Visualizing Qudit Controls With Sheets

Scott Wesley

Dalhousie University, Halifax NS, Canada

Abstract. Graphical methods have a long history in mathematics, dating back to Euclid's *Elements*. However, this tradition was abandoned in favour of Hilbert's program. More recently, research in topology and categorical algebra have driven a resurgence in the development of graphical techniques. Of particular interest are the two-dimensional *string diagrams* of categorical algebra, which share deep connections with quantum computation. This paper reviews some important categorical concepts in the theory of quantum circuit diagrams, and describes an ongoing effort to study *controlled qudit operations* through the lens of three-dimension *sheet diagrams*.

1 Introduction

Graphical techniques have a long history in mathematics. For example, the work of Euclid in *The Elements* relied heavily on the use of geometric figures [10]. Similarly, the field of graph theory (as pioneered by Euler in 1736 [11]) reduces mathematical relations to diagrams of vertices and the edges between them. However, graphical methods fell out of favour in mathematics at the end of the nineteenth century, as Hilbert attempted to formalize all of mathematics [17]. It was not until the second half of the twentieth century, that graphical methods would return with more rigorous foundations. One well-known example is the use of Cayley graphs to study geometric properties of groups [6]. Another example is the use of commuting diagrams in categorical algebra, which visualize equations as paths through graphs [19]. As category theory has developed over time, the simple insight of viewing equations as graphs has evolved into formal graphical languages for the manipulation of equations up to isomorphism [21].

One such family of graphical languages, known as string diagrams, visually resemble the graphical languages for logical circuit design used in computer science. In fact, the semantics of string diagrams precisely capture the notion of sequential and parallel composition, such as the application of logical gates to wires in a sequential circuit [14]. Surprisingly, these same diagrams capture the notion of unitary operators in quantum mechanics [16]. It turns out that this connection between computation and physics is fundamental, and has been well-studied relative to graphical languages in prior work [1]. This connection lead to many practical contributions, such as Feynman's proposal for quantum computation as a means to simulate physical systems [12]. More recently, quantum computation has been shown to solve computational problems as well [20].

This paper serves as a review for some important concepts in the theory of quantum circuit diagrams. However, the story told by this paper deviates from the conventional narrative, by emphasizing the *bimonoidal structure* of unitary quantum mechanics. This approach yields a new perspective on *controlled unitary operations* using recent developments in string diagrams. For the extended version of this paper see [26].

2 Background and Notation

This section reviews linear algebra and introduces some common definitions from category theory. For a comprehensive introduction to category theory, see [19]. For more applied perspectives, [3] introduces category theory for computer science, [5] introduces category theory for logicians, [16] introduces category theory from the perspective of quantum mechanics, [23] provides a more general introduction for scientists, and [13] outlines applications of category theory in probability.

This paper is concerned with unitary operators acting on finite-dimensional $\mathbb{C}$-vector spaces. For the discussion that follows let A be a n-dimensional $\mathbb{C}$-vector space with standard basis $e_1, e_2, \dots, e_d$. Given a vector $v = a_1 e_1 + a_2 e_2 + \cdots + a_d e_n$ in A, the *norm of* v is $||v|| = \sqrt{\sum_{j=1}^{n} |a_j|^2}$ where $|a_j|$ is the complex modulus of a_j. If $|v| = 1$, then v is said to be a *unit vector*. Two vectors u and v

$$\begin{tikzcd} A \arrow[r, "1_A"] \arrow[dr, "f"'] & A \arrow[d, "f"] \\ & B \end{tikzcd}$$

(a) Right identity.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "f"'] & B \arrow[d, "1_B"] \\ & B \end{tikzcd}$$

(b) Left identity.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "h \circ f"'] & B \arrow[d, "h \circ g"] \\ C \arrow[r, "g"'] & D \end{tikzcd}$$

(c) Commutativity.

Fig. 1: The commuting diagrams for the definition of a category.

$$\begin{tikzcd} F(A) \arrow[r, "F(f)"] \arrow[dr, "F(g \circ f)"'] & F(B) \arrow[d, "F(g)"] \\ & F(C) \end{tikzcd}$$

(a) Respecting composition.

$$\begin{tikzcd} F(A,B) \arrow[r, "F(f{,}1_B)"] \arrow[d, "F(1_A{,}g)"'] & F(A',B) \arrow[d, "F(1_A{,}g)"] \\ F(A,B') \arrow[r, "F(f{,}1_B)"'] & F(A',B') \end{tikzcd}$$

(b) Bifunctoriality

$$\begin{tikzcd} F(A) \arrow[r, "\alpha_A"] \arrow[d, "F(f)"'] & G(A) \arrow[d, "G(f)"] \\ F(B) \arrow[r, "\alpha_B"'] & G(B) \end{tikzcd}$$

(c) Naturality.

