unbiased*: measuring a Z-basis state in the X-basis gives a uniformly random outcome, and vice versa. In the ZX-calculus, this complementarity is encoded algebraically through *bialgebra* and *Hopf algebra* rules between the Z and X Frobenius algebras.

11.2 Bialgebras

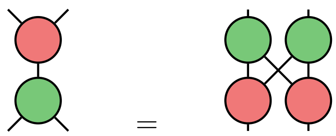
Definition 11.1 (Bialgebra)**.** Let $(A, \mu_{\color{green}\bullet}, \eta_{\color{green}\bullet})$ and $(A, \delta_{\color{red}\bullet}, \varepsilon_{\color{red}\bullet})$ be a monoid and a comonoid on the same object A in a symmetric monoidal category. They form a *bialgebra* if the following compatibility conditions hold:

1. The comultiplication $\delta_{\color{red}\bullet}$ is a monoid homomorphism (equivalently, the multiplication $\mu_{\color{green}\bullet}$ is a comonoid homomorphism).

2. $\varepsilon_{\color{red}\bullet} \circ \mu_{\color{green}\bullet} = \varepsilon_{\color{red}\bullet} \otimes \varepsilon_{\color{red}\bullet}$ and $\delta_{\color{red}\bullet} \circ \eta_{\color{green}\bullet} = \eta_{\color{green}\bullet} \otimes \eta_{\color{green}\bullet}$.

3. $\varepsilon_{\color{red}\bullet} \circ \eta_{\color{green}\bullet} = \mathrm{id}_I$.

The key bialgebra rule in string diagrams uses two colors to distinguish the two Frobenius structures. In ZX-calculus notation:



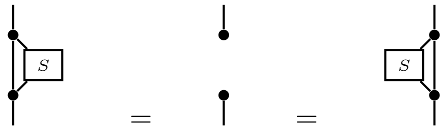
This is the *bialgebra rule* of the ZX-calculus: a green merge followed by a red split equals two red splits followed by two green merges (with a crossing in between). In van de Wetering's notation, this is the (B2) rule.

11.3 Hopf Algebras

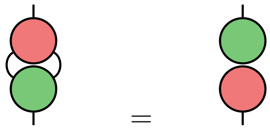
Definition 11.2 (Hopf algebra)**.** A bialgebra $(A, \mu, \eta, \delta, \varepsilon)$ is a *Hopf algebra* if there exists a morphism $S\colon A \to A$ called the *antipode* satisfying:

$$\mu \circ (\mathrm{id} \otimes S) \circ \delta = \eta \circ \varepsilon = \mu \circ (S \otimes \mathrm{id}) \circ \delta.$$

In string diagrams, the Hopf algebra axiom is:



In the ZX-calculus, the Hopf rule takes a particularly elegant form. When the two Frobenius algebras (Z and X) interact, the antipode is related to the Pauli-Z gate. The Hopf rule states:



A green spider connected to a red spider by two parallel wires simplifies to an isolated red spider below and green spider above (each with one input and one output). This directly corresponds to van de Wetering §4.5 and §7.5.

11.4 Complementarity of Bases

The bialgebra and Hopf algebra structures encode the *complementarity* of the Z and X bases. Two orthonormal bases $\{|a_i\rangle\}$ and $\{|b_j\rangle\}$ of $\mathbb{C}^n$ are *mutually unbiased* if $|\langle a_i|b_j\rangle|^2 = 1/n$ for all i, j. The computational and Hadamard bases of $\mathbb{C}^2$ satisfy this: $|\langle 0|+\rangle|^2 = 1/2$.

Categorically, two SCFAs on the same object are *complementary* if their combined monoid-comonoid structure forms a bialgebra (with a Hopf algebra structure). This algebraic condition is equivalent to mutual unbiasedness of the corresponding bases. The ZX-calculus exploits this complementarity as its primary computational tool.

Chapter 12

The Big Picture → Props and Graphical Theories

12.1 Props: Product and Permutation Categories

Definition 12.1 (Prop)**.** A *prop* (short for "**pro**duct and **p**ermutation category") is a strict symmetric monoidal category whose objects are the natural numbers $0, 1, 2, \ldots$ and whose tensor product on objects is addition: $m \otimes n = m + n$.

Props provide a convenient framework for algebraic theories with multiple inputs and outputs. A prop is specified by:

- **Generators**: a set of morphisms ("basic operations").
- **Relations**: equations between composites of generators.

Example 12.2 (The prop of matrices)**. Mat** is a prop: objects are natural numbers, morphisms from n to m are $m \times n$ complex matrices, tensor product is direct sum (on objects: addition; on morphisms: block-diagonal matrices).

12.2 ZX-Calculus as a Prop

The ZX-calculus can be formalized as a prop ZX with:

- **Generators**: Z-spiders (with phase parameter α), X-spiders (with phase parameter β), the Hadamard box, cups, and caps.
- **Relations**: the spider rule, bialgebra rule, Hopf rule, π-copy rule, color change rule, and supplementarity (among others).

