Shaded tangles for the design and verification of quantum programs

Reutter, David; Vicary, Jamie

Shaded tangles for the design and verification of quantum programs

David Reutter
Department of Computer Science
University of Oxford
david.reutter@cs.ox.ac.uk

Jamie Vicary
Department of Computer Science
University of Oxford
jamie.vicary@cs.ox.ac.uk

March 6, 2019

Abstract

We give a scheme for interpreting shaded tangles as quantum programs, with the property that isotopic tangles yield equivalent programs. We analyze many known quantum programs in this way—including entanglement manipulation, error correction and teleportation—and in each case present a fully-topological formal verification, yielding in several cases substantial new insight into how the program works. We also use our methods to identify several new or generalized procedures.

1 Introduction

1.1 Overview

In this paper we introduce a new knot-based language for designing and verifying quantum programs. Terms in this language are *shaded tangles*, which look like traditional knot diagrams, possibly involving multiple strings and strings with open ends, and decorated with a shading pattern. Examples of shaded tangles are given in Figure 2. The overall goal of this paper is to use this language to identify a number of new or generalized protocols; an extensive summary of our results is given in Section 1.2.

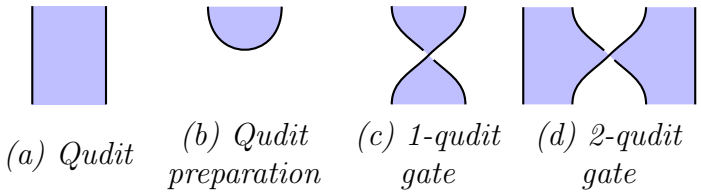


Figure 1: Part of the graphical language along with its interpretation in terms of quantum structures.

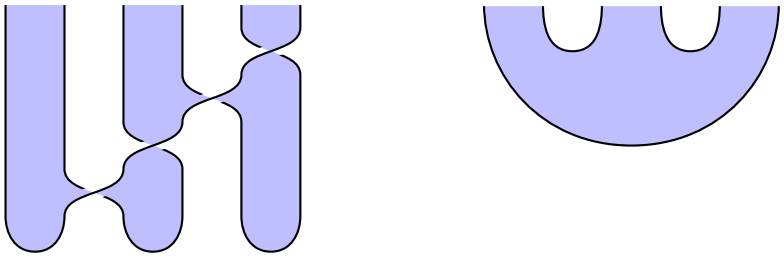
We give an operational semantics in which a shaded tangle is interpreted as a linear map between Hilbert spaces. Since this is the basic mathematical foundation for quantum information, this enables us to interpret our shaded tangles as *quantum programs*. Under this interpretation, we read our shaded tangles as quantum circuits, with time flowing from bottom to top, and with individual geometrical features of the diagrams—such as shaded regions, cups and caps, and crossings—interpreted as distinct quantum circuit components, such as qudits (a *qudit* is a d-dimensional quantum system; a *qubit* is a qudit for $d = 2$), qudit preparations, and 1- and 2-qudit gates (see Figure 1 for this part of the graphical language.)

Given two shaded tangles with the same shading pattern on their boundaries, we say they are *isotopic* just when, ignoring shading and considering them as ordinary knotted strings, one can be deformed topologically into the other. We show that our semantics is *sound* with respect to this isotopy relation: that is, if two shaded tangles are isotopic, then they have equal interpretations as quantum programs.

This yields a powerful method for the design and verification of quantum procedures. We draw one shaded tangle for the *program*, describing the exact steps the quantum computer would perform, and another shaded tangle for the *specification*, describing the intended computational effect. The program is then verified simply by showing that the two shaded tangles are isotopic. Since humans have an innate skill for visualizing knot isotopy, this verification procedure can often be performed immediately by eye, even in sophisticated cases. We illustrate this idea in Figure 2, which illustrates the program and specification for constructing a *GHZ state*, an important primitive resource in quantum information. It can be seen by inspection that, ignoring shading, the tangles are isotopic, and hence the program is correct. By reference to Figure 1, we see that the program Figure 2(a) involves three qudit preparations, two 1-qudit gates, and two 2-qudit gates.

1.2 Main results

In our main results, we apply this new high-level technique to represent and verify 13 quantum programs, some generalized from their form in the literature, and some completely new. We give a summary here of the programs we analyze. In this summary, we make use of some basic concepts from quantum information theory. A *Hadamard matrix* is a unitary matrix with all coefficients having the same absolute



(a) Program *(b) Specification*

Figure 2: Shaded tangles giving the program and specification for constructing a tripartite GHZ state.

value; they are important primitive structures in quantum information, playing a central role in quantum key distribution and many other phenomena [**?**]. A *unitary error basis* is a basis of unitary operators on a finite-dimensional Hilbert space, orthogonal with respect to the trace inner product; they provide the basic data for all quantum teleportation and dense coding procedures [**?**], and some error correction procedures [**? ?**].

- Section 4.1. A generalization of a program due to Uchida et al [**?**] for constructing GHZ states.
- Section 4.2. Programs due to Raussendorf and others [**? ?**] for constructing *cluster chains* [**?**], resources of central importance in quantum information.
- Section 4.3. Programs due to Briegel and others [**? ?**] for interconverting certain GHZ and cluster states.
- Sections 4.4 and 4.5. Programs due to van den Nest and others [**?**] for cutting and splicing cluster chains, which play an important role in measurement-based quantum computation [**?**]. We generalize these programs to qudits and to new classes of cluster chains.
- Section 4.6. A new program for robust state transfer in a cluster state–based quantum computer.
- Section 5.1. A generalization of programs due to Karlsson, Hillery, Grudka and others [**? ? ?**] for measurement-based teleportation along a GHZ state.
- Section 5.2. A generalization of a program due to Raussendorf and Briegel [**?**] for measurement-based teleportation along a cluster chain.
- Section 5.3. A program for teleportation along an n-party GHZ state, which we believe is folklore, with a new robustness property against a broad family of errors.
- Section 5.4. A program due to Yu, Jaffe and others [**? ?**] which reduces resource requirements for executing a distributed controlled quantum gate.
- Sections 6.1 and 6.2. The phase code and the Shor code [**? ? ?**], important error correcting codes in quantum information, which are built from Hadamard matrices.
- Section 6.3. New generalizations of the phase code and Shor code, based on unitary error bases rather than Hadamard matrices.

1.3 Significance

We outline some areas of potential significance of our work.

Novelty. Several of the programs we verify are in fact generalizations of those described in the literature, or are completely new, thereby making our results potentially of interest in mainstream quantum information science. Perhaps most

significantly, we highlight: the new program for robust state transfer in a cluster-state quantum computer (Section 4.6); the new constructions of error correcting codes based on unitary error bases (Section 6.3); and our identification of the $\frac{\pi}{2}$-rotations around the X-axis on the Bloch sphere as having privileged topological properties among all qubit Hadamard matrices (Section 2.3). In particular, we aim to rebut a common criticism that categorical methods in quantum information are useful only to recast known phenomena in a formal setting, without generating new knowledge [**?**].

Insight. Throughout, the shaded tangle syntax gives considerable new insight into why each procedure works. For example, in our verification of error correcting codes, the errors are literally 'trapped by bubbles' and removed from the diagram, and in our verification of cluster chain surgery procedures the qubits are literally untangled from the chain. In both cases, this gives a powerful intuition for these schemes which we believe to be new. This stands in contrast to traditional verification methods in quantum computer science [**?**], where a program is often given as a series of linear maps encoded algebraically (for example, as matrices of complex numbers), and verification involves composing the maps and examining the result; from this perspective, high-level structure can be difficult to perceive, and it may be unclear whether a program can be generalized.

Efficiency. Where our methods apply, we can often give the program, specification and verification in a concise way; compare for example our discussion of Figure 2 above with the traditional verification of a related program due to Uchida et al [**?**], which requires a page of algebra, and is also less general. As a consequence, even in this short extended abstract, we are able to give detailed analyses of 13 distinct procedures. We suggest that our methods would therefore be suitable for reasoning about large-scale quantum programs, such as architectures for quantum computers.

1.4 Criticism

Completeness. We define our semantics to be *sound* if topological isotopy implies computational equivalence, and *complete* if computational equivalence implies topological isotopy. The main semantics we give is sound, allowing the verification method for quantum programs that we use throughout the paper. However, it is not complete, meaning that there exist quantum programs that cannot be verified by our methods. Achieving completeness is an important focus of future work. We note that the *ZX calculus* (see Section 1.5), a dominant existing high-level approach to quantum information, shares this property of being sound but not complete [**?**], although it is complete for the stabilizer fragment [**?**].

Universality. Our language is also not *universal*, meaning that not all quantum programs can be constructed. It would be easy to make it universal by adding additional 1-qubit generators; however, without completeness, this has limited value.

Algorithms. All of our examples are in the broad area of quantum communication; we do not study quantum algorithms, such as Grover's or Shor's algorithms [**?**]. These algorithms have been analyzed in the related CQM approach [**?**]; in future work we aim to analyze them using our new syntax.

1.5 Related work

Categorical quantum mechanics (CQM). Our work emerges from the CQM research programme, initiated by Abramsky and Coecke [**?**] and developed by them and others [**? ? ? ? ? ? ? ? ? ? ? ? ? ?**], which uses monoidal categories with duals to provide a high-level language for quantum programs, using in particular a graph-based language called the *ZX calculus* [**? ?**]. CQM verifications have been given for some programs related to those that we analyze, including the Steane code [**?**], and cluster state arguments [**? ?**]. Many of the advantages of our calculus over traditional techniques—such as the power of the diagrammatic language, and its topological flavour—inherit directly from the CQM programme.

The current authors have previously shown that CQM methods can be extended to a higher-categorical setting [**? ? ?**], developing the work of Baez on a categorified notion of Hilbert space [**?**], and this paper develops these ideas further.

We give here some important points of distinction between traditional CQM techniques and our present work. Unlike the ZX calculus, our calculus is purely topological, and hence complex deductions can sometimes be perceived by eye in a single step. Also, our calculus is incomparable in strength to the ZX calculus, which is restricted (in its basic form) to Clifford quantum theory; neither calculus can simulate the other in general. As a result, we are able to analyze many protocols that have not previously been analyzed with ZX methods, as well as discover a number of new and generalized protocols.