Fig. 2: The commuting diagrams for functors and natural transformations.

are said to be *orthogonal* if $u^\dagger v = 0$ where $(-)^\dagger$ is the conjugate transpose of u. A basis is said to be *orthonormal* if it is composed from unit vectors which are pairwise orthogonal. Then a *unitary operator on A* is simply a complex matrix whose columns form an orthonormal basis for A.

Two common operations on vector spaces (and their operators) are the direct sum and Kronecker tensor product. The direct sum $\oplus$ and the Kronecker tensor product $\otimes$ are defined as follows, where u is an n-dimensional vector, v is any vector, N is an $n \times m$ matrix, and M is any matrix.

$$u \oplus v = \begin{bmatrix} u \\ v \end{bmatrix} \qquad u \otimes v = \begin{bmatrix} u_1 v \\ \vdots \\ u_n v \end{bmatrix} \qquad N \oplus M = \begin{bmatrix} N & 0 \\ 0 & M \end{bmatrix} \qquad U \otimes V = \begin{bmatrix} N_{1,1}M & \cdots & N_{1,m}M \\ \vdots & \ddots & \vdots \\ N_{n,1}M & \cdots & N_{n,m}M \end{bmatrix}$$

First note that $\mathbb{C}^n \oplus \mathbb{C}^m \cong \mathbb{C}^{n+m}$ whereas $\mathbb{C}^n \otimes \mathbb{C}^m \cong \mathbb{C}^{nm}$. If u and v are vectors, then $||u \oplus v|| = ||u|| + ||v||$ whereas $||u \otimes v|| = ||v|| \cdot ||v||$. Furthermore, if U and V are unitary operators, then $U \oplus V$ and $U \otimes V$ are also unitary operators. The tensor and direct sum are *bilinear* in the sense that $U \otimes (V \circ W) = (U \otimes V) \circ (U \otimes W)$ and $(U \circ V) \otimes W = (U \otimes W) \circ (V \otimes W)$.

Matrix multiplication, the tensor product, and the direct sum, each define a unique way to compose matrices. This yields equations with many connectives. This complex syntax leads to complicated equations, which are challenging for humans to read and write inline [17]. One solution to this problem is *commuting diagrams*, which depict function composition in terms of directed graphs [19]. In a commuting diagrams, the vertices depict (co)domains, and the edges depict functions between these vertices (for example, see Fig. 1). Any pair of paths with a common start and end point compose to the same function. In the language of category theory, the vertices are *objects*, and the edges are are *morphisms* [19]. Category theory studies collections of objects and the morphisms they satisfy.

Definition 1 (Category [19]). *A category $\mathcal{C}$ is defined by a collection of objects $\mathcal{C}_0$ and the following information.*

1. *For objects $A \in \mathcal{C}_0$ and $B \in \mathcal{C}_0$, a collection of morphisms $\mathcal{C}(A, B)$.*
2. *For each object $A \in \mathcal{C}$, an identity morphism $1_A : A \to A$.*
3. *For morphisms $f : A \to B$ and $g : B \to C$, a morphism $g \circ f : A \to C$.*

This information is subject to the conditions that Figs. 1a and 1b commute for all $f : A \to B$ and Fig. 1c commutes for all $f : A \to B$, $g : B \to C$, $h : C \to D$.

Example 1. The category **Unitary** has complex vector spaces as objects, unitary matrices as morphisms, and matrix multiplication as composition.

Many constructions in mathematics can be formalized as well-behaved mappings between categories. Well-behaved means that morphisms and their composition are respected by the mapping. A common example from linear algebra is the embedding of a vector space into its double dual. Naturally, the corresponding mapping would send both vector spaces and linear maps to their double duals. In category theory, these well-behaved mappings are referred to as functors.

(a) $g \circ f$ (b) $A \odot B$ (c) $f \odot g$ (d) $\sigma_{A,B}$

Fig. 3: The syntax and semantics of symmetric string diagrams [21]. Each morphism 1_A is denoted by an empty wire of type A. Wires of type I are omitted.

