A Complete Equational Presentation of Qudit Circuits via Polycontrolled PROPs

Colin Blake

Université de Lorraine, CNRS, Inria, LORIA, Nancy, France

Abstract

High-dimensional quantum computation needs a native circuit-level equational theory for qudits. We give the first finite schematic equational theory that is sound and complete for exact unitary qudit circuits in every finite dimension at least two. The result is entirely circuit-level: circuits are built from local gates, sequential and parallel composition, and value-controls, and equality is derivable exactly when two circuits have the same standard unitary semantics. For each dimension, the theory is presented by a finite family of local bounded-arity axiom schemata whose diagrammatic shapes are uniform in the dimension. The key syntactic ingredient is primitive value-control, which builds control on a chosen basis value directly into the language. This gives the language a useful internal algebra of controlled operations from local rules while keeping the presentation native to qudit circuits. The result provides a finite, dimension-uniform foundation for exact equational reasoning about qudit circuits.

2012 ACM Subject Classification Theory of computation → Quantum computation theory; Theory of computation → Categorical semantics; Theory of computation → Equational logic and rewriting

Keywords and phrases Qudit circuits, Quantum circuits, Completeness, Control, Categorical quantum mechanics

Digital Object Identifier 10.4230/LIPIcs...

Funding This work is supported by the Plan France 2030 through the PEPR integrated project EPiQ (ANR-22-PETQ-0007) and the HQI platform (ANR-22-PNCQ-0002); by the European Union through the MSCA Staff Exchanges project QCOMICAL (Grant Agreement ID: 101182520); and by the Maison du Quantique MaQuEst.

Acknowledgements I thank my PhD advisors, Simon Perdrix and Miriam Backens, for their helpful feedback and for reviewing earlier versions of this article.

1 Introduction

Qudit circuits make higher-dimensional quantum data visible at circuit level: basis values of a d-level carrier can be named, controlled, and rewritten directly, which matters both for hardware where extra levels are native and for compilers that exploit those levels as structure. For $d > 2$, high-dimensional carriers are a native resource [33], and multilevel entangling operations have already been demonstrated, for example, in trapped-ion processors [19]. The extra levels also matter when the algorithm is logically qubit-based: qutrit workspace can implement generalized Toffoli gates without external ancillas and with lower depth, while ququart encodings can shorten three-qubit constructions [16, 24]; the small example in Figure 1 shows the mechanism: packing two qubits into one four-level system turns two binary controls into one value-control and turns a two-qubit target subspace into an adjacent two-level ququart subspace.

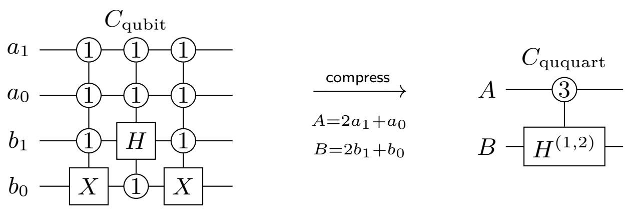


Figure 1 Compressing four qubits into two ququarts via $|a_1a_0\rangle\,|b_1b_0\rangle \mapsto |A\rangle\,|B\rangle$. The branch $a_1 = a_0 = 1$ becomes $A = 3$, while the target subspace $\text{span}\{|01\rangle\,,|10\rangle\}$ becomes $\text{span}\{|1\rangle\,,|2\rangle\}$, giving $\mathtt{C}_3(H^{(1,2)})$.

This kind of compression is useful only if the packed circuits can still be reasoned about locally. A compiler may introduce value-controls, adjacent two-level gates, or swaps between encoded levels; a verifier then has to show that these local changes preserve the intended unitary. That is the setting of this paper: circuit-level rewriting for qudit circuits.

Quantum circuits provide the standard low-level syntax for finite-dimensional quantum computation [13]: wires carry d-level quantum data, boxes represent unitary gates, and an n-wire circuit denotes a unitary operator on $(\mathbb{C}^d)^{\otimes n}$. A rewrite system for such circuits should therefore be stable under the two ways circuits are built: sequential composition and parallel composition. In PROP language, where objects are wire counts, this asks for local equations between diagrams, closed under those compositions and under the structural symmetries. The question is then sharp: can equality of unitary qudit circuits be axiomatized by a *finite schematic*, syntax-directed equational theory in the circuit PROP?

For qubits ($d = 2$), the answer is known to be yes [6], with later simplifications and minimal presentations [7, 4]. The qubit proofs also identify the pressure point. Control cannot be treated as an afterthought: if it is absent from the syntax, completeness tends to reintroduce it through rules with arbitrarily many explicit control wires. Controlled PROPs avoid that blow-up by treating control functorially, which restores finite, bounded-arity axiom sets [12] and connects with the categorical account of control constructors in [17].

Graphical calculi provide complementary complete languages for quantum processes, including qudit and finite-dimensional variants of ZX/ZW-style systems [9, 20, 1, 31, 29, 32, 27, 11]. Their natural target is a broad category of linear maps, so returning to a circuit-level unitary can require a non-local extraction step [10]. Existing circuit-level qutrit and qudit results are more specialised: phase-gadget methods treat qutrit diagonal gates [30], complete rule sets cover multi-qutrit Clifford fragments [21], and generalized-Clifford formalisms handle multi-qudit computations [22, 23]. The remaining gap is the full unitary qudit circuit category itself: wire arities as objects, circuits as morphisms, and, for every $d \geq 2$, a finite schematic uniform-in-d presentation by bounded-arity local axiom schemata.

Beyond qubits, the difficulty is to find a qudit-native circuit presentation on which completeness can still be proved by finite local rules. Several binary shortcuts in the qubit completeness proof of [6] become case-sensitive for adjacent two-level gates: a local optical relation may decode either to a single adjacent-level chain or to a mixed row/column configuration guarded by basis controls. The proof therefore needs structured control management while remaining inside a native qudit circuit language.

Our answer is to make each value-control primitive. For every basis value $k \in [d]$, a control functor $\mathtt{C}_k$ sends an endomorphism $f : n \to n$ to its k-controlled extension $\mathtt{C}_k(f) : 1{+}n \to 1{+}n$. The payoff is concrete: axioms may mention $\mathtt{C}_k(-)$ directly, rather than simulate it by a family of rules with a variable number of control wires.

Main result. For each finite $d \geq 2$, we define a polycontrolled circuit PROP $\mathbf{CQC}_d$ and a finite schematic equational theory QC_d generated by local axiom schemata uniform in d whose instances involve at most three wires; compatibility, commutativity of distinct branches, and exhaustivity are derived as the global control algebra in QC_d, not imposed as extra structure on $\mathbf{CQC}_d$. The main completeness theorem states that QC_d is sound and complete for the standard unitary semantics $[\![-]\!] : \mathbf{CQC}_d \to \mathbf{Qudit}_d$: for any n-qudit circuits $C_1, C_2 : n \to n$, $[\![C_1]\!] = [\![C_2]\!]$ if and only if $\mathrm{QC}_d \vdash C_1 = C_2$.

The proof is organized around three interfaces: the finite bounded-arity presentation of the axiom schemata (Theorem 3.1), the branch-control algebra derived from the support cells (Theorems 3.9–3.11), and the Gray-ordered linear-optical transfer proving completeness (Theorem 4.17). The transfer encodes n-qudit circuits as d^n-mode single-photon **LOPP** circuits, serialises tensor to match the optical direct-sum tensor, decodes optical derivations back into $\mathbf{CQC}_d$, and proves an encoding/decoding retraction. When $d = 2$, this specializes to the qubit transfer pattern of [6]; for larger d, reflected Gray order keeps optical neighbours adjacent in one qudit coordinate.

2 Qudit circuits as a polycontrolled PROP

Fix a finite dimension $d \geq 2$ and write $[d] = \{0, \dots, d-1\}$. A PROP is a strict symmetric monoidal category whose objects are natural numbers and whose tensor on objects is addition [25]. PROPs give the ambient circuit calculus used throughout the paper: objects are wire counts and morphisms are diagrams modulo strict symmetric-monoidal coherence. Since all gates considered here are unitary, the only non-empty hom-sets are endomorphism hom-sets.

The extra structure is value control. In ordinary circuit notation one freely draws a control and reads it by its action on a branch; here this operation is part of the categorical syntax. For a detailed analysis of controlled and polycontrolled PROPs and their circuit applications, see [12]; we only use the following interface.

▶ **Definition 2.1.** *For a PROP* $\mathbf{P}$*, let* $\mathbf{P}_{\mathrm{endo}}$ *be the wide subcategory with the same objects and only endomorphism hom-sets. A* control functor *is a functor* $\mathtt{C} : \mathbf{P}_{\mathrm{endo}} \to \mathbf{P}_{\mathrm{endo}}$ *such that* $\mathtt{C}(n) = 1 + n$ *on objects. For every* $f, g : n \to n$*,* $m \geq 0$*, and permutation* $\pi : n \to n$*, it satisfies*

$$\mathtt{C}(\mathrm{id}_n) = \mathrm{id}_{1+n}, \quad \mathtt{C}(g \circ f) = \mathtt{C}(g) \circ \mathtt{C}(f), \quad \mathtt{C}(f \otimes \mathrm{id}_m) = \mathtt{C}(f) \otimes \mathrm{id}_m,$$
$$\mathtt{C}(\pi^{-1} \circ f \circ \pi) = (\mathrm{id}_1 \otimes \pi^{-1}) \circ \mathtt{C}(f) \circ (\mathrm{id}_1 \otimes \pi),$$
$$\mathtt{C}(\mathtt{C}(f)) \circ (\sigma_{1,1} \otimes \mathrm{id}_n) = (\sigma_{1,1} \otimes \mathrm{id}_n) \circ \mathtt{C}(\mathtt{C}(f)).$$

A d-ary polycontrolled PROP *is a PROP equipped with control functors* $(\mathtt{C}_k)_{k\in[d]}$.

In this syntax, $\mathtt{C}_k(f) : 1+n \to 1+n$ is a circuit with one additional distinguished control wire. The equations say that unused wires remain unused, target-wire permutations commute with adding a control, and two nested controls of the same value may be swapped with the two control wires.

For a word $u = u_1 \cdots u_m \in [d]^*$ we write $\mathtt{C}_u(f) := \mathtt{C}_{u_1}(\cdots \mathtt{C}_{u_m}(f)\cdots)$, with $\mathtt{C}_\epsilon(f) = f$. Beyond the single-value control functor axioms, three global principles will be needed: compatibility between different values, commutation of different branches, and exhaustivity over all values. These principles are valid for the unitary interpretation of $\mathbf{CQC}_d$ defined below, but they are not placed in the structural congruence of circuits. Instead, Section 3.3 derives them from the finite axiom system.

▶ **Definition 2.2.** *Let* $(\mathtt{C}_k)_{k\in[d]}$ *be a family of control functors. For well-typed endomorphisms* $f, g : n \to n$*, we use the following three control-algebra laws.*

1. *The family is* compatible *when, for any* $a, b \in [d]$*, the two ways of nesting a- and b-controls around the same circuit agree up to the control-wire symmetry* $\sigma_{1,1} : 2 \to 2$*, i.e.* $\mathtt{C}_a(\mathtt{C}_b(f)) \circ (\sigma_{1,1} \otimes \mathrm{id}_n) = (\sigma_{1,1} \otimes \mathrm{id}_n) \circ \mathtt{C}_b(\mathtt{C}_a(f))$.
2. *The family is* commutative *when, for distinct* $a, b \in [d]$*, operations guarded by different values of the same control wire commute, i.e.* $\mathtt{C}_a(f) \circ \mathtt{C}_b(g) = \mathtt{C}_b(g) \circ \mathtt{C}_a(f)$.
3. *The family is* exhaustive *when all value-controlled branches of the same circuit compose to the uncontrolled target operation, i.e.* $\mathtt{C}_0(f) \circ \mathtt{C}_1(f) \circ \cdots \circ \mathtt{C}_{d-1}(f) = \mathrm{id}_1 \otimes f$*. The product in the exhaustivity law is composed in increasing order of the control value; once commutativity is available, this order is immaterial.*

In the matrix model, these branch laws say that on each computational-basis value of the control wire exactly one guarded operation is active.

▶ **Definition 2.3.** *We define a d-ary polycontrolled PROP to be* support-sensitive *when each endomorphism* $f : n \to n$ *is assigned a basis-support envelope* $\mathrm{supp}(f) \subseteq [d]^n$ *such that endomorphisms with disjoint supports commute: if* $\mathrm{supp}(f) \cap \mathrm{supp}(g) = \varnothing$*, then* $f \circ g = g \circ f$*. Supports must also be stable under the circuit constructors: for* $f, g : n \to n$ *and* $h : m \to m$*,* $\mathrm{supp}(f \circ g) \subseteq \mathrm{supp}(f) \cup \mathrm{supp}(g)$*,* $\mathrm{supp}(f \otimes h) \subseteq (\mathrm{supp}(f) \times [d]^m) \cup ([d]^n \times \mathrm{supp}(h))$*, wire symmetries transport supports by the corresponding permutation of basis words, and* $\mathrm{supp}(\mathtt{C}_k(f)) = \{k\} \times \mathrm{supp}(f)$.

▶ Remark 2.4. Support-sensitivity is stronger than distinct-control commutativity: if $a \neq b$, then $\mathrm{supp}(\mathtt{C}_a(f))$ and $\mathrm{supp}(\mathtt{C}_b(g))$ are disjoint, so the controlled operations commute.

The structural PROP supplies the empty diagram, identity wires, symmetries, sequential composition, and tensor. Set $\mathsf{G}_d := \{ \textcircled{\theta} : 0 \to 0 \mid \theta \in \mathbb{R}\} \cup \{ -\boxed{H^{(r,r+1)}}- : 1 \to 1 \mid 0 \le r < d-1\}$. This non-structural gate family consists of scalar phases indexed by θ and adjacent two-level Hadamards indexed by r. A phase on one basis level is expressed as $\mathtt{C}_i(\textcircled{\theta})$, which semantically multiplies $|i\rangle$ by $e^{i\theta}$ and fixes the other basis states.

▶ **Definition 2.5.** *Let* $\mathbf{CQC}_d^{\mathrm{raw}}$ *be the raw endomorphism syntax generated by the structural PROP operations, the gates in* G_d*, and the unary constructors* $\mathtt{C}_k(-)$ *for* $k \in [d]$*, with no non-endomorphism homs.*

▶ **Definition 2.6.** *Let* $\equiv$ *be the least congruence, closed under* $\circ$*,* $\otimes$*, and every* $\mathtt{C}_k(-)$*, containing the strict PROP coherence laws and the control-functor equations for each* $\mathtt{C}_k$.

▶ **Definition 2.7.** *The resulting d-ary polycontrolled circuit PROP is* $\mathbf{CQC}_d(n,n) := \mathbf{CQC}_d^{\mathrm{raw}}(n,n)/{\equiv}$*, with* $\mathbf{CQC}_d(m,n) := \varnothing$ *for* $m \neq n$*.*

The quotient $\equiv$ is structural rather than branch-semantic: compatibility between different values, branch commutativity, and exhaustivity must be proved inside QC_d.

Let $\mathbf{Qudit}_d$ be the semantic polycontrolled PROP with $\mathbf{Qudit}_d(n,n) = U(d^n)$ and no non-endomorphism homs. Symmetries are interpreted by wire swaps, and the remaining clauses are

$$\begin{aligned} [\![C_2 \circ C_1]\!] &= [\![C_2]\!]\,[\![C_1]\!], \qquad [\![C_1 \otimes C_2]\!] = [\![C_1]\!] \otimes [\![C_2]\!], \qquad [\![\textcircled{\theta}]\!] = e^{i\theta}, \\ \left[\!\left[-\boxed{H^{(r,r+1)}}- \right]\!\right] &= \text{Hadamard on } \mathrm{span}\{|r\rangle, |r+1\rangle\} \text{ and identity elsewhere}, \\ [\![\mathtt{C}_k(C)]\!] &= |k\rangle\langle k| \otimes [\![C]\!] + \textstyle\sum_{\ell\in[d],\,\ell\neq k} |\ell\rangle\langle\ell| \otimes I. \end{aligned}$$

▶ **Proposition 2.8.** *The interpretation respects the structural congruence and hence factors through a strict symmetric monoidal functor* $[\![-]\!] : \mathbf{CQC}_d \to \mathbf{Qudit}_d$*.*

Proof. The clauses are the standard matrix interpretation of strict monoidal circuits, so identities, composition, tensor, and symmetries respect the PROP coherence laws. The displayed block formula for $[\![\mathtt{C}_k(C)]\!]$ is functorial in C. It is also stable under padding and target-wire permutations, and it makes two equal-value controls symmetric after swapping the control wires. Thus every generator of the structural congruence has equal denotation, and the interpretation descends to a strict symmetric monoidal functor on $\mathbf{CQC}_d$. ◀

The signature is deliberately small, but it still reaches all finite-dimensional unitaries.

▶ **Proposition 2.9.** *For every* $n \geq 0$*,* $[\![-]\!]$ *is surjective on* n*-wire endomorphisms.*

Proof. For $n = 0$, scalar phases realise $U(1)$. For one qudit, adjacent two-level decompositions reduce an arbitrary element of $U(d)$ to adjacent supports [26, 8, 28, 14], each realised by Euler decompositions into the adjacent Hadamard, level phases, and a scalar phase [26, 2]. For $n > 1$, these one-qudit unitaries together with the controlled scalar $[\![\mathtt{C}_0(\mathtt{C}_0(\textcircled{\pi}))]\!]$ give the exact universal family of [3]: the controlled scalar is entangling for $d \geq 2$, scalar phases supply global phases, and tensors with identities and symmetries place the generators on any chosen wires. ◀

▶ **Definition 2.10.** *An equational theory* Γ *on* $\mathbf{CQC}_d$ *is a set of well-typed equations closed under equivalence closure, sequential composition, tensor, and every control constructor* $\mathtt{C}_k(-)$*. We write* $\Gamma \vdash f = g$ *for the generated congruence.*

▶ **Definition 2.11.** *For each* $d \geq 2$*,* QC_d *is the equational theory on* $\mathbf{CQC}_d$ *generated by the local axiom schemata displayed in Figures 2 and 3.*

At this stage QC_d is only a syntactic congruence generated by the displayed rules. The transfer argument of Section 4 proves that this congruence is exact: Theorem 4.17 identifies it with equality of unitary denotations.

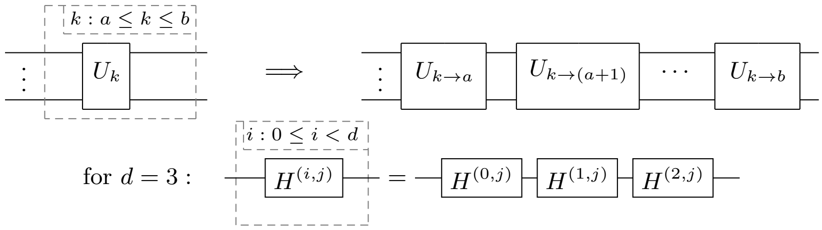
3 A Finite Axiom System for Qudit Circuits

The presentation below packages exactly the two ingredients needed for the completeness proof: the gate algebra of the qudit signature and the finite branch-control rules used by the later normalisation arguments. All equations in Figures 2–3 are read in $\mathbf{CQC}_d$, where the strict PROP laws and same-control coherence equations have already been quotiented out.

3.1 Reading the axiom schemata

The schemata below use a slightly higher-level notation, but no extra generators are being added. The non-structural generators are the gates G_d of Section 2; identities, symmetries, and the control constructors $\mathtt{C}_k(-)$ come from the polycontrolled PROP structure. Here $X^{(i,j)}$, $H^{(i,j)}$, and $R_x^{(i,j)}(\theta)$ are derived one-qudit circuits: semantically they are, respectively, a transposition of levels i, j, a Hadamard on $\mathrm{span}\{|i\rangle, |j\rangle\}$, and the corresponding two-level beam-splitter rotation. Their recursive definitions are given in Appendix B; in the main text we only need the interface. For instance, when $d = 3$, the non-adjacent swap is the adjacent walk $X^{(0,2)} = X^{(1,2)} \circ X^{(0,1)} \circ X^{(1,2)}$; the same pattern gives $H^{(0,2)}$ and $R_x^{(0,2)}(\theta)$.

The figures use *indexed product frames* as meta-notation for finite sequential products. A frame labelled $\vec{k} : \varphi(\vec{k})$ binds the displayed index tuple and expands to the composite of the enclosed diagram over all satisfying tuples, in lexicographic increasing order:



A tuple label such as $(a, b) : 0 \leq a < b < d$ is read lexicographically, with a the outer index and b the inner index. Product frames expand on the same wires, so they add repetitions without increasing arity.

3.2 The axiom schemata

▶ **Theorem 3.1.** *For every $d \geq 2$, QC_d is a finite schematic equational theory on $\mathbf{CQC}_d$. The diagrammatic shape of the rules is independent of d, and every expanded instance acts on at most three wires.*

Proof. The only d-dependent data are level indices and product frames. Product frames expand to sequential products on the same wires, while the displayed diagrams themselves contain at most three wires. The angle variables are ordinary schematic parameters; finiteness refers to the finite list of rule shapes. ◀

Figure 2 contains the gate algebra: phase arithmetic, basic two-level identities, the swap decomposition, and the local decomposition/rotation principles. Figure 3 contains the branch-control checks used to recover the global control principles of Definition 2.2. These checks matter because the later proof uses those principles as rewrites, while the structural congruence of $\mathbf{CQC}_d$ only contains the same-control functor laws.

The next proposition is only a semantic soundness check for the support cells, not an extra structural rule of $\mathbf{CQC}_d$.

▶ **Proposition 3.2.** *The semantic polycontrolled PROP $(\mathbf{Qudit}_d, (\mathsf{C}_k)_{k\in[d]})$ is support-sensitive.*

Proof. For $U \in \mathbf{Qudit}_d(n, n)$, set $\mathrm{supp}(U) := \{x \in [d]^n \mid U|x\rangle \neq |x\rangle\}$. The support inclusions for composition and tensor, and the formulas for symmetries and controls, follow from their matrix semantics. If U and V have disjoint supports S and T, then they act on different direct summands in the decomposition by S, T, and the remaining basis states, hence commute. ◀

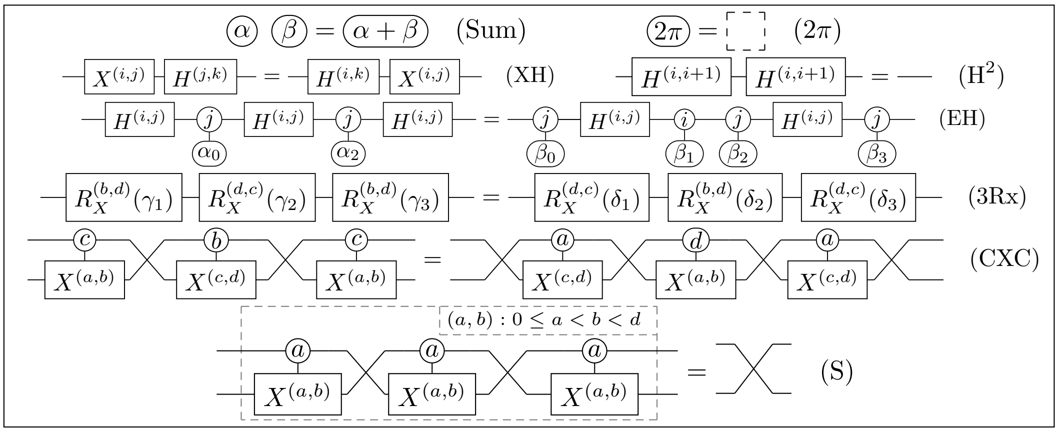


Figure 2 Core axiom schemata for QC_d**.** In (EH) and (3Rx) the right-hand-side angles are chosen according to the explicit admissible relations recorded in Appendix C.

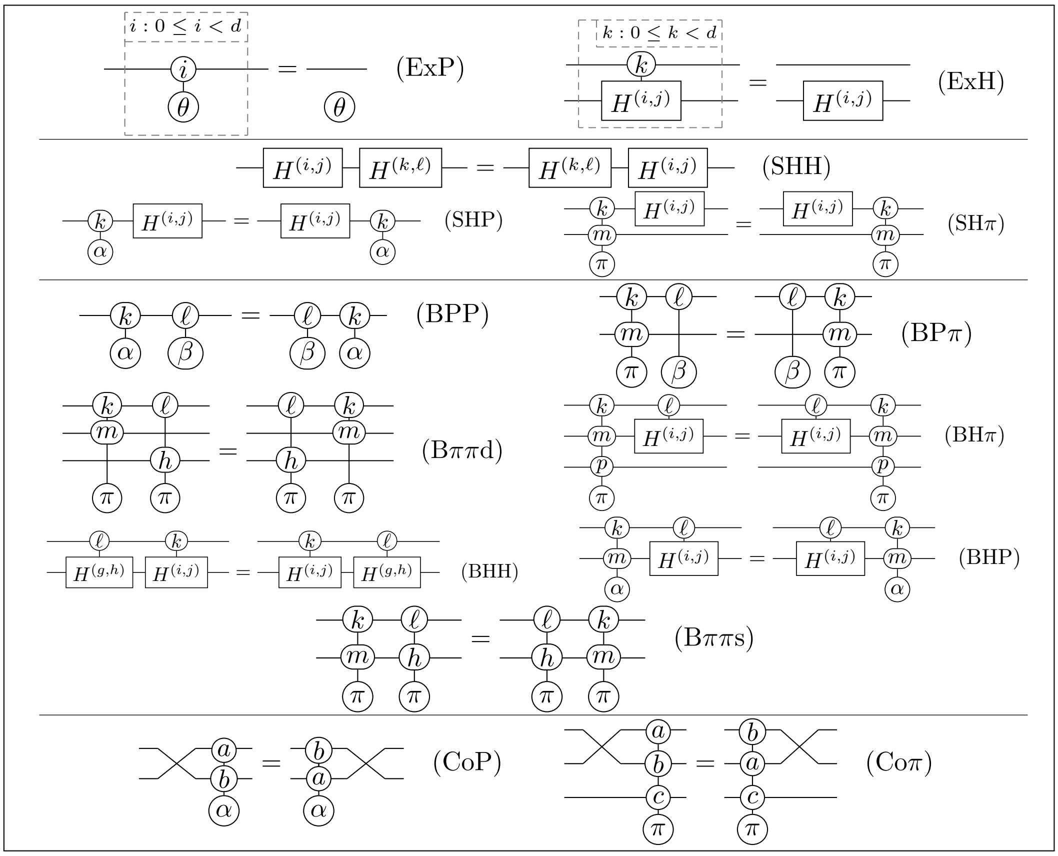


Figure 3 Control-support axiom schemata for QC_d**.** The prefixes Ex, S, B, and Co mark totalisation, support-locality, branch-commutation, and compatibility rules. The support side conditions are i, j, k pairwise distinct, and i, j, k, ℓ pairwise distinct for (SHH); the branch-commutation rules assume $k \neq \ell$.

The next four lemmas are sanity checks: if one temporarily admitted the relevant global control law, or the stronger semantic support principle, the displayed local schemata of

Figure 3 would follow. Section 3.3 proves the converse direction needed for completeness.

▶ **Lemma 3.3.** *If distinct-control commutativity from Definition 2.2 were admitted as an admissible rewrite in* $\mathbf{CQC}_d$*, then the schemata* (BPP)*,* (BPπ)*,* (B$\pi\pi$d)*,* (BHπ)*,* (BHH)*,* (BHP)*, and* (B$\pi\pi$s) *would be derivable.*

▶ **Lemma 3.4.** *If* $\mathbf{CQC}_d$ *were assumed to be support-sensitive, with the displayed generators carrying their semantic supports, then the branch-commutation schemata listed above would be derivable. Moreover,* (SHH)*,* (SHP)*, and* (SHπ) *would also be derivable.*

▶ **Lemma 3.5.** *If mixed-control compatibility from Definition 2.2 were admitted as an admissible rewrite in* $\mathbf{CQC}_d$*, then the schemata* (CoP) *and* (Coπ) *would be derivable.*

▶ **Lemma 3.6.** *If exhaustivity from Definition 2.2 were admitted as an admissible rewrite in* $\mathbf{CQC}_d$*, then the totalisation schemata* (ExP) *and* (ExH) *would be derivable.*

Proof of Lemmas 3.3–3.6. The branch-commutation rules are direct instances of distinct-control commutativity, up to tensoring with identities and PROP coherence. Support-sensitivity gives the extra Hadamard commutations because the displayed supports are disjoint: $\{i, j\}$ from $\{k, \ell\}$ or $\{k\}$, and $\{i, j\} \times [d]$ from $\{k\} \times \{m\}$. Compatibility gives (CoP) and (Coπ) by applying it to a scalar phase and to a once-controlled π-phase. Finally, exhaustivity gives (ExP) and (ExH) by taking, respectively, the scalar phase and the derived two-level Hadamard as the controlled circuit. ◀

3.3 Derived control-algebra laws

Starting only from the finite schemata in Figures 2–3, the lifting argument recovers compatibility, distinct-control commutativity, and exhaustivity for arbitrary circuits. The technical step is the Separate normalisation of Appendix E.1.