Statistical mechanics. There is a rich interplay between quantum information (QI), knot theory (KT) and statistical mechanics (SM). The KT-SM and SM-QI relationships are quite well-explored in the literature, unlike the KT-QI relationship, which is our focus here.

The KT-SM relationship was first studied by Kauffman, Jones and others [**? ? ?**], who showed how to obtain knot invariants from certain statistical mechanical models. Much of the mathematical foundations of our paper are already present in the paper [**?**], including the shaded knot notation. Work on the SM-QI relationship has focused on finding efficient quantum algorithms for approximating partition functions of statistical mechanical systems [**? ? ? ?**], for which the best known classical algorithm is often exponential. 'Chaining' these relationships allows one to take a knot, obtain from it a statistical mechanical model, and then write down a quantum circuit approximating the model's partition function, giving overall a mapping from knots to quantum circuits, which ends up closely matching the construction we present. In particular, the relation between Hadamard matrices and shaded tangles, and the specific Hadamard matrices used in this paper are well-known, going back to results of Jones [**?**] on building link invariants from statistical mechanical models. The novelty here is of course the large range of new and existing quantum programs which we are able to analyze using this knot-theoretic notation.

The direct KT-QI relationship has also been emphasized by Kauffman and collaborators [**? ?**], by Jaffe, Liu and Wozniakowski [**? ? ? ? ?**] and also in the field of topological quantum computing [**?**], where (as here) a strong analogy is developed between topological and quantum entanglement.

Planar algebras. The graphical notation we employ can be described formally

as a *shaded planar algebra*, although we do not use that terminology in this paper, preferring a more elementary presentation. The relationship between shaded planar algebras and Hadamard matrices was first suggested by Jones [**?**], and developed by the present authors [**?** **?** **?**]. In particular, although we are using 2-categorical terminology, most planar diagrams in this work can alternatively be understood in terms of Jones' spin model planar algebra [**?** , Example 2.8]. (This is not the case for the diagrams with overlapping regions in Section 5, which explicitly use the monoidal structure of the underlying 2-category, which is not a part of the basic spin model planar algebra.)

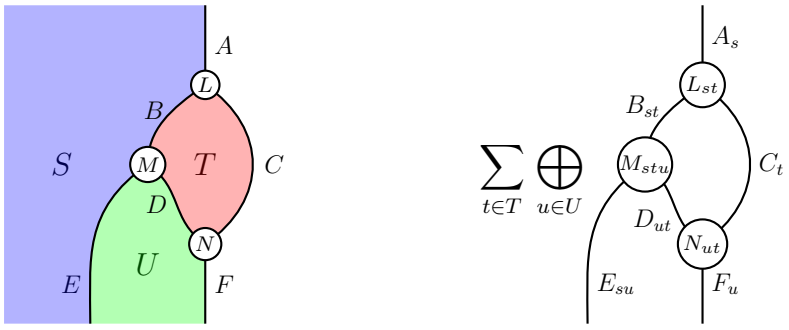
Recently, Jaffe, Liu and Wozniakowski have described a related tangle-based approach to quantum information based on *planar para algebras* [**?** **?** **?** **?** **?**]. In particular, up to a Fourier transform, the Hadamard matrix in Figure 7(b) coincides with the one used for the braiding in [**?**]. Nevertheless, there are several points of distinction that can be drawn between our work and theirs. Firstly, their planar para algebra setting is quite different to the mathematical structure we use. Secondly, their diagrams must be decorated with additional indices encoding measurement results, while our diagrams do not require such indices. Thirdly, there is little overlap between the quantum procedures analyzed so far in each setting (although see Section 5.4.)

Classical verification. There has been some work on using knot-theoretic methods for verification in linear logic [**?** **?**] and separation logic [**?**]. We do not see a direct technical connection to our results, although we expect this to be a fruitful direction for further investigation.

2 Mathematical foundations

2.1 Graphical calculus

The graphical calculus for describing composition of multilinear maps was proposed by Penrose [**?**], and is today widely used [**?** **?** **?** **?** **?**]. In this scheme, wires represent Hilbert spaces and vertices represent multilinear maps between them, with



(a) A diagram in our generalized calculus.

(b) Its interpretation for any $s \in S$.

Figure 3: The generalized graphical calculus.

wiring diagrams representing composite linear maps.

In this article we use a generalized calculus that involves *regions*, as well as wires and vertices; see Figure 3(a) for an example. This is an instance of the graphical calculus for symmetric monoidal 2-categories [**?** **?** **?** **?**] applied to the 2-category **2Hilb** of finite-dimensional 2-Hilbert spaces [**?**]. (Here and throughout, we use the term '2-category' to refer to the weak structure, which is sometimes called 'bicategory'.) The 2-category of 2-Hilbert spaces can be described as follows [**?** **?**]:

- objects are natural numbers;
- 1-morphisms are matrices of finite-dimensional Hilbert spaces;
- 2-morphisms are matrices of linear maps.

We represent composite 2-morphisms in this 2-category using a graphical notation involving regions, wires and vertices, which represent objects, 1-morphisms and 2-morphisms respectively. In Section 5, we also use the monoidal structure of **2Hilb**, represented graphically by 'layering' diagrams above each other. Here we use a particular fragment of this language, which—except for the monoidal structure—corresponds to the *spin model planar algebra* of Jones [**?** , Example 2.8]. Nonetheless, we emphasize that we use the techniques of monoidal 2-categories here, and not the techniques of planar algebras.

Elementary description. While these structures are widely used in higher representation theory, they are not yet prevalent in the quantum computing community. To help the reader understand these new concepts, we also give a direct account of the formalism in elementary terms, that can be used without reference to the higher categorical technology (see also [**?**]).

In this direct perspective, regions are labelled by *finite sets*. Wires and vertices now represent *families* of Hilbert spaces and linear maps respectively, indexed by the elements of the sets labelling all adjoining regions. A composite surface diagram represents a family of composite linear maps, indexed by the elements of all regions open on the left or right. For regions open only at the top or bottom of the diagram, we take the direct sum over elements of the indexing set, while for closed regions, we take the vector space sum over elements of the indexing set.

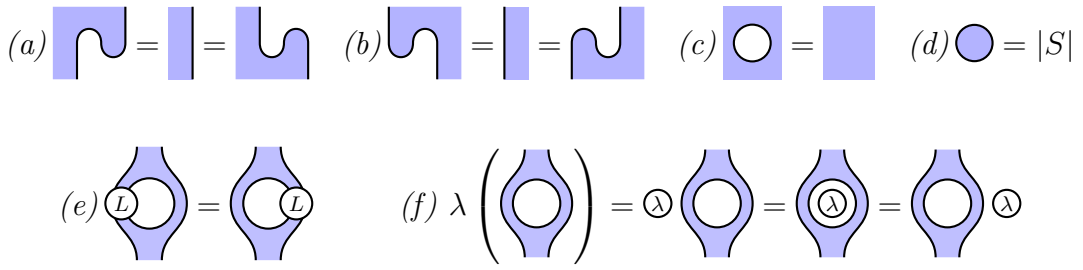


Figure 4: Some identities in the graphical calculus.

We give an example in Figure 3. In the diagram on the left, regions are labelled by finite sets (S, T, U), with unshaded regions labelled implicitly by the 1-element set; wires are labelled by families of finite-dimensional Hilbert spaces (A, B, C, D, E, F); and vertices are labelled by families of linear maps (L, M, N). For wires and vertices, the families are indexed by the sets associated to all neighbouring regions: for example, for $s \in S$ and $t \in T$, we have Hilbert spaces A_s, B_{st} and C_t, and $L_{st} : B_{st} \otimes C_t \to A_s$ is a linear map. The single diagram on the left represents an entire family of linear maps, with the maps comprising this family given by the right-hand diagram for different values of $s \in S$. We take the direct sum over index $u \in U$, since its region is open only at the bottom of the diagram, and the vector space sum over index $t \in T$, since its region is closed.

Given this interpretation of diagrams D as families of linear maps D_i, we define two diagrams D, D' to be *equal* when all the corresponding linear maps D_i, D'_i are equal; we define the *scalar product* λD as the family of linear maps λD_i; we define the *adjoint* $D^\dagger$ as the family of adjoint linear maps $(D_i)^\dagger$; and we say that D is unitary if all the maps D_i are unitary. Following convention [**?**], we depict the adjoint of a vertex by flipping it about a horizontal axis.

Restricted calculus. We use a highly restricted part of this calculus; it is this restricted part that agrees with the spin model planar algebra, as we mention above. Every shaded region we assume to be labelled by a single fixed finite set S. All wires bound precisely one shaded region and one unshaded region, and these wires are always labelled by a family of 1-dimensional Hilbert spaces $\mathbb{C}$. Nonetheless, the calculus is not trivial. For example, we can build the identity on a nontrivial Hilbert space as the diagram Figure 1(a); under the rules set out above, this is the identity map on $\oplus_{s \in S}(\mathbb{C} \otimes \mathbb{C}) \simeq \mathbb{C}^{|S|}$.

Also, we add the following components to our language. In the first case there is an open region, and we use the obvious isomorphism $\mathbb{C} \simeq \mathbb{C} \otimes \mathbb{C}$ to build the associated families of linear maps.

$\forall s \in S, \mathbb{C} \simeq \mathbb{C} \otimes \mathbb{C}$

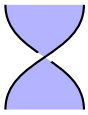
$1 \mapsto \sum_{i \in S} |i\rangle$ (1)

Flipping these components about a horizontal axis denotes the adjoint of these maps, as discussed above. With these definitions the equations illustrated in Figure 4 can be demonstrated; in that figure, the vertex L and the scalar $\lambda \in \mathbb{C}$ are arbitrary.

2.2 Shaded tangles

The *Reidemeister moves* [**?** , Section 2.4] are the basic relations of classical knot theory. In this section we present an equational theory of shaded knots, which use shaded versions of the Reidemeister moves. This theory follows work of Jones [**?**] on shaded tangle invariants from statistical mechanical models.

We begin by supposing the existence of a *shaded crossing*, depicted as follows:



(2)

We say that this crossing satisfies the *basic calculus* when it satisfies the equations of Figure 5(a)–(d), and the *extended calculus* when it additionally satisfies equation

Figure 5(e)–(f). In presenting this calculus, λ is an arbitrary nonzero constant, and we implicitly use the rule described in Section 2.1 regarding the representation of the adjoint as a reflected diagram, which causes the crossing type to change; this calculus also defines a rotated crossing in Figure 5(a). A *shaded tangle diagram* is a diagram constructed from the components of this calculus, the shaded cups (1), and their adjoints.