Definition 2 (Functor [19]). *A functor $F : \mathcal{C} \to \mathcal{D}$ between categories $\mathcal{C}$ and $\mathcal{D}$ is defined by the following information.*

1. *For each $A \in \mathcal{C}_0$, an object $F_0(A) \in \mathcal{D}_0$.*
2. *For each $f : A \to B$ in $\mathcal{C}$, a morphism $F(f) : F_0(A) \to F_0(B)$.*

This information is subject to the condition that $F(1_A) = 1_{F(A)}$ for all $A \in \mathcal{C}_0$ and that Fig. 2a commutes for all $f : A \to B$ in $\mathcal{C}$.

Definition 3 (Bifunctor [19]). *A functor $F : \mathcal{C} \times \mathcal{D} \to \mathcal{K}$ is called a binfunctor if Fig. 2b commutes for all $f : A \to A'$ in $\mathcal{C}$ and $g : B \to B'$ in $\mathcal{D}$.*

Example 2. The Kronecker tensor product and the direct sum are both examples of bifunctors from **Unitary** $\times$ **Unitary** to **Unitary**. This follows almost immediately from the bilinearity of the Kronecker tensor product and direct sum.

In mathematics, it is often necessary to transform information between two functors. For example, the isomorphism between a finite-dimensional vector space and its double dual is a structure preserving transformation between the identity functor and the double dual functor. These structure preserving transformations between functor are known as natural transformations. Using the language of natural transformations, it is also possible to define when bifunctors are unital, associative, and commutative.

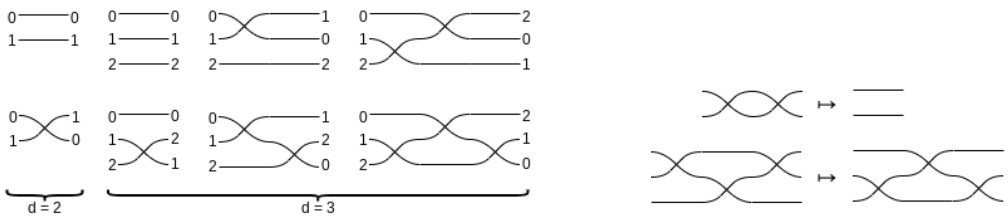
Definition 4 (Natural Transformation [19]). *Let $F, G : \mathcal{C} \to \mathcal{D}$ be functors. A natural transformation $\alpha : F \Rightarrow G$ assigns to each object $A \in \mathcal{C}_0$ a morphism $\alpha_A : F_0(A) \to G_0(A)$ such that for all morphisms $f : A \to B$, the diagram in Fig. 2c commutes. If each α_A is invertible, then α is called a natural isomorphism.*

Example 3. Bifunctors generalize the notion of a binary operators to categories. From this perspective, it makes sense to ask that a bifunctor satisfy properties such as associativity and unitality. This is done by defining a natural isomorphism which encodes the desired property up to isomorphism. In the cases of unitality and associativity, these natural isomorphisms are called unitors and associators, respectively. The standard unitors and associators for the direct sum and tensor product can be found in the extended version of this paper.

Many categories admit additional structures, such as sums and products. For example, the category of vector spaces is endowed with the direct sum and the Kronecker tensor product. If a category is endowed with a unital, associative, bifunctorial structure, then the category is *monoidal*. There exists a graphical language for monoidal categories, known as string diagrams (see Fig. 3 for the syntax and semantics). In this graphical language, equations hold up to planar deformations (this is equivalent to the coherence conditions) [21]. However, most graphical languages trade away some semantic information to achieve efficient graphical representations [17]. In the case of string diagrams, the unitors and associators are neglected. However, the *Strictification Theorem* of [19] allows for the unitors and associators to be neglected without loss of generality, provided that they are not the objects of study.

Definition 5 (Monoidal Category [19]). *A monoidal category is a category $\mathcal{C}$ together with the following information.*

1. ***Monoidal Product.*** *A bifunctor $\odot : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$.*
2. ***Unit Object.*** *An object $I \in \mathcal{C}_0$.*
3. ***Left Unitor.*** *A natural isomorphism $\lambda : I \odot (-) \Rightarrow (-)$.*



(a) All permutations for $d = 2$ and $d = 3$.

(b) Minimal relations.

Fig. 4: Relations for all permutation, as generated by transpositions and $\oplus$.

4. ***Right Unitor.*** *A natural isomorphism $\rho : (-) \odot I \Rightarrow (-)$.*
5. ***Associator.*** *A natural isomorphism $\alpha : ((-) \odot (-)) \odot (-) \Rightarrow (-) \odot ((-) \odot (-))$.*

The isomorphisms λ, ρ, and α are subject to the coherence conditions as stated in the extended version of this paper.