▶ **Definition 3.7.** *Let* $\mathcal{G} = \{\textcircled{\theta},\ -\boxed{H^{(r,r+1)}}-,\ \mathtt{C}_k(\textcircled{\theta}),\ \mathtt{C}_k(\mathtt{C}_\ell(\textcircled{\pi}))\}$*, with the adjacent Hadamard indexed by* $0 \leq r < d-1$ *and the controls indexed by* $k, \ell \in [d]$*. A controlled* $\mathcal{G}$*-layer is an instance of one of these four shapes, placed in a structural context and equipped with an explicit surrounding control word.*

Given a circuit C, Separate serialises tensor, pushes controls through composites, and reduces controlled generators to controlled $\mathcal{G}$-layers.

▶ **Lemma 3.8.** *Every morphism of* $\mathbf{CQC}_d$ *is provably equal in* QC_d *to a possibly empty finite sequential product of controlled* $\mathcal{G}$*-layers in structural context.*

Proof. This is proved by the normalisation procedure Separate in Appendix E.1. It expands the derived one-qudit notation using Appendix B, reduces controlled adjacent Hadamards and higher controlled phases using Appendix D, and terminates by the lexicographic measure given there. When the input is an identity, Separate returns the empty product at the same ambient arity, i.e. the PROP unit. ◀

The normal form above reduces compatibility and distinct-control commutativity to controlled $\mathcal{G}$-layers; exhaustivity is then propagated structurally from the primitive totalisation cells.

▶ **Theorem 3.9.** *In the equational theory* QC_d *on* $\mathbf{CQC}_d$*, the family* $(\mathtt{C}_k)_{k=0}^{d-1}$ *satisfies compatibility in the sense of Definition 2.2.*

Proof. Appendix E proves the statement by applying SEPARATE to the two nested-control circuits. Layers touching at most one distinguished control pass through the outer swap by structural naturality; the remaining layers are exactly the doubly controlled π-phase cases covered by (CoP). ◀

▶ **Theorem 3.10.** *In the equational theory* QC_d *on* $\mathbf{CQC}_d$*, the family* $(\mathtt{C}_k)_{k=0}^{d-1}$ *satisfies commutativity of distinct branches in the sense of Definition 2.2.*

Proof. Appendix F proves the statement by reducing both circuits to finite products of controlled $\mathcal{G}$-layers and commuting the layers pairwise using the finite table of derived commutation cells. ◀

▶ **Theorem 3.11.** *In the equational theory* QC_d *on* $\mathbf{CQC}_d$*, the family* $(\mathtt{C}_k)_{k=0}^{d-1}$ *satisfies exhaustivity in the sense of Definition 2.2.*

Proof. Appendix G proves the statement after Theorems 3.9 and 3.10 are available. The primitive totalisation cells give the phase and adjacent-Hadamard cases, and the closure argument propagates the equation through sequential composition, tensor product, and added controls. ◀

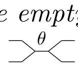
From this point on, these three laws are used as derived rewrite principles inside QC_d: products of branch-controlled layers may be reordered and collapsed without reopening the normalisation proof.

4 Embedding Qudit Circuits into LOPP Circuits

Single-photon linear optics gives a complete equational calculus for finite unitary matrices generated by phases, beam splitters, and swaps [5, 18]. The transfer uses that calculus as a proof engine, then returns to the circuit language. We place the d^n computational basis states of an n-qudit register in reflected Gray order, so adjacent optical generators decode to basis-controlled phases or adjacent two-level qudit gates. The only structural mismatch is tensor: $\mathbf{CQC}_d$ uses Hilbert-space tensor product, while single-photon **LOPP** uses block-diagonal direct sum. Thus the translations are contextual maps on a fixed d^n-mode space: they preserve sequential composition, serialise tensor, and are tied together by an encode-decode retraction in QC_d. The proof obligations below follow this shape: encoding preserves denotations after the Gray-basis change, decoded optical rewrites can be mimicked in QC_d, and decoding an encoding retracts to the original qudit circuit.

4.1 Linear optics and Gray-ordered semantics

The decoding map later inspects raw optical subterms, including the consecutive mode interval a subcircuit occupies. The quotient **LOPP** is the language to which the imported completeness theorem applies.

▶ **Definition 4.1.** *Let* $\mathbf{LOPP}^{\mathrm{raw}}$ *be the free endomorphism PROP generated by the empty diagram, identity wires, swap, one-mode phases* $-\boxed{\theta}-$*, and two-mode beam splitters* *, with* $\mathbf{LOPP}^{\mathrm{raw}}(m,n) = \varnothing$ *for* $m \neq n$*. Let* **LOPP** *be the quotient of* $\mathbf{LOPP}^{\mathrm{raw}}$ *by strict PROP coherence.*

▶ **Definition 4.2.** *The single-photon interpretation* $[\![-]\!]_{\mathrm{sp}} : \mathbf{LOPP}^{\mathrm{raw}}(m,m) \to U(m)$ *is defined by*

$$[\![C_2 \circ C_1]\!]_{\mathrm{sp}} = [\![C_2]\!]_{\mathrm{sp}}\,[\![C_1]\!]_{\mathrm{sp}}\,, \quad [\![C_1 \otimes C_2]\!]_{\mathrm{sp}} = [\![C_1]\!]_{\mathrm{sp}} \oplus [\![C_2]\!]_{\mathrm{sp}}\,,$$

$$\left[\!\!\left[-\boxed{\theta}- \right]\!\!\right]_{\mathrm{sp}} = (e^{i\theta}), \qquad \left[\!\!\left[\overset{\theta}{\rtimes\!\ltimes} \right]\!\!\right]_{\mathrm{sp}} = \begin{pmatrix} \cos\theta & i\sin\theta \\ i\sin\theta & \cos\theta \end{pmatrix}.$$

The structural generators are interpreted by $[\![\,\boxed{}\,]\!]_{\mathrm{sp}} = I_0$, $[\![-]\!]_{\mathrm{sp}} = I_1$, *and* $[\![\times]\!]_{\mathrm{sp}} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$. *More generally, identities and swaps use the corresponding identity and permutation matrices. This interpretation respects strict PROP coherence and therefore descends to* **LOPP**.

The non-structural equations of the calculus are listed in Figure 10 of Appendix K; we write LOPP $\vdash C_1 = C_2$ for the congruence generated by those equations together with strict PROP coherence.

▶ **Theorem 4.3** ([5, 18]). *The equational theory* LOPP *is sound and complete for the single-photon semantics: for* $C_1, C_2 : m \to m$, $[\![C_1]\!]_{\mathrm{sp}} = [\![C_2]\!]_{\mathrm{sp}}$ *if and only if* LOPP $\vdash C_1 = C_2$.

To compare d^n-mode optical circuits with n-qudit circuits, we identify optical modes with computational basis states using a reflected d-ary Gray code.

▶ **Definition 4.4.** *The reflected Gray order* $G^d_n : \{0, \ldots, d^n - 1\} \to [d]^n$ *is defined by* $G^d_0(0) = \epsilon$ *and, writing* $k = qd^{n-1} + r$, *by* $G^d_n(k) = q \cdot G^d_{n-1}(r)$ *if* q *is even and* $G^d_n(k) = q \cdot G^d_{n-1}(d^{n-1} - 1 - r)$ *if* q *is odd.*

▶ **Lemma 4.5.** *Consecutive Gray words differ in exactly one digit, and that digit changes by* ± 1. *In particular, two adjacent optical modes always correspond to an adjacent two-level qudit transition, possibly under controls on the remaining digits.*

Proof. Induct on n, the case $n = 0$ being empty. Inside a leading block $qd^{n-1}, \ldots, (q+1)d^{n-1} - 1$, the suffix is listed either in G^d_{n-1}-order or in reverse order, so the induction hypothesis gives a single adjacent change in the suffix. At the boundary between the q- and $(q+1)$-blocks, reflection makes the terminal suffix of one block equal to the initial suffix of the next; hence only the leading digit changes, from q to $q + 1$. Thus every consecutive pair differs in one digit, by ± 1. ◀

For $d = 3, n = 2$, the order is $00, 01, 02, 12, 11, 10, 20, 21, 22$. The edges $02 \to 12$ and $12 \to 11$ show the two cases: a fixed suffix or prefix guards the move, and odd blocks reverse the orientation, while the changed digit stays adjacent. Let $\mathfrak{G}_{n,d} : \mathbb{C}^{d^n} \to (\mathbb{C}^d)^{\otimes n}$ be the permutation unitary determined by $\mathfrak{G}_{n,d}(|t\rangle) = \left|G^d_n(t)\right\rangle$. The native single-photon semantics $[\![-]\!]_{\mathrm{sp}}$ is the one used by Theorem 4.3; for comparison with qudit circuits we conjugate it by this fixed Gray-code basis change.

▶ **Definition 4.6.** *For* $C : d^n \to d^n$ *in* **LOPP**, *define its Gray-ordered qudit semantics by* $[\![C]\!]_{\mathrm{LOPP}} := \mathfrak{G}_{n,d} \circ [\![C]\!]_{\mathrm{sp}} \circ \mathfrak{G}^{-1}_{n,d}$.

4.2 Encoding and decoding

Encoding and decoding are defined before quotienting and carry context parameters: encoding records where a qudit circuit sits inside a larger register, and decoding records where a raw optical subcircuit sits in the mode list. The following lemmas state where those raw choices become invariant under the relevant quotients.

Three optical gadget families are used by the encoding. The block permutation $\sigma^d_{a,n,b} \in \mathbf{LOPP}^{\mathrm{raw}}(d^{a+n+b}, d^{a+n+b})$ reorders Gray-coded modes so that, after decoding, it becomes the qudit wire permutation $\mathrm{id}_a \otimes \sigma_{n,b}$. The lift $\mathrm{Lift}^p_d(C)$ copies a d^n-mode circuit C through p additional Gray digits, mirroring the copy on odd reflected blocks. The selected lift $\mathrm{Lift}^p_{d,j}(C)$ does the same on the branch where the added distinguished digit has value j, with identities on other branches. Their raw constructions are in Appendices I and L; their decoding interface is Lemma 4.13.

▶ **Definition 4.7.** *For a raw qudit circuit $C : n \to n$, the contextual encoding $\mathrm{E}^a_b(C) \in \mathbf{LOPP}^{\mathrm{raw}}(d^{a+n+b}, d^{a+n+b})$ represents C on the middle n qudits of an $(a+n+b)$-qudit register. The recursive clauses are*

$$
\begin{array}{rcl}
\mathrm{E}^a_b(C_2 \circ C_1) & := & \mathrm{E}^a_b(C_2) \circ \mathrm{E}^a_b(C_1), \\
\mathrm{E}^a_b(C_1 \otimes C_2) & := & \mathrm{E}^{a+n_1}_b(C_2) \circ \mathrm{E}^a_{b+n_2}(C_1) \quad (C_i : n_i \to n_i), \\
\mathrm{E}^a_b(\mathrm{id}_n) & := & \mathrm{id}_{d^{a+n+b}}, \\
\mathrm{E}^a_b(\textcircled{\theta}) & := & (-\boxed{\theta}-)^{\otimes d^{a+b}}, \\
\mathrm{E}^a_b(-\boxed{H^{(r,r+1)}}-) & := & \sigma^d_{a,b,1} \circ \mathrm{Lift}^{a+b}_d\big(B^{(i,i+1)}_d\big) \circ \sigma^d_{a,1,b}, \\
\mathrm{E}^a_b(\times) & := & \sigma^d_{a,b,2} \circ \sigma^d_{a+b,1,1} \circ \sigma^d_{a,2,b}, \\
\mathrm{E}^a_b(\mathsf{C}_j(C)) & := & \sigma^d_{a,b,m+1} \circ \mathrm{Lift}^{a+b}_{d,j}(\mathrm{E}^0_0(C)) \circ \sigma^d_{a,m+1,b} \quad (C : m \to m).
\end{array}
$$

Here $B^{(i,i+1)}_d$ denotes the d-mode optical network implementing the adjacent Hadamard, fixed in Definition I.1 of Appendix I. We write $\mathrm{E}(C) := \mathrm{E}^0_0(C)$.

For example, if $g, h : 1 \to 1$, then $\mathrm{E}^0_0(g \otimes h) = \mathrm{E}^1_0(h) \circ \mathrm{E}^0_1(g)$ on the same d^2 modes: tensor is serialised over context branches.

▶ **Lemma 4.8.** *For all $a, b \geq 0$, if raw qudit circuits $C, C' : n \to n$ satisfy $C \equiv C'$ in the structural congruence defining $\mathbf{CQC}_d$, then $\mathrm{LOPP} \vdash \mathrm{E}^a_b(C) = \mathrm{E}^a_b(C')$.*

Proof. This is Lemma J.9 in Appendix J. The appendix first proves the raw semantic invariant for the contextual encoding and then checks each generator of the structural congruence, including the control-functor equations. ◀

▶ **Lemma 4.9.** *The contextual encoding preserves semantics: for $C : n \to n$ in $\mathbf{CQC}_d$, $[\![\mathrm{E}^a_b(C)]\!]_{\mathrm{LOPP}} = I_{d^a} \otimes [\![C]\!] \otimes I_{d^b}$. In particular, $[\![\mathrm{E}(C)]\!]_{\mathrm{LOPP}} = [\![C]\!]$.*

Proof. The proof is by induction on C. Gray lifts duplicate the same local action on every context branch, selected lifts restrict that action to one value of the distinguished digit, and block permutations move the relevant Gray blocks to the position of the qudit wire on which the generator acts. Appendix I gives the full lift constructions, and Appendix L gives the block-permutation construction. ◀

Decoding goes in the opposite direction, reading a raw **LOPP** subcircuit in a consecutive mode interval $t, \ldots, t+\ell-1$ inside a d^n-mode system. When $G^d_n(t) = u\,v\,w$ and $G^d_n(t+1) = u\,(v+\varepsilon)\,w$, with $\varepsilon \in \{-1, +1\}$, write $\Lambda^u_w(g) := \sigma_{|u|,|w|,1} \circ \mathsf{C}_{uw}(g) \circ \sigma_{|u|,1,|w|}$ for the one-qudit gate g placed on the changed digit and controlled by the surrounding basis word.

▶ **Definition 4.10.** *For $0 \leq t$ and $t + \ell \leq d^n$, the contextual decoding $\mathrm{D}^t_n : \mathbf{LOPP}^{\mathrm{raw}}(\ell, \ell) \to \mathbf{CQC}^{\mathrm{raw}}_d(n, n)$ reads a raw optical subcircuit as occupying global modes $t, \ldots, t+\ell-1$. Its* *recursive clauses are*

$$
\begin{aligned}
\mathrm{D}^t_n(C_2 \circ C_1) &:= \mathrm{D}^t_n(C_2) \circ \mathrm{D}^t_n(C_1),\\
\mathrm{D}^t_n(C_1 \otimes C_2) &:= \mathrm{D}^{t+\ell_1}_n(C_2) \circ \mathrm{D}^t_n(C_1) \quad (C_1 : \ell_1 \to \ell_1),\\
\mathrm{D}^t_n(\,[\;]\,) &:= \mathrm{id}_n, \qquad \mathrm{D}^t_n(\text{—}) := \mathrm{id}_n,\\
\mathrm{D}^t_n(-\boxed{\theta}-) &:= \mathtt{C}_{G^d_n(t)}(\textcircled{\theta}),\\
\mathrm{D}^t_n(\times) &:= \Lambda^u_w(-\boxed{X^{(v,v+\varepsilon)}}-),\\
\mathrm{D}^t_n(\rangle\!\overset{\theta}{\times}\!\langle) &:= \Lambda^u_w(-\boxed{R_X^{(v,v+\varepsilon)}(\theta)}-).
\end{aligned}
$$

In the last two clauses, u, v, w, ε are determined by Lemma 4.5 for the consecutive modes $t, t+1$. The figures $X^{(v,v+\varepsilon)}$ and $R_x^{(v,v+\varepsilon)}(\theta)$ use $\varepsilon = +1$ for the adjacent transition $v \leftrightarrow v+1$ and $\varepsilon = -1$ for $v \leftrightarrow v-1$; the displayed order records the direction of the Gray edge. We write $\mathrm{D}(C) := \mathrm{D}^0_n(C)$ for a circuit on all d^n modes.

For $d = 3, n = 2$, mode 4 has Gray word 11, so its phase decodes to a phase controlled on both qutrits being 1; beam splitters on modes $2, 3$ and $3, 4$ decode to adjacent transitions along $02 \leftrightarrow 12$ and $12 \leftrightarrow 11$, with the unchanged digit as control.

▶ **Lemma 4.11.** *For every n, t, the decoding of a one-mode phase is supported on the single basis word $G^d_n(t)$, and the decoding of a swap or beam splitter on consecutive modes $t, t+1$ is supported on the two Gray-neighbouring basis words $G^d_n(t)$ and $G^d_n(t+1)$.*

Proof. The one-mode phase clause of Definition 4.10 is $\mathtt{C}_{G^d_n(t)}(\textcircled{\theta})$, so its support is exactly the branch word $G^d_n(t)$. For swaps and beam splitters, Lemma 4.5 writes the adjacent Gray words as $u\,v\,w$ and $u\,(v{+}\varepsilon)\,w$. The decoding clause is then $\Lambda^u_w(g) = \sigma_{|u|,|w|,1} \circ \mathtt{C}_{uw}(g) \circ \sigma_{|u|,1,|w|}$, where g is the adjacent X- or R_x-gate on the changed digit. Hence the only non-identity action is on those two basis words. ◀

▶ **Lemma 4.12.** *Let $P[-]$ be a raw* **LOPP** *context whose hole occupies a consecutive mode interval $t, \ldots, t+\ell-1$ inside a d^n-mode circuit. If $C, C' : \ell \to \ell$ are equal by strict PROP coherence in* $\mathbf{LOPP}^{\mathrm{raw}}$*, then $\mathrm{QC}_d \vdash \mathrm{D}^0_n(P[C]) = \mathrm{D}^0_n(P[C'])$. In particular, $\mathrm{QC}_d \vdash \mathrm{D}(C) = \mathrm{D}(C')$ for coherent d^n-mode circuits C, C'.*

Proof. This is Lemma K.4 in Appendix K. Units and associativity follow from the recursive definition of D; interchange uses disjoint Gray supports and the derived distinct-control commutativity of Theorem 3.10. ◀

▶ **Lemma 4.13.** *The optical gadgets used in the encoding decode as follows. For every $n, p \geq 0$, every raw $C : d^n \to d^n$ in* **LOPP**, *and every $j \in [d]$, $\mathrm{QC}_d \vdash \mathrm{D}^0_{n+p}(\mathrm{Lift}^p_d(C)) = \mathrm{id}_p \otimes \mathrm{D}^0_n(C)$ and $\mathrm{QC}_d \vdash \mathrm{D}^0_{n+1+p}(\mathrm{Lift}^p_{d,j}(C)) = \mathrm{id}_p \otimes \mathtt{C}_j(\mathrm{D}^0_n(C))$. For all $a, n, b \geq 0$, $\mathrm{QC}_d \vdash \mathrm{D}^0_{a+n+b}(\sigma^d_{a,n,b}) = \mathrm{id}_a \otimes \sigma_{n,b}$.*

Proof. The lift equations are Lemmas J.6 and J.7 in Appendix J; the block-permutation equation is Lemma L.10 in Appendix L. ◀

▶ **Theorem 4.14.** *If $\mathrm{LOPP} \vdash C_1 = C_2$ for $C_1, C_2 : d^n \to d^n$, then $\mathrm{QC}_d \vdash \mathrm{D}(C_1) = \mathrm{D}(C_2)$.*

Proof. Induct on the LOPP-derivation. Strict PROP steps use Lemma 4.12. The non-structural axioms decode to the matching controlled phase, swap, and Euler rules of QC_d. The three-beam-splitter axiom is the only case split: same-digit Gray moves give (3Rx), while two-coordinate corners give the derived rule (3CRx). Appendix K gives the axiom-by-axiom derivations. ◀

4.3 Completeness via transfer

▶ **Theorem 4.15.** *For all $n, a, b \geq 0$ and all $C : n \to n$ in* $\mathbf{CQC}_d$, $\mathrm{QC}_d \vdash \mathrm{D}^0_{a+n+b}(\mathrm{E}^a_b(C)) = \mathrm{id}_a \otimes C \otimes \mathrm{id}_b$.

This internal retraction is the reason the non-monoidal translation is usable: in a fixed surrounding register, encoding and then decoding returns the original circuit with only identity padding.

Proof. Choose a raw representative by Lemma 4.8 and argue by structural induction; Appendix J gives the full case analysis. Empty diagrams and identities are immediate, scalar phases collapse by exhaustivity, adjacent Hadamards use Lemma 4.13, and wire symmetries use the block-permutation equation. Composition and tensor follow from the recursive clauses. For $\mathtt{C}_j(C')$, decoding the selected lift gives $\mathrm{id}_a \otimes \mathtt{C}_j(\mathrm{D}(\mathrm{E}(C'))) \otimes \mathrm{id}_b$, which the induction hypothesis reduces to $\mathrm{id}_a \otimes \mathtt{C}_j(C') \otimes \mathrm{id}_b$. ◀

▶ **Corollary 4.16.** *For every $C : n \to n$ in* $\mathbf{CQC}_d$, $\mathrm{QC}_d \vdash \mathrm{D}(\mathrm{E}(C)) = C$.

Proof. Take $a = b = 0$ in Theorem 4.15; the identity padding is then the monoidal unit on both sides. ◀

Completeness is now the encode-mimic-decode round trip.

▶ **Theorem 4.17.** *For every $d \geq 2$, QC_d is sound and complete for exact unitary qudit semantics: for all $C_1, C_2 : n \to n$, $[\![C_1]\!] = [\![C_2]\!]$ if and only if $\mathrm{QC}_d \vdash C_1 = C_2$.*

Proof. Soundness is checked axiom by axiom against the displayed unitary semantics in Appendix M: phase arithmetic, 2π-periodicity, and the Euler/rotation rules are the corresponding one- or two-level matrix identities, while the support, compatibility, commutation, and totalisation cells are direct block-diagonal control calculations. If $[\![C_1]\!] = [\![C_2]\!]$, then Lemma 4.9 gives equal single-photon semantics for the encodings after Gray conjugation, so $\mathrm{LOPP} \vdash \mathrm{E}(C_1) = \mathrm{E}(C_2)$. Theorem 4.14 and Corollary 4.16 yield $\mathrm{QC}_d \vdash C_1 = \mathrm{D}(\mathrm{E}(C_1)) = \mathrm{D}(\mathrm{E}(C_2)) = C_2$. ◀

5 Conclusion

We have given a finite schematic, bounded-arity equational theory for exact unitary qudit circuits, uniform in the dimension and including global phases. Primitive value-control keeps the schemata within three wires, while compatibility, branch commutativity, and exhaustivity are derived internally. The Gray-ordered **LOPP** transfer, via mimicking and encode-decode retraction, gives a circuit-preserving basis for certified qudit rewriting: reflected Gray order, selected lifts, and derived control algebra keep optical locality compatible with native tensor structure.

The result gives a circuit-level foundation for higher-dimensional compilation, where value controls and adjacent two-level operations can be reasoned about directly. Natural next steps include orienting rule fragments as rewrite systems and specialising the presentation to fault-tolerant or hardware-native qudit gate sets.

References

1 Miriam Backens and Aleks Kissinger. ZH: A complete graphical calculus for quantum computations involving classical non-linearity. *Electronic Proceedings in Theoretical Computer Science*, 287:23–42, 2019. Preprint: arXiv:1805.02175. doi:10.4204/EPTCS.287.2.

2 Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, 1995. doi:10.1103/PhysRevA.52.3457.

3 Jean-Luc Brylinski and Ranee Brylinski. Universal quantum gates. In *Mathematics of Quantum Computation*, pages 101–116. Chapman & Hall/CRC, Boca Raton, FL, USA, 2002. doi:10.1201/9781420035377-5.

4 Alexandre Clément, Noé Delorme, and Simon Perdrix. Minimal equational theories for quantum circuits. In *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2024)*, pages 27:1–27:14, Tallinn, Estonia, July 2024. IEEE Computer Society Press. See also arXiv:2311.07476. doi:10.1145/3661814.3662088.

5 Alexandre Clément, Nicolas Heurtel, Shane Mansfield, Simon Perdrix, and Benoît Valiron. LOv-calculus: A graphical language for linear optical quantum circuits. In *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*, volume 241 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 35:1–35:16, Dagstuhl, Germany, 2022. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2022.35.

6 Alexandre Clément, Nicolas Heurtel, Shane Mansfield, Simon Perdrix, and Benoît Valiron. A complete equational theory for quantum circuits. In *38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2023), Boston, MA, USA, June 26–29, 2023*, pages 1–13, Boston, MA, USA, 2023. IEEE. doi:10.1109/LICS56636.2023.10175801.

7 Alexandre Clément, Simon Perdrix, Noé Delorme, and Renaud Vilmart. Quantum circuit completeness: Extensions and simplifications. In *Computer Science Logic (CSL 2024)*, volume 288 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:23, Dagstuhl, Germany, 2024. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CSL.2024.20.

8 William R. Clements, Peter C. Humphreys, Benjamin J. Metcalf, W. Steven Kolthammer, and Ian A. Walmsley. Optimal design for universal multiport interferometers. *Optica*, 3(12):1460–1465, 2016. doi:10.1364/OPTICA.3.001460.

9 Bob Coecke and Ross Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13:043016, 2011. doi:10.1088/1367-2630/13/4/043016.

10 Niel de Beaudrap, Aleks Kissinger, and John van de Wetering. Circuit extraction for zx-diagrams can be #p-hard. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 119:1–119:19, Dagstuhl, Germany, 2022. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2022.119.

11 Marc de Visme and Renaud Vilmart. Minimality in finite-dimensional ZW-calculi. In *Computer Science Logic (CSL 2025)*, volume 326 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:22, Dagstuhl, Germany, 2025. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CSL.2025.49.

12 Noé Delorme and Simon Perdrix. Diagrammatic reasoning with control as a constructor, applications to quantum circuits. In *Proceedings of the International Conference on Foundations of Software Science and Computation Structures (FOSSACS 2026)*, Lecture Notes in Computer Science. Springer, 2026. Accepted for publication. arXiv:2508.21756.

13 David Deutsch. Quantum computational networks. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 425(1868):73–90, September 1989. doi:10.1098/rspa.1989.0099.

14 Yao-Min Di and Hai-Rui Wei. Synthesis of multivalued quantum logic circuits by elementary gates. *Physical Review A*, 87(1):012325, 2013. doi:10.1103/PhysRevA.87.012325.

15 David Eberly. Euler angle formulas, 2020. URL: https://www.geometrictools.com/Documentation/EulerAngles.pdf.

16 Pranav Gokhale, Jonathan M. Baker, Casey Duckering, Natalie C. Brown, Kenneth R. Brown, and Frederic T. Chong. Asymptotic improvements to quantum circuits via qutrits, 2019. arXiv:1905.10481, doi:10.1145/3307650.3322253.

17 Chris Heunen, Robin Kaarsgaard, and Louis Lemonnier. One rig to control them all, 2025. arXiv:2510.05032.

18 Nicolas Heurtel. A complete graphical language for linear optical circuits with finite-photon-number sources and detectors. In *Computer Science Logic (CSL 2025)*, volume 326 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 38:1–38:23, Dagstuhl, Germany, 2025. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CSL.2025.38.

19 Pavel Hrmo, Benjamin Wilhelm, Lukas Gerster, Martin W. van Mourik, Marcus Huber, Rainer Blatt, Philipp Schindler, Thomas Monz, and Martin Ringbauer. Native qudit entanglement in a trapped ion quantum processor, 2022. arXiv:2206.04104, doi:10.1038/s41467-023-37375-2.

20 Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Completeness of the zx-calculus. *Logical Methods in Computer Science*, 16(2):11:1–11:72, June 2020. doi:10.23638/LMCS-16(2:11)2020.

21 Sarah Meng Li, Michele Mosca, Neil J. Ross, John van de Wetering, and Yuming Zhao. A complete and natural rule set for Multi-Qutrit Clifford circuits. *Electronic Proceedings in Theoretical Computer Science*, 426:23–78, August 2025. doi:10.4204/EPTCS.426.2.

22 Robert Lin. An algebraic framework for multi-qudit computations with generalized clifford algebras, 2022. arXiv:2103.15324.

23 Robert Lin. A graphical calculus for quantum computing with multiple qudits using generalized clifford algebras. *Quantum*, 9:1913, November 2025. doi:10.22331/q-2025-11-17-1913.

24 Andrew Litteken, Lennart Maximilian Seifert, Jason D. Chadwick, Natalia Nottingham, Tanay Roy, Ziqian Li, David Schuster, Frederic T. Chong, and Jonathan M. Baker. Dancing the quantum waltz: Compiling three-qubit gates on four level architectures, 2024. arXiv:2303.14069, doi:10.1145/3579371.3589106.

25 Saunders Mac Lane. Categorical algebra. *Bulletin of the American Mathematical Society*, 71(1):40–106, 1965. doi:10.1090/S0002-9904-1965-11234-4.

26 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 10th anniversary edition, 2010. See pp. 189–192.

27 Boldizsár Poór, Razin A. Shaikh, and Quanlong Wang. ZX-calculus is complete for finite-dimensional hilbert spaces. *Electronic Proceedings in Theoretical Computer Science*, 426:127–158, August 2025. doi:10.4204/EPTCS.426.5.

28 D. J. Rowe, B. C. Sanders, and H. de Guise. Representations of the weyl group and wigner functions for su(3). *Journal of Mathematical Physics*, 40(7):3604–3615, 1999. doi:10.1063/1.532911.

29 Patrick Roy, John van de Wetering, and Lia Yeh. The qudit zh-calculus: Generalised toffoli+hadamard and universality. *Electronic Proceedings in Theoretical Computer Science*, 384:142–170, 2023. Preprint: arXiv:2307.10095. doi:10.4204/EPTCS.384.9.