The extended calculus has the following attractive property.

Theorem 2.1. *Two shaded tangle diagrams with the same upper and lower boundaries are equal under the axioms of the extended calculus (up to an overall scalar factor) just when their underlying tangles, obtained by ignoring the shading, are isotopic as classical tangles.*

We emphasize that there is no corresponding full isotopy statement for the basic calculus since it only satisfies a subset of the shaded Reidemeister moves.

In **2Hilb**, we can classify representations of the basic calculus as follows. Note that from the discussion of Section 2.1, a vertex of type (2) represents in **2Hilb** a linear map of type $\mathbb{C}^{|S|} \to \mathbb{C}^{|S|}$, and is therefore canonically represented by a matrix, which we assume to have matrix entries $H_{a,b}$.

Theorem 2.2. *In* **2Hilb**, *a shaded crossing yields a solution of the basic calculus just when it is equal to a self-transpose Hadamard matrix.*

The following theorem identifies the additional constraint given by the extended calculus.

Theorem 2.3. *In* **2Hilb**, *a self-transpose Hadamard matrix satisfies the extended calculus just when:*

$$\sum_{r=0}^{|S|-1} \overline{H}_{ar} H_{br} H_{cr} = \sqrt{|S|}\, \overline{H}_{ab} \overline{H}_{ac} H_{bc} \tag{3}$$

A full classification of representations of this extended calculus is not known. However, it is known that solutions exist in all finite dimensions; we present this in Appendix A.

2.3 Programs and specifications

Scalar factors. From this point onwards we drop the scalar factors appearing in the shaded tangle calculus, since they complicate the diagrams. More formally, every

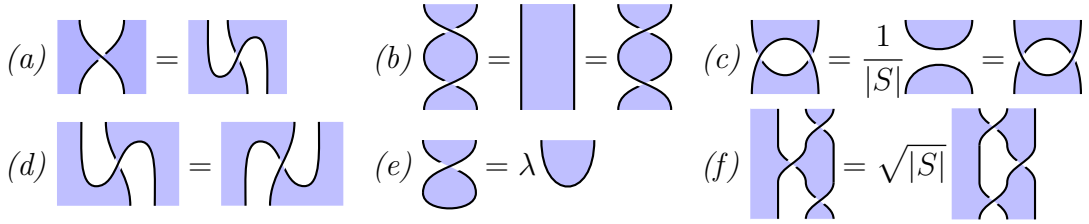


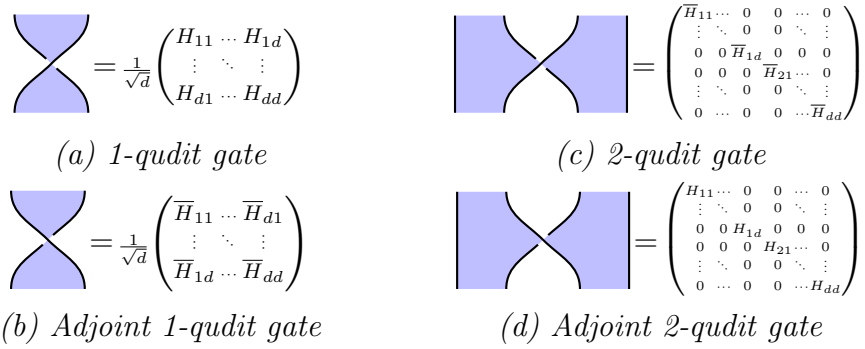
Figure 5: The shaded tangle calculus.

component we use in the remainder of the paper is proportional to an isometry, and we silently replace it with its isometric equivalent.

Programs. We write our quantum programs in terms of four basic components of this shaded tangle language.

- **Qudits.** As mentioned above, Figure 1(a) is interpreted as the identity map on $\mathbb{C}^{|S|}$, some finite-dimensional Hilbert space. This gives us our qudit.
- **Qudit preparations.** In expression (1) above we draw a blue 'cup' to indicate the state $\sum_{i \in S} |i\rangle \in \mathbb{C}^{|S|}$, which we interpret as a *qudit preparation* (see Figure 1(b).)
- **Qudit gates.** Applying our graphical calculus, given that (2) corresponds to a self-transpose Hadamard with matrix entries $H_{ij} = H_{ji}$, we obtain concrete representations for our 1- and 2-qubit gates, given in Figure 6.

Specifications. We can write our specifications using the entire language, including all the cups and caps arising from (1) and their adjoints. We excluded some of these components from the program language illustrated in Figure 1 because they are not directly interpretable as circuit components. This does not prevent us using them in specifications, however, since these will not be directly executed; they exist only to define the mathematical behaviour of the overall program.



(a) 1-qudit gate *(c) 2-qudit gate*

(b) Adjoint 1-qudit gate *(d) Adjoint 2-qudit gate*

Figure 6: Explicit expressions for the 1- and 2-qudit gates.

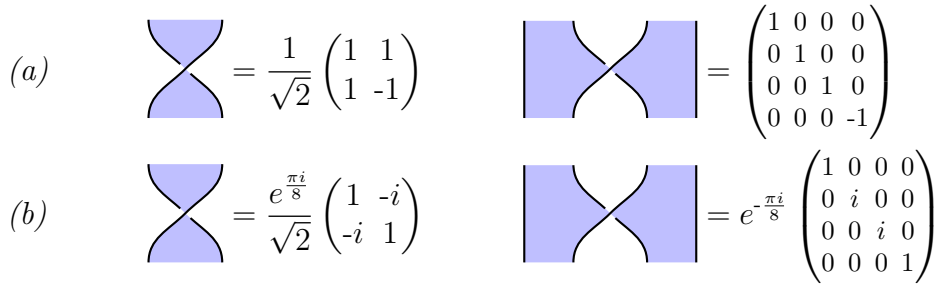


Figure 7: Different examples of our circuit elements.

Examples. We give some concrete examples of our basic circuit components. Some programs we analyze require only the basic calculus to be satisfied, in which case we can choose any self-transpose Hadamard matrix as our basic data. A standard choice is the *qubit Fourier Hadamard*, which satisfies the basic calculus but not the extended calculus, illustrated in Figure 7(a). Other programs require a Hadamard representing the extended calculus; an example is the metaplectic Hadamard illustrated in Figure 7(b) constructed using the methods of Appendix A. This Hadamard has been used in the cluster state literature for neighbourhood inversion on a cluster graph [**?** , Proposition 5], an operation we verify in Section 4.5 for a linear graph.

3 Entangled states

In this section we describe several forms of entanglement and their representations in our graphical calculus.

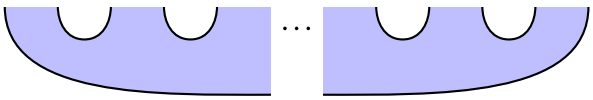
3.1 GHZ states

GHZ states were introduced by Greenberger, Horne and Zeilinger [**?**] to give a simplified proof of Bell's theorem. We define the unnormalized n-partite qudit GHZ state as follows:

$$|\mathrm{GHZ}_n\rangle := \sum_{k=0}^{d-1} |k, \cdots, k\rangle \tag{4}$$

Proposition 3.1. GHZ *states are represented as follows:*

$|\mathrm{GHZ}_n\rangle$ =



(5)

This arises directly from the representation of GHZ states in the CQM programme [**?**]. Important special cases for qubits are the $|+\rangle$ state $|\mathrm{GHZ}_1\rangle = |0\rangle + |1\rangle$, and the Bell state $|\mathrm{GHZ}_2\rangle = |00\rangle + |11\rangle$.

3.2 Cluster chains

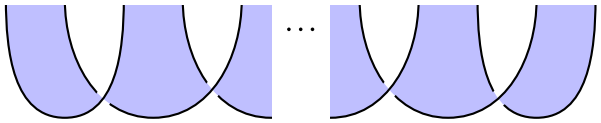
Another important class of entangled states are the *cluster states* or *graph states* [**?** **?** **?**] and their qudit generalizations associated to Hadamard matrices [**?**]. Cluster states have numerous applications, most prominently in the theory of measurement based quantum computation [**?** **?**] and quantum error correction [**?**]. Here we will focus on qudit *cluster chains*, cluster states entangled along a chain.

Given a self-transpose d-dimensional Hadamard matrix H, the n-partite qudit cluster chain associated to H is the following, where we conjugate the matrix due to our conventions:

$$|\mathrm{C}_n\rangle := \sum_{a_1,\ldots,a_n=0}^{d-1} \overline{H}_{a_1,a_2} \cdots \overline{H}_{a_{n-1},a_n} |a_1 \cdots a_n\rangle \tag{6}$$

Proposition 3.2. *Cluster chains are represented as follows:*

$|\mathrm{C}_n\rangle$ =



(7)

This is essentially the same as the representation of cluster states used in the CQM programme [**?** **?**]. If all Hadamard matrices in (6) are the Fourier Hadamard, then this recovers conventional cluster chains [**?**].

3.3 Tangle gates and tangle states

More generally, a *tangle gate* is any circuit built from 1- and 2-qudit gates and their adjoints, and a *tangle state* is a tangle with no inputs built from qudit preparations and tangle gates (see Figure 8.) Such tangle states and gates can be arbitrarily complex, and have all the algebraic richness of knot topology. If a Hadamard represents the extended calculus, then two tangle states or gates are equal just when the corresponding tangles are isotopic, as established by Theorem 2.1.

4 Manipulating quantum states

In this section we verify a wide variety of programs for creating and manipulating entangled states, including a new program for robust state transfer within a cluster chain–based quantum computer.

4.1 Constructing GHZ states (Figure 2)

Overview. We can use our formalism to design and verify a program for constructing n-partite GHZ states.

Program Figure 2(a). Begin by preparing n qudits, then apply a sequence of 2- and 1-qudit gates as indicated in Figure 2(a) for $n = 3$ qudits.

Specification Figure 2(b). This is the 3-qudit instance of (5), the tangle state corresponding to a GHZ-state.

Verification. Immediate by isotopy: the middle and rightmost qudit preparations in Figure 2(a) move up and left, underneath the diagonal strand, producing Figure 2(b).