Definition 6 (Symmetric Monoidal Category [19]). *A monoidal category $\mathcal{C}$ with monoidal product $\odot$ is symmetric if there exists a natural isomorphism $\sigma : (-) \odot (-) \Rightarrow (-) \odot (-)$ such that $\sigma_{A,B} : A \odot B \to B \odot A$, $\sigma_{A,B}^{-1} = \sigma_{B,A}$. Furthermore, σ must satisfy the coherence conditions as stated in the extended version of this paper.*

Example 4. **Unitary** is monoidal with respect to the direct sum (the unit object is $\mathbb{C}^0$) and the Kronecker tensor product (the unit object is $\mathbb{C}$).

3 Building Qudits Systems Through Sums and Products

Quantum computation studies finite-dimensional quantum systems [20]. For the purposes of this discussion, the state of a quantum system can be thought of as a unit vector in $\mathbb{C}^d$ for some $d > 1$. The standard basis states for this vector space are denoted $\{|0\rangle, |1\rangle, \ldots, |d-1\rangle\}$. When $d = 2$, the basis vectors $|0\rangle$ and $|1\rangle$ are obtained, which can be thought of as the 0 and 1 state of a bit in traditional computing. For this reason, unit vectors in $\mathbb{C}^2$ are referred to as *qubits*, which is short for *quantum bit*. More generally, a unit vector in $\mathbb{C}^d$ is referred to as a *qudit*.

A qudit can exist in a *superposition* of multiple basis states [20]. In general, a qudit is a vector of the form $\sum_{n=0}^{d-1} \alpha_n |n\rangle$ where each $\alpha_n \in \mathbb{C}$ and $\sum_{n=0}^{d-1} |\alpha_n|^2 = 1$. The value $|\alpha_n|^2$ can be interpreted as the probability of observing basis state $|n\rangle$ when the quantum system is measured. Of course, measurement is a physical process. This means that basis states are properties of physical systems. For example, when $d = 2$, a qubit may be realized as the spin of an electron, which has basis states *up* and *down* [20].

A qudit system can be decomposed into its constituent components using the direct sum of vector spaces. For example, $\mathbb{C}^2 \cong \mathbb{C} \oplus \mathbb{C}$. More generally, $\mathbb{C}^d \cong \oplus^d \mathbb{C}$ where $\oplus^d A := \oplus_{n=1}^d A$. The direct sum of vector spaces is symmetric in the sense of monoidal categories. The symmetries for $\oplus$ are as follows.

$$\tau_{A,B} : A \oplus B \to B \oplus A \qquad\qquad \tau_{A,B} : |\varphi\rangle \mapsto \begin{bmatrix} 0 & 1_B \\ 1_A & 0 \end{bmatrix} |\varphi\rangle$$

Given a quantum system with d basis states, there are d-factorial $\oplus$-symmetries of the system. These symmetries permute the basis states. For example, the permutations of $\mathbb{C}^2$ and $\mathbb{C}^3$ are illustrated in Fig. 4a. In both examples, the permutations are written as sequences of transpositions between adjacent basis states. It was shown in [18] that this decomposition is always possible, and that equality in this form is decided by repeated application of the rules in Fig. 4a. We say that the $\oplus$-symmetries of $\mathbb{C}^d$ are presented by $\oplus$ and $\tau_{\mathbb{C},\mathbb{C}}$ with respect to the relations in Fig. 4b.

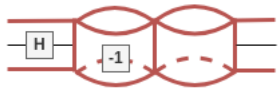
Unfortunately, **Unitary** with $\oplus$ does not capture all properties of quantum mechanics. In particular, $\oplus$ not describe how to compose two distinct quantum systems into a larger quantum system. To see why this is the case, simply note that the direct sum of two unit vectors is not

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

(a) Hamadard.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \tau_{\mathbb{C},\mathbb{C}}$$

(b) Pauli-X.



$$= \tau_{\mathbb{C},\mathbb{C}} \cdot \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot H$$

(c) A sheet diagram in **Unitary**.

Fig. 5: The unitaries H and X, and constant -1, assembled into a sheet diagram.

a unit vector. However, the tensor product does preserve unit vectors, and also aligns with the postulates of quantum mechanics [20].