30 John van de Wetering and Lia Yeh. Building qutrit diagonal gates from phase gadgets. *Electronic Proceedings in Theoretical Computer Science*, 394:46–65, 2023. Preprint: arXiv:2204.13681. doi:10.4204/EPTCS.394.4.

31 Quanlong Wang. Qufinite ZX-calculus: a unified framework of qudit ZX-calculi, 2022. arXiv:2104.06429.

32 Quanlong Wang, Boldizsár Poór, and Razin A. Shaikh. Completeness of qufinite zxw calculus, a graphical language for finite-dimensional quantum theory, 2024. arXiv:2309.13014.

33 Yuchen Wang, Zixuan Hu, Barry C. Sanders, and Sabre Kais. Qudits and high-dimensional quantum computing, 2020. arXiv:2008.00959, doi:10.3389/fphy.2020.589504.

A Proof dependency guide

The proof dependencies are acyclic in the following order. This appendix is only a reference point for checking where each later argument gets its inputs.

The raw circuit syntax and the structural control-functor laws are fixed in Section 2. No compatibility, commutativity, or exhaustivity principle for distinct controls is assumed there.

The local schemata of Figures 2–3 generate QC_d. Appendix B expands the derived one-qudit notation, Appendix C fixes the angle conventions for the rotation schemata, and Appendix M records the semantic soundness checks for the displayed axiom schemata.

Appendix D proves the first support catalogue, including the layer-expansion and generator-commutation rules used by normalisation and commutativity. Appendix E.1 then proves the controlled-layer normalisation used in Lemma 3.8 and Corollary E.3, reducing arbitrary controlled circuits to the finite non-identity layer family $\mathcal{G}$ named in Section 3.3.

The control algebra is derived from those normalised layers: Appendix E proves compatibility, Appendix F proves commutativity of differently controlled circuits, and Appendix G proves exhaustivity using those two principles together with the totalisation support rules.

Once the control algebra is available, Appendix H proves the post-control-algebra identities, including the mixed controlled three-rotation rule used in the LOPP mimicking step.

Appendices I, L, and J establish the Gray-code interface for encoding and decoding. Appendix L proves the block-permutation decoding interface used both by the retraction proof and by structural congruence in the mimicking proof. At this point the derived control algebra is available, so products over Gray branches can be reordered and collapsed by commutativity and exhaustivity.

Appendix K checks that every LOPP axiom is preserved by decoding. The mixed three-mode optical case uses the mixed controlled three-rotation rule derived in Appendix H.

Section 4 combines the pieces: encoding preserves semantics, LOPP completeness gives an optical derivation, Theorem 4.14 decodes it into QC_d, and Theorem 4.15 removes the encode-decode round trip.

B Derived notation for the axiom schemata

The axiom figures use several derived one-qudit circuits as notation. The definitions below pin down those abbreviations inside $\mathbf{CQC}_d$; no generator is added.

▶ **Definition B.1.** *Write $H^{(r,r+1)}$ for the adjacent Hadamard generator $-\boxed{H^{(r,r+1)}}-$. For each adjacent pair $(r, r+1)$, the adjacent swap is $X^{(r,r+1)} := H^{(r,r+1)} \circ \mathsf{C}_{r+1}(\textcircled{\pi}) \circ H^{(r,r+1)}$.*

▶ **Definition B.2.** *Set $X^{(i,i)} := \mathrm{id}_1$. For the adjacent-walk conjugator, set $S_{r,r} := \mathrm{id}_1$ and, if $i < j$, $S_{i,j} := \prod_{k=0}^{j-i-1} X^{(i+k,i+k+1)}$, where $\prod_{k=0}^{m} f_k := f_m \circ \cdots \circ f_0$, and define $X^{(i,j)} := S_{i+1,j} \circ X^{(i,i+1)} \circ S_{i+1,j}^{-1}$. For $i > j$, set $X^{(i,j)} := X^{(j,i)}$.*

▶ **Definition B.3.** *The derived Hadamards are obtained by sliding the relevant levels into an adjacent position. Set $H^{(i,i)} := \mathrm{id}_1$; if $i < j$, set $H^{(i,j)} := X^{(j,i+1)} \circ H^{(i,i+1)} \circ X^{(j,i+1)}$; and if $i > j$, set $H^{(i,j)} := X^{(j,i)} \circ H^{(j,i)} \circ X^{(j,i)}$. For distinct $i, j \in [d]$, define $R_x^{(i,j)}(\theta) := H^{(i,j)} \circ \mathsf{C}_i(\textcircled{\theta}) \circ \mathsf{C}_j(\textcircled{-\theta}) \circ H^{(i,j)}$.*

Under $[\![-]\!]$, $X^{(i,j)}$ swaps $|i\rangle$ and $|j\rangle$, $H^{(i,j)}$ acts as the usual Hadamard on $\mathrm{span}\{|i\rangle, |j\rangle\}$, and $R_x^{(i,j)}(\theta)$ acts on that same two-dimensional subspace as $\begin{pmatrix} \cos\theta & i\sin\theta \\ i\sin\theta & \cos\theta \end{pmatrix}$, while all three act as the identity on the orthogonal complement.

▶ **Definition B.4.** *The product frames used in the axiom figures are also meta-notation. A frame labelled $\vec{k} : \varphi(\vec{k})$, where $\vec{k} = (k_1, \dots, k_r)$, denotes the sequential composite of the enclosed circuit over all tuples $\vec{k} \in [d]^r$ satisfying φ, in lexicographic increasing order. Equivalently, k_1 is the outer loop variable and k_r is the inner loop variable. For a single variable this is just the increasing order of that index; if the predicate is omitted, the default range is $0 \leq k < d$.*

C Angle relations for rotation axioms

The angle-bearing schemata (EH), (3Rx), and the derived mixed rule (3CRx) need explicit right-hand-side parameter relations. This appendix fixes those *meta-level parameter relations*: the left-hand side carries freely chosen real parameters, while the right-hand side is chosen from the stated admissible solutions. Away from the usual Euler degeneracies the solution is deterministic; at degenerate points we record a canonical choice, but the axiom schema may be instantiated with any admissible degenerate representative.

In Delorme's one-qubit Euler parametrisations, the decomposition of a two-level unitary naturally comes with a block scalar factor (*global phase* in $U(2)$). In the present paper we apply such decompositions *inside a fixed two-dimensional subspace* $\mathrm{span}\{|i\rangle, |j\rangle\}$ of a d-level system. A phase factor that is "global" for the 2×2 block is *not* a scalar on the full d-dimensional space: rather, it is the operation that multiplies both $|i\rangle$ and $|j\rangle$ by the same phase while leaving the other basis levels unchanged. In our syntax this is realised by applying the same level-phase on both levels, i.e. by $\mathtt{C}_i(\alpha) \circ \mathtt{C}_j(\alpha)$. On the (i, j)-subspace this acts as $e^{\mathrm{i}\alpha} I_2$, hence it *commutes* with every gate supported on (i, j) (in particular with $H^{(i,j)}$ and with all level phases). For this reason, whenever the qubit formulas isolate a scalar phase, we are free to slide the corresponding two-level phase and absorb it into neighbouring level phases. This is the only reason our explicit angle formulas do not coincide syntactically with the qubit case; semantically they implement the same underlying $U(2)$ parametrisation on $\mathrm{span}\{|i\rangle, |j\rangle\}$.

Throughout, equalities between angles are understood modulo 2π. We write $\arg(z)$ for the principal argument of a non-zero complex number $z \in \mathbb{C}$, taking values in $(-\pi, \pi]$. When we restrict an angle to a range (e.g. $\beta \in [0, 2\pi)$), we always mean its canonical representative in that interval. We also use the standard two-argument arctangent $\mathrm{atan2}(y, x)$ with values in $(-\pi, \pi]$.

Euler-type relation (EH)

The axiom (EH) is a two-level Euler-type identity (applied in the paper on a subspace $\mathrm{span}\{|i\rangle, |j\rangle\}$ of a qudit). Syntactically, the left-hand side is parameterised by two real angles α_0, α_2, while the right-hand side uses four angles $\beta_0, \beta_1, \beta_2, \beta_3$. For every choice of (α_0, α_2), an admissible choice of $(\beta_0, \beta_1, \beta_2, \beta_3)$ in the prescribed ranges makes (EH) sound on the relevant two-dimensional subspace, and hence as a qudit equation acting as the identity outside that subspace. The formulas below choose a canonical deterministic representative; in degenerate cases the admissible family has the usual Euler one-parameter freedom.

We follow Delorme's explicit Euler parametrisation (stated for qubits) and use the same extraction function, with the "two-level global phase" convention explained above.

Given $\alpha_0, \alpha_2 \in \mathbb{R}$, define $z = -\sin\big((\alpha_0 + \alpha_2)/2\big) + i \cos\big((\alpha_0 - \alpha_2)/2\big)$ and $z' = \cos\big((\alpha_0 + \alpha_2)/2\big) - i \sin\big((\alpha_0 - \alpha_2)/2\big)$.

Define $\beta_0, \beta_1, \beta_2, \beta_3 \in [0, 2\pi)$ by cases. The degenerate cases $z = 0$ or $z' = 0$ are treated separately to avoid dividing by 0; the generic case uses the ratio z/z'.

- If $z' = 0$, set $\beta_0 = 2\arg(z)$, $\beta_1 = (\pi + \alpha_0 + \alpha_2)/2 - \arg(z)$, $\beta_2 = (\pi + \alpha_0 + \alpha_2)/2 - \arg(z)$, and $\beta_3 = 0$.
- If $z = 0$, set $\beta_0 = 2\arg(z')$, $\beta_1 = (\alpha_0 + \alpha_2)/2 - \arg(z')$, $\beta_2 = \pi + (\alpha_0 + \alpha_2)/2 - \arg(z')$, and $\beta_3 = 0$.
- Otherwise (when $z \neq 0$ and $z' \neq 0$), set $\beta_0 = \arg(z) + \arg(z')$, $\beta_1 = -\arg\big(i + |z/z'|\big) + (\pi + \alpha_0 + \alpha_2)/2 - \arg(z)$, $\beta_2 = \arg\big(i + |z/z'|\big) + (\pi + \alpha_0 + \alpha_2)/2 - \arg(z)$, and $\beta_3 = \arg(z) - \arg(z')$.

Soundness is the property used later; in the degenerate cases one endpoint phase remains free.

▶ **Lemma C.1.** *The canonical choices above make Equation* (EH) *sound on the relevant two-dimensional subspace. If $z = 0$ or $z' = 0$, then β_3 may be prescribed arbitrarily and the other three angles may be chosen so that the same two-level matrix is obtained.*

Proof. Write

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad P_j(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}, \qquad P_i(\theta) = \begin{pmatrix} e^{i\theta} & 0 \\ 0 & 1 \end{pmatrix}.$$

With the left-to-right convention of the diagrams, the left-hand side of (EH) is

$$HP_j(\alpha_2)HP_j(\alpha_0)H = \frac{e^{iS}}{\sqrt{2}}\begin{pmatrix} i\overline{z} & z' \\ \overline{z'} & iz \end{pmatrix}, \qquad S = \frac{\alpha_0 + \alpha_2}{2}.$$

The right-hand side is

$$P_j(\beta_3)HP_j(\beta_2)P_i(\beta_1)HP_j(\beta_0) = \begin{pmatrix} A & Be^{i\beta_0} \\ Be^{i\beta_3} & Ae^{i(\beta_0+\beta_3)} \end{pmatrix},$$

where $A = (e^{i\beta_1} + e^{i\beta_2})/2$ and $B = (e^{i\beta_1} - e^{i\beta_2})/2$.

Assume first that $z \neq 0$ and $z' \neq 0$. Put $\rho = |z/z'|$ and $t = \arg(i + \rho)$. Since $|z|^2 + |z'|^2 = 2$, the generic formula gives $A = \frac{ie^{iS}\overline{z}}{\sqrt{2}}$ and $B = \frac{e^{iS}|z'|e^{-i\arg z}}{\sqrt{2}}$. Together with $\beta_0 = \arg z + \arg z'$ and $\beta_3 = \arg z - \arg z'$, the four entries of the right-hand side become

$$\frac{e^{iS}}{\sqrt{2}}\begin{pmatrix} i\overline{z} & z' \\ \overline{z'} & iz \end{pmatrix},$$

which is the left-hand side.

If $z' = 0$, then $|z| = \sqrt{2}$. The canonical choice has $B = 0$, $A = e^{i((\pi+\alpha_0+\alpha_2)/2 - \arg z)} = \frac{ie^{iS}\overline{z}}{\sqrt{2}}$, and $e^{i\beta_0} = e^{2i\arg z}$, so the same diagonal matrix is obtained. If $z = 0$, then $|z'| = \sqrt{2}$. The canonical choice has $A = 0$, $B = e^{i(S - \arg z')}$, and $e^{i\beta_0} = e^{2i\arg z'}$, and again the two off-diagonal entries agree with the displayed left-hand-side matrix.

Finally let τ be any prescribed value for β_3, understood modulo 2π. When $z' = 0$, keep $\beta_1 = \beta_2 = (\pi + \alpha_0 + \alpha_2)/2 - \arg z$ and set $\beta_0 = 2\arg z - \tau$. When $z = 0$, set $\beta_1 = S - \arg z' - \tau$, $\beta_2 = \beta_1 + \pi$, and $\beta_0 = 2\arg z' + \tau$. After taking representatives in $[0, 2\pi)$, these choices give the same matrix as the canonical representative and have $\beta_3 = \tau$. ◀

Since every occurrence of (EH) in the main development is applied to a two-level gate supported on $\mathrm{span}\{|i\rangle, |j\rangle\}$ and acts as the identity on the other computational levels, Lemma C.1 applies systematically for all $d \geq 2$. The canonical formulas above are kept as a deterministic rewrite convention, while derivations may use any admissible degenerate representative.

Three-rotation identities (3Rx) and derived (3CRx)

The axiom (3Rx) and the derived mixed rule (3CRx) express the standard fact that the same real 3×3 rotation (hence an element of SO(3)) admits both a *ZXZ* and an *XZX* Euler decomposition. They are the qudit/circuit-side reflections of the "(E3)" principle used in the linear-optical setting: (3Rx) is the uncontrolled identity, and (3CRx) is its mixed controlled consequence. The parameter relations are identical in both cases.

Let $R_x(\theta)$ and $R_z(\theta)$ be the standard real rotation matrices about the x- and z-axes. The left-hand side angles $\gamma_1, \gamma_2, \gamma_3$ are the freely chosen parameters of (3Rx). Write $c_r := \cos(\gamma_r)$ and $s_r := \sin(\gamma_r)$. Define

$$R_{E3} := R_z(\gamma_1)\, R_x(\gamma_2)\, R_z(\gamma_3) = \begin{pmatrix} c_1c_3 - s_1c_2s_3 & -c_1s_3 - s_1c_2c_3 & s_1s_2 \\ s_1c_3 + c_1c_2s_3 & -s_1s_3 + c_1c_2c_3 & -c_1s_2 \\ s_2s_3 & s_2c_3 & c_2 \end{pmatrix},$$

and write $R_{E3} = (r_{\mu,\nu})_{\mu,\nu \in \{1,2,3\}}$. The right-hand side angles $\delta_1, \delta_2, \delta_3$ are the deterministic XZX Euler angles extracted from this same matrix, so $R_{E3} = R_x(\delta_1)\, R_z(\delta_2)\, R_x(\delta_3)$.

The extraction formulas below are exactly those used in the linear-optical literature, following standard Euler-angle extraction conventions [15]; they are stated with explicit "generic" and "gimbal-lock" cases, since Euler angles are not unique in degenerate configurations.

Assume first that we are not in a gimbal-lock configuration for XZX, i.e. $-1 < r_{1,1} < 1$. Then set $\delta_2 = \arccos(r_{1,1})$, $\delta_1 = \operatorname{atan2}(r_{3,1},\, r_{2,1})$, and $\delta_3 = \operatorname{atan2}(r_{1,3},\, -r_{1,2})$. Equivalently, in terms of the freely chosen γ-angles:

$$\begin{aligned} \delta_2 &= \arccos(c_1c_3 - s_1c_2s_3), \\ \delta_1 &= \operatorname{atan2}(s_2s_3,\, s_1c_3 + c_1c_2s_3), \\ \delta_3 &= \operatorname{atan2}(s_1s_2,\, c_1s_3 + s_1c_2c_3). \end{aligned}$$

The remaining cases correspond to gimbal lock for XZX, i.e. $\delta_2 \in \{0, \pi\}$. To keep the schema functional, we fix the canonical representative with last X-rotation equal to 0:

- If $r_{1,1} = 1$, set $\delta_2 = 0$, $\delta_3 = 0$, and $\delta_1 = \operatorname{atan2}(r_{3,2},\, r_{3,3})$.
- If $r_{1,1} = -1$, set $\delta_2 = \pi$, $\delta_3 = 0$, and $\delta_1 = \operatorname{atan2}(-r_{3,2},\, r_{3,3})$.

These choices are one representative among the usual one-parameter family of gimbal-lock solutions, but they give a deterministic right-hand side for the schematic rule.

The extraction is sound in the following sense.

▶ **Lemma C.2.** *With the choices of $\delta_1, \delta_2, \delta_3$ above, the identity*

$$R_z(\gamma_1)R_x(\gamma_2)R_z(\gamma_3) = R_x(\delta_1)R_z(\delta_2)R_x(\delta_3)$$

holds. Consequently the same parameter relation is sound for the derived mixed rule (3CRx)*, whose proof reduces to the uncontrolled three-rotation pattern.*

Proof. For arbitrary a, b, c, direct multiplication gives

$$R_x(a)R_z(b)R_x(c) = \begin{pmatrix} \cos b & -\sin b\cos c & \sin b \sin c \\ \sin b \cos a & -\sin a \sin c + \cos a \cos b \cos c & -\sin a \cos c - \sin c \cos a \cos b \\ \sin a \sin b & \sin a \cos b \cos c + \sin c \cos a & -\sin a \sin c \cos b + \cos a \cos c \end{pmatrix}.$$

In the generic case $-1 < r_{1,1} < 1$, we have $\sin(\delta_2) > 0$. The displayed matrix then shows that $\cos(\delta_2) = r_{1,1}$, $(\cos\delta_1, \sin\delta_1) = \frac{(r_{2,1}, r_{3,1})}{\sin\delta_2}$, and $(\cos\delta_3, \sin\delta_3) = \frac{(-r_{1,2}, r_{1,3})}{\sin\delta_2}$, which is exactly the stated arccos / atan2 extraction. Substituting these values into the matrix for $R_x(\delta_1)R_z(\delta_2)R_x(\delta_3)$ recovers all entries of R_{E3}.

If $r_{1,1} = 1$, then the first row and column force the matrix to be a pure x-rotation on the last two coordinates, and $\operatorname{atan2}(r_{3,2}, r_{3,3})$ is its angle. Thus the canonical choice $\delta_2 = \delta_3 = 0$ is sound. If $r_{1,1} = -1$, then the same displayed matrix with $b = \pi$ shows that the free invariant is $\delta_1 - \delta_3$, and the canonical choice $\delta_2 = \pi, \delta_3 = 0, \delta_1 = \operatorname{atan2}(-r_{3,2}, r_{3,3})$ is sound.

The rule (3CRx) uses the same real rotation identity after the surrounding controlled X-moves expose the uncontrolled three-rotation subdiagram in Appendix H; therefore no additional angle relation is needed for the controlled version. ◀

D Support diagrammatic derivations

All derivations in this appendix take place in the equational theory QC_d. We write $\mathrm{QC}_d \vdash C_1 = C_2$ when the circuit identity $C_1 = C_2$ is derivable from the axioms of Figures 2–3. Unlabelled equalities in the displayed diagram chains are definitional unfoldings of the derived notation or strict PROP coherence; labelled equalities cite the axiom or derived rule doing the substantive rewrite.

D.1 Catalogue of derived rules

The catalogue is split into three batches. Figure 4 gathers the elementary support identities used for scalar phases, involutions, totalisations, controlled transpositions, and the basic X-interaction rules. Figure 5 adds the local Hadamard–phase conjugation rules together with the decomposition identities for controlled H, controlled phase, and higher-controlled phase gates. Figure 6 records the remaining commutation laws between differently value-controlled phases, controlled Hadamards, and controlled transpositions. Subsection D.2 proves these rules in the same order.

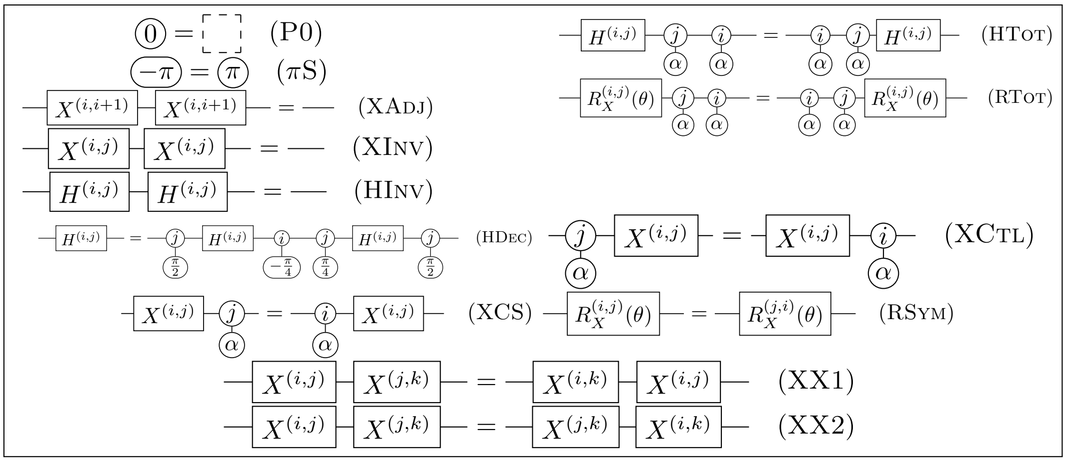


Figure 4 Derived rules in QC_d (batch 1). The rules (XX1) and (XX2) assume that i, j, k are pairwise distinct.

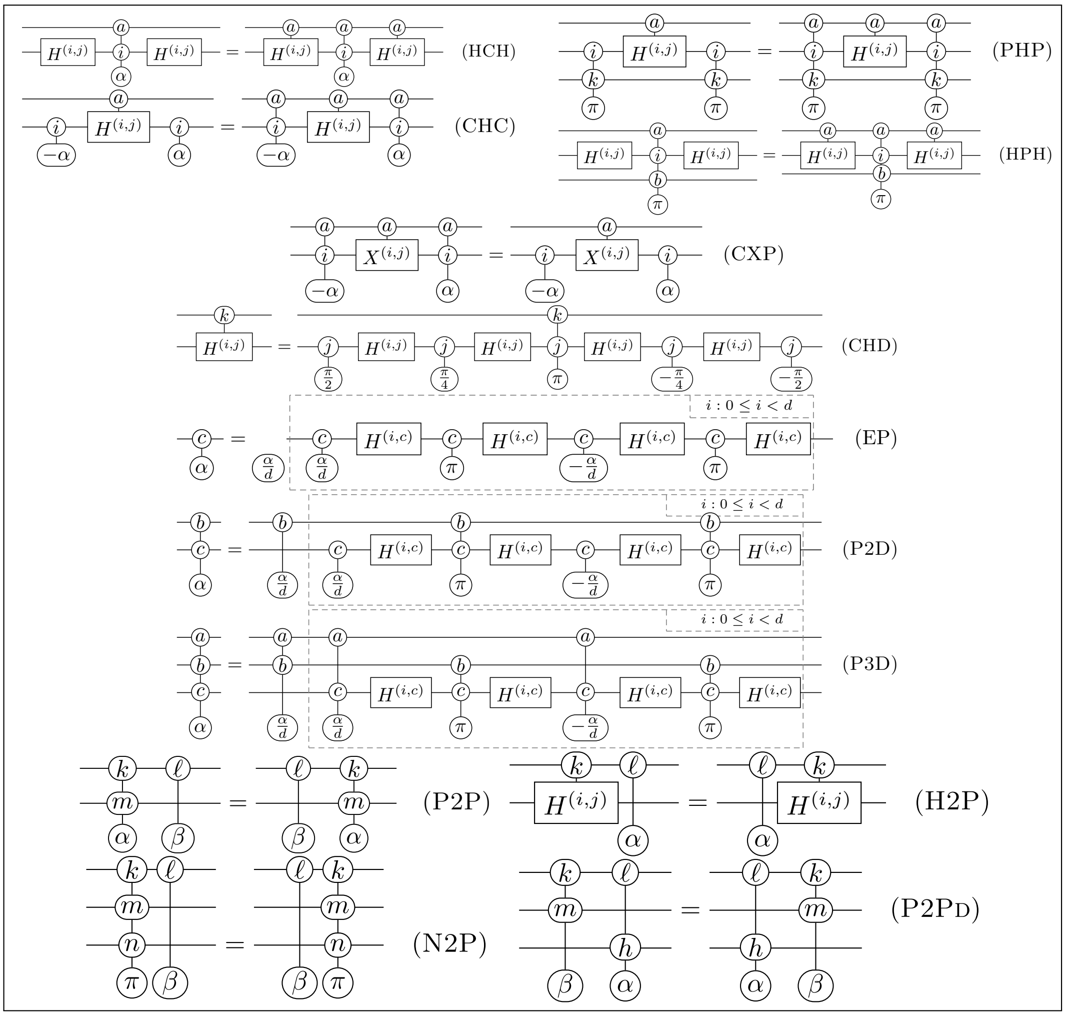
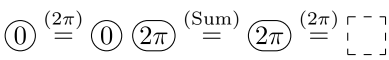


Figure 5 Derived rules in QC_d (batch 2). The rules (P2P), (N2P), (H2P), and (P2PD) assume $k \neq \ell$.

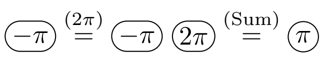
D.2 Proofs of the derived rules

Proof of (P0).



◀

Proof of (πS).



◀

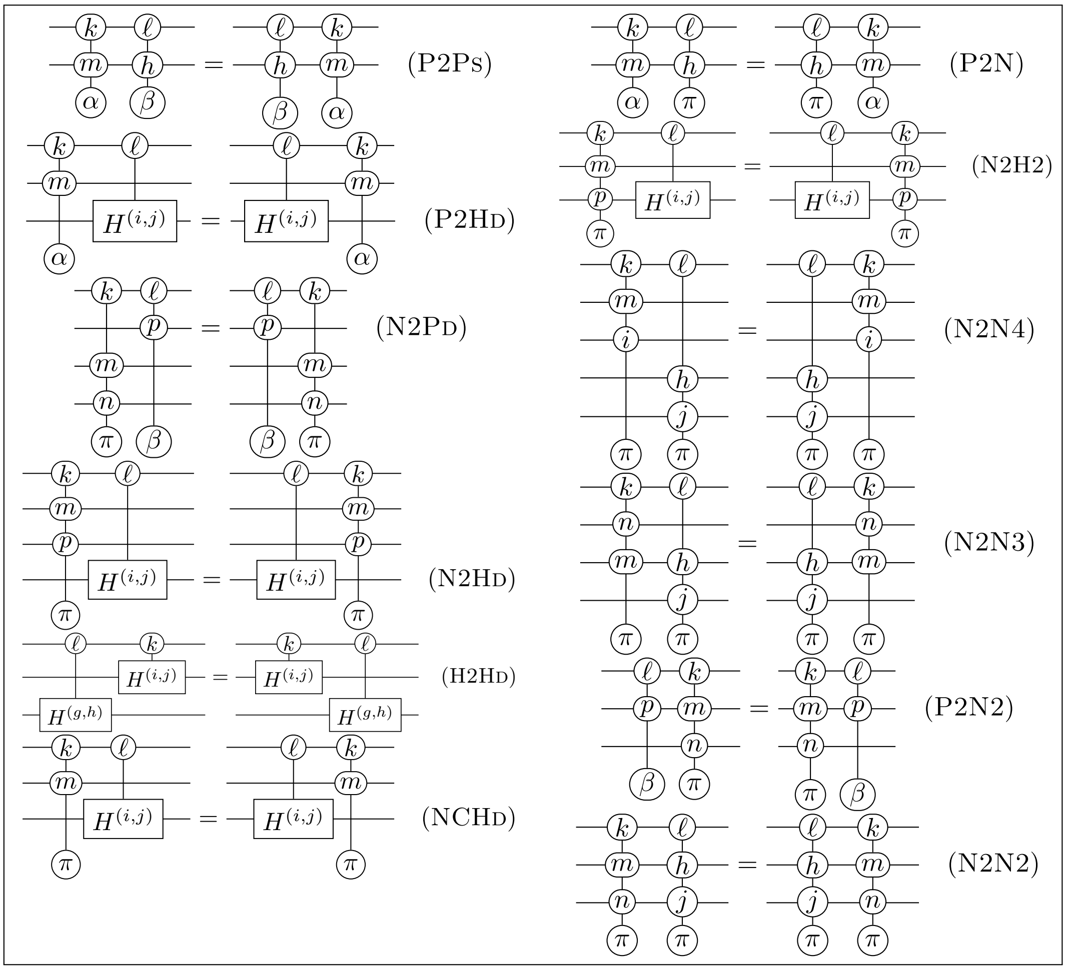
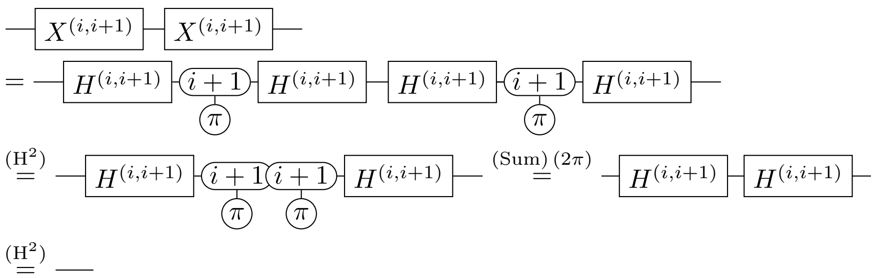


Figure 6 Derived rules in QC_d (batch 3). All rules in this batch assume $k \neq \ell$.