Reidemeister moves. This only requires the basic calculus.

Novelty. The GHZ version is known for the qubit Fourier Hadamard and was described very recently [**?**] for the *qudit Fourier matrices* $H_{ab} = e^{\frac{2\pi i}{d}ab}$. For the self-transpose qudit Hadamard case covered here, the procedure seems new.

4.2 Creating cluster chains

Overview. Analogously to Section 4.1, we can consider the design of a program to create a cluster chain.

Figure 8: A tangle gate and a tangle state.

Program. We illustrate this in (7): we begin with n qudit preparations, then perform 2-qudit gates a total of $n - 1$ times.

Specification. We also take expression (7) to be the specification, Proposition 3.2 shows that this recovers the standard algebraic definition (6).

Verification. Trivial, the program and specification are equal.

Novelty. This procedure is well known for conventional and generalized cluster states [**?** **?**]. Our treatment here is not fundamentally different to the CQM analysis of Coecke, Duncan and Perdrix [**?** **?**].

4.3 Local unitary equivalence (Figures 9 and 10)

Overview. In the case of 2 or 3 parties, cluster chains can be converted into GHZ states by applying 1-qudit gates on certain sites. This means that, in a strong sense, they are equivalent computational resources. The reverse process, converting GHZ states to cluster chains, could be just as easily described.

Program. For 2 and 3 parties, we illustrate the programs in Figure 9(a) and Figure 10(a), respectively. ① Construct a cluster state. ② Perform a 1-qubit gate at certain sites.

Specification. Illustrated in Figure 9(b) and Figure 10(b), these are instances of the general GHZ specification (5).

Verification. Immediate by isotopy. For the 2-party case, a loop of string in the lower-left of Figure 9(a) contracts to the top of the diagram, giving Figure 9(b). For the 3-party case we perform similar contractions for loops of string at the lower-left and lower-right of Figure 10(a), which move above and below a third strand respectively, giving Figure 10(b).

Reidemeister moves. This only requires the basic calculus.

(a) Program *(b) Specification*

Figure 9: Converting a 2-party cluster state into a GHZ state.

(a) Program *(b) Specification*

Figure 10: Converting a 3-party cluster state into a GHZ state.

Novelty. This is known both for conventional [**?**] and generalized [**?**] cluster chains. For more than 3 parties it is known to be false, and indeed our method fails in these instances.

4.4 Cutting cluster chains (Figure 11)

Overview. Given a cluster chain of length n we can *cut* a target node from the chain, yielding two chains of total length $n-1$ and the target node in the $|+\rangle$ state. In some variants the target node is instead destroyed by a projective measurement, and controlled operations performed on the adjacent qudits [**?** , Section 3]; the mathematical structure is identical to the version we analyze. A similar comment applies to the splicing procedure of Section 4.5.

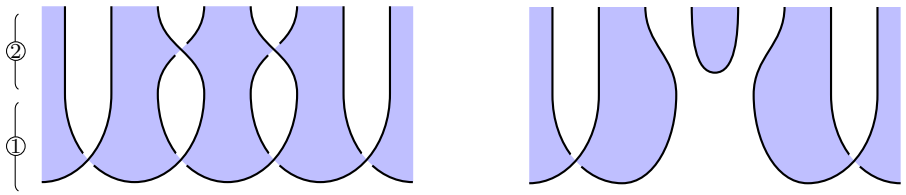
Program Figure 11(a). ① Prepare a cluster state, of which only a central part is shown. ② Perform two adjoint 2-qudit gates both involving a central target qudit.

Specification Figure 11(b). Prepare two separate cluster chains, and separately prepare the target node in the $|+\rangle$ state.

Verification. Immediate by isotopy: starting with Figure 11(a), we cancel the inverse pairs of crossings on the left and right of the central qudit, yielding Figure 11(b).

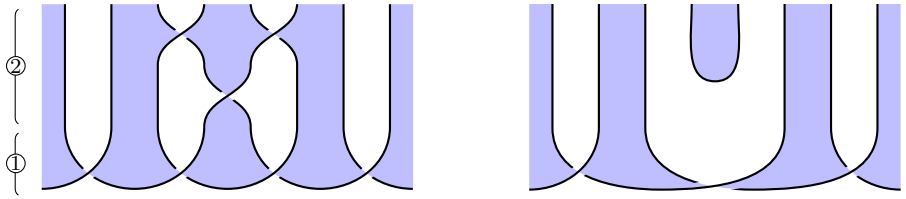
Reidemeister moves. This only requires the basic calculus.

Novelty. For the qubit Fourier Hadamard matrix this is well known; see [**?**] and [**?** , Section 3]. Here, and in the next subsection, our verification provides new insight into how these chain manipulation programs work.



(a) Program *(b) Specification*

Figure 11: Cutting a cluster chain.



(a) Program *(b) Specification*

Figure 12: Splicing a cluster chain.

4.5 Splicing cluster chains (Figure 12)

Overview. Given a cluster chain of length n we can *splice* a target node from the chain, yielding a single chain of length $n-1$, and the target node in the $|+\rangle$ state.

Program Figure 12(a). ① Prepare a cluster state, of which only a central part is shown. ② Perform a 1-qudit gate on the target qudit, and then two 2-qudit gates involving the target qudit and each of its adjacent qudits.

Specification Figure 12(b). Prepare a cluster chain of length $n-1$, and separately prepare the target node in the $|+\rangle$ state.

Verification. By isotopy, although harder to see by eye than previous examples. Looking closely, one can see that the target qudit in Figure 12(a) is unlinked from the other strings, and so the entire shaded tangle can be deformed to give Figure 12(b).

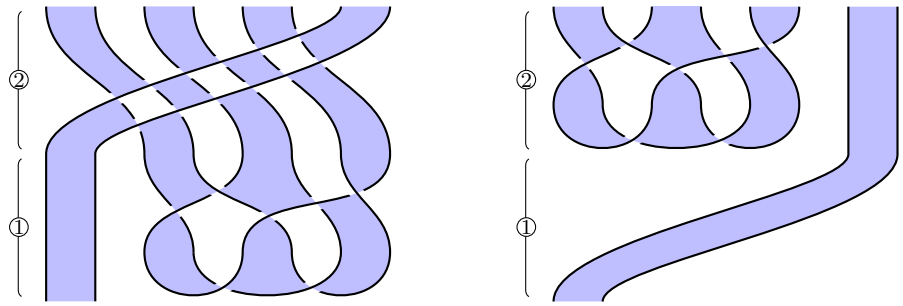
Reidemeister moves. This requires the extended calculus.

Novelty. It is well-known that certain local operations on cluster chains splice the chain (*neighbourhood inversion* on graph states, see [**?**] and [**?** , Prop. 5]); our analysis is more general since it applies for any qudit Hadamard satisfying the extended calculus. The standard procedures use cluster chains based on the qubit Fourier Hadamard, and require additional phase corrections, which effectively serve to convert the Hadamard into one representing the extended calculus. We avoid this by building the cluster chain itself from a Hadamard representing the extended calculus.

4.6 Cluster-based quantum state transfer (Figure 13)

Overview. In real quantum computing architectures that make heavy use of cluster states, such as the ion trap model [**?**], qubits are encoded in individual atomic structures, often arranged in a linear chain. One may want to move a target qubit to a different position in the chain—for example, to enable a multi-qubit gate to be applied, or to put the target qubit into position to be measured—but physically moving individual atoms may be impractical [**?**], and the 2-qubit swap gate $|ij\rangle \mapsto |ji\rangle$ may be hard to implement.

Here we introduce a state transfer program for moving a target qubit along a



(a) Program *(b) Specification*

Figure 13: Cluster-based quantum state transfer.

cluster chain, which uses only the tangle interaction used for generating the cluster states which the machine may be optimized to perform, and which is robust against tangle gate errors on the non-target qubits.

Program Figure 13(a). ① Begin with a target qudit on the left, and a cluster state to the right, corrupted with an arbitrary tangle gate. ② Perform a repeating sequence of 1- and 2-qudit tangle gates along the chain as indicated.

Specification Figure 13(b). ① Move the target qudit to the rightmost position. ② Recreate the cluster state with tangle gate error on the remaining qudits.

Verification. Immediate by isotopy: the tangle state in the lower-right of Figure 13(a) moves up and left, underneath the diagonal wires, producing the shaded tangle Figure 13(b).

Reidemeister moves. This requires the extended calculus.

Novelty. We believe this procedure is new.

5 Teleportation

Teleportation is a major theme in quantum information, playing an important structural role in the design of quantum computers. Here we use our topological calculus to verify a wide range of teleportation protocols. Our analysis in this section requires some small modifications to our graphical language, which we briefly describe.

Partitions. In this section it will often be important that the resources are *partitioned*, with qudits controlled by a number of different agents. Where appropriate we use vertical dashed lines to indicate this partitioning as an informal visual aid.

Measurement. It is standard that if a qudit is used only as the control side of a controlled 2-qudit unitary, then it may be considered as having been *measured*, and the controlled unitary interpreted as a classically controlled family of 1-qubit gates [**?** , Exercise 4.35]. To indicate this in our formalism, we shade the corresponding qudit red, and interpret it as a classical dit. As with partitions above, this is an informal visual aid; with respect to the mathematics, the red and blue shaded regions are equivalent. When a red region meets a vertical dashed partitioning line, we interpret this as classical communication.

Overlapping. When multiple red regions exist at once, we are sometimes forced by the protocol topology to draw them as *overlapping*. This will necessitate the use of new sorts of crossings, such as those in the upper-left of Figure 14(a). Unlike the crossings of blue regions which represent Hadamards, these crossings of red regions are mathematically trivial, and simply encode the reordering of classical data. In terms of our categorical semantics, this overlapping is described by the monoidal structure of the 2-category. In terms of knot theory this corresponds to the *shaded virtual knots* of Kauffman and others [**?**], which also have two kinds of crossing. In this extended abstract we treat aspects of virtual shaded knot theory, and the corresponding Reidemeister moves, informally.

5.1 Measurement-based GHZ teleportation (Figure 14)

Overview. Teleport a state from agent 1 to agent n using a shared n-partite GHZ state and classical communication, with all corrections performed by agent n. (Note relationship to Section 5.2.)