The tensor product of two basis states, say $|0\rangle$ and $|1\rangle$, is denoted by $|01\rangle$. An interesting observation is that there exists states such as $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ which exist in $\mathbb{C}^2 \otimes \mathbb{C}^2$, but cannot be written as the tensor product of two elements in $\mathbb{C}^2$. Physically, the two systems cannot be decomposed without losing information. This phenomenon is known as *entanglement* [20]. A unitary that gives rise to entanglement is referred to as an entangling operator. For example, given H as defined in Fig. 5a, the operator $M := 1 \oplus H \oplus 1$ is entangling with respect to $\mathbb{C}^2 \otimes \mathbb{C}^2$. This is because $M|01\rangle = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle$.

Since the monoidal category **Unitary** equipped with tensor product describes the composition of quantum systems, then its string diagrams describe quantum computation with multiple qudits of mixed dimensions. Dense generating sets are known for qudit systems of various dimensions, given the $\otimes$-operator [25], though their relations are still an open area of research. For example, H, $1 \oplus 1 \oplus X$, and $1 \oplus e^{i\pi/8}$ generate all qubit operators up to arbitrary precision, though their relations are only known for systems of up to 2 qubits [4]. It should come as no surprise that this category is symmetric monoidal, since permutations of $\mathbb{C}^n \otimes \mathbb{C}^m$ correspond to certain permutations of the basis states in $\bigoplus^{nm} \mathbb{C}$ Unfortunately, there is no general technique to write string diagrams for multiple monoidal products[1]. Despite this, many $\oplus$-constructions still appear in quantum computation.

4 Interacting Structures: Sums and Products

The direct sum and tensor product on **Unitary** enjoy many nice properties. For example, the tensor product of vector spaces distributes over the direct product of vector spaces, just as multiplication distributes over addition. The monoidal unit for the direct sum also acts as an annihilator for the tensor product, just as $0 \cdot a = 0 = a \cdot 0$ for all $a \in \mathbb{C}$. When two monoidal products interact in this way, we say that the category is *bimonoidal*.

Definition 7 (Bimonoidal [9]). *A category $\mathcal{C}$ is bimonoidal with respect to a symmetric monoidal structure $(\mathcal{C}, \oplus, \mathbb{0}, \dots)$ and a monoidal structure $(\mathcal{C}, \otimes, \mathbb{1}, \dots)$ given the following information.*

1. ***Left Distributor.*** *A natural isomorphism $d^l : A \otimes (B \oplus C) \Rightarrow (A \otimes B) \oplus (A \otimes C)$.*
2. ***Right Distributor.*** *A natural isomorphism $d^r : (A \oplus B) \otimes C \Rightarrow (A \otimes C) \oplus (B \otimes C)$.*
3. ***Left Annihilator.*** *The natural isomorphism $a^l : \mathbb{0} \otimes A \Rightarrow A$.*
4. ***Right Annihilator.*** *The natural isomorphism $a^r : A \otimes \mathbb{0} \Rightarrow A$.*

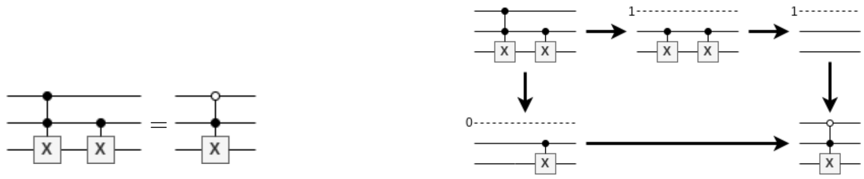
The natural isomorphisms d^l, d^r, a^l, and a^r are subject to the 24 coherence axioms of [9].

It was shown in [9] that bimonoidal categories admit a generalized notion of string diagrams, referred to as the language of sheet diagrams. However, the language requires that the equations are in a normal form. First, the distributors d^l and d^r are used to rewrite each object as a sum of products. For example, $(A \oplus B) \otimes (C \oplus D)$ is represented by the naturally isomorphic object $(A \otimes C) \oplus (A \otimes D) \oplus (B \otimes C) \oplus (B \otimes D)$. Each term of the direct sum is then represented by a vertical plane in $\mathbb{R}^3$. Symmetries of the direct sum are expressed by crossing sheets, as illustrated in Fig. 6a. A morphism of type $A \to B \oplus C$ is representing by a branching sheet, as in Fig. 6b. A

[1] There do exist general constructions to obtain graphical languages for categories with two monoidal products, such as proof nets [15, 21]. However, proof nets are not string diagrams since the monoidal products are not depicted using spatial juxtaposition.