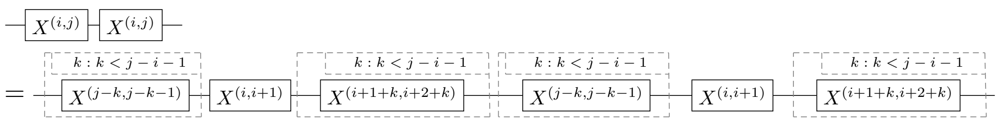
Proof of (XAdj).

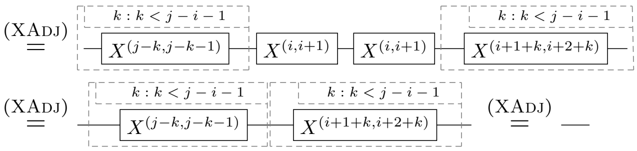


◀

Proof of (XInv). If $i = j$, this is trivial.

If $i < j$

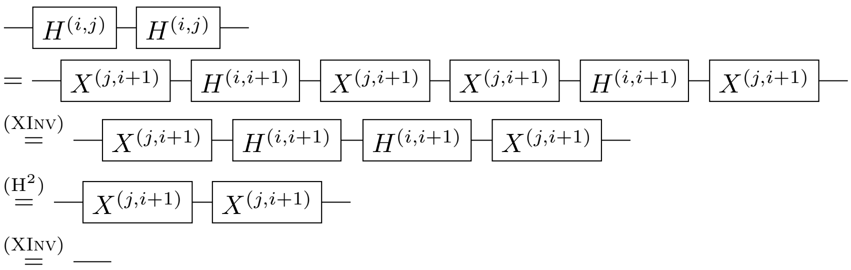




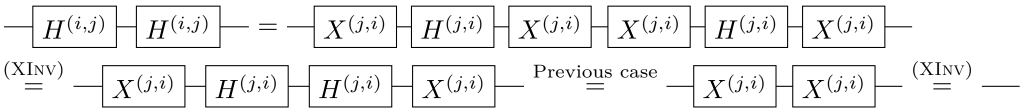
If $i > j$, by definition $X^{i,j} = X^{j,i}$ so this is covered by the previous case. ◀

Proof of (HInv). If $i = j$, this is trivial.

If $i < j$

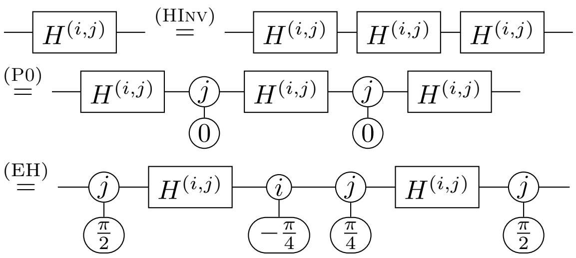


If $i > j$,



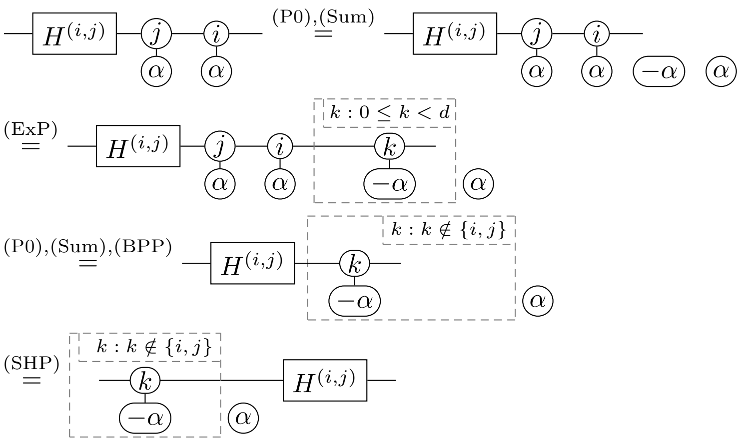
◀

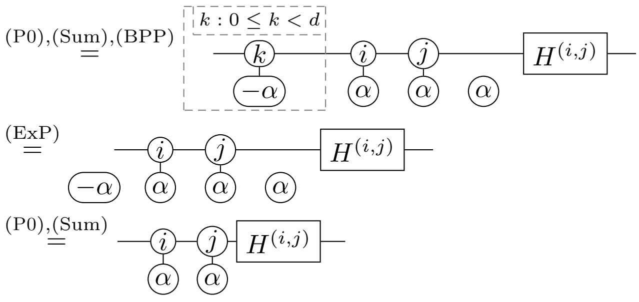
Proof of (HDec).



◀

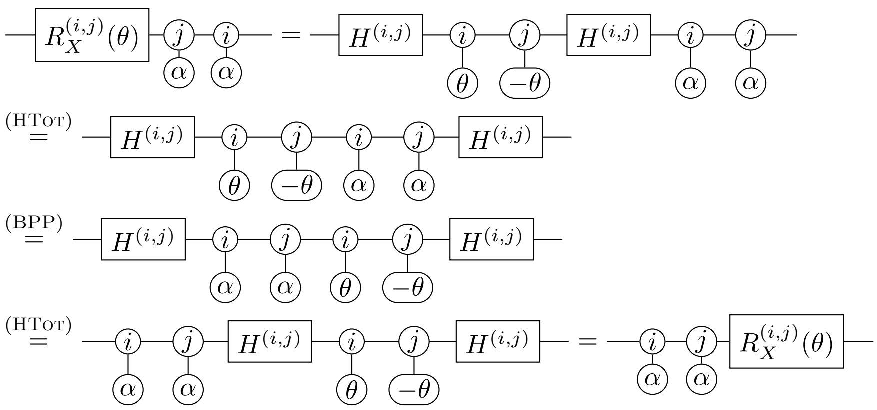
Proof of (HTot).





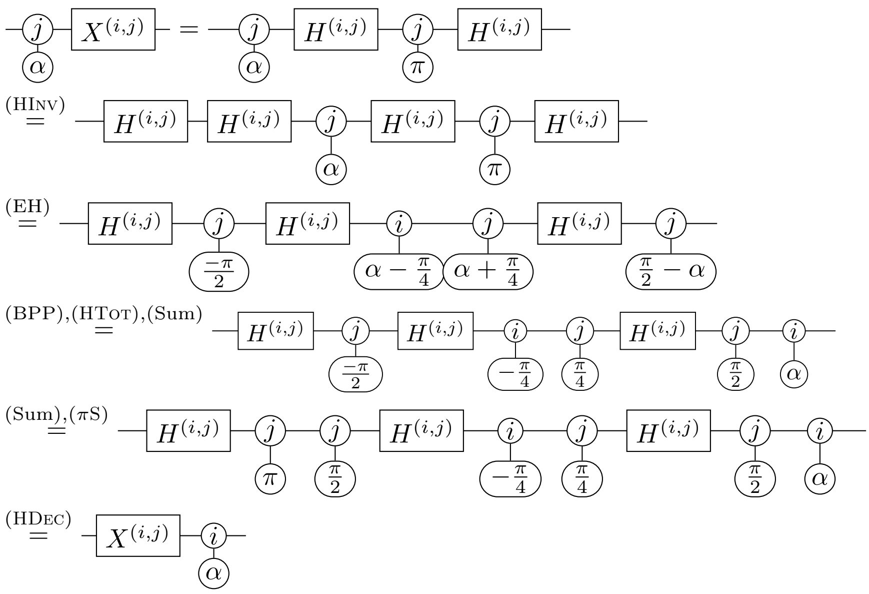
◀

Proof of (RTOT).



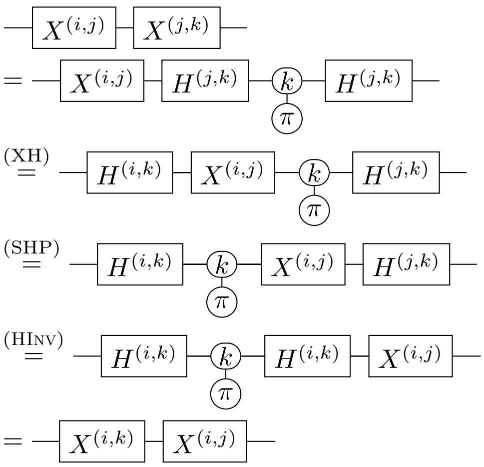
◀

Proof of (XCTL).



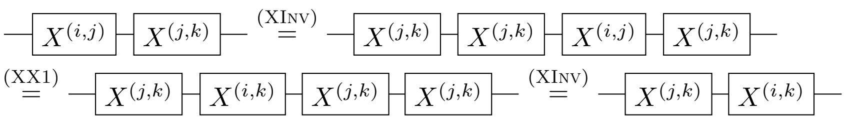
◀

Proof of (XX1).



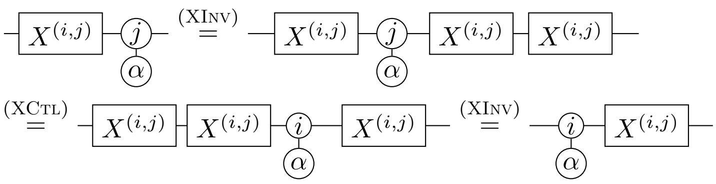
◀

Proof of (XX2).



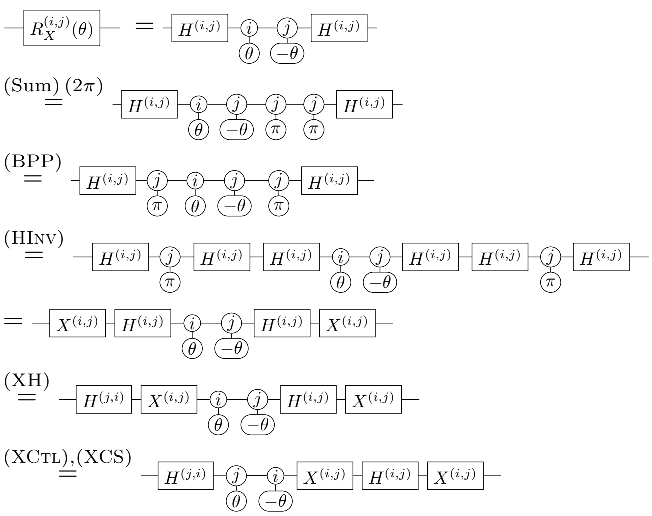
◀

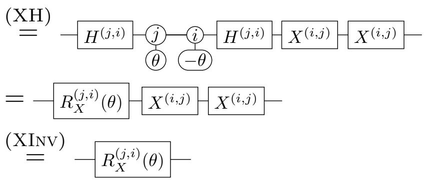
Proof of (XCS).



◀

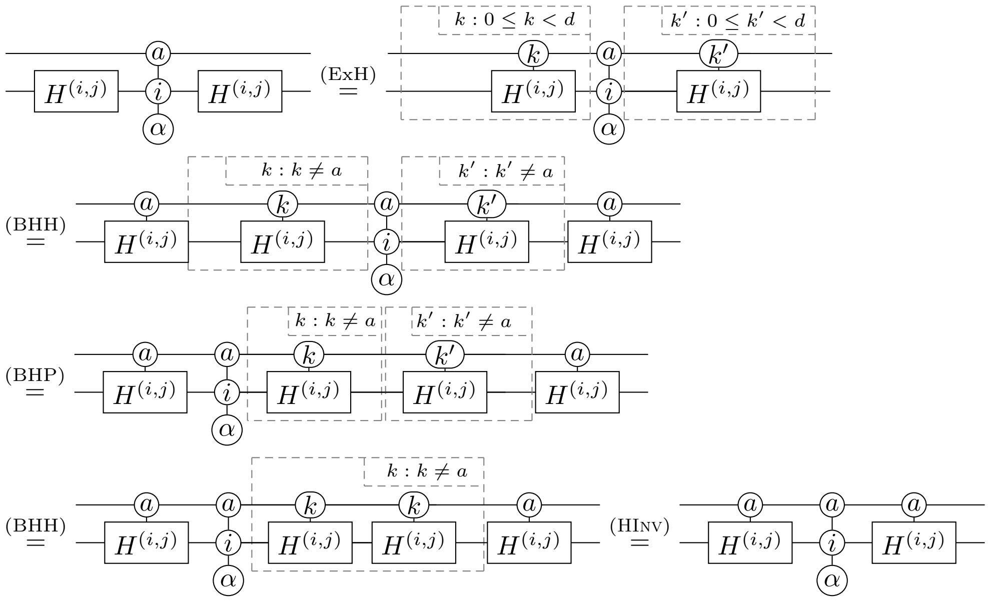
Proof of (RSym).





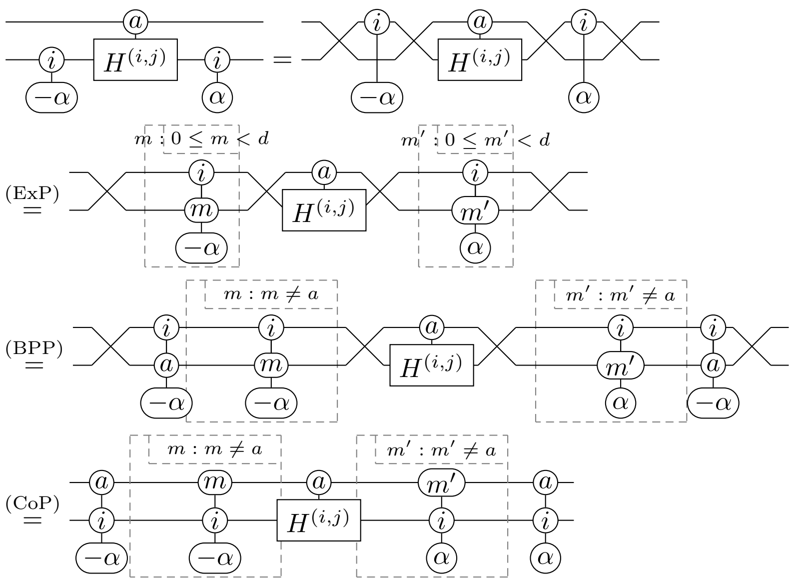
◀

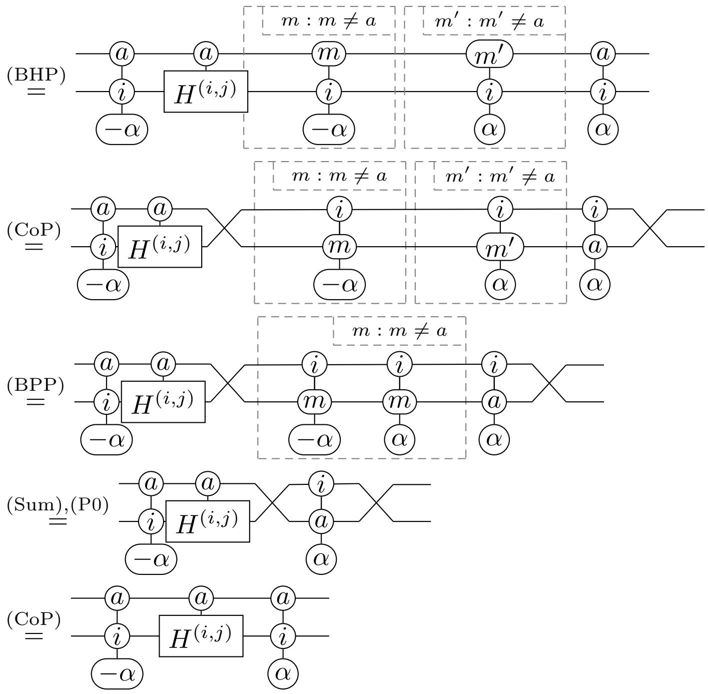
Proof of (HCH).



◀

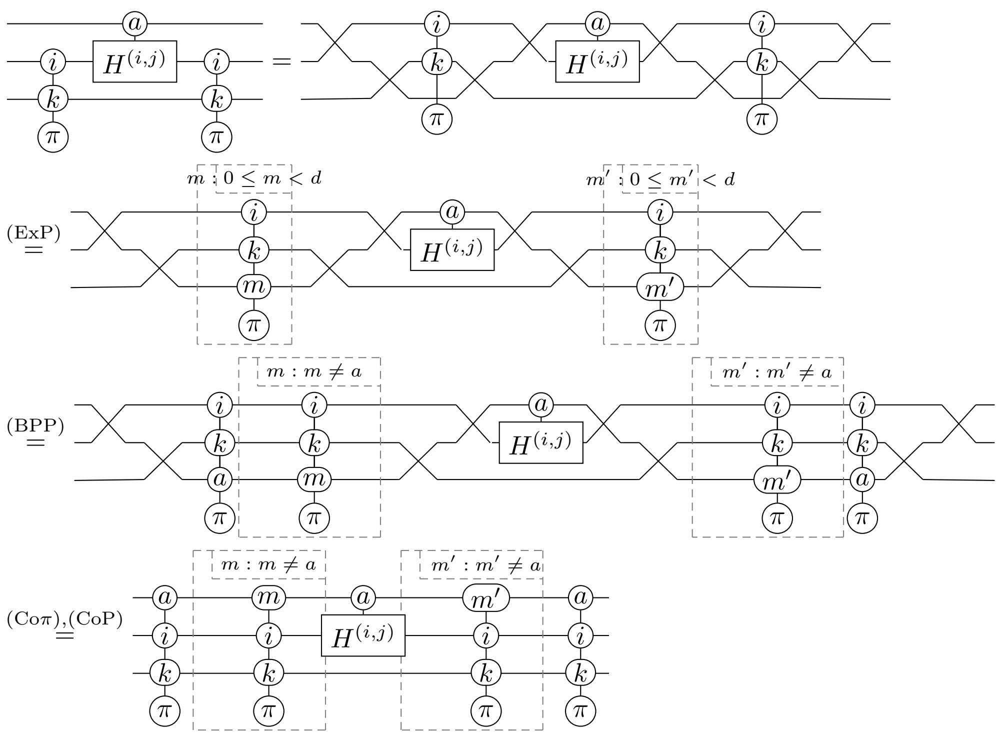
Proof of (CHC).

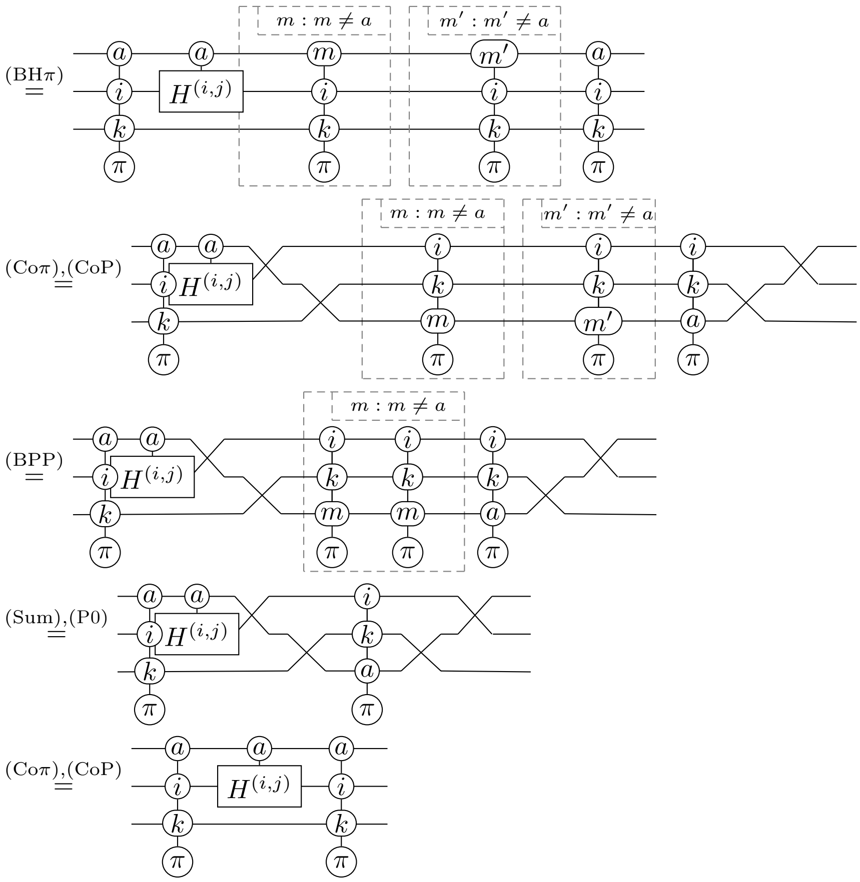




◀

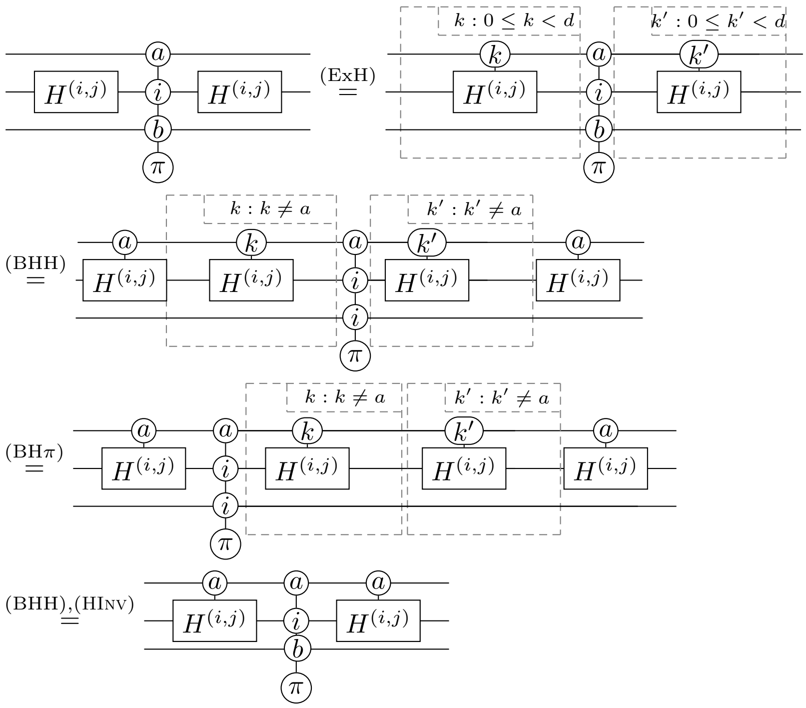
Proof of (PHP).




(BHπ) =
m : m ≠ a
m′ : m′ ≠ a
(Coπ),(CoP) =
(BPP) =
(Sum),(P0) =
(Coπ),(CoP) =

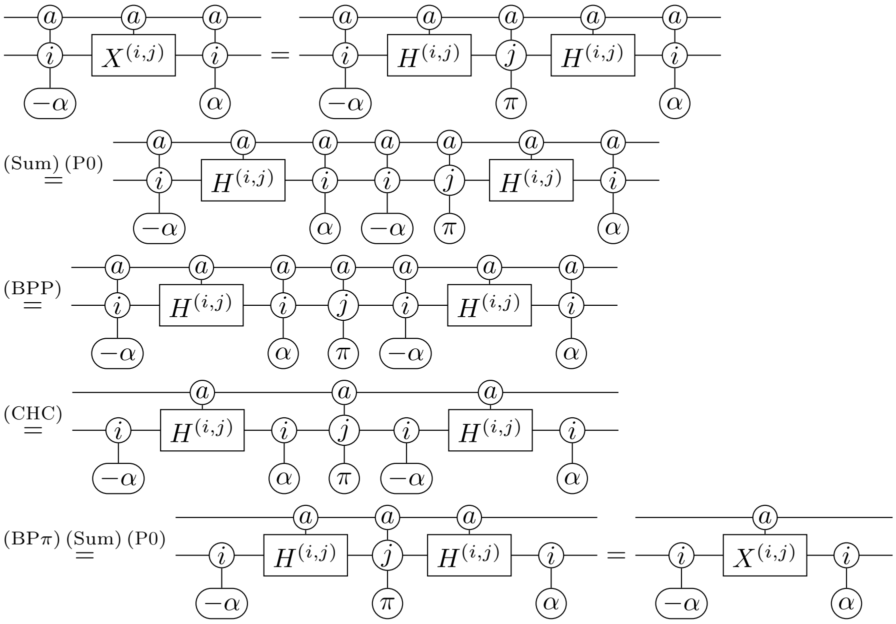
◄

Proof of (HPH).


(ExH) =
k : 0 ≤ k < d
k′ : 0 ≤ k′ < d
(BHH) =
k : k ≠ a
k′ : k′ ≠ a
(BHπ) =
(BHH),(HInv) =

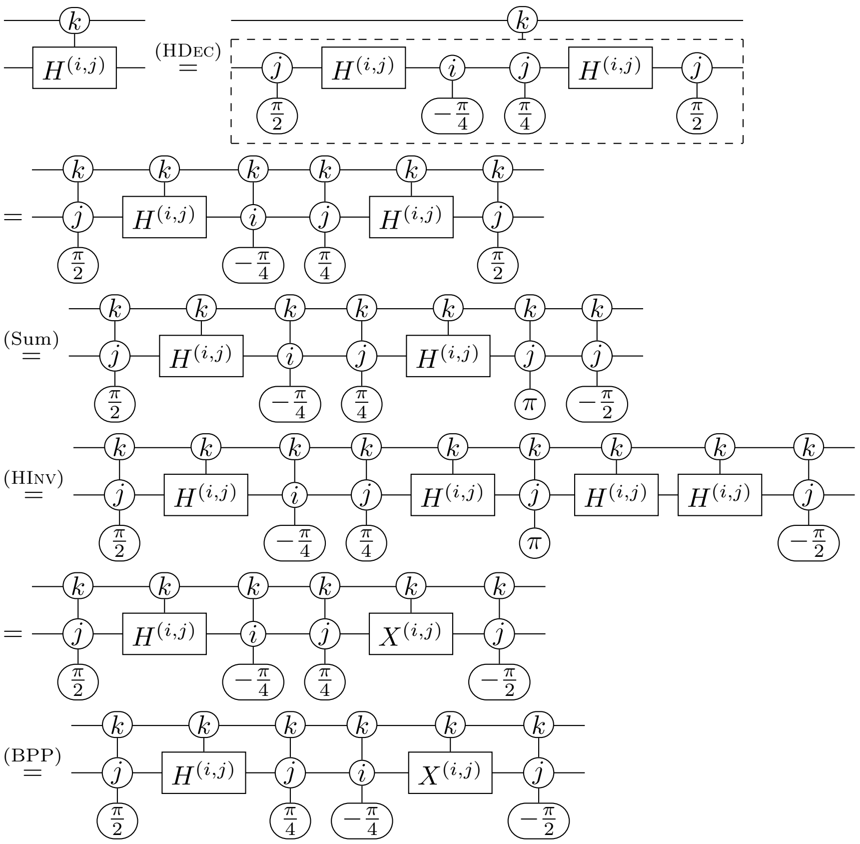
◀

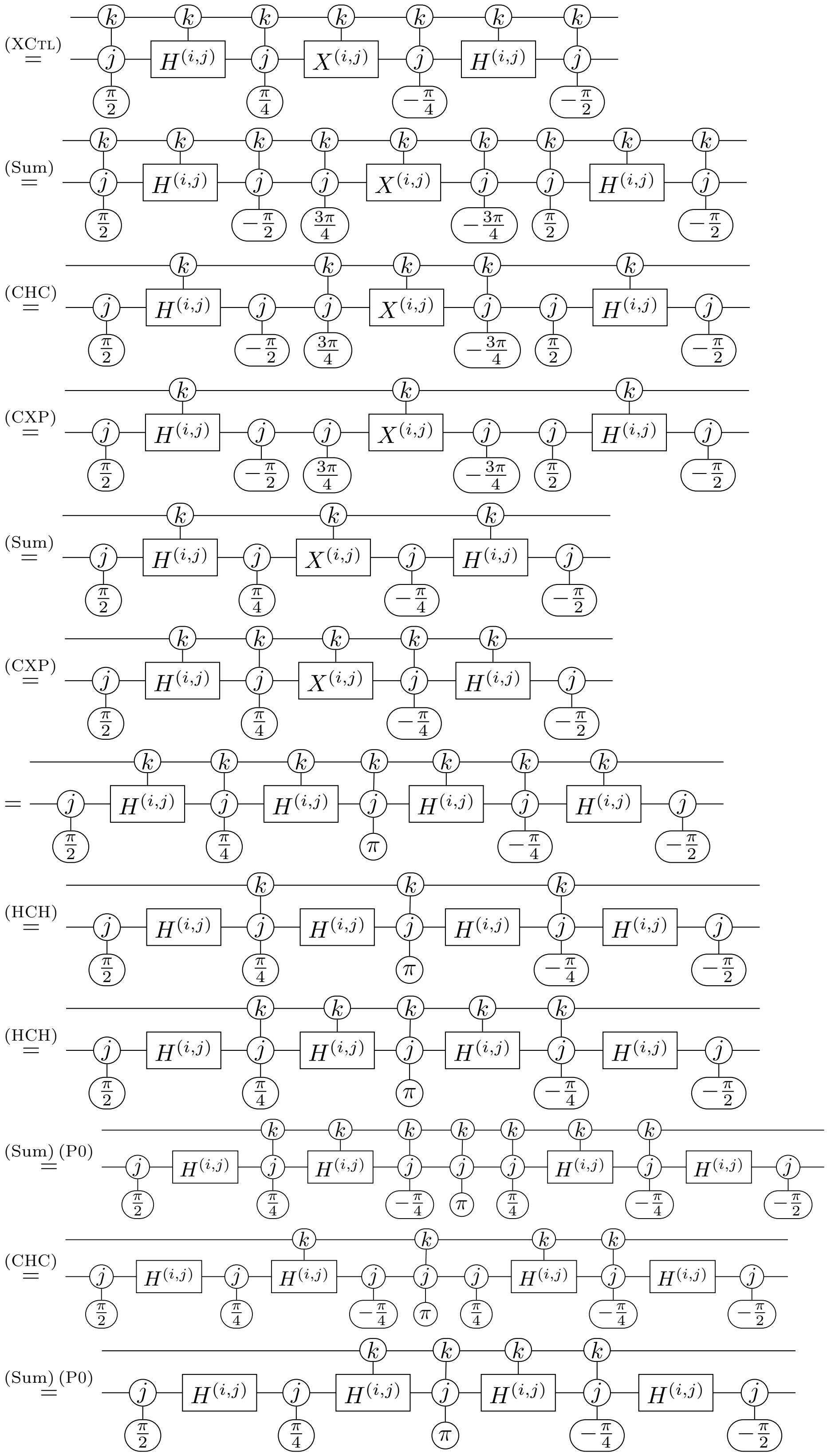
Proof of (CXP).