Program Figure 14(a). We illustrate the program for three agents: Alice, Bob and Charlie. ① Alice has a qudit to be teleported, and all agents share a GHZ state, perhaps generated according to Section 4.1. ② Alice applies a 2-qudit gate. ③ Alice and Bob measure all their qudits in the complementary basis determined by the Hadamard, and send the results classically to Charlie. ④ Charlie performs unitaries on his qudit, dependent on Alice and Bob's measurement results.

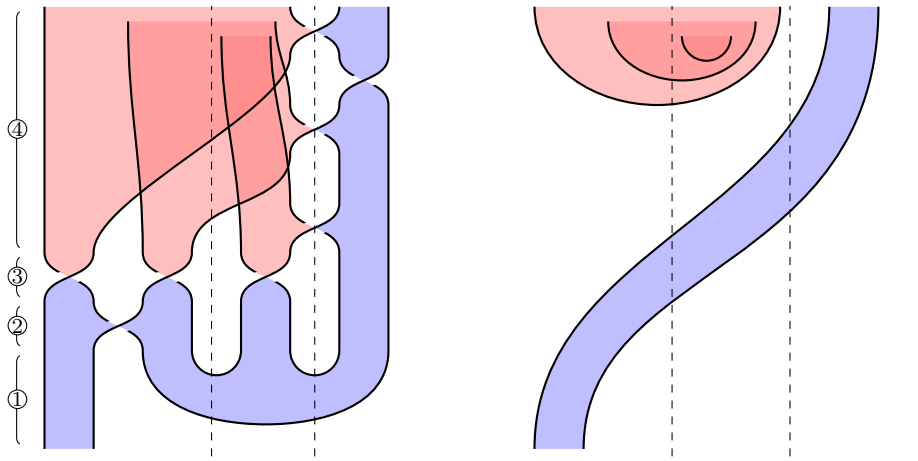
Specification Figure 14(b). Alice's qudit is passed to Charlie, and the classical dits are produced by measuring $|+\rangle$ states.

Verification. By isotopy; the three 'cups' forming the GHZ state can be pulled up one at a time.

Reidemeister moves. This only requires the basic calculus.

Novelty. This protocol was first described by Karlssen [**?**] for the qubit 3-party case, by Hillery [**?**] for the qubit n-party case and by Grudka [**?**] for qudit Fourier matrices. For general self-transpose Hadamards this seems new.

Discussion. This program can be understood as distributing Alice's initial state across all parties. Only after all parties cooperate and reveal their measurement results can Alice's state be reconstructed by Charlie. From this perspective the procedure is reminiscent of a secret sharing protocol, and it has been discussed in these terms by Hillery [**?**].



(a) Program *(b) Specification*

Figure 14: Measurement-based GHZ teleportation.

5.2 Measurement-based cluster chain teleportation (Figure 15)

Overview. Teleport a state from agent 1 to agent n using a shared n-party cluster chain and classical communication, with all corrections performed by agent n. (Note relationship to Section 5.1.)

Program Figure 15(a). Almost identical to the program of Section 5.1, except with an initial cluster chain rather than GHZ state, and Charlie's steps are slightly modified.

Specification Figure 15(b). Alice's qudit is passed to Charlie, and the classical dits are produced by measuring $|+\rangle$ states.

Verification. Similar to Section 5.1.

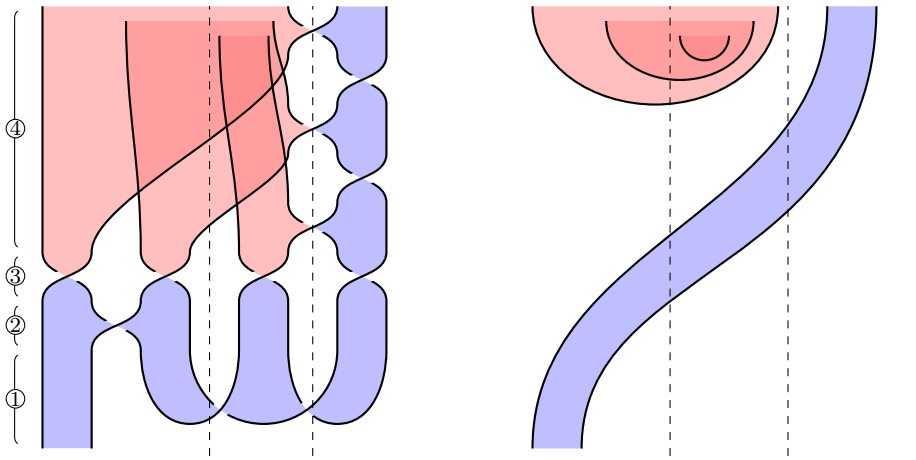
Reidemeister moves. This only requires the basic calculus.

Novelty. This program is known in the case of conventional qubit cluster states [**? ?**]. We believe this program is new for the generalized cluster chains considered here, based on arbitrary self-transpose Hadamards, and in fact a further generalization to arbitrary Hadamards is straightforward.

5.3 Robust GHZ teleportation (Figure 16)

Overview. Given a chain of n agents sharing a GHZ resource state, teleport a qudit from agent 1 to n, in a way which is robust against a large class of errors in the resource state.

Program Figure 16(a). We illustrate the program for 3 agents Alice, Bob and Charlie. ① Alice begins with a qudit to be teleported, and all 3 agents share a GHZ state, perhaps generated according to Section 4.1. ② An error in the form of an arbitrary tangle gate (see Section 3.3) acts on part of the GHZ state as shown. ③ Alice performs a 2-qudit gate, then measures her qudits in the complementary



(a) Program *(b) Specification*

Figure 15: Measurement-based cluster chain teleportation.

basis determined by the Hadamard, sending the left and right qudit results to Bob and Charlie respectively. ④ Bob applies a controlled unitary to his qudit, then measures his qudit in the complementary basis, and sends his result to Charlie. ⑤ Charlie performs unitaries on his qudit dependent on Alice's and Bob's results.

Specification Figure 16(b). Alice's qudit is passed to Charlie, and the classical dits are produced by applying a tangle gate (in fact, the shading-reversal of the tangle gate used in Figure 16(a)) to $|+\rangle$ states and measuring in the computational basis.

Verification. By isotopy, the entire tangle error can be pulled up, 'underneath' the lower diagonal strand, inverting its shading. The lower 'cup' of the GHZ state can then be pulled up similarly.

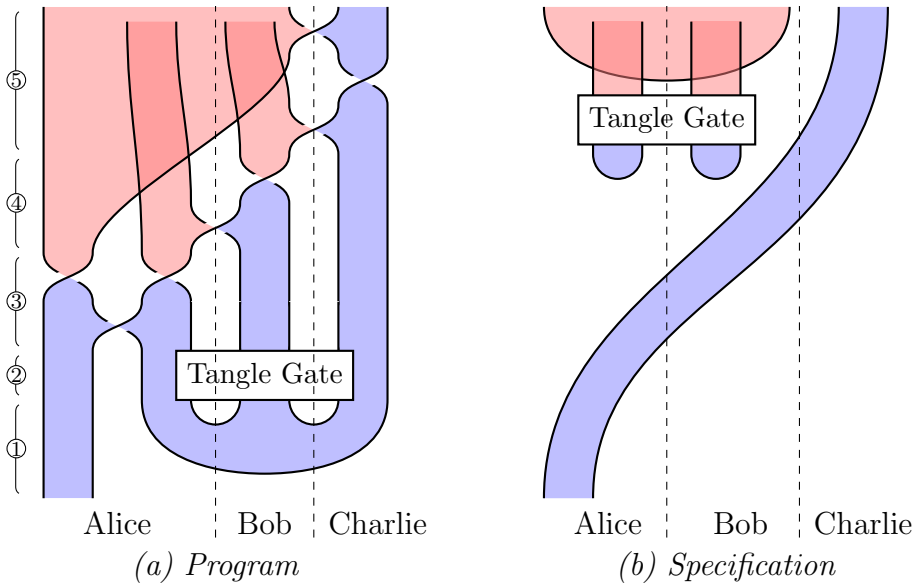
Reidemeister moves. The GHZ teleportation program requires the basic calculus, and robustness under tangle errors requires the extended calculus.

Novelty. Ignoring the robustness property, we believe this program to be folklore; note that for 2 parties it corresponds to ordinary Bell state teleportation, and the measure-correct pattern repeated here serves to convert $|\mathrm{GHZ}_n\rangle$ to $|\mathrm{GHZ}_{n-1}\rangle$. The generalization here to arbitrary self-transpose qudit Hadamards, and (with the extended calculus) the robustness property, seems to be new.

5.4 Nonlocal controlled unitaries (Figure 17)

Overview. Suppose that Alice and Bob have separate qudits, and they want to perform a 2-qudit controlled unitary of the following form, where Bob's qudit is the control:

$$U = \sum_{i=0}^{d-1} U_i \otimes |i\rangle \langle i|$$



(a) Program *(b) Specification*

Figure 16: Robust GHZ teleportation.

Both qudits are to be kept coherent throughout. The naive solution would be for one party to transport their system to the other party; for the 2-qudit unitary to be performed; and for the system to then be transported back. We describe a protocol to achieve this task with only one quantum transport required.

Program Figure 17(a). ① Bob prepares a $|+\rangle$ state, entangles it with his qudit and transports it to Alice. ② Alice performs a 1-qubit gate, performs the controlled unitary C, measures in the complementary basis, then sends the result to Bob. ③ Bob performs a correction.

Specification Figure 17(b). Alice's qudit is transported to Bob, who performs the controlled unitary, then passes it back. The classical dit arises from measuring a $|+\rangle$ state.