(a) $\tau_{A,B}$ (b) $A \to B \oplus C$ (c) $A \oplus B \to C$ (d) $(W \circ V) \oplus U$

Fig. 6: The syntax and semantics of sheet diagrams [9]. The rules of Fig. 3 still apply for the multiplicative structure. The red lines denote sheet boundaries.



(a) A reversible circuit equality. (b) A proof sketch.

Fig. 7: A proof sketch for a non-trivial quantum circuit identity. This graphical language is imprecise, but illustrate a case-based argument.

morphism of type $A \oplus B \to C$ is representing by the merging of two sheets, as in Fig. 6c. Along each sheet is a string diagram for the monoidal category $(\mathbb{C}, \otimes, \mathbb{1})$ as in Fig. 6d. As in monoidal categories, pulling gates along wires preserves equality of diagrams. Graphical rules exist to move gates through direct sums, though these rules are not used in this paper. A sheet diagram for a non-trivial unitary can be found in Fig. 5c.

5 Visualizing Controlled Qubit Gates

The direct sum of **Unitary** can be interpreted computationally as the switch statement from sequential programming. For example, if U_1 through to U_d are unitaries acting on some vector space A, then $M := U_1 \oplus U_2 \oplus \cdots \oplus U_d$ is a unitary acting on $\mathbb{C}^d \otimes A$ such that $M\,|j\rangle \otimes |\varphi\rangle = |j\rangle \otimes U_j\,|\varphi\rangle$. In [22], this construction is referred to as a *generalized control.* Given a unitary U acting on A, the *positively controlled* version of U is $C(U) := 1_A \oplus U$, and the *negatively controlled* version of U is defined to be $C^{-}(U) = U \oplus 1_A$. Positive (resp. negative) controls are indicated by vertical wires with black (resp. white) dots as in Fig. 7a

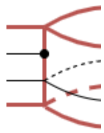
Recall the Pauli-X gate from Fig. 5b. Since $X\,|0\rangle = |1\rangle$ and $X\,|1\rangle = |0\rangle$, then X has a computational interpretation as the logical NOT-gate. The gates X, $C(X)$, and $C(C(X))$ are known to be universal for reversible computation, meaning that invertible circuit can be written using only X, $C(X)$, and $C(C(X))$ gates [24]. Unsurprisingly, networks of $C(C(X))$ gates are common in quantum computation, and simplifying these networks is an important problem [20]. For example, consider the identity in Fig. 7a. This can be proven using circuit identities as outlined in the extended version of this paper, though these identities lack any geometric intuition. In Fig. 7b, a more graphical argument in given, by considering the case where the control qubit is in state $|0\rangle$, and when the control qubit is in state $|1\rangle$. It is shown that if the control qubit is in state $|0\rangle$, then $C(X)$ is applied to the target, otherwise, the identity is applied to the target. It is concluded, by linearity, that the gates on the left-hand side of Fig. 7a compose to $C^{-1}(C(X))$. However, this proof is very ad-hoc, and lacks a set of formal derivation rules. The goal of this section is to make the argument of Fig. 7b precise via sheet diagrams.

The first step is to introduce a graphical notation for generalized controls. For each $d > 1$, it follows by dimensional analysis that $\mathbb{C}^d \otimes A$ is isomorphic to $\oplus^d A$. To give an explicit isomorphism, first note that each element of $\mathbb{C}^d \otimes A$ can be written as $|0\rangle \otimes |\varphi_0\rangle + |1\rangle \otimes |\varphi_1\rangle + \cdots + |d-1\rangle \otimes |\varphi_{d-1}\rangle$ for some φ_0 through to φ_{d-1} in A. Using this example, the following map defines an isomorphism between $\mathbb{C}^d \otimes A$ and $\oplus^d A$.

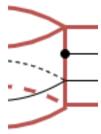
$$\delta_A : \sum_{n=0}^{d-1} |n\rangle \otimes |\varphi_n\rangle \mapsto (|\varphi_0\rangle, |\varphi_1\rangle, \dots, |\varphi_{d-1}\rangle) \qquad \delta_A^{-1} : (|\varphi_0\rangle, |\varphi_1\rangle, \dots, |\varphi_{d-1}\rangle) \mapsto \sum_{n=0}^{d-1} |n\rangle \otimes |\varphi_n\rangle$$

$$\begin{tikzcd} \mathbb{C}^d \otimes A \arrow[r, "\delta_A"] \arrow[d, "1\otimes f"'] & \oplus^d A \arrow[d, "\otimes^d f"] \\ \mathbb{C}^d \otimes B \arrow[r, "\delta_B"'] & \oplus^d B \end{tikzcd}$$

(a) Isomorphism.