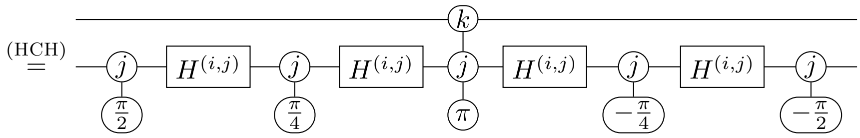
◀

Proof of (CHD).



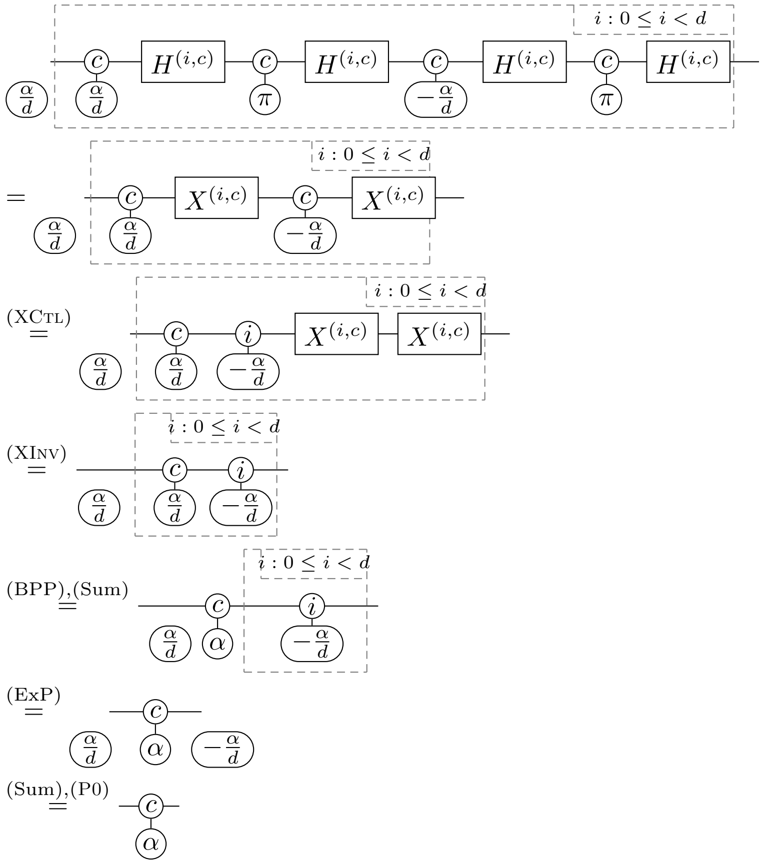


(XCtl) =
(Sum) =
(CHC) =
(CXP) =
(Sum) =
(CXP) =
=
(HCH) =
(HCH) =
(Sum) (P0) =
(CHC) =
(Sum) (P0) =



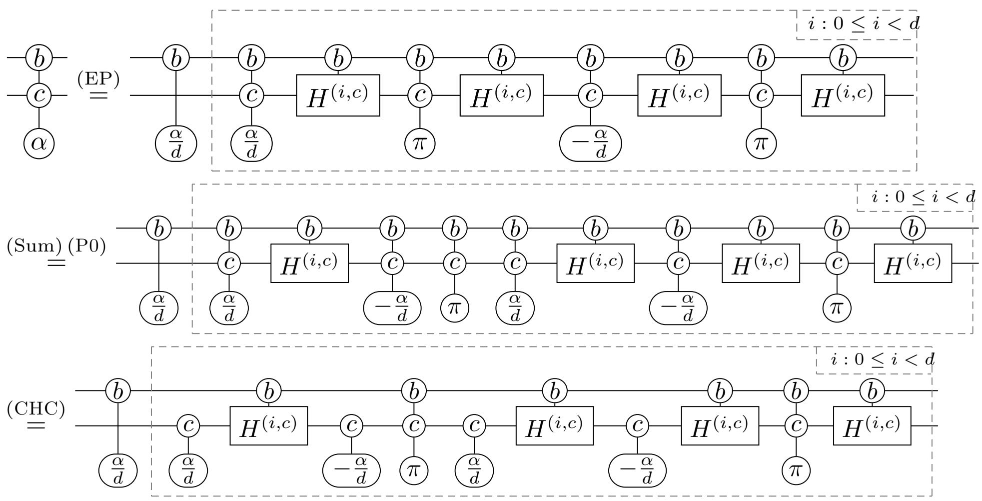
◀

Proof of (EP).

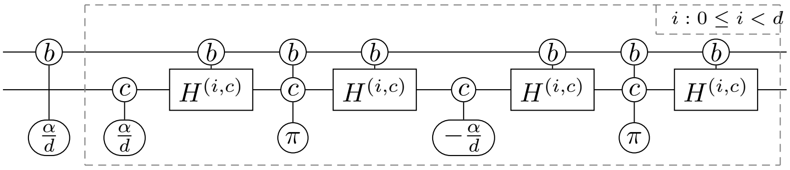


◀

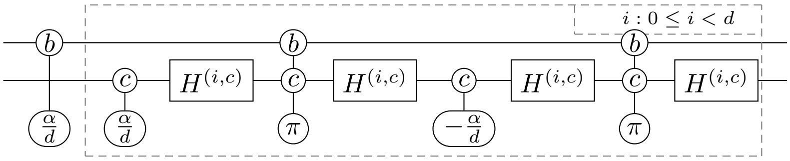
Proof of (P2D).



(HCH) =

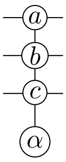


(Sum) (P0) =

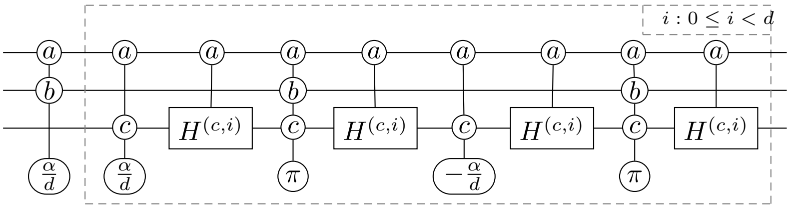


◀

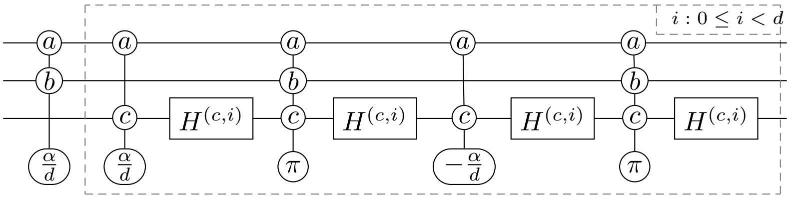
Proof of (P3D).



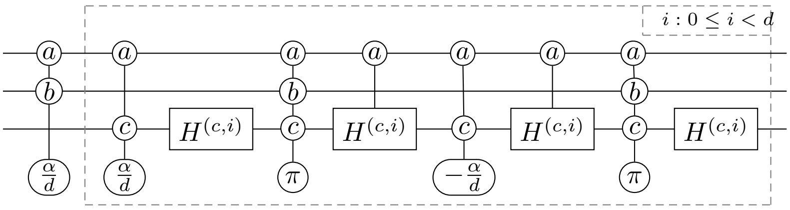
(P2D) =



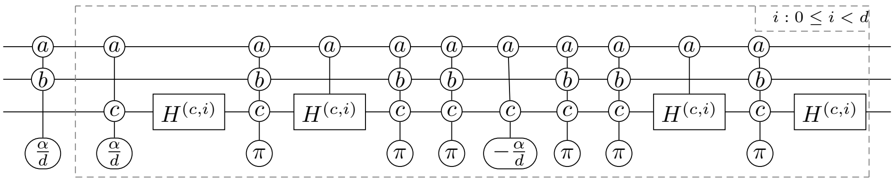
(HPH) =



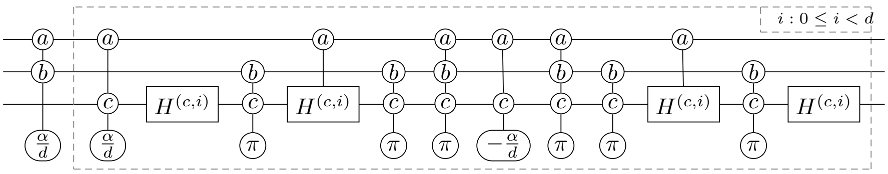
(HCH) =



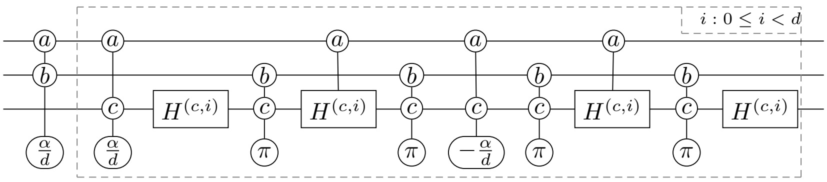
(Sum) (2π) =

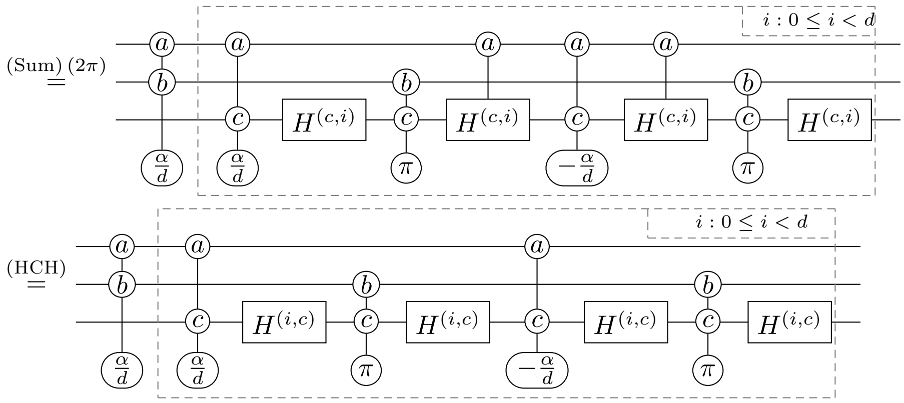


(PHP) =

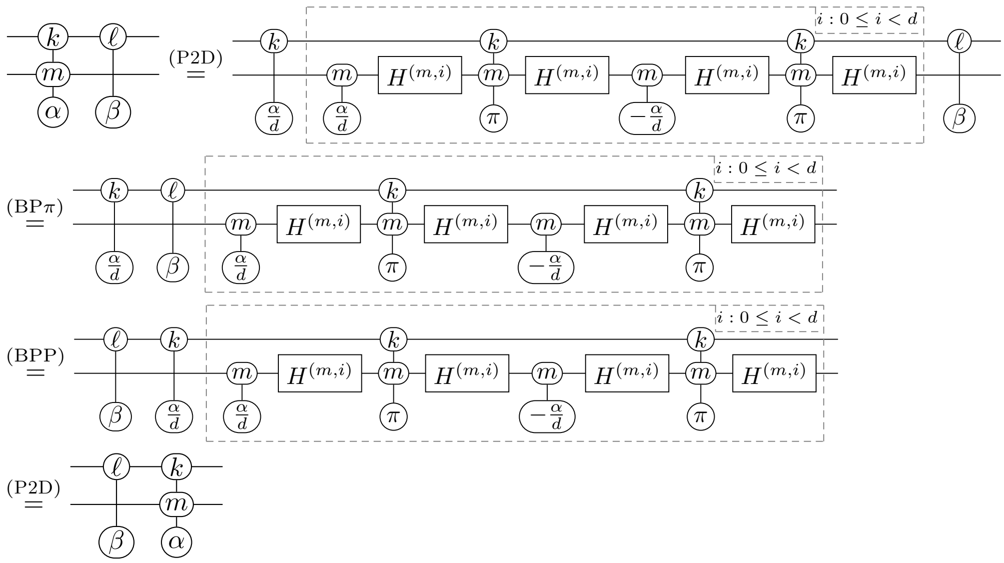


(Sum) (2π) =

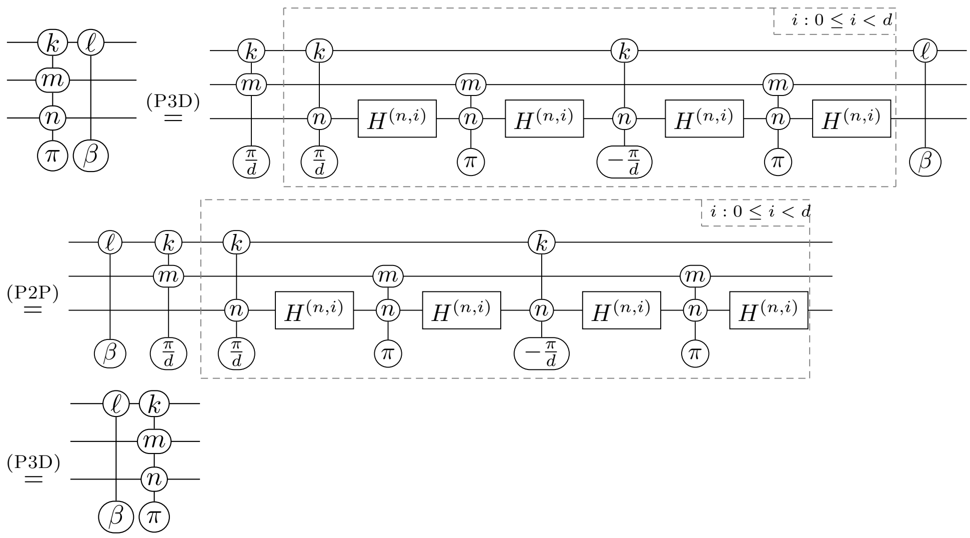




Proof of (P2P).

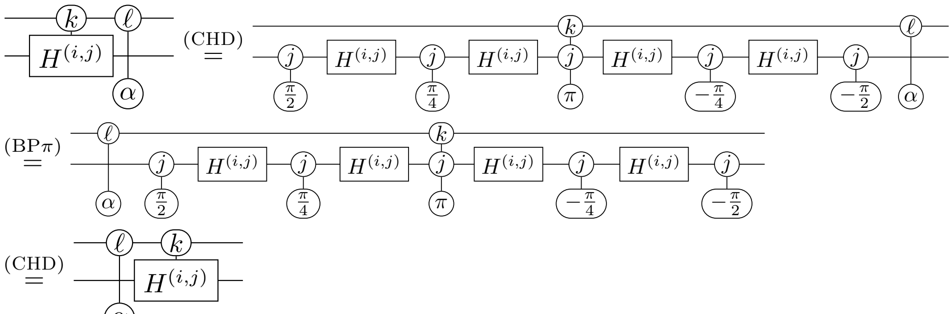


Proof of (N2P).



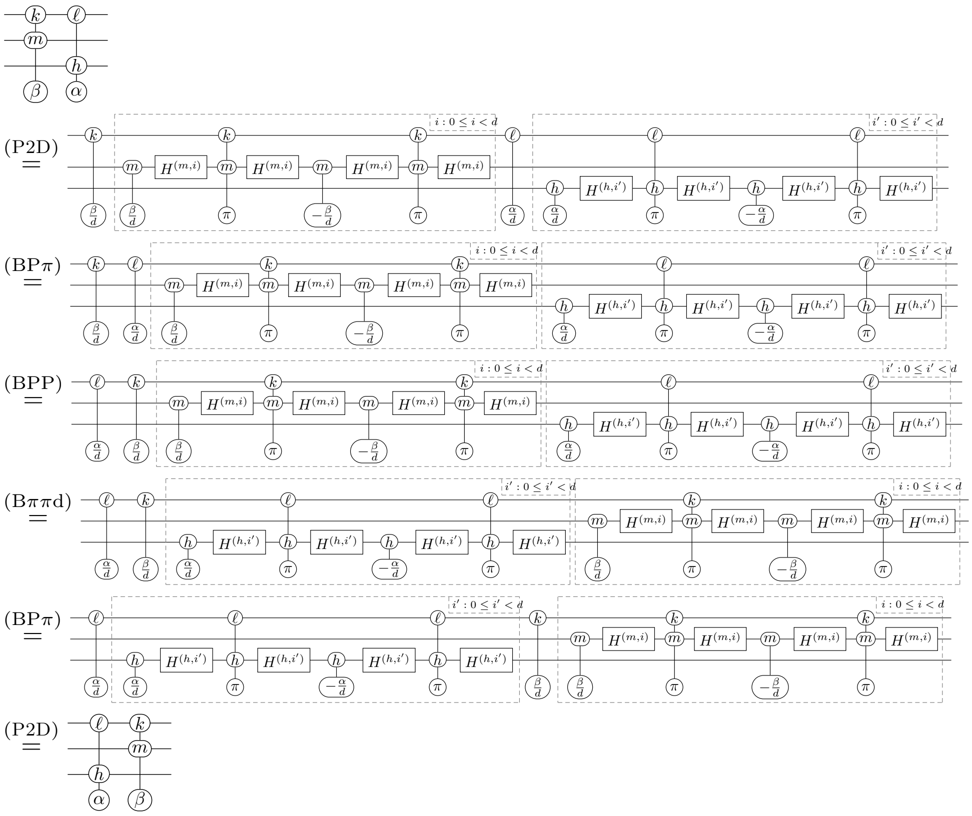
◀

Proof of (H2P).



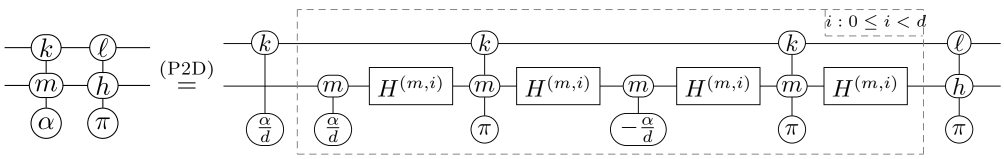
◀

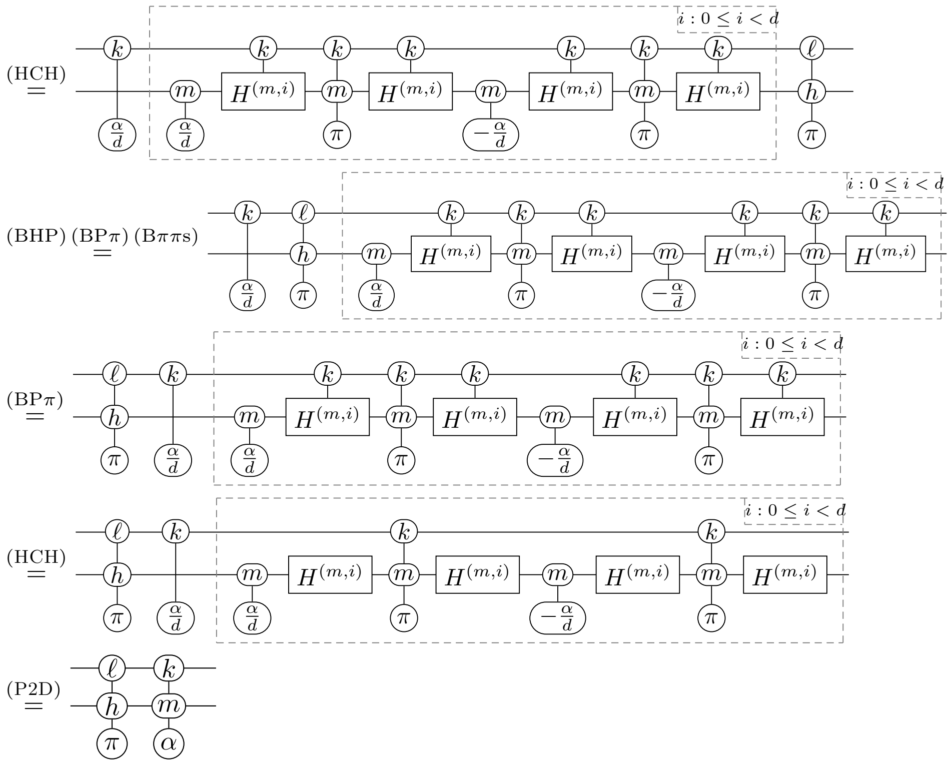
Proof of (P2PD).



◀

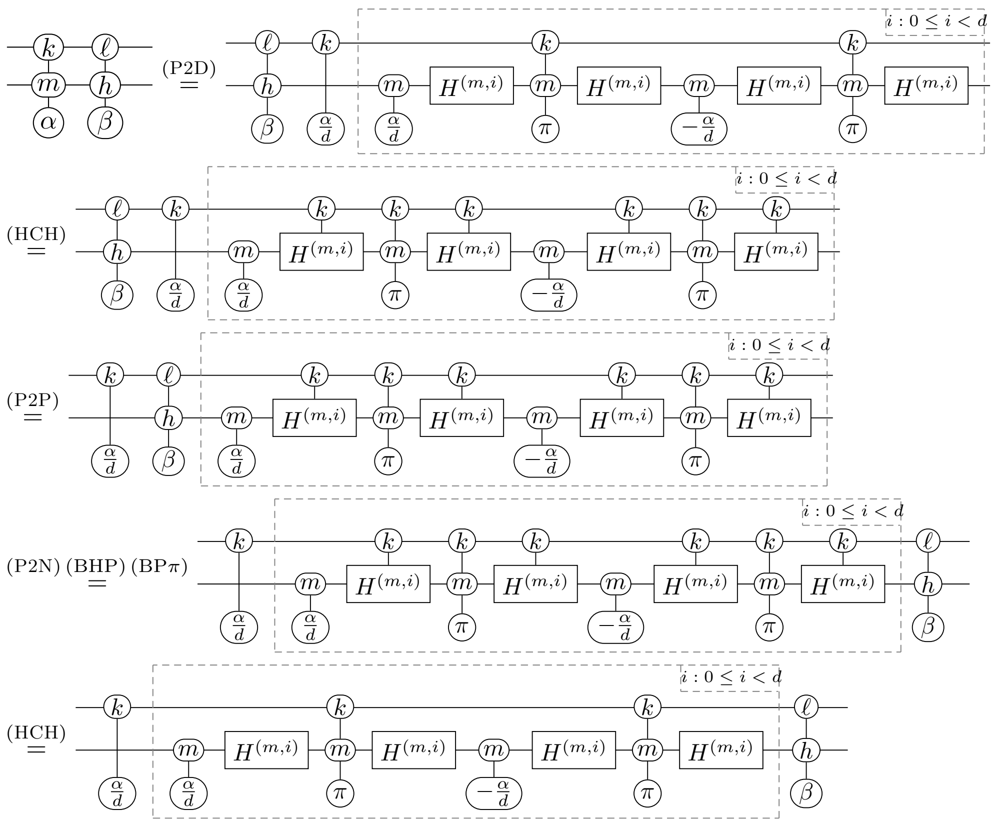
Proof of (P2N).

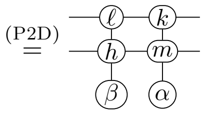




◄

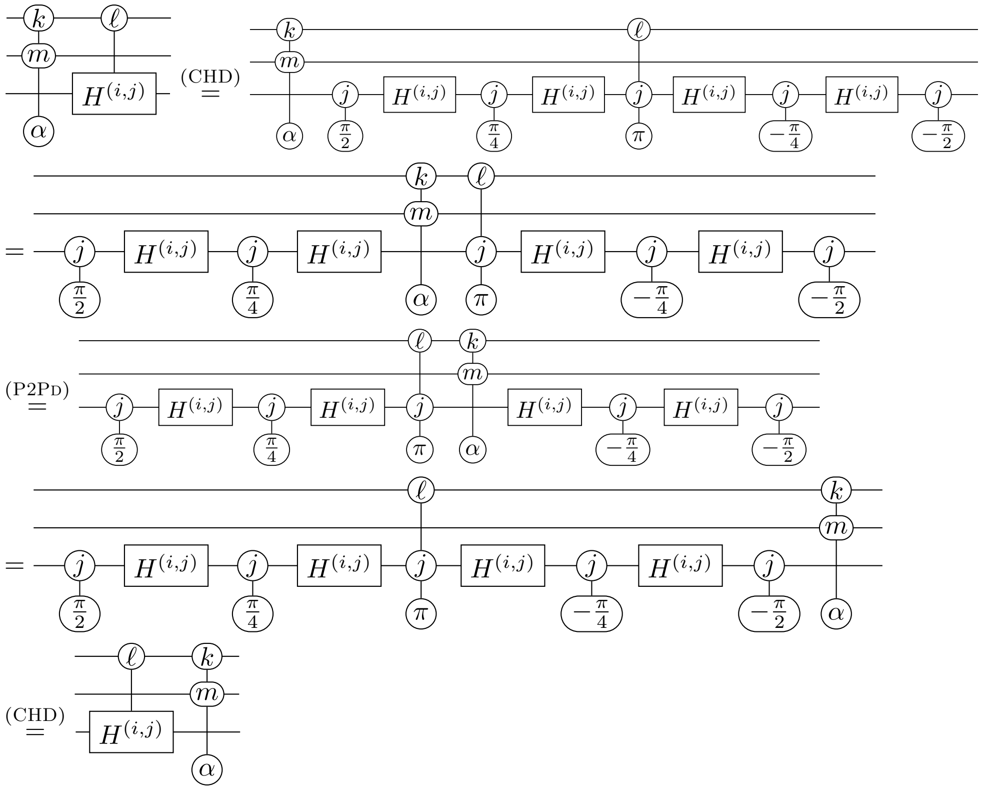
Proof of (P2Ps).





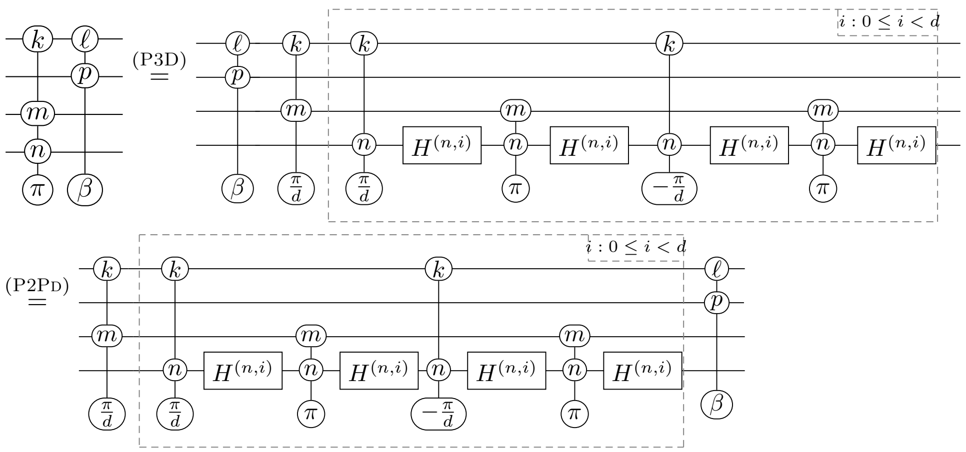
◀

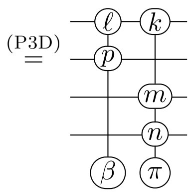
Proof of (P2HD).