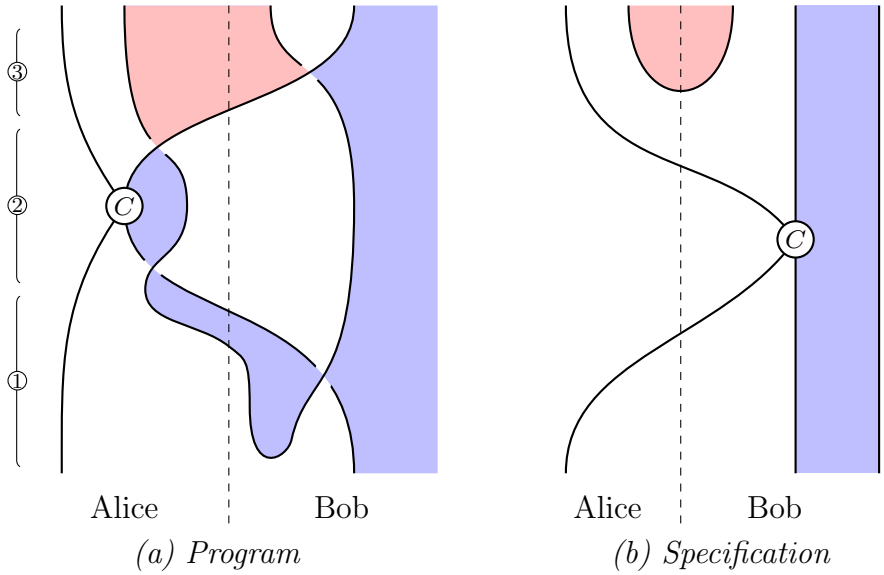
Verification. Immediate by isotopy.

Reidemeister moves. This only requires the basic calculus.

Novelty. A version of this program based on qudit Fourier Hadamards and involving an initial teleportation step was considered by Yu et al. [**?**] and described graphically by Jaffe et al. [**?**]. We generalize this here to arbitrary self-transpose Hadamard matrices. However, Jaffe et al describe a different generalization to multiple agents, which we cannot capture.

6 Quantum error correction

We now apply our shaded tangle calculus to the theory of quantum error correction. We give a graphical verification of the phase and Shor codes, and we give a substantial new generalization of both based on unitary error bases. This verification is based on the Knill-Laflamme theorem [**?**], a powerful theorem in quantum



(a) Program *(b) Specification*

Figure 17: Execution of a nonlocal controlled unitary.

information which establishes a correspondence between error correcting codes and mathematical properties of the encoding isometry. Given an encoding map satisfying these properties, a full error correcting protocol can be constructed.

Basic definitions. We begin by establishing notation. For $n, k, p, d \in \mathbb{N}$, an $[[n, k, p]]_d^{\mathcal{E}}$ code uses n physical qudits to encode k logical qudits, in a way which is robust against errors occurring on at most $\lfloor (p-1)/2 \rfloor$ physical qudits, such that each error is drawn from the subgroup $\mathcal{E} \subseteq U(d)$. We will be concerned with two types of errors: *full qudit errors*, for which $\mathcal{E} = U(d)$, and *phase errors*, for which $\mathcal{E} = \mathcal{P} \subset U(d)$, the subgroup of diagonal unitary matrices. The Knill-Laflamme theorem [**?**] gives a way to identify these codes.

Definition 6.1. An operator $e : (\mathbb{C}^d)^n \to (\mathbb{C}^d)^n$ is $(p, \mathcal{E})$-*local* when it is of the form $e = U_1 \otimes \cdots \otimes U_n$, such that $U_i \in \mathcal{E}$ for all $1 \leq i \leq n$, and such that at most $p-1$ of the operators U_i are not the identity.

Theorem 6.2 (Knill-Laflamme [**?**])**.** *An isometry $i : (\mathbb{C}^d)^k \to (\mathbb{C}^d)^n$ gives an $[[n, k, p]]_d^{\mathcal{E}}$ code just when, for any $(p, \mathcal{E})$-local operator $e : (\mathbb{C}^d)^n \to (\mathbb{C}^d)^n$, the composite*

$$(\mathbb{C}^d)^k \xrightarrow{i} (\mathbb{C}^d)^n \xrightarrow{e} (\mathbb{C}^d)^n \xrightarrow{i^\dagger} (\mathbb{C}^d)^k \tag{8}$$

is proportional to the identity.

Informally, the Knill-Laflamme theorem says that we have a $[[n, k, p]]_d^{\mathcal{E}}$ code just when, if we perform the encoding map, then perform a $(p, \mathcal{E})$-local error, then perform the adjoint of the encoding map, the result is proportional to our initial state. To be clear, any proportionality factor is allowed, even 0.

Representing errors. Following the general rules of our graphical calculus presented in Section 2.1, we represent arbitrary qudit phases and qudit gates as follows, respectively:

We draw them in red as they are interpreted here as errors.

6.1 The phase code

Overview. We present a $[[n, 1, n]]_d^{\mathcal{P}}$ code: that is, a code which uses n physical qudits to encode 1 logical qudit in a way that corrects $\lfloor (n-1)/2 \rfloor$ phase errors on the physical qudits. The data is a family of n d-dimensional Hadamard matrices.

Program. The encoding map i is depicted in Figure 18(a).

Specification. Satisfaction of the conditions of Theorem 6.2.

Verification. In Figure 19 we illustrate the $n = 3$ version of the code. We must therefore show that the composite $i^\dagger \circ e \circ i$, for any 3-local phase error in which 2 qudits are corrupted by arbitrary phases, is proportional to the identity. Given the symmetry of the encoding map, there are two cases: the errors can occur on

adjacent or nonadjacent qudits. We analyze the case of adjacent errors here; the verification for nonadjacent errors is analogous. In the first image of Figure 19 we represent the composite $i^\dagger \circ e \circ i$, using some artistic licence to draw the closed curves as circles. We apply RII moves to cause the errors to become 'captured' by bubbles floating in unshaded regions, which therefore (see Figure 4(f)) give rise to overall scalar factors.

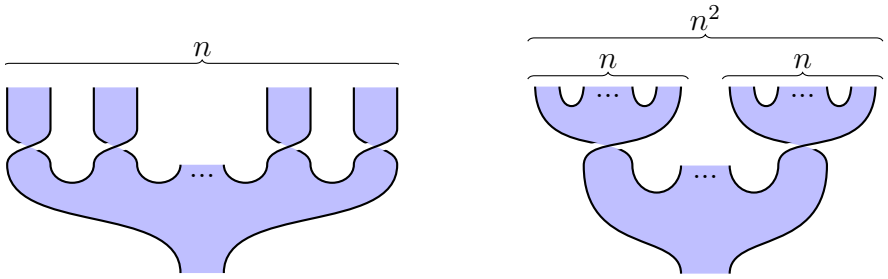
Reidemeister moves. This only requires the basic calculus.

Novelty. A major novel feature is the visceral sense of how the protocol works that Figure 19 conveys: the phase errors are 'captured by bubbles' and turned into scalar factors. We believe this intuition has not been described elsewhere.

In terms of the mathematics, for the qubit Fourier Hadamard, this code is very well known [**?** **?**]. The generalization to arbitrary qudit Hadamard follows from work of Ke [**?**]. Our treatment reveals a further generalization: each of the n Hadamards used to build the encoding map i may be distinct, since throughout the verification, we only ever apply the basic calculus moves to a Hadamard and its own adjoint. Furthermore, note that our usual requirement for the Hadamards to be self-transpose is not necessary here, since we never rotate the crossings.

6.2 The Shor code

Overview. We present a $[[n^2, 1, n]]_d^{U(d)}$ code: that is, a code which uses n^2 physical qudits to encode 1 logical qudit in a way that corrects $\lfloor (n-1)/2 \rfloor$ arbitrary physical qudit errors. The data is a family of n d-dimensional Hadamard matrices.



(a) Phase code *(b) Shor code*

Figure 18: The encoding maps i for the phase and Shor codes.

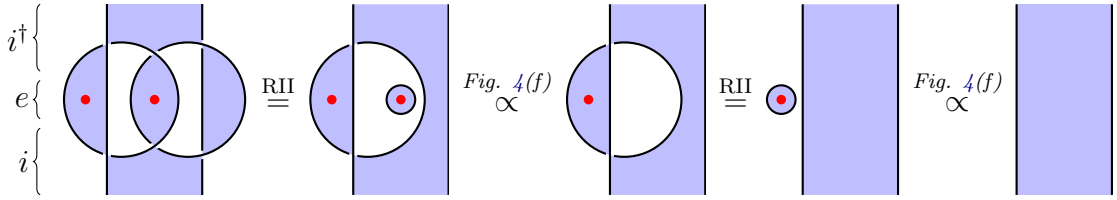


Figure 19: Verification of the $[[3, 1, 3]]_d^{\mathcal{P}}$ phase code.

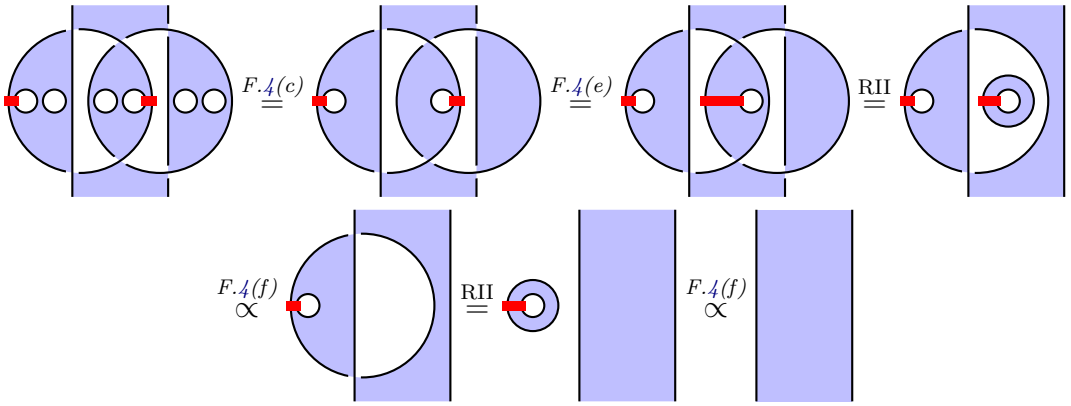


Figure 20: Verification of the $[[9, 1, 3]]_d^{U(d)}$ *Shor code.*

Program. We choose the encoding map from Figure 18(b).

Specification. Satisfaction of the conditions of Theorem 6.2.

Verification. In Figure 20 we illustrate one error configuration for the $n = 3$ case, where e encodes two full qudit errors. All other cases work similarly. The general principle is the same as for Section 6.1.

Reidemeister moves. This only requires the basic calculus.

Novelty. For $d = 2$, $n = 3$ and the qubit Fourier Hadamard, this is exactly Shor's 9-qubit code [**?**]. The qudit generalization for a general Hadamard is discussed in [**?**]. As with the phase code, our version is more general still, since each of the Hadamards can be different.