(b) Graphical notation for δ ($d = 2$).

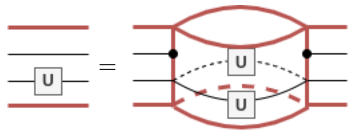


(c) Graphical notation for δ^{-1} ($d = 2$).

Fig. 8: A natural isomorphism between the direct sum and the Kronecker tensor product.

$$\begin{tikzcd} \mathbb{C}^d \otimes A \arrow[r, "\delta_A"] \arrow[d, "1\otimes f"'] & \oplus^d A \arrow[r, "\delta_A^{-1}"] \arrow[d, "\otimes^d f"] & \mathbb{C}^p \otimes A \arrow[d, "1\otimes f"] \\ \mathbb{C}^d \otimes B \arrow[r, "\delta_B"'] & \oplus^d B \arrow[r, "\delta_B^{-1}"'] & \mathbb{C}^p \otimes B \end{tikzcd}$$

(a) Derivation of rule.



(b) Graphical depiction of rule ($d = 2$).

Fig. 9: A graphical rule to introduce controls derived from the isomorphism δ.

This pair gives rise to the natural isomorphism in Fig. 8a, as proven in the extended version of this paper. Graphical notation is given for this isomorphism in Figs. 8b and 8c. From this construction, one can derive the common identity that given any $d > 1$ and any unitary U acting on A, the d-fold generalized control built from d copies of U is equivalent to $1_B \otimes U$ where $B = \mathbb{C}^d$. A proof is given in the extended version of this paper. The key insight for this proof is that $\delta_A^{-1} \circ \oplus^d U \circ \delta_A = 1 \otimes U$, which follows from the commuting diagram in Fig. 9a and corresponds to the equality of sheet diagrams in Fig. 9b.

Another common circuit identity is that conjugating a positive qubit control by an X gate is equivalent to negating the control. More generally, permuting the control qubit of a generalized control corresponds to permuting the operators in the direct sum. The first step in proving this result is to show that the diagram in Fig. 10a commutes. A proof is given in the extended version of this paper. From this commuting diagram, the equality of sheet diagram in Fig. 10 is obtained. This sheet diagram rule is used to prove the qubit case in the extended version of this paper.

At this point, it is possible to revisit Fig. 7a. A graphical proof is provided in the extended version of this paper, using only the coherent deformations of sheet diagrams, together with the rules derived thus far. It is shown at the end of the extended version of this paper, that each step of the conventional proof aligns precisely with one or more consecutive steps of the graphical proof. However, unlike the conventional proof, which follows entirely from circuit relations, the graphical proof is motivated geometrically by the structure of the sheets.

6 Prospects and Limitations

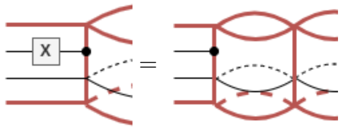
In this paper, a graphical language was proposed for unitary circuits with generalized controls. Sec. 5 illustrated how this language allows for case-based reasoning about controlled operations in a purely categorical fashion. This reasoning was used to re-derive some well-known results about qubits controls, together with an equational proof concerning $C(C(X))$ circuits. The primary advantage of this new language is that the syntactic notion of a control wire becomes a geometric property of the sheet diagrams used throughout the proofs.

However, this language does have limitations. For example, the notation introduced in Sec. 5 allows for constructions which lack physical motivation, such as in the extended version of this paper. We suspect that this problem can be solved by refining the type system associated with objects. For example, the direct summands in the definition of δ could be tagged with metadata about the control qubit, to avoid their conflation with qudits in a composition system. However, this type system would likely introduce challenges when typing the generators for the category.

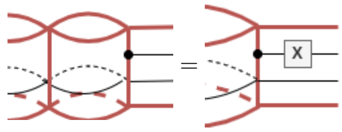
To better understand this design space, there are further examples to consider. As outlined in the extended version of this paper, one interesting family of examples are the Toffoli circuits of [8], in terms of their relation to the inductively defined controlled operations of [2]. More generally, it

$$\begin{tikzcd} \mathbb{C}^2 \otimes A \arrow[r, "\delta_A"] \arrow[d, "X \otimes 1_A"'] & A \oplus A \arrow[d, "\tau_{A,A}"] \\ \mathbb{C}^2 \otimes A \arrow[r, "\delta_A"'] & A \oplus A \end{tikzcd}$$

(a) Derivation of rule.