◀

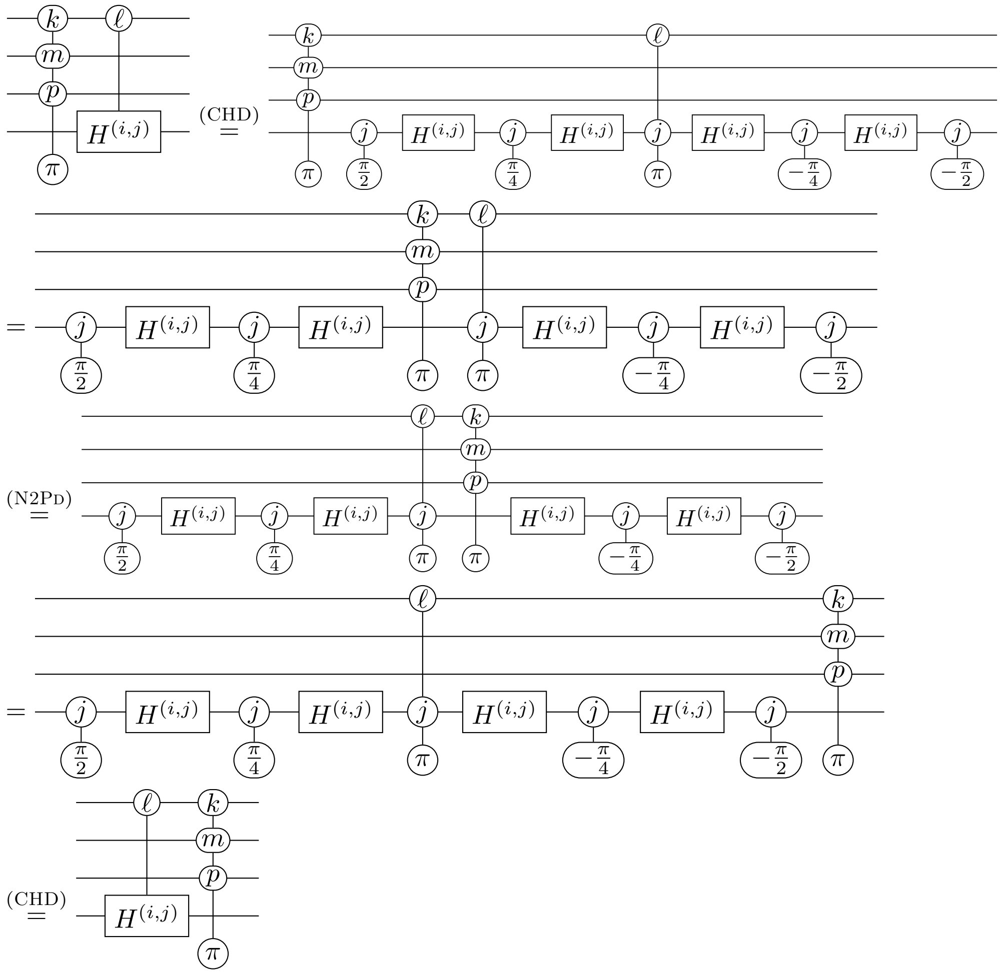
Proof of (N2PD).





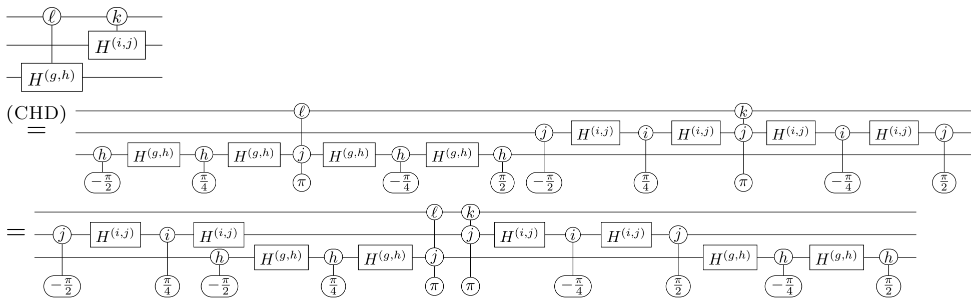
◀

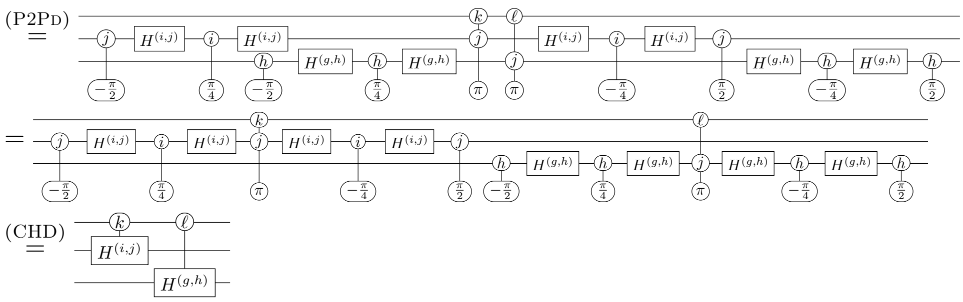
Proof of (N2HD).



◀

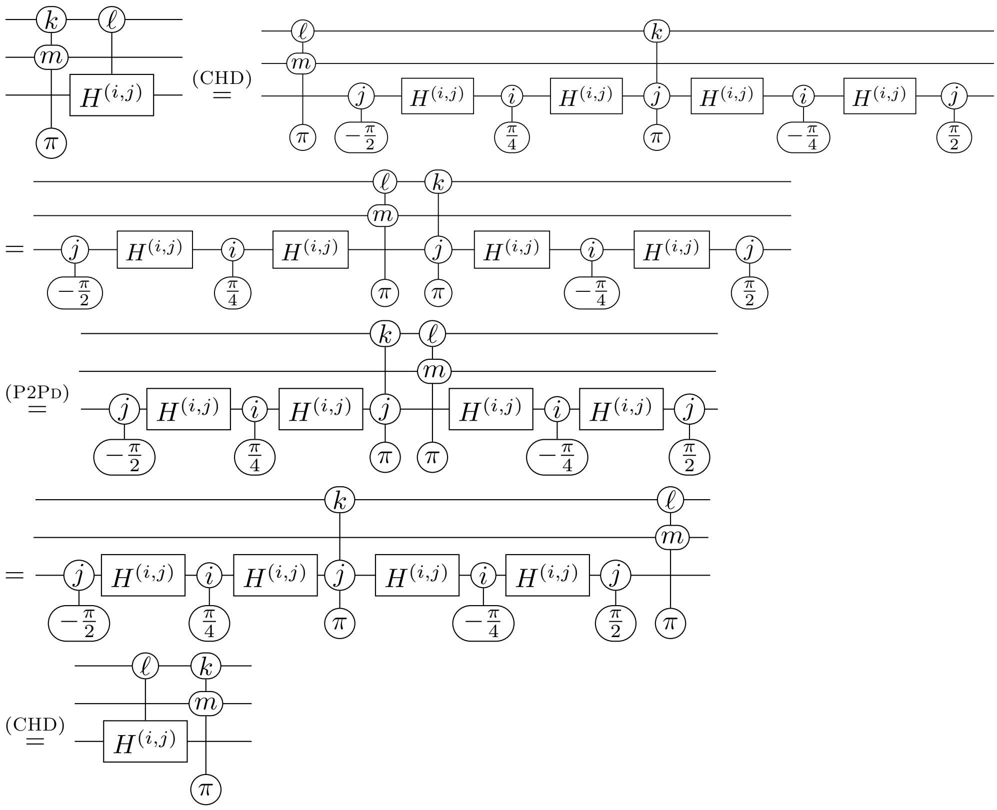
Proof of (H2HD).





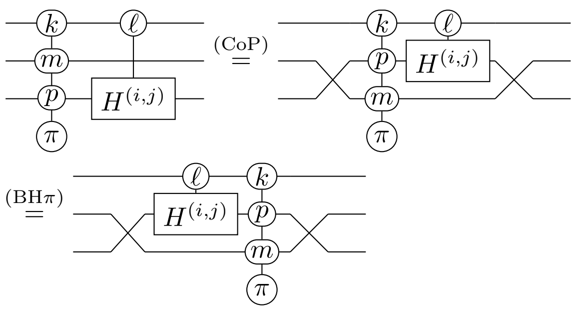
◀

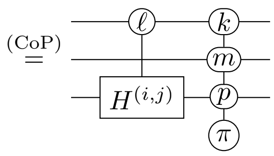
Proof of (NCHD).



◀

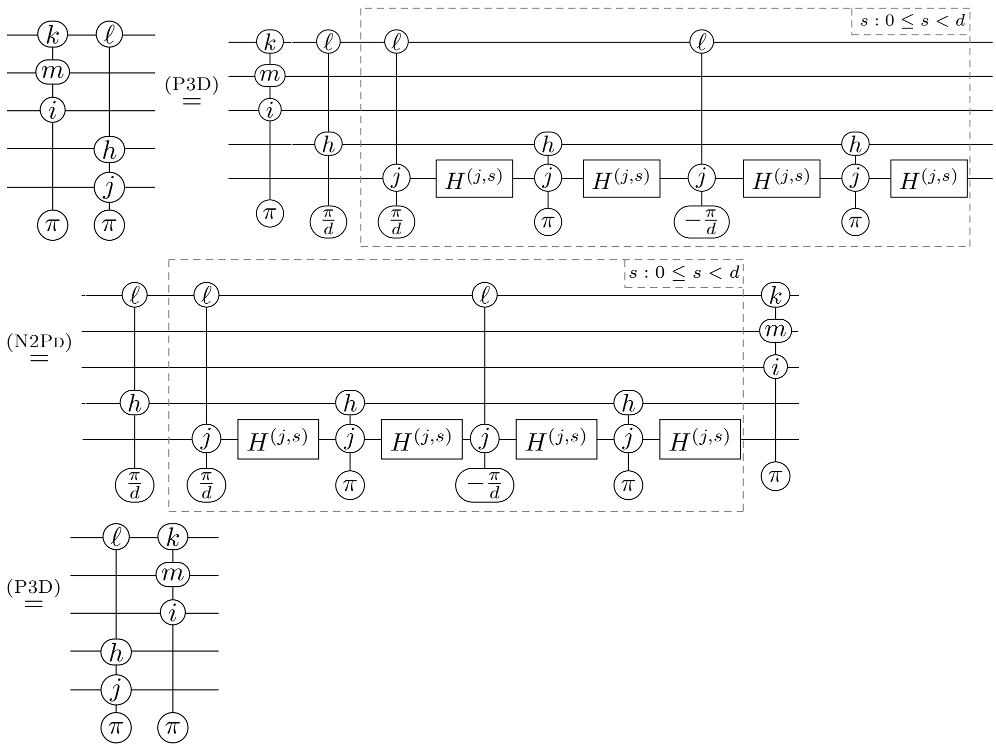
Proof of (N2H2).





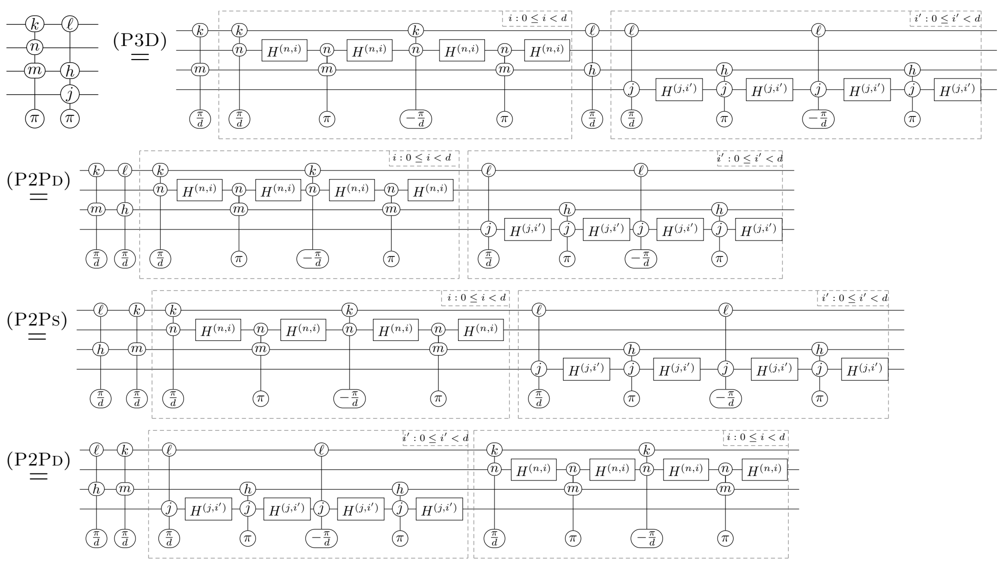
◄

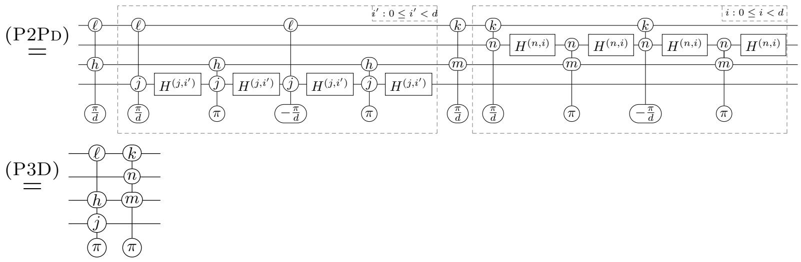
Proof of (N2N4).



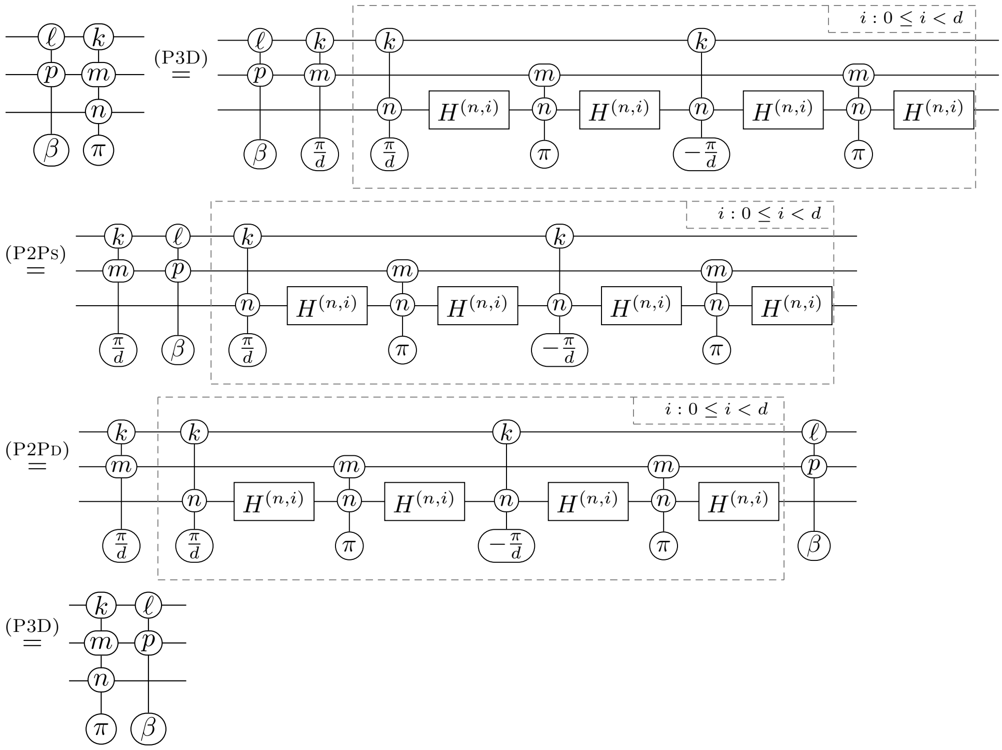
◄

Proof of (N2N3).

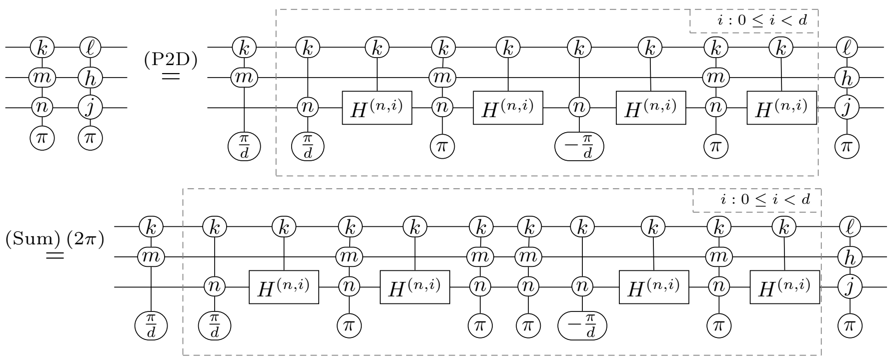




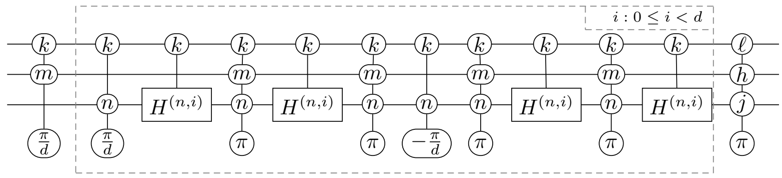
Proof of (P2N2).



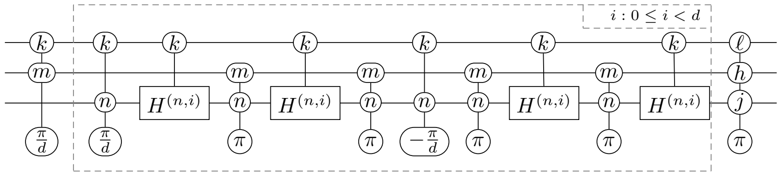
Proof of (N2N2).



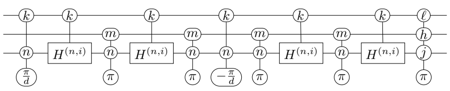
(P2N2) $=$



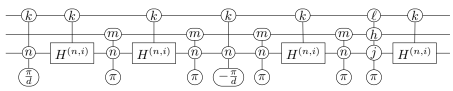
(PHP) $=$



For any i we have

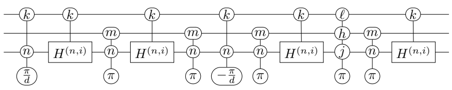


(N2H2) $=$



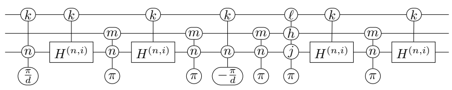
(1)

(P2N2) $=$



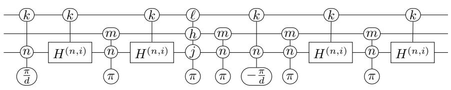
(2)

(N2H2) $=$



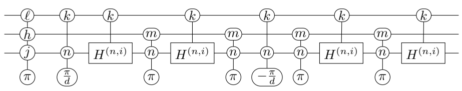
(3)

(P2N2) $=$



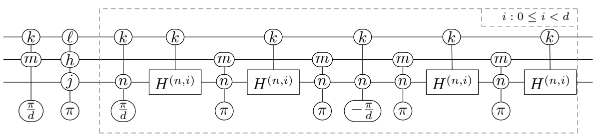
(4)

(P2N2) (N2H2) $=$

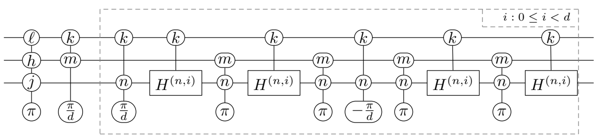


(5)

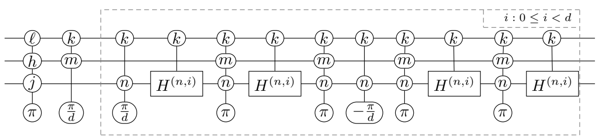
and hence, applying this for all i,



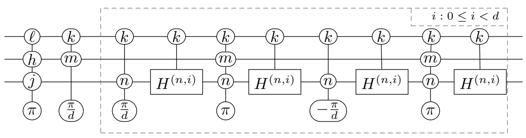
(P2N2) $=$



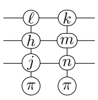
(PHP) $=$



(P2N2) (Sum) (2π) $=$



(P2D) $=$



◀

E Compatibility

We prove Theorem 3.9: mixed-control compatibility, in the sense of Definition 2.2, is derivable in QC_d. The argument is purely syntactic and uses the same normalisation step as Appendix F. Throughout we work modulo:

- the strict symmetric monoidal coherence in $\mathbf{CQC}_d$ (associativity, unit, symmetry, interchange), so wire permutations can be freely inserted and removed; and
- the coherence laws for each control functor $\mathtt{C}_k$ (functoriality, strength, and naturality), so controls can be pushed through structural contexts.

The reduction is a controlled-layer normal form. Once the two nested-control circuits are in that form, compatibility becomes a local swap-through calculation: every layer except the doubly controlled π-phase case passes through the outer swap by structural naturality.

E.1 Normalisation into controlled layers

Algorithm 1 SEPARATE: push parallelism out, isolate single-gate layers with explicit controls

function SEPARATE(f)
  replace every structural swap in f by the fixed decomposition (S)
  use SMC coherence only for reassociation, units, and wire-permutation bookkeeping; call the result f'
  unfold derived one-qudit notation in f' into adjacent $H^{(r,r+1)}$ gates and controlled phases
  match f' **with**
    case $f_1 \otimes f_2$:
      return $(\textsc{Separate}(f_1) \otimes \mathrm{id}_{|f_2|}) \circ (\mathrm{id}_{|f_1|} \otimes \textsc{Separate}(f_2))$
    case $f_1 \circ f_2$:
      return $\textsc{Separate}(f_1) \circ \textsc{Separate}(f_2)$
    case id_n: **return** the empty product at arity n
    case $\mathtt{C}_m(f_1)$:
      match f_1 **with**
        case $g \circ h$: **return** $\textsc{Separate}(\mathtt{C}_m(g)) \circ \textsc{Separate}(\mathtt{C}_m(h))$
        case $g \otimes h$:
          return $(\textsc{Separate}(\mathtt{C}_m(g)) \otimes \mathrm{id}_{|h|}) \circ (\mathrm{id}_1 \otimes \sigma_{|g|,|h|}) \circ$
            $(\textsc{Separate}(\mathtt{C}_m(h)) \otimes \mathrm{id}_{|g|}) \circ (\mathrm{id}_1 \otimes \sigma_{|h|,|g|})$
        case $-\boxed{H^{(r,r+1)}}-$:
          rewrite f' by the adjacent instance of CHD
          return $\textsc{Separate}(f')$
        case (α): **return** f'
        case id_n: **return** the empty product at arity $n+1$
        case $\mathtt{C}_n(g)$:
        match g **with**
          case (π): **return** f'
          case (β): **return** $\textsc{Separate}($**Apply P2D to rewrite** $f')$
          case $\mathtt{C}_k((\alpha))$: **return** $\textsc{Separate}($**Apply P3D to rewrite** $f')$
          otherwise return $\textsc{Separate}\big(\mathtt{C}_m(\textsc{Separate}(f_1))\big)$
    otherwise: **return** f'

This subsection isolates a small family of one-gate layers, each with an explicit control word. Every circuit can be rewritten in QC_d into a sequential composition of such layers, with all wire permutations explicit. That is what reduces compatibility to the single axiom CoP, and commutativity to a finite table of generator commutations.

▶ **Definition E.1.** *An atomic layer is a morphism $L := (id_p \otimes G \otimes id_q) : n \to n$, where G is one of: $\textcircled{\theta}$ for $\theta \in \mathbb{R}$, $-\boxed{H^{(r,r+1)}}-$ for $0 \le r < d-1$, $\mathtt{C}_k(\textcircled{\theta})$ for k, θ, or $\mathtt{C}_k(\mathtt{C}_\ell(\textcircled{\pi}))$ for k, ℓ. A* controlled layer *is any circuit of the form $\mathtt{C}_u(L)$ for a control word u.*

If a circuit C is written as a sequential composition of controlled layers $C = L_m \circ \cdots \circ L_1$, where $L_t = \mathtt{C}_{u_t}(id_{p_t} \otimes G_t \otimes id_{q_t})$, its control-depth multiset *is $\mathtt{cd}(C) := \{|u_t| \mid t = 1, \ldots, m\}$. We order such multisets by the usual multiset extension of the natural order on $\mathbb{N}$. The empty sequential product is allowed, denotes the identity at the ambient arity, and has empty control-depth multiset.*

The procedure first removes implicit swaps, then unfolds derived one-qudit notation until only adjacent $H^{(r,r+1)}$ gates and controlled phases remain, then serialises $\otimes$, and finally pushes controls down to generators using the bounded-arity axioms of QC_d.

▶ **Lemma E.2.** SEPARATE *terminates on all inputs f and returns a possibly empty sequential composition* SEPARATE$(f) = L_m \circ \cdots \circ L_1$*, where each L_t is a controlled layer. Equivalently, after expanding the derived notation, every non-structural factor is one of the generators in the menu $\mathcal{G}$, placed in a structural context and equipped with an explicit control word.*

Proof. We use a staged well-founded induction. First remove explicit structural swaps by the swap decomposition (S); after this step, swaps contribute only fixed finite products of controlled X-layers and do not reappear as recursive inputs. Next unfold the derived one-qudit notation $X^{(i,j)}$, $H^{(i,j)}$, and $R_x^{(i,j)}(\theta)$ using Appendix B, except that the adjacent Hadamards $H^{(r,r+1)}$ are left as generators. This is a finite definitional expansion into adjacent Hadamards $H^{(r,r+1)}$, controlled phases, composition, and identities. The remaining recursion is ordered lexicographically by $\mu(f) := (s(f), b(f))$, where $s(f)$ is the number of constructor nodes $\circ, \otimes, \mathtt{C}_k$ in the raw syntax of f, and $b(f)$ is the number of occurrences $\mathtt{C}_u(C)$ in which the controlled body C is not one of the layer cores listed in Definition E.1. The order is the usual lexicographic order on $\mathbb{N}^2$.

The clauses for $f_1 \otimes f_2$, $f_1 \circ f_2$, $\mathtt{C}_m(g \circ h)$, and $\mathtt{C}_m(g \otimes h)$ call SEPARATE only on proper syntactic subcircuits, possibly after adding the same outer control to one of them. In each case the composition or tensor node under inspection has disappeared, so s strictly decreases. A top-level identity returns the empty product at its arity.

The remaining clauses are those where a control has reached a generator or a nested control. The phase and doubly controlled π-phase cases are already controlled layers and return immediately. The controlled identity case returns the empty product at the ambient arity, using $\mathtt{C}_k(\mathrm{id}_n) \equiv \mathrm{id}_{1+n}$ and the PROP unit laws. Since all derived Hadamards except the adjacent generators were unfolded in the first stage, the only Hadamard generator that can occur here is adjacent. The controlled-adjacent-Hadamard, controlled phase, and triply controlled phase clauses use the derived decompositions (CHD), (P2D), and (P3D). These identities are oriented here as layer expansions: inspecting their right-hand sides shows a finite sequential product of controlled layers whose active cores all lie in $\mathcal{G}$. This removes the offending occurrence counted by b, and no new occurrence with the same controlled body is created.

In the final nested-control "otherwise" branch, the inner circuit is first separated by the induction hypothesis. Functoriality of $\mathtt{C}_m$ then turns $\mathtt{C}_m(\text{SEPARATE}(f_1))$ into a sequential product of controls applied to already separated layers. Each resulting recursive call is either one of the layer cases just discussed or has strictly smaller b, because the non-layer body f_1 has been replaced by its finite product of layer cores. The measure therefore decreases in every recursive branch.

Every rewrite used in this procedure is either a definitional expansion from Appendix B or an equation of QC_d: structural coherence, the control-functor laws, the swap decomposition, or one of the three displayed layer-expansion lemmas. The returned circuit is therefore provably equal to the input and, by construction, is a finite sequential product of controlled $\mathcal{G}$-layers in structural context, possibly empty for an identity. ◀

▶ **Corollary E.3.** *Every morphism of* $\mathbf{CQC}_d$ *is provably equal in* QC_d *to a circuit generated, using sequential and parallel composition together with structural identities, by the reduced family* $\mathcal{G}$*: phases* (θ) *for* $\theta \in \mathbb{R}$*, adjacent Hadamards* $-\boxed{H^{(r,r+1)}}-$ *for* $0 \le r < d-1$*, controlled phases* $\mathtt{C}_k((\theta))$*, and doubly controlled* π*-phases* $\mathtt{C}_k(\mathtt{C}_\ell((\pi)))$*.*

▶ **Lemma E.4.** *Let* $a \neq b$*, let* $f : n \to n$*, and put* $F_{ab} := \mathtt{C}_a(\mathtt{C}_b(f))$*,* $F_{ba} := \mathtt{C}_b(\mathtt{C}_a(f))$*, and* $S := \sigma_{1,1} \otimes \mathrm{id}_n$*. Then* $\mathrm{QC}_d \vdash \textsc{Separate}(F_{ab}) \circ S = S \circ \textsc{Separate}(F_{ba})$*.*

Proof. We inspect the recursion defining Separate and argue by induction on the termination measure used in Lemma E.2. The induction invariant is slightly more general: if a recursive call is made under the two distinguished outer controls a, b, then exchanging those two controls before running the same recursive clause exchanges the two controls in every produced layer and preserves the sequential order of the produced layers.

The clauses for composition and tensor are functorial. For $\mathtt{C}_m(g \circ h)$, Separate produces the separated g-part followed by the separated h-part on both sides, so the induction hypotheses for the two subcalls compose in the same order. For $\mathtt{C}_m(g \otimes h)$, the procedure serialises tensor by inserting only target-wire symmetries and identity padding. These structural maps do not involve the two distinguished outer control values; sliding S past them is strict symmetric-monoidal naturality, and the two recursive subcalls are handled by induction.

The identity and phase-generator clauses are immediate. The clauses that expand controlled adjacent Hadamards, controlled phases, and triply controlled phases use the derived decompositions (CHD), (P2D), and (P3D). These decompositions are closed under adding surrounding controls and under target-wire permutation, so the expansion obtained after exchanging the two outer controls is exactly the exchanged expansion of the original call; the induction hypothesis applies to the recursive calls created by these expansions.