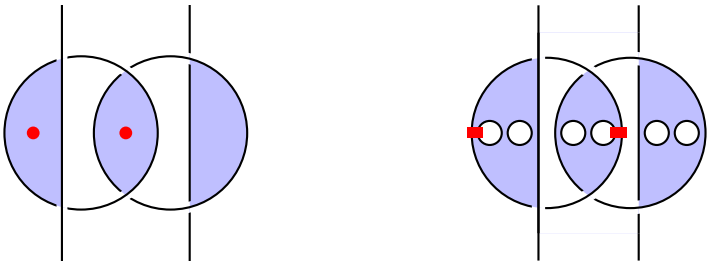
6.3 Unitary error basis codes

Overview. We show that the phase and Shor codes described above still work correctly when the Hadamards are replaced by *unitary error bases* (UEBs). These new codes have the same types $[[n, 1, n]]_{d^2}^{\mathcal{P}}$, $[[n^2, 1, n]]_{d^2}^{U(d)}$ as the phase and Shor codes, except with the additional restriction that the systems are of square dimension, since unitary error bases always have a square number of elements.

Unitary error bases (UEBs). UEBs are fundamental structures in quantum information which play a central role in quantum teleportation and dense coding [**?**], and also in error correction when they satisfy the additional axioms of a *nice error basis* [**?**]. However, the new UEB codes we present here are seemingly unrelated, and do *not* require the additional nice error basis axioms. UEBs are defined as follows.

Definition 6.3. On a finite-dimensional Hilbert space H, a *unitary error basis* is a basis of unitary operators $U_i : H \to H$ such that $\mathrm{Tr}(U_i^\dagger U_j) = \delta_{ij} \dim(H)$.

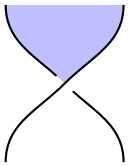
UEBs have an elegant presentation in terms of our graphical calculus [**?** **?**].



(a) Phase code *(b) Full qudit code*

Figure 21: The Knill-Laflamme condition for UEB codes.

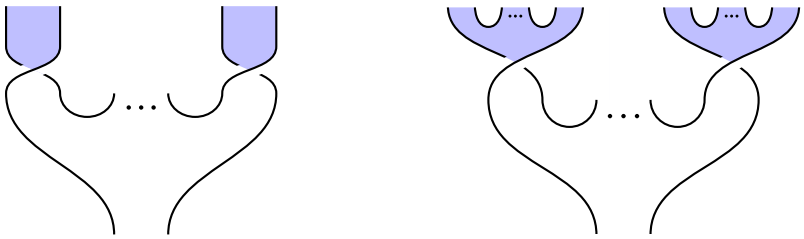
Theorem 6.4 (See [**?**])**.** *Unitary error bases correspond to vertices of the following type, satisfying equations analogous to Figure 5(b) and (c):*



(9)

For a precise description of the necesasry equations see [**?** , Proposition 9]. The wires with unshaded regions on both sides represent the Hilbert space $\mathbb{C}^d$, and the shaded region is labelled by a set of cardinality d^2.

Program. We choose the following encoding maps to generalize the phase and Shor codes, respectively:



Specification. Satisfaction of the conditions of Theorem 6.2.

Verification. The procedure is identical to the phase and Shor code verifications, the only difference being that some regions are differently shaded. To make this clear, in Figure 21 we give the graphical representations of the Knill-Laflamme $i^\dagger \circ e \circ i$ composites for these new codes; compare these images to the first graphics in Figures 19 and 20.

Novelty. As error correcting codes, these have precisely the same strength as the traditional phase and Shor codes. However, they are constructed from completely different data, and therefore push the theory of quantum error correcting codes in a new direction. (Although some UEBs can be constructed from Hadamards, they do not all arise in that way [**?** **?**].) This showcases the power of our approach to uncover new paradigms in quantum information.

Data accessibility. This paper has no additional data.

Authors' contributions. Both authors contributed equally to the current work.

Competing interests. We declare we have no competing interests.

Funding. J.V. is funded by a Royal Society University Research Fellowship.

Ethics. This work did not involve any collection of human data.

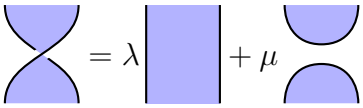
Acknowledgements. We thank Paul-André Mellies for suggesting the shaded tangle representation, Amar Hadzihasanovic for detailed conversations about the protocol in Section 5.4, Matty Hoban, Nathan Bowler and Niel de Beaudrap for telling us some useful things about cluster states, and Arthur Jaffe, Zhengwei Liu and Alex Wozniakowski for discussions about planar para algebras.

References

A Reidemeister III Hadamard matrices

The additional RIII condition (3) induces substantial constraints on a self-transpose Hadamard matrix. Here, we show that these equations have solutions in all finite dimensions. We consider two different families of solutions: *Potts-Hadamard matrices*, and *metaplectic invariants*. Almost everything in this section follows directly from results of Jones [**?**] on building link invariants from statistical mechanical models.

Potts-Hadamard matrices. A *Potts-Hadamard matrix* is a self-transpose Hadamard matrix of the following form, that satisfies (3):



(10)

In tensor notation, this means that $H_{a,b} = \lambda\,\delta_{a,b} + \mu$. We can classify Potts-Hadamard matrices exactly.

Theorem A.1. *Every Potts-Hadamard matrix has* $\mu = \frac{1}{\sqrt{d}}\,\overline{\lambda}$ *with*

$$\lambda \in U(1) \quad \textit{and} \quad \lambda^2 + \overline{\lambda}^2 = -\sqrt{d} \tag{11}$$

where d *is the dimension of the Hadamard matrix. This has the following solutions:*

- $d = 2$ *and* $\lambda \in \{e^{\frac{3\pi i}{8}}, e^{-\frac{3\pi i}{8}}, e^{-\frac{5\pi i}{8}}, e^{\frac{5\pi i}{8}}\}$;
- $d = 3$ *and* $\lambda \in \{e^{\frac{5\pi i}{12}}, e^{-\frac{5\pi i}{12}}, e^{-\frac{7\pi i}{12}}, e^{\frac{7\pi i}{12}}\}$;
- $d = 4$ *and* $\lambda \in \{i, -i\}$.

The $d = 2$ Potts-Hadamard matrices have the following form:

$$\frac{e^{\text{-}\frac{\pi i}{8}}}{\sqrt{2}}\begin{pmatrix}1 & i\\ i & 1\end{pmatrix} \qquad \frac{e^{\frac{\pi i}{8}}}{\sqrt{2}}\begin{pmatrix}1 & \text{-}i\\ \text{-}i & 1\end{pmatrix} \qquad \frac{e^{\frac{7\pi i}{8}}}{\sqrt{2}}\begin{pmatrix}1 & i\\ i & 1\end{pmatrix} \qquad \frac{e^{\text{-}\frac{7\pi i}{8}}}{\sqrt{2}}\begin{pmatrix}1 & \text{-}i\\ \text{-}i & 1\end{pmatrix}$$

In fact, it can be shown by direct calculation that these are the only two dimensional self-transpose Hadamard matrices fulfilling (3).

Equation (10) (together with (11)) is a rescaled version of the defining relation of Kauffman's bracket polynomial [**?**]; evaluating one of these matrices on a closed link diagram therefore yields (after suitable renormalization) the Jones polynomial of the link at certain roots of unity.

Metaplectic invariants. Following Jones and others [**?** **?** **?**], given $d \in \mathbb{N}$ with $d > 0$, we make the following definitions:

$$\xi := -e^{\frac{\pi i}{d}} \qquad\qquad \omega := \tfrac{1}{\sqrt{d}} \textstyle\sum_{k=0}^{d-1} \xi^{k^2} \tag{12}$$

Let λ be a square root of ω, and for $0 \le a, b \le d-1$, define $H_{a,b}$ as follows:

$$H_{a,b} = \frac{\overline{\lambda}}{\sqrt{d}}\,\xi^{(a-b)^2} \tag{13}$$

Then we have the following.

Theorem A.2. *The coefficients* $H_{a,b}$ *define a self-transpose Hadamard satisfying* (3).

This establishes that solutions to our graphical equations can be found in all finite dimensions.

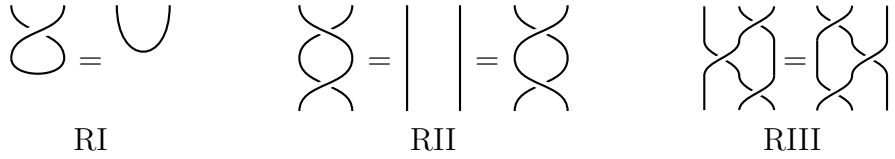


Figure 22: The unshaded Reidemeister moves up to rotations and reflections.

B Omitted proofs

Theorem 2.1. *Two shaded tangle diagrams with the same upper and lower boundaries are equal under the axioms of the extended calculus (up to an overall scalar factor) just when their underlying tangles, obtained by ignoring the shading, are isotopic as classical tangles.*

Proof. It is well known that two tangle diagrams are isotopic just when they can be transformed into each other using local Reidemeister moves. All Reidemeister moves can be obtained from arbitrary rotations and reflections of the moves depicted in Figure 22. Since our tangles are shaded, they transform under *shaded Reidemeister moves* - ordinary Reideimeister moves with a choice of checkerboard shading. Thus, up to rotations and reflections, there are 2 shaded versions of RI, 4 shaded versions of RII and 2 shaded versions of RIII. To prove Theorem 2.1, we therefore have to show that (up to scalar factors) all these shaded Reidemeister moves are implied by the basic axioms of the extended calculus presented in Figure 5. Using the shaded RII equations, it can be shown that the two shaded RIII equations are equivalent. Using shaded RII and RIII, it can be shown that the two shaded RI equations are equivalent. Therefore, two shaded tangles are isotopic if and only if they can be transformed into each other using all four shaded RII equations and one shaded RI and RIII equation, respectively. □

Theorem 2.2. *In* **2Hilb***, a shaded crossing yields a solution of the basic calculus just when it is equal to a self-transpose Hadamard matrix.*