The standard interpretation is a prop morphism (a "strict symmetric monoidal functor")

$$- : \mathsf{ZX} \to \mathbf{Mat}$$

sending each ZX-diagram to its corresponding matrix.

12.3 Soundness and Completeness

Definition 12.3 (Soundness and completeness of a graphical calculus)**.** Let $\mathcal{G}$ be a prop (the "graphical theory") and $- : \mathcal{G} \to \mathcal{T}$ an interpretation into a target category $\mathcal{T}$.

- The calculus is *sound* if equal diagrams (in $\mathcal{G}$) map to equal morphisms in $\mathcal{T}$: the rewrite rules don't produce false equations.
- The calculus is *complete* if the converse holds: whenever two diagrams have the same interpretation in $\mathcal{T}$, they can be proved equal using the rules of $\mathcal{G}$.

Theorem 12.4 (Completeness of ZX-calculus)**.** *The ZX-calculus (with the complete set of rules) is both sound and complete for the prop of matrices over* $\mathbb{C}$*: two ZX-diagrams represent the same linear map if and only if one can be transformed into the other using the ZX rewrite rules.*

This is a remarkable result: it means the graphical rules *capture all of linear algebra* over qubits. The proof of completeness is non-trivial and was achieved incrementally (first for the Clifford fragment, then for Clifford+T, then universally).

12.4 Connections to the Source Material

We now have the full categorical toolkit to engage with each of the source documents:

Source	**Category theory used**
van de Wetering	Dagger compact categories (Ch. 7–8), Frobenius algebras (Ch. 10), bialgebras and Hopf algebras (Ch. 11), props (Ch. 12). See especially §4 (rewrite rules = prop relations), §7.5 (categorical semantics).
Genovese	Symmetric monoidal categories (Ch. 4–6), string diagrams (Ch. 5), Frobenius algebras (Ch. 10). The slides present ZX-calculus as string diagrams in a symmetric monoidal category.
PennyLane	String diagrams (Ch. 5), spider theorem (Ch. 10, Thm. 10.5). The tutorial uses spider fusion implicitly in every diagram simplification.
Wakeham (SIQP)	Abstract wiring diagrams = string diagrams for monoidal categories (Ch. 5), tensor products (Ch. 4), dagger structure (Ch. 7). The C*-algebraic framework uses compact closed structure (Ch. 8).

12.5 Where to Go from Here

With the categorical foundations in place, the reader is equipped to:

1. **Learn the ZX-calculus**: study the rewrite rules as equations in a prop, using spider fusion, bialgebra, and Hopf rules.
2. **Prove circuit identities**: use string diagram reasoning to verify quantum circuit equivalences graphically.
3. **Understand quantum error correction**: categorical frameworks (stabilizer codes, lattice surgery) use Frobenius algebras and compact closed structure.
4. **Explore categorical quantum mechanics**: the Abramsky–Coecke program reconstructs quantum theory from categorical axioms (dagger compact categories with enough structure).

The graphical language of category theory transforms quantum computing from matrix arithmetic into topology—and that is where the deepest insights live.

Appendix A

Notation Summary

Symbol	Meaning	Introduced
$\mathcal{C}, \mathcal{D}$	Categories	Def. 2.1
$\mathrm{Hom}(A, B)$	Morphisms from A to B	Def. 2.1
$g \circ f$	Composition of f then g	Def. 2.1
id_A	Identity on A	Def. 2.1
$F : \mathcal{C} \to \mathcal{D}$	Functor	Def. 3.1
$\otimes$	Tensor product	Def. 4.1
I	Monoidal unit	Def. 4.1
$\sigma_{A,B}$	Braiding / symmetry	Def. 6.2
$f^\dagger$	Dagger / adjoint of f	Def. 7.1
A^*	Dual object	Def. 8.1
η, ε	Cup, cap (or unit, counit)	Def. 8.1
μ, η	Monoid multiplication, unit	Def. 9.1
δ, ε	Comonoid comultiplication, counit	Def. 9.2
FdHilb	Finite-dim. Hilbert spaces	Ex. 2.4
Set, **Vect**, **Mat**	Standard categories	Ch. 2

Appendix B

Suggested Reading

1. B. Coecke and A. Kissinger, *Picturing Quantum Processes* (Cambridge, 2017). The definitive textbook on categorical quantum mechanics and graphical reasoning.

2. S. Mac Lane, *Categories for the Working Mathematician* (Springer, 1998). The classic reference for category theory.

3. J. van de Wetering, "ZX-calculus for the working quantum computer scientist" (2020). Comprehensive review of ZX-calculus—one of the source documents for this tutorial.

4. S. Abramsky and B. Coecke, "A categorical semantics of quantum protocols," *Proc. LICS* (2004). The foundational paper on categorical quantum mechanics.

5. P. Selinger, "A survey of graphical languages for monoidal categories," *New Structures for Physics*, Lecture Notes in Physics 813 (2011). Comprehensive survey of string diagrams for various categorical structures.

6. C. Heunen and J. Vicary, *Categories for Quantum Theory* (Oxford, 2019). A modern textbook covering dagger categories, compact closed categories, and Frobenius algebras with a quantum information focus.