It remains to consider the one branch where Separate deliberately leaves a nested control undecomposed: a doubly controlled scalar π-phase. If the layer involves at most one of the distinguished outer controls, S passes it by structural naturality. If the layer is supported on both distinguished controls, it has the form $\mathtt{C}_a(\mathtt{C}_b((\pi)))$ in structural context on one side and $\mathtt{C}_b(\mathtt{C}_a((\pi)))$ in the corresponding context on the other. The required swap-through step is precisely the compatibility axiom (CoP), tensored with identities and conjugated by the surrounding target-wire symmetry.

Every layer produced by Separate falls under one of the cases above. Pushing S through the finite separated product from right to left therefore transforms the layers of $\textsc{Separate}(F_{ab})$ into the corresponding layers of $\textsc{Separate}(F_{ba})$, in the same sequential order, giving the stated derivation. ◀

E.2 Compatibility of mixed controls

Mixed-control compatibility now follows from the finite axiom system QC_d, using the normalisation machinery above.

Proof of Theorem 3.9. Write $S := \sigma_{1,1} \otimes \mathrm{id}_n : 2+n \to 2+n$. Consider the two $(2+n)$-wire circuits $F_{ab} := \mathtt{C}_a(\mathtt{C}_b(f))$ and $F_{ba} := \mathtt{C}_b(\mathtt{C}_a(f))$. If $a = b$, the desired equality $F_{aa} \circ S = S \circ F_{aa}$ is exactly the same-control nested-swap coherence included in the structural congruence of $\mathbf{CQC}_d$. Hence assume $a \neq b$ for the rest of the proof.

Apply the normalisation procedure Separate (Algorithm 1) to F_{ab} and F_{ba}. By construction, each rewrite step performed by Separate is justified either by the definitional expansions of Appendix B, by strict PROP coherence, by control-functoriality/strength/naturality, or by one of the bounded-arity axiom schemata of Figures 2–3 or the derived layer-expansion rules cited above. Therefore $\mathrm{QC}_d \vdash F_{ab} = \textsc{Separate}(F_{ab})$ and $\mathrm{QC}_d \vdash F_{ba} = \textsc{Separate}(F_{ba})$.

By Lemma E.4, $\mathrm{QC}_d \vdash \textsc{Separate}(F_{ab}) \circ S = S \circ \textsc{Separate}(F_{ba})$. Finally, replacing $\textsc{Separate}(F_{ab})$ by F_{ab} and $\textsc{Separate}(F_{ba})$ by F_{ba} using the equalities above yields $F_{ab} \circ S = S \circ F_{ba}$, as required. ◀

F Commutativity

Fix $d \geq 2$ and the polycontrolled PROP $\mathbf{CQC}_d$ of §2. Theorem 3.10 says that circuits controlled on *distinct* computational-basis values commute.

As in Appendix E, we work modulo strict symmetric monoidal coherences and the coherence laws of each control functor. We use the controlled-layer normal form of Lemma E.2 to reduce the commutativity statement to a finite list of generator-level commutations.

Proof of Theorem 3.10. By Lemma E.2, the proof reduces to commuting two placed controlled $\mathcal{G}$-layers; repeating these pairwise swaps commutes the finite products returned by Separate, by induction on the two product lengths. By SMC coherence we may therefore assume that $f = id_p \otimes f' \otimes id_{n-n_f-p}$ and $g = id_q \otimes g' \otimes id_{n-n_g-q}$, with $f' : n_f \to n_f$ and $g' : n_g \to n_g$ belonging to $\mathcal{G}$, and where $0 \leq p \leq q$. Identity factors have already been erased as empty products in Lemma E.2, so neither local core is an identity.

There is one point where the shared control wire must be kept visible. If the target windows of f' and g' are disjoint, the two controlled layers do not commute by plain tensor interchange: they still test the same outer control wire, on two distinct branch values. Instead, we close any irrelevant gap between the two target windows by target-wire symmetries. Control naturality for target permutations moves those symmetries through the outer controls, so a commutation with separated target windows is conjugate to the same commutation with adjacent target windows. Scalar cores have $n_f = 0$ (or $n_g = 0$) and are placed at the edge of the same local window. Thus it is enough to check the finite adjacent-window offsets recorded in the table below: $\Delta = 0$ for coincident left edge or scalar-core cases, $\Delta = 1$ for the first shifted one-wire cases, and $\Delta = 2$ for the first shifted two-wire cases. Larger target gaps are obtained from these by tensoring identities and undoing the target permutation.

The remaining possibilities are summarised in the following table. Each non-trivial entry cites either a generator commutation from Section 3.3 or a derived commutation proved in Appendix D. The label *sym.* means that the transposed generator pair is handled by the same cited equality, used in the opposite direction after exchanging the names of the two layers and tensoring with identities on untouched wires. This does not conflict with the convention $p \leq q$: that convention only fixes the left edge of the local window, while the cited diagrammatic equations are equalities in the symmetric monoidal theory and may be reflected by coherence. The symbols i, j, k, ℓ inside the table are local level/control indices of the displayed cores, while Δ is the relative offset of the two placed cores. A dash means that after the gap-closing reduction and, if necessary, exchanging the names of the two layers, the case is one of the displayed non-dash entries.

	$f' = (\theta)$	$f' = \mathtt{C}_i((\theta))$	$f' = \mathtt{C}_i(\mathtt{C}_j((\pi)))$	$f' = -\boxed{H^{(r,r+1)}}-$
$g' = (\gamma)$	$\Delta = 0$: (BPP)	$\Delta = 0$: (P2P)	$\Delta = 0$: (N2P)	$\Delta = 0$: (H2P)
$g' = \mathtt{C}_k((\gamma))$	$\Delta = 0$: (P2P) *sym.* –	$\Delta = 0$: (P2Ps) $\Delta = 1$: (P2PD)	$\Delta = 0$: (P2N2) $\Delta = 1$: (N2PD)	$\Delta = 0$: (BHP) $\Delta = 1$: (P2HD)
$g' = \mathtt{C}_k(\mathtt{C}_\ell((\pi)))$	$\Delta = 0$: (N2P) *sym.* – –	$\Delta = 0$: (P2N2) *sym.* $\Delta = 1$: (N2PD) *sym.* –	$\Delta = 0$: (N2N2) $\Delta = 1$: (N2N3) $\Delta = 2$: (N2N4)	$\Delta = 0$: (BHπ) $\Delta = 1$: (N2H2) $\Delta = 2$: (N2HD)
$g' = -\boxed{H^{(r,r+1)}}-$	$\Delta = 0$: (H2P) *sym.* –	$\Delta = 0$: (BHP) *sym.* $\Delta = 1$: (P2HD) *sym.*	$\Delta = 0$: (BHπ) *sym.* $\Delta = 1$: (N2H2) *sym.*	$\Delta = 0$: (BHH) $\Delta = 1$: (H2HD)

The Hadamard row and column use only adjacent instances $H^{(r,r+1)}$, as required by $\mathcal{G}$. Each cited lemma states precisely that the corresponding pair of controlled generators commutes for the indicated offset Δ. Tensoring with identities on the untouched wires and inserting coherence isomorphisms as needed yields $\mathtt{C}_k(f) \circ \mathtt{C}_\ell(g) = \mathtt{C}_\ell(g) \circ \mathtt{C}_k(f)$ for each of the finitely many generator pairs. Repeating these swaps over the normal forms of the two circuits gives the general case. ◀

G Exhaustivity

The exhaustivity component of Theorem 3.11 uses the two control-algebra principles already established in Appendices E and F. Semantically it is the familiar partition-of-the-identity calculation on the control wire; the syntactic proof must also show that this calculation survives composition, tensoring, and further controls.

We use the convention $\prod_{k=0}^{m} f_k = f_m \circ \cdots \circ f_0$. For a circuit f, write $\mathsf{E}(f)$ for the equation $\prod_{k=0}^{d-1} \mathtt{C}_k(f) = — \otimes f$. The proof is arranged so that the source of each step is visible: the totalisation support rules give the generators, while commutativity and compatibility propagate the equation through the circuit constructors.

▶ **Lemma G.1.** *The property* $\mathsf{E}(f)$ *holds when* f *is the monoidal unit, a single wire, a scalar phase, or an adjacent two-level Hadamard.*

Proof. For the unit and one-wire identity, the control functor laws give $\prod_{k=0}^{d-1} \mathtt{C}_k(\,[\,]\,) = — = — \otimes [\,]$ and $\prod_{k=0}^{d-1} \mathtt{C}_k(—) = \mathrm{id}_2 = — \otimes —$. The general identity id_n then follows by repeated tensoring, using Lemma G.2.

The remaining primitive cases are exactly the two totalisation support rules. For a scalar phase, (ExP) gives $\prod_{k=0}^{d-1} \mathtt{C}_k((\theta)) = — \otimes (\theta)$. For an adjacent two-level Hadamard, the adjacent instance of (ExH) gives $\prod_{k=0}^{d-1} \mathtt{C}_k(-\boxed{H^{(r,r+1)}}-) = — \otimes -\boxed{H^{(r,r+1)}}-$. ◀

▶ **Lemma G.2.** *The property* E *is closed under sequential composition, tensor product, and adding one more control.*

Proof. We check the three constructors separately.

Sequential composition. Assume $\mathsf{E}(g)$ and $\mathsf{E}(h)$. Functoriality gives $\mathtt{C}_k(g \circ h) = \mathtt{C}_k(g) \circ \mathtt{C}_k(h)$. In the product over k, Appendix F allows the differently controlled factors to be collected into the two blocks belonging to g and h:

$$\begin{aligned}\prod_{k=0}^{d-1} \mathtt{C}_k(g\circ h) &= \prod_{k=0}^{d-1}\big(\mathtt{C}_k(g)\circ \mathtt{C}_k(h)\big)\\ &= \left(\prod_{k=0}^{d-1}\mathtt{C}_k(g)\right)\circ\left(\prod_{k=0}^{d-1}\mathtt{C}_k(h)\right)\\ &= (— \otimes g)\circ(— \otimes h) = — \otimes (g\circ h).\end{aligned}$$

Therefore $\mathsf{E}(g\circ h)$ holds.

Tensor product. Assume $\mathsf{E}(g)$ and $\mathsf{E}(h)$, and put $a := |g|$, $b := |h|$. Strength for unused wires and symmetry naturality express a controlled tensor as the serialised composite

$$\mathtt{C}_k(g\otimes h) = (\mathtt{C}_k(g)\otimes \mathrm{id}_b)\circ(— \otimes \sigma_{a,b})\circ(\mathtt{C}_k(h)\otimes \mathrm{id}_a)\circ(— \otimes\sigma_{b,a}).$$

Collecting the controlled g-parts and controlled h-parts, again using Appendix F, gives

$$\begin{aligned}\prod_{k=0}^{d-1}\mathtt{C}_k(g\otimes h) &= \left(\prod_{k=0}^{d-1}(\mathtt{C}_k(g)\otimes\mathrm{id}_b)\right)\circ(— \otimes\sigma_{a,b})\\ &\quad\circ\left(\prod_{k=0}^{d-1}(\mathtt{C}_k(h)\otimes\mathrm{id}_a)\right)\circ(— \otimes\sigma_{b,a})\\ &= (— \otimes g\otimes\mathrm{id}_b)\circ(— \otimes\sigma_{a,b})\circ(— \otimes h\otimes\mathrm{id}_a)\\ &\quad\circ(— \otimes\sigma_{b,a}) = — \otimes(g\otimes h).\end{aligned}$$

Therefore $\mathsf{E}(g\otimes h)$ holds.

Nested control. Assume $\mathsf{E}(g)$, fix $m\in\{0,\dots,d-1\}$, and write $S := \sigma_{1,1}\otimes\mathrm{id}_{|g|}$. Compatibility from Appendix E gives $\mathtt{C}_k(\mathtt{C}_m(g)) = S\circ\mathtt{C}_m(\mathtt{C}_k(g))\circ S$. Since $S^2 = \mathrm{id}$, the outer swaps survive only at the two ends of the product:

$$\begin{aligned}\prod_{k=0}^{d-1}\mathtt{C}_k(\mathtt{C}_m(g)) &= S\circ\left(\prod_{k=0}^{d-1}\mathtt{C}_m(\mathtt{C}_k(g))\right)\circ S\\ &= S\circ\mathtt{C}_m\left(\prod_{k=0}^{d-1}\mathtt{C}_k(g)\right)\circ S\\ &= S\circ\mathtt{C}_m(— \otimes g)\circ S = — \otimes\mathtt{C}_m(g).\end{aligned}$$

The second equality uses functoriality of $\mathtt{C}_m$ together with the product convention $\prod_{k=0}^{d-1} f_k = f_{d-1}\circ\cdots\circ f_0$, so the order of the k-indexed factors is preserved inside $\mathtt{C}_m$. The last equality is just the strength and coherence law for $\mathtt{C}_m$, with the two control wires exchanged back by S. Therefore $\mathsf{E}(\mathtt{C}_m(g))$ holds. ◀

Proof of Theorem 3.11. By the swap decomposition (S) and strict symmetric-monoidal coherence, it suffices to treat presentations built from the non-structural generators without taking swaps as primitive constructors. Lemma G.1 gives the base cases, and Lemma G.2 gives the induction steps for $\circ$, $\otimes$, and $\mathtt{C}_m$. Structural induction therefore proves $\mathsf{E}(f)$ for every such presentation.

The same argument applies to the fixed decomposition of each swap $\sigma_{m,n}$, so structural symmetries are covered as well. Therefore $\prod_{k=0}^{d-1}\mathtt{C}_k(f) = — \otimes f$ for every circuit $f\in\mathbf{CQC}_d$. ◀

▶ **Corollary G.3.** *For every $m \geq 0$ and every circuit f, the product over all control words $u \in [d]^m$, taken in lexicographic order, satisfies $\prod_{u\in[d]^m} \mathtt{C}_u(f) = \mathrm{id}_m \otimes f$.*

Proof. The case $m = 0$ is the empty control word. For the induction step, write each word as ku with $k \in [d]$ and $u \in [d]^m$. Functoriality of $\mathtt{C}_k$ and the induction hypothesis give $\prod_{u\in[d]^m} \mathtt{C}_k(\mathtt{C}_u(f)) = \mathtt{C}_k\left(\prod_{u\in[d]^m} \mathtt{C}_u(f)\right) = \mathtt{C}_k(\mathrm{id}_m \otimes f)$. Taking the product over k and applying Theorem 3.11 once more yields $\mathrm{id}_1 \otimes (\mathrm{id}_m \otimes f) = \mathrm{id}_{m+1} \otimes f$. ◀

▶ Remark G.4. Semantically, this says that the projectors $(|k\rangle\langle k|)_{k=0}^{d-1}$ on the control wire form a partition of the identity. The syntactic form above is the one used later when decoding Gray-code gadgets produces products over all control branches.

H Further derived identities

The identities in this appendix are derived after the control algebra of Section 3.3 is available. We write $\mathrm{QC}_d \vdash C_1 = C_2$ when the circuit identity $C_1 = C_2$ follows from the axioms of Figures 2–3.

H.1 Catalogue of derived rules

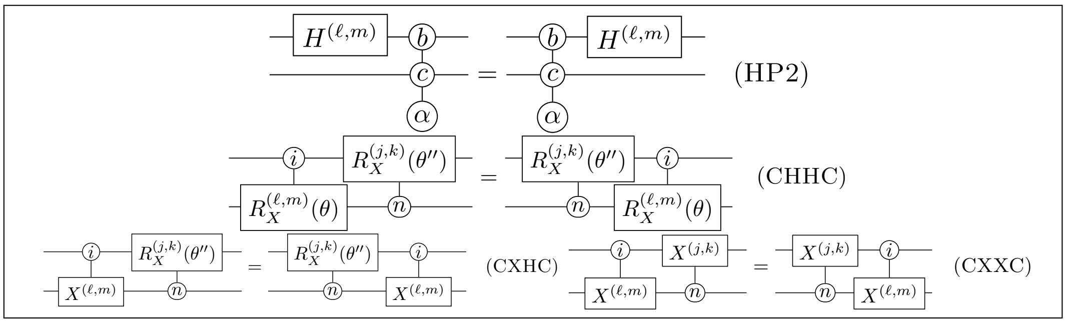


Figure 7 Derived rules in QC_d (batch 4). The rule (HP2) assumes that m, ℓ, b are pairwise distinct. The rules (CHHC), (CXHC), and (CXXC) hold when either i, j, k are pairwise distinct or ℓ, m, n are pairwise distinct.

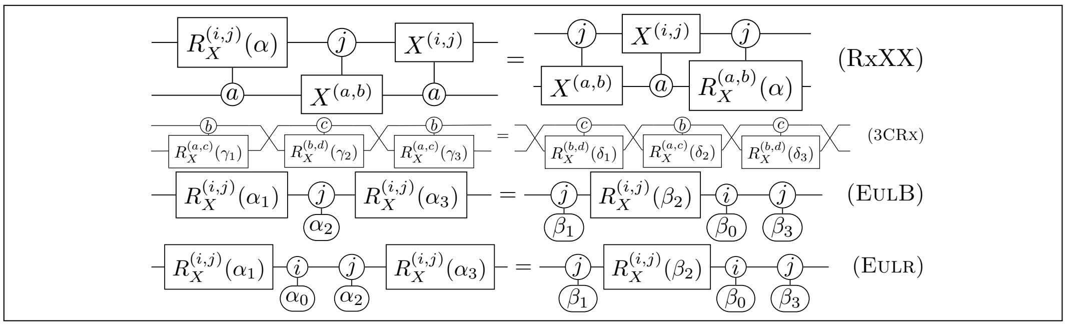
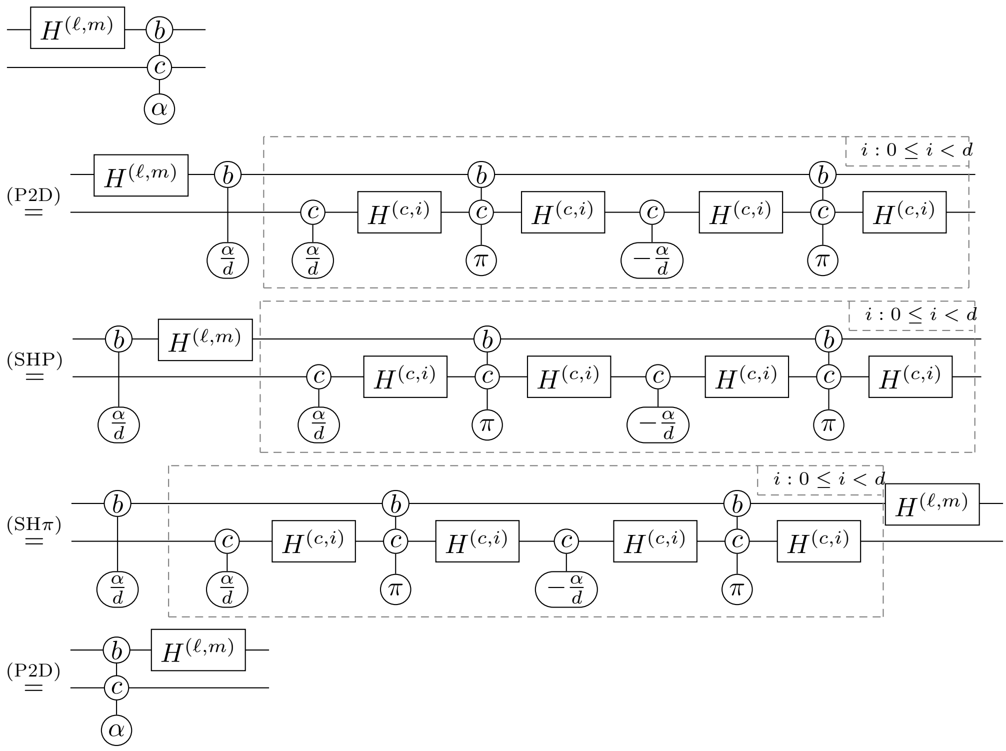


Figure 8 Derived rules in QC_d (batch 5).

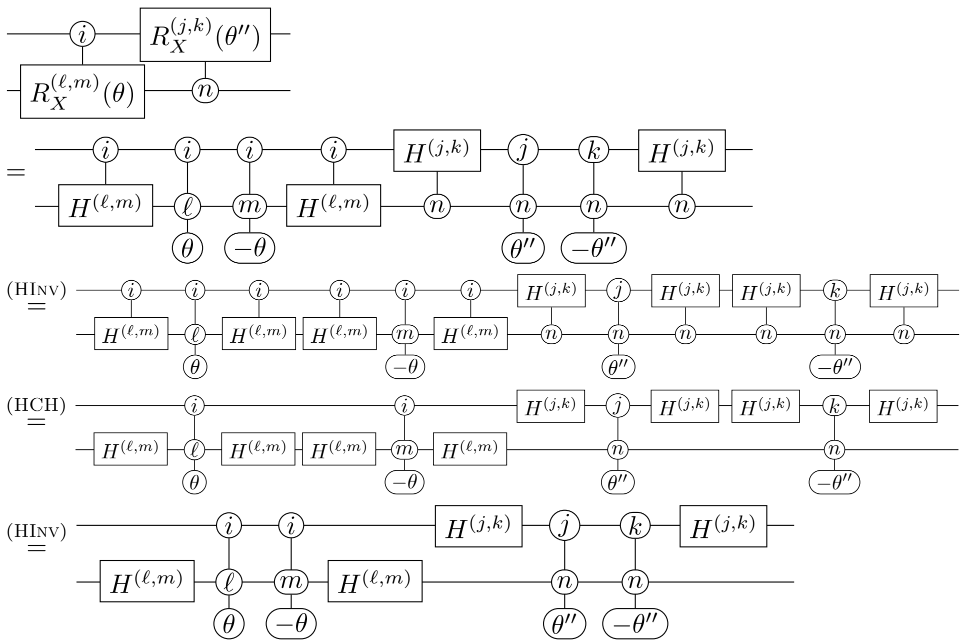
H.2 Proofs of the derived rules

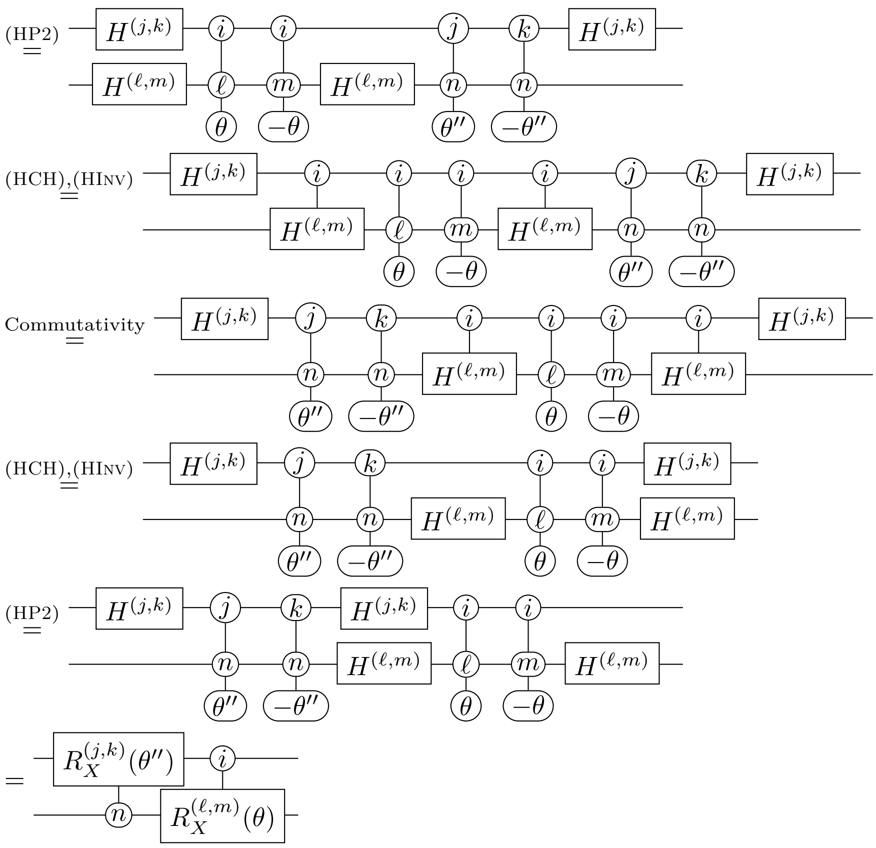
Proof of (HP2).



◀

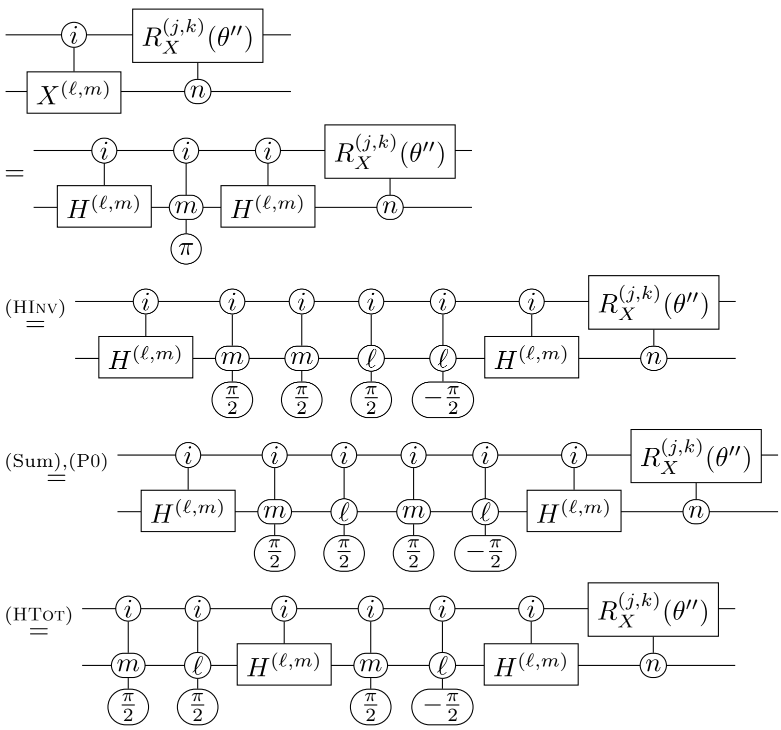
Proof of (CHHC). We treat the case where i, j, k are pairwise distinct; the case with ℓ, m, n is analogous.

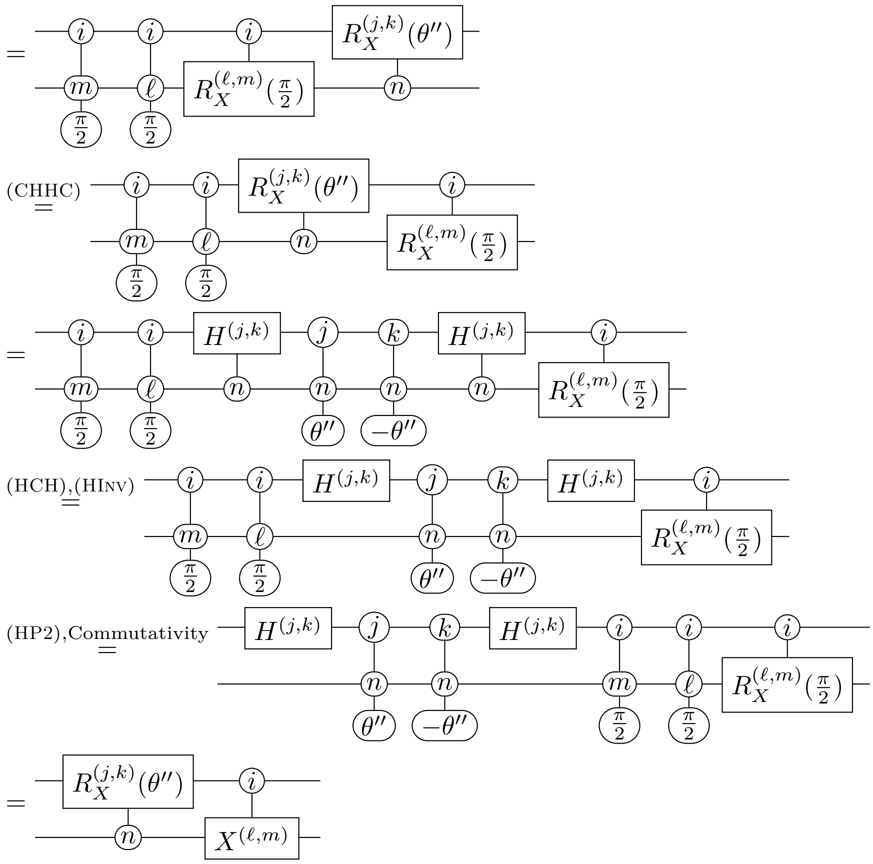




◀

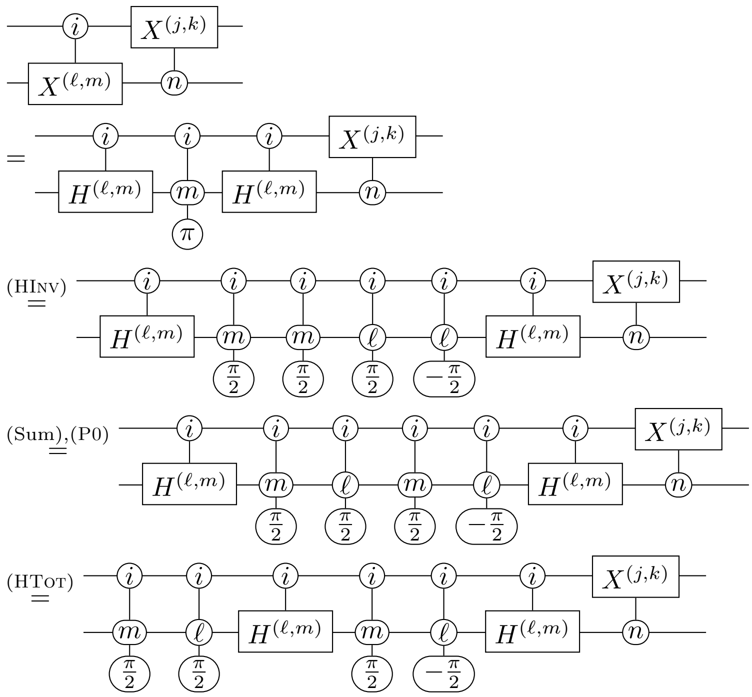
Proof of (CXHC)**.** Again consider i, j, k pairwise distinct; the case with ℓ, m, n is analogous.