(b) Graphical depiction of rule.



(c) Graphical depiction of inverse rule.

Fig. 10: A graphical rule via natural isomorphism to pass an $\oplus$-symmetry through a control.

would be interesting to study the application of sheet diagrams to other bimonoidal categories with computational interpretations, such as [7]. In both cases, the resulting diagrams would be difficult to read, and challenging to typeset. This motivates the development of three-dimensional modelling tools, such as in [9]. To enable automated reasoning with these tools, a bimonoidal presentation for qudit unitary circuits would also be desirable.

References

1. Baez, J., Stay, M.: Physics, topology, logic and computation: A Rosetta stone. In: Coecke, B. (ed.) New Structures for Physics, pp. 95–172. Springer Berlin Heidelberg (2011)
2. Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., Sleator, T., Smolin, J.A., Weinfurter, H.: Elementary gates for quantum computation. Phys. Rev. A **52**, 3457–3467 (1995)
3. Barr, M., Wells, C.: Category Theory for Computing Science. Prentice Hall (1999)
4. Bian, X., Selinger, P.: Generators and relations for 2-qubit Clifford+T operators (2022)
5. Blute, R., Scott, P.: Category theory for linear logicians. In: Ehrhard, T., Girard, J.Y., Ruet, P., Scott, P. (eds.) Linear Logic in Computer Science, p. 3–64. London Mathematical Society Lecture Note Series, Cambridge University Press (2004)
6. Cayley, A.: Desiderata and suggestions: No. 2. the theory of groups: graphical representation. Amer. J. of Math. **195**, 179–195 (1878)
7. Choudhury, V., Karwowski, J., Sabry, A.: Symmetries in reversible programming: From symmetric rig groupoids to reversible programming languages. Proc. of the ACM on Prog. Lang. **6**(POPL), 1–32 (2022)
8. Cockett, J.R.B., Comfort, C.: The category TOF. Elec. Proc. in Theor. Comp. Sci. **287**, 67–84 (2019)
9. Comfort, C., Delpeuch, A., Hedges, J.: Sheet diagrams for bimonoidal categories (2020)
10. Euclid, Fitzpatrick, R.: Euclid's Elements of Geometry (2008)
11. Euler, L.: Solutio problematis ad geometriam situs pertinentis. In: Commentarii Academiae scientiarum imperialis Petropolitanae. vol. 8, pp. 124–140. Typis Academiae Petropolis (1726)
12. Feynman, R.: Simulating physics with computers. Int. J. of Theor. Phys. **21**, 467–488 (1982)
13. Furber, R., Jacobs, B.: Towards a categorical account of conditional probability. Elec. Proc. in Theor. Comp. Sci. **195**, 179–195 (2015)
14. Ghica, D.R., Jung, A., Lopez, A.: Diagrammatic semantics for digital circuits. In: Comp. Sci. Logic. vol. 82, pp. 24:1–24:16 (2017)
15. Girard, J.Y.: Linear logic. Theor. Comp. Sci. **50**(1), 1–101 (1987)
16. Heunen, C., Vicary, J.: Categories for Quantum Theory: An Introduction. Oxford University Press (2019)
17. Kulpa, Z.: On diagrammatic representation of mathematical knowledge. In: Asperti, A., Bancerek, G., Trybulec, A. (eds.) Mathematical Knowledge Management. pp. 190–204. Springer Berlin Heidelberg (2004)
18. Lafont, Y.: Towards an algebraic theory of Boolean circuits. J. of Pure and Applied Alg. **184**(2), 257–310 (2003)
19. Lane, S.M.: Categories for the Working Mathematician. Springer NY (2010)
20. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2011)
21. Selinger, P.: A survey of graphical languages for monoidal categories. In: Coecke, B. (ed.) New Structures for Physics, pp. 289–355. Springer Berlin Heidelberg (2011)
22. Selinger, P.: The Quipper System. https://www.mathstat.dal.ca/ selinger/quipper/doc/ (2014)
23. Spivak, D.I.: Category Theory for the Sciences. MIT Press (2014)
24. Toffoli, T.: Reversible computing. Tech. Rep. MIT/LCS/TM-151, MIT (1980)
25. Wang, Y., Hu, Z., Sanders, B.C., Kais, S.: Qudits and high-dimensional quantum computing. Frontiers in Phys. **8** (2020)
26. Wesley, S.: Visualizing qudit controls with sheets (extended) (2023)