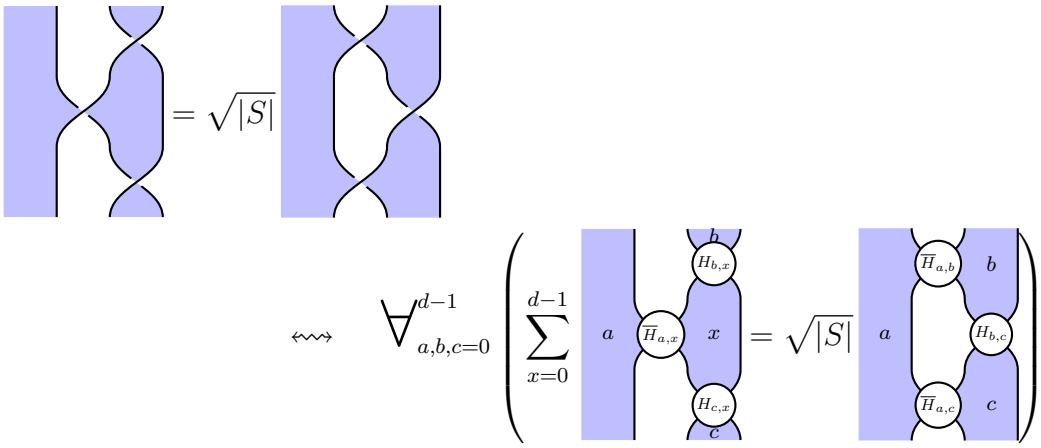
Proof. Solutions to the shaded Reimeister II equations in **2Hilb** Figure 5(b) and (c) were classified in terms of Hadamard matrices in [**?** , Proposition 7]. The additional equation Figure 5(d) implies that the corresponding Hadamard matrix is self-transpose. An equivalent classification using slightly different terminology can be found in [**?**]. □

Theorem 2.3. *In* **2Hilb***, a self-transpose Hadamard matrix satisfies the extended calculus just when:*

$$\sum_{r=0}^{|S|-1} \overline{H}_{ar} H_{br} H_{cr} = \sqrt{|S|}\, \overline{H}_{ab} \overline{H}_{ac} H_{bc} \tag{3}$$

Proof. Translating the Reimeister III equation Figure 5(f) into the corresponding

family of tensor diagrams (as described in Figure 3) yields the following:



Here a, b, and c label the left, top right, and bottom right shaded region, respectively. The central shaded region is labelled by x and summed over. Note that the Hadamard matrix H is self-transpose. Thus, this results in equation (3). Similarly, the Reidemeister I equation Figure 5(e) translates into the following equation which is a direct algebraic consequence of (3) for $a = b$ (with $\lambda = \sqrt{|S|}\,\overline{H}_{a,a}$): $\sum_{r=0}^{|S|-1} H_{c,r} = \lambda$. An equivalent classification using slightly different terminology can be found in [**?**]. □

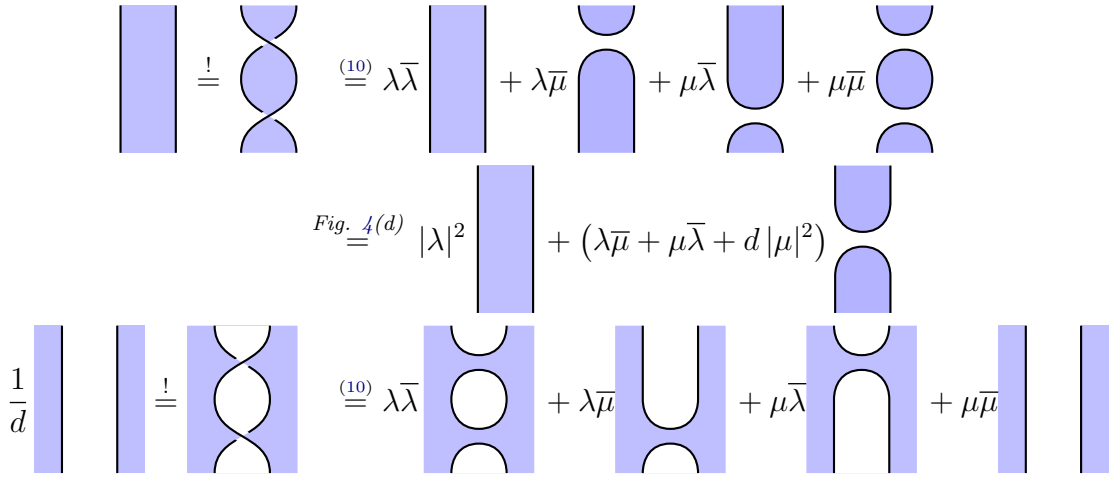
Theorem A.1. *Every Potts-Hadamard matrix has* $\mu = \frac{1}{\sqrt{d}}\overline{\lambda}$ *with*

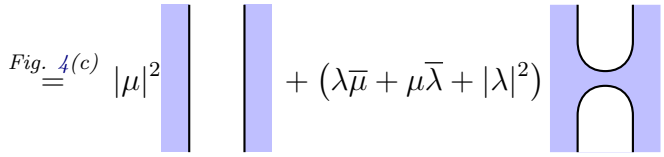
$$\lambda \in U(1) \quad \textit{and} \quad \lambda^2 + \overline{\lambda}^2 = -\sqrt{d} \tag{11}$$

where d *is the dimension of the Hadamard matrix. This has the following solutions:*

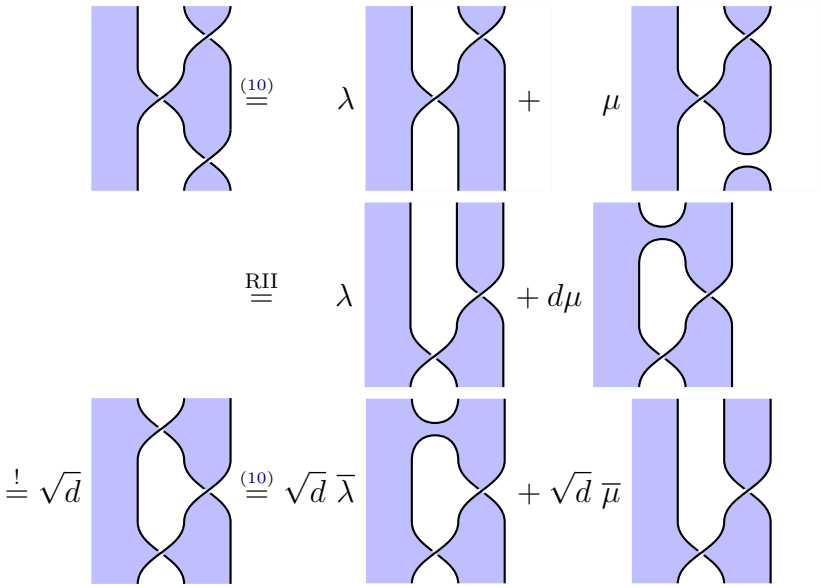
- $d = 2$ *and* $\lambda \in \{e^{\frac{3\pi i}{8}}, e^{-\frac{3\pi i}{8}}, e^{-\frac{5\pi i}{8}}, e^{\frac{5\pi i}{8}}\}$;
- $d = 3$ *and* $\lambda \in \{e^{\frac{5\pi i}{12}}, e^{-\frac{5\pi i}{12}}, e^{-\frac{7\pi i}{12}}, e^{\frac{7\pi i}{12}}\}$;
- $d = 4$ *and* $\lambda \in \{i, -i\}$.

Proof. For a shaded crossing of the form (10) the two Reidemeister II equations look as follows:





In other words, $|\lambda| = 1$, $|\mu| = \frac{1}{\sqrt{d}}$, and $\lambda\overline{\mu} + \mu\overline{\lambda} = -1$. Reidemeister III yields the following:



In short, $\mu = \frac{\overline{\lambda}}{\sqrt{d}}$. Together with the constraints from RII this proves the theorem. □

Theorem A.2. *The coefficients $H_{a,b}$ define a self-transpose Hadamard satisfying (3).*

Proof. Note that ω (defined in (12)) and its square root λ have modulus one. A proof of this fact using the discrete Fourier transform can be found in [**?** , Proposition 2.15]. Therefore, $|H_{a,b}| = \frac{1}{\sqrt{d}}$. The matrix H is unitary, since

$$\sum_{c=0}^{d-1} H_{a,c}\overline{H}_{b,c} \overset{(13)}{=} \frac{1}{d}\sum_{c=0}^{d-1} \xi^{(a-c)^2-(b-c)^2} = \frac{1}{d}\xi^{a^2-b^2}\sum_{c=0}^{d-1}\xi^{2bc-2ac} \overset{(12)}{=} \frac{1}{d}\xi^{a^2-b^2}\sum_{c=0}^{d-1} e^{\frac{2\pi i}{d}(b-a)c} = \delta_{a,b}.$$

It satisfies (3), since

$$\begin{aligned}\sum_{r=0}^{d-1}\overline{H}_{a,r}H_{b,r}H_{c,r} &\overset{(13)}{=} \frac{\overline{\lambda}}{d^{\frac{3}{2}}}\sum_{r=0}^{d-1}\xi^{-(a-r)^2+(b-r)^2+(c-r)^2} = \frac{\overline{\lambda}}{d^{\frac{3}{2}}}\xi^{b^2+c^2-a^2}\sum_{r=0}^{d-1}\xi^{r^2+2r(a-b-c)} \\ &= \frac{\overline{\lambda}}{d^{\frac{3}{2}}}\xi^{b^2+c^2-a^2-(a-b-c)^2}\sum_{r=0}^{d-1}\xi^{(r+(a-b-c))^2} = \frac{\overline{\lambda}}{d^{\frac{3}{2}}}\xi^{b^2+c^2-a^2-(a-b-c)^2}\sum_{r=0}^{d-1}\xi^{r^2} \\ &\overset{(12)}{=} \frac{\lambda}{d}\xi^{-2a^2+2ab+2ac-2bc}.\end{aligned} \tag{14}$$

The second equality in (14) holds since $\xi^{d^2} = 1$. On the other hand,

$$\sqrt{d}\,\overline{H}_{a,b}\overline{H}_{a,c}H_{b,c} = \frac{\lambda}{d}\xi^{-(a-b)^2-(a-c)^2+(b-c)^2} = \frac{\lambda}{d}\xi^{-2a^2+2ab+2ac-2bc}.$$

Thus, H is a self-transpose Hadamard matrix fulfilling (3).

□