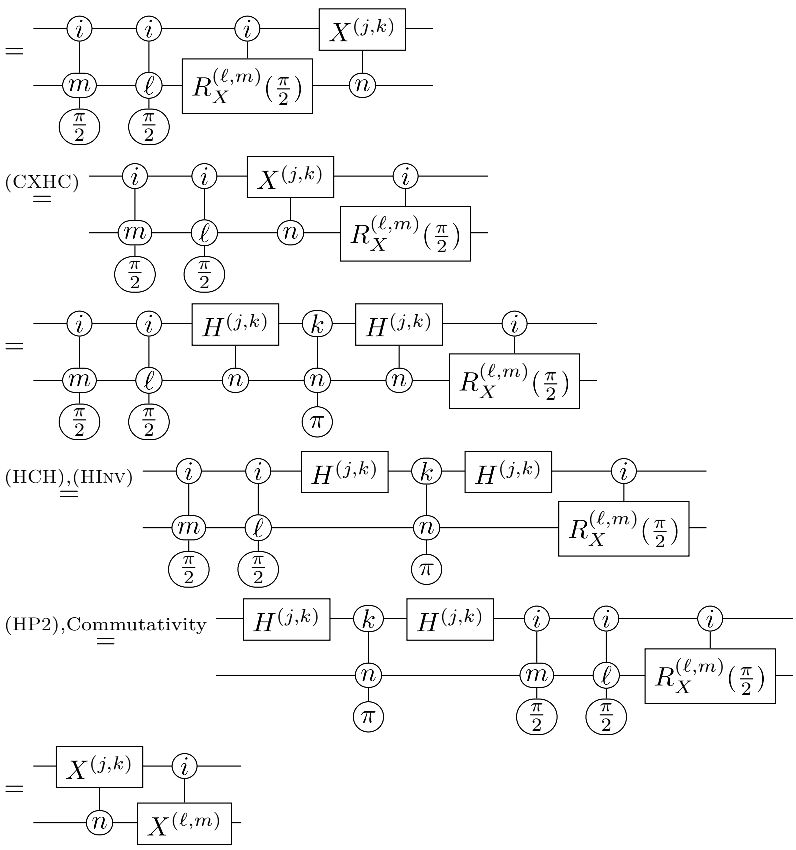




◄

Proof of (CXXC). For i, j, k pairwise distinct (the case with ℓ, m, n is similar),

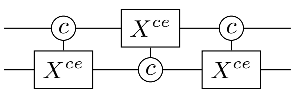




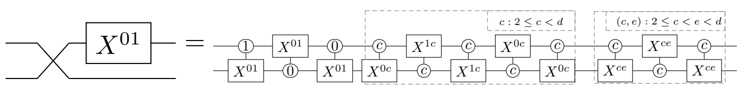
◀

▶ **Lemma H.1.** *On two wires we use the following three-gate T-block:*

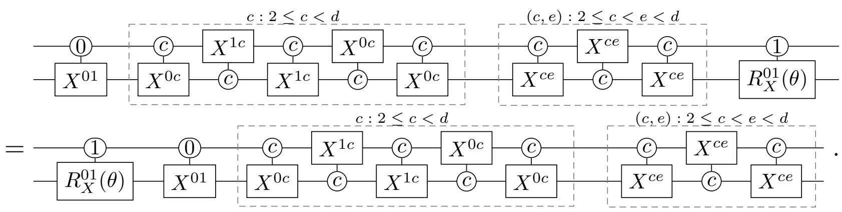
T_{ce}, *for* $2 \le c < e < d$



In this lemma, every single c-frame has $2 \le c < d$, and every T-frame has $2 \le c < e < d$. Then the following two identities are derivable:

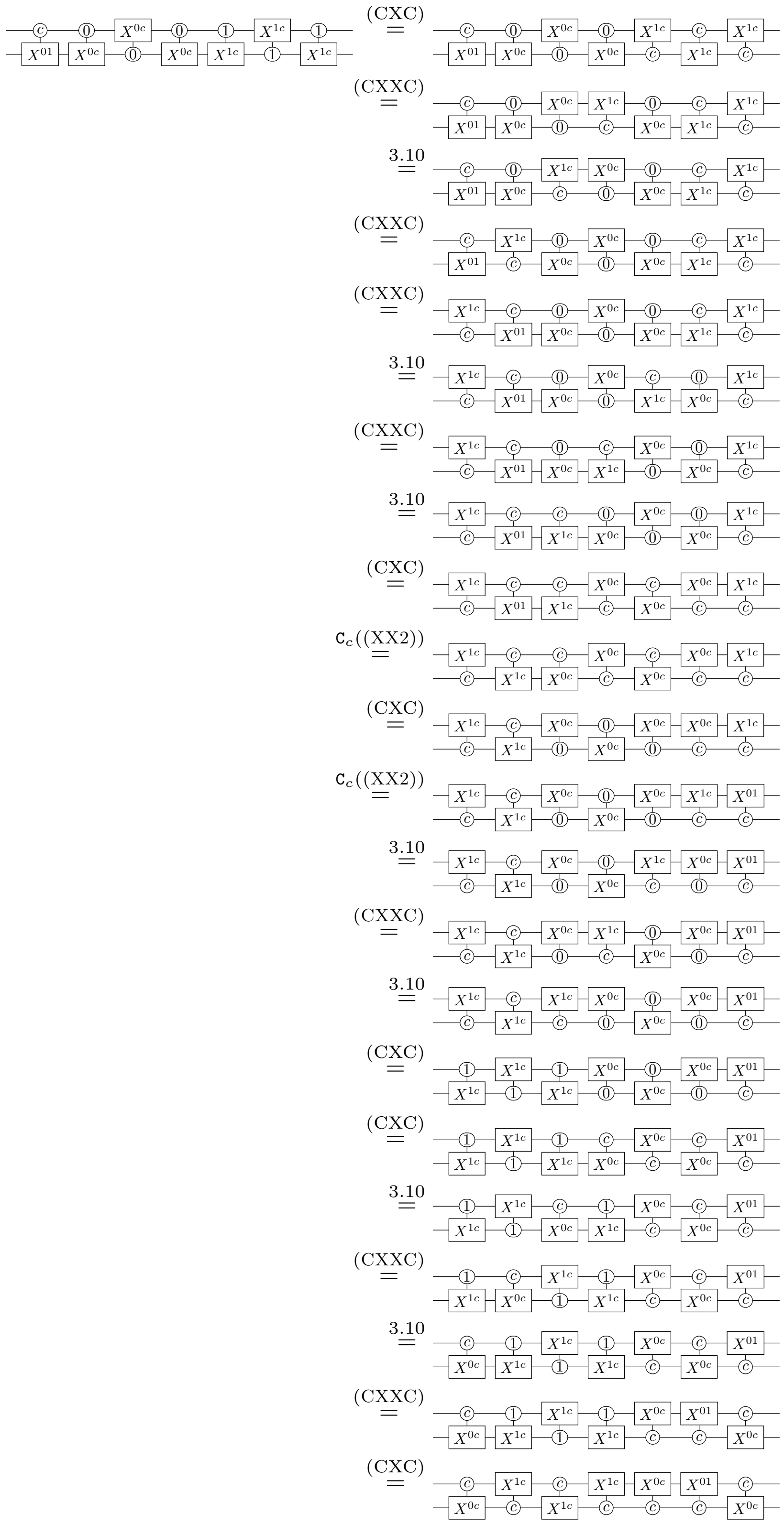


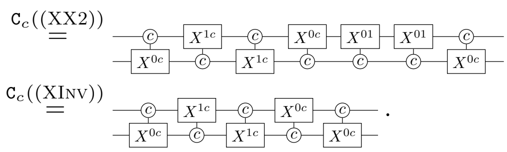
(G1a)



(G1b)

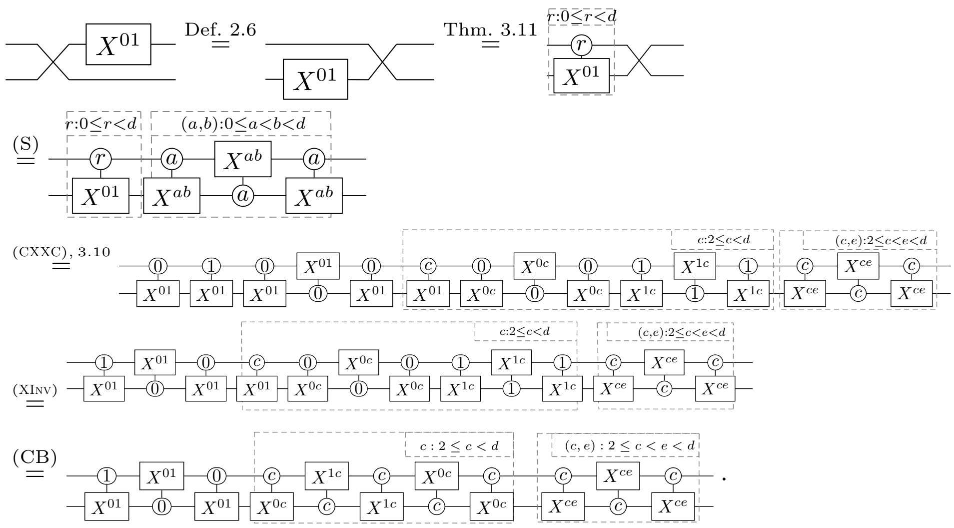
Proof. First we record the local c-block calculation in circuits. The condition $2 \le c < d$ is used throughout, so $0, 1, c$ are pairwise distinct.



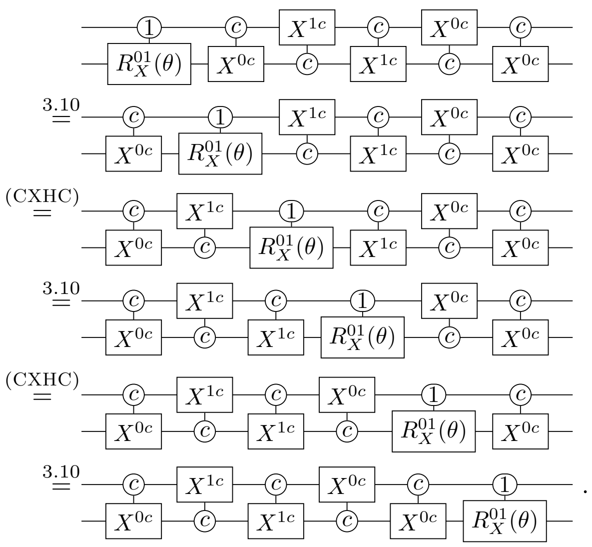


(CB)

For (G1a), use naturality of the structural symmetry before expanding anything. This avoids regrouping by interpretation: $\sigma_{1,1}$ carries the final X^{01} on the first wire to the same X^{01} on the second wire.

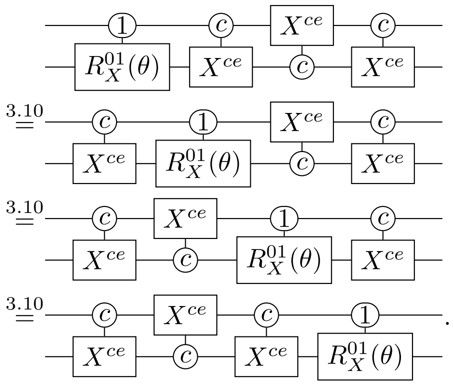


This proves (G1a).



Finally, when $2 \le c < e < d$, every level touched by T_{ce} is outside $\{0, 1\}$, so the three local moves are ordinary commutations. The rotation is moved past $C^c(X^{ce})$, then $C_c(X^{ce})$, then

$C^c(X^{ce})$; the control labels c, e are never 0 or 1:

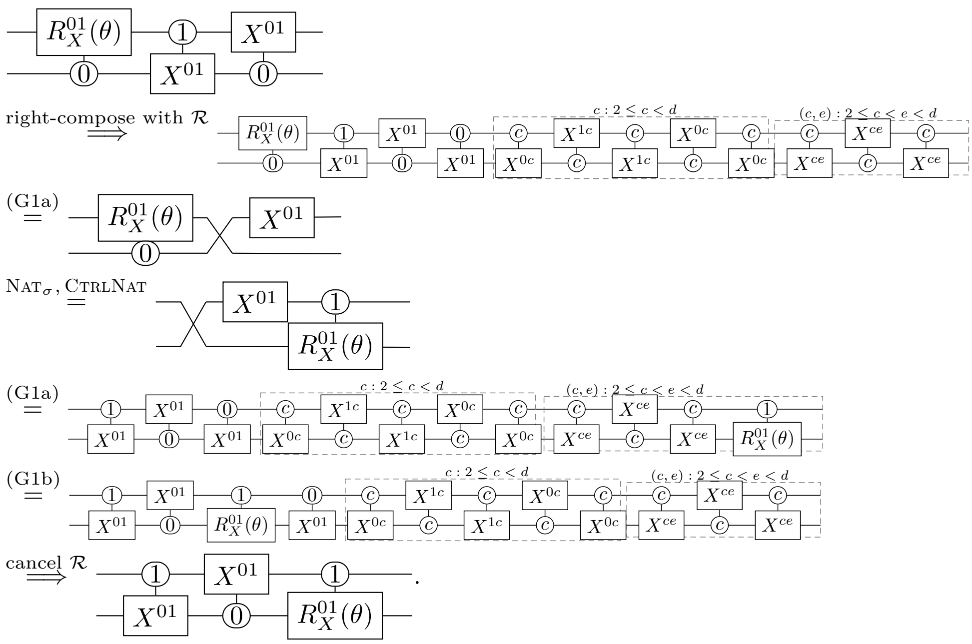


Applying these local slides inside the product frames, in their increasing lexicographic order, gives (G1b). The two steps labelled by (CXHC) are its vertical reflections: a first-wire controlled $R_X^{01}(\theta)$ commutes past a second-wire controlled X, with the support side condition supplied by the distinct levels $0, 1, c$. ◀

Proof of (RxXX)**.** Since the rule is schematic in the two levels, and using (RSym) to fix the orientation, it suffices to prove the instance with levels $0, 1$. Let $\mathcal{R}$ be the residual circuit

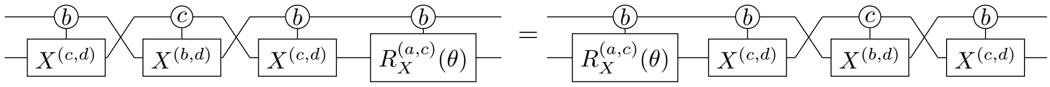
$$C^0(X^{01}); \prod_c (C^c(X^{0c}); C_c(X^{1c}); C^c(X^{1c}); C_c(X^{0c}); C^c(X^{0c})); \prod_{c<e} T_{ce},$$

namely the part of the right-hand side of (G1a) after $C^1(X^{01}); C_0(X^{01})$. This residual is reversible and equal to its own inverse read backwards. The derivation below starts with $C_0(R_X^{01}(\theta)); C^1(X^{01}); C_0(X^{01})$, right-composes with $\mathcal{R}$, moves through the factorisation and naturality steps, and then cancels the same residual at the end. The fourth equality uses (G1b) to move the rotation left through exactly $\mathcal{R}$. The first and last arrows below are not standalone rewrites: they indicate post-composition with $\mathcal{R}$ and, after the displayed equalities, post-composition with $\mathcal{R}^{-1}$:



The cancellation step is composing by $\mathcal{R}^{-1}$ on the right and using (XInv) factor by factor, in reverse order because circuits are read left to right. This is the $0, 1$ instance of (RxXX); relabelling the two levels gives the stated rule. ◀

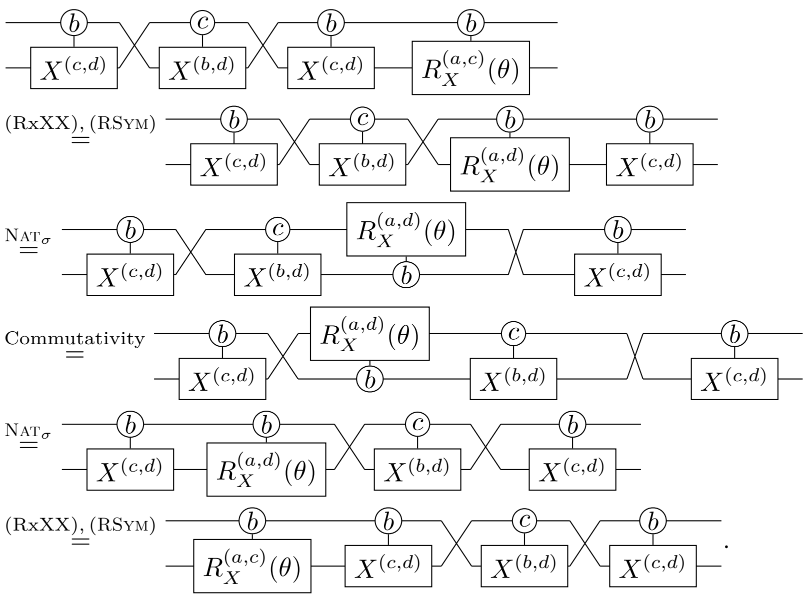
Proof of (3CRx). *Auxiliary equality* (PA).



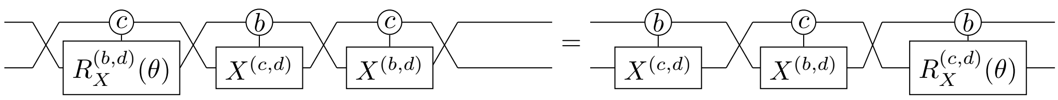
(PA)

In this derivation, the two Nat_σ steps are conjugations through a wire swap: the controlled rotation changes orientation when it passes through the swap, becoming a lower-controlled, upper-target gate, and changes back at the second swap. The two intermediate diagrams display this flipped orientation explicitly; the middle commutativity step then has disjoint support.

Derivation.

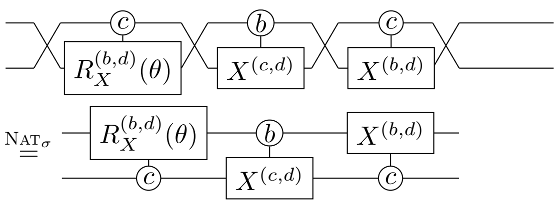


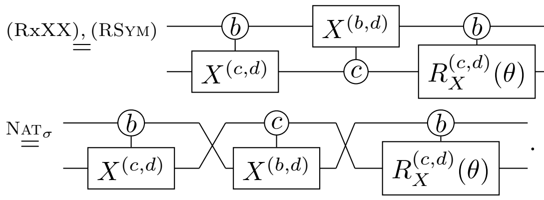
Auxiliary equality (BQ).



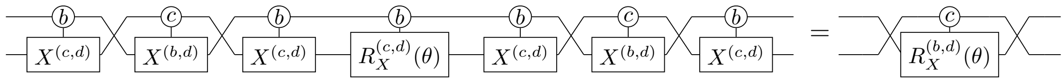
(BQ)

Derivation.



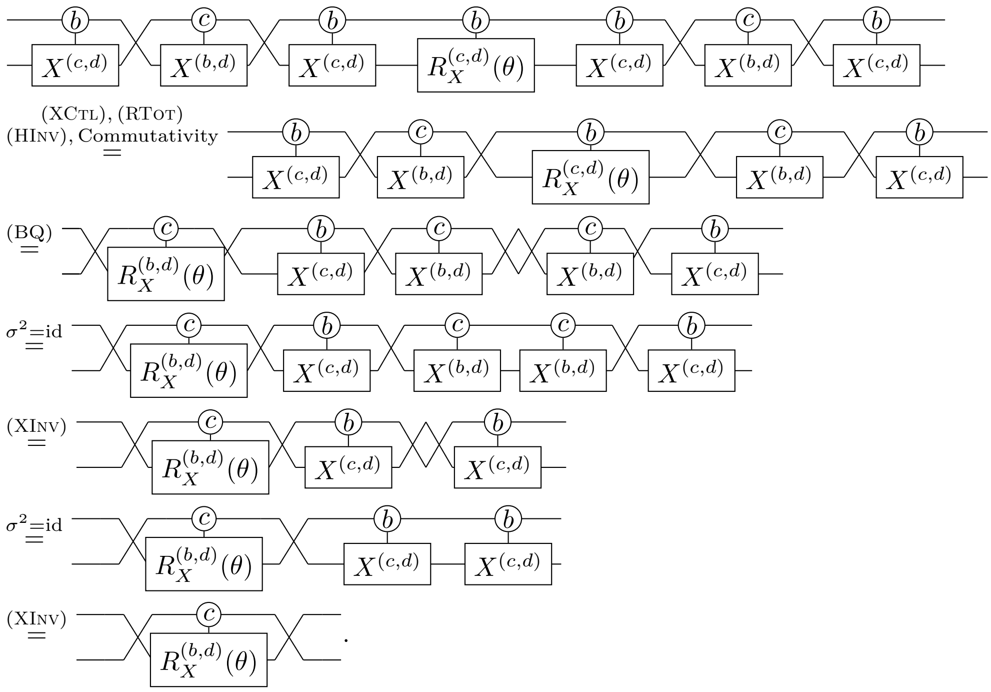


Auxiliary equality (PB).

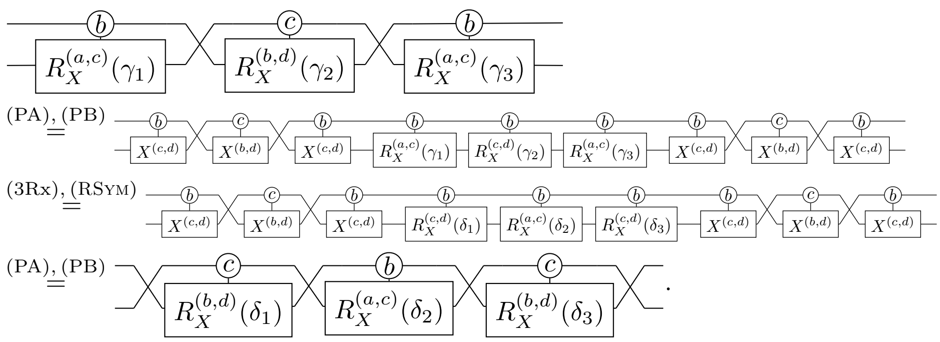


(PB)

Derivation.



Apply (PA) and (PB) to expose the uncontrolled three-rotation pattern, use (3Rx) together with the symmetry of R_X, and then reverse the same auxiliary equalities.



◀

The following π-normalisation lemma is used in the proof of (EulB).

We introduce an auxiliary real parameter x (by splitting a middle rotation/phase into two pieces x and $\alpha_2 - x$). This produces a *family* of circuit identities indexed by x, in which the angles $\beta_i(x)$ and $\gamma_i(x)$ produced by the Euler-extraction depend on x (while the input angles $\alpha_1, \alpha_2, \alpha_3$ are fixed).

To continue the purely diagrammatic part of the derivation, we need one additional property: we can choose a value x_0 and admissible representatives for the two (EH) decompositions such that the *bridge phase* $\beta_3(x_0) + \gamma_0(x_0)$ is a multiple of π. Since angles are always understood modulo 2π, this means that $\beta_3(x_0) + \gamma_0(x_0)$ is either 0 or π modulo 2π, which is exactly what justifies the two cases treated in the proof of (EulB).

The following analytic lemma is the normalisation argument needed for the Euler splitter.

▶ **Lemma H.2.** *Fix $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}$ and consider the continuous function $N : [-\frac{\pi}{2}, \frac{\pi}{2}] \to \mathbb{R}$ defined by $N(x) := \sin(\alpha_1)\cos(x)\cos(\alpha_3) + \cos(\alpha_1)\sin(\alpha_3)\cos(\alpha_2 - x)$. Then there exists $x_0 \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ such that $N(x_0) = 0$.*

In the Euler-extraction step used in the proof of (EulB)*, the two* (EH) *instances can be chosen from their admissible parameter families so that the corresponding bridge phase satisfies $\beta_3(x_0) + \gamma_0(x_0) \in \pi\mathbb{Z}$; hence (modulo 2π) we necessarily have $\beta_3(x_0) + \gamma_0(x_0) \equiv 0$ or $\equiv \pi$.*

Proof. First note that $N(-\pi/2) = \cos(\alpha_1)\sin(\alpha_3)\cos(\alpha_2 + \pi/2) = -\cos(\alpha_1)\sin(\alpha_3)\sin(\alpha_2)$, while $N(\pi/2) = \cos(\alpha_1)\sin(\alpha_3)\cos(\alpha_2 - \pi/2) = \cos(\alpha_1)\sin(\alpha_3)\sin(\alpha_2)$. Hence $N(-\frac{\pi}{2}) = -N(\frac{\pi}{2})$, so $N(-\frac{\pi}{2}) \cdot N(\frac{\pi}{2}) \le 0$. By continuity of N, the intermediate value theorem yields $x_0 \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ with $N(x_0) = 0$.

For the second part, use the notation of Appendix C for the two Euler extractions. Let z_β, z'_β be the quantities z, z' for the first extraction, with input angles $(-\alpha_1, x)$, and let z_γ, z'_γ be the corresponding quantities for the second extraction, with input angles $(\alpha_2 - x, -\alpha_3)$. Put $Z(x) := \big(\sin(\alpha_1)\cos(x) + i\cos(\alpha_1)\big)\big(\sin(\alpha_3)\cos(\alpha_2 - x) + i\cos(\alpha_3)\big)$. In the generic case where none of $z_\beta, z'_\beta, z_\gamma, z'_\gamma$ vanishes, the formulas for (EH) give

$$e^{i(\beta_3(x)+\gamma_0(x))} = \frac{z_\beta\,\overline{z'_\beta}\,z_\gamma\,z'_\gamma}{\left|z_\beta\,\overline{z'_\beta}\,z_\gamma\,z'_\gamma\right|}.$$

The two products in the numerator reduce to

$$z_\beta\,\overline{z'_\beta} = \sin(\alpha_1)\cos(x) + i\cos(\alpha_1),$$
$$z_\gamma\,z'_\gamma = \sin(\alpha_3)\cos(\alpha_2 - x) + i\cos(\alpha_3).$$

Thus the numerator is $Z(x)$, and

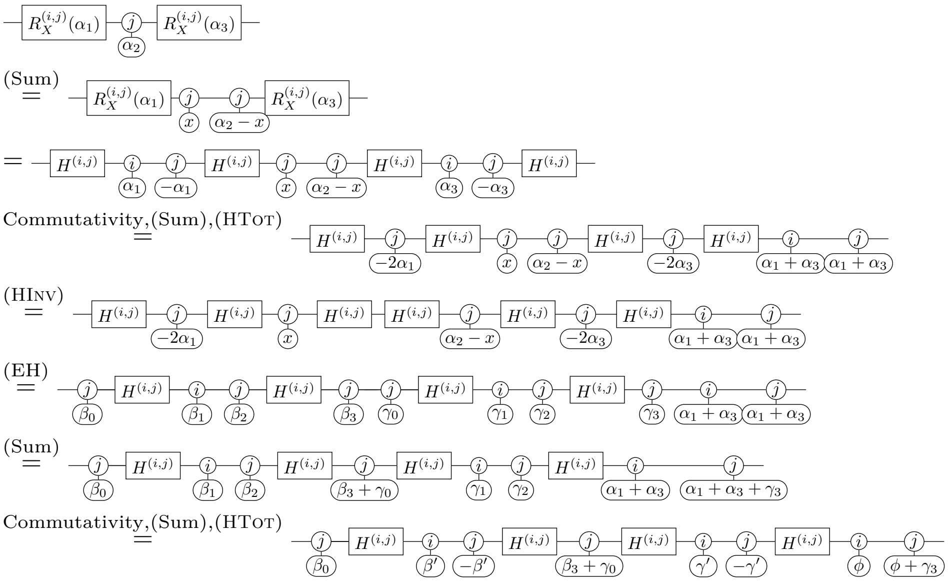
$$\tan(\beta_3(x) + \gamma_0(x)) = \frac{\sin(\alpha_1)\cos(x)\cos(\alpha_3) + \cos(\alpha_1)\sin(\alpha_3)\cos(\alpha_2 - x)}{\sin(\alpha_1)\sin(\alpha_3)\cos(x)\cos(\alpha_2 - x) - \cos(\alpha_1)\cos(\alpha_3)}.$$

The numerator is exactly $N(x)$. If $Z(x_0) \neq 0$, then $N(x_0) = 0$ implies that $Z(x_0)$ is a non-zero real number. Hence $Z(x_0)/|Z(x_0)| \in \{1, -1\}$, so the bridge phase is a multiple of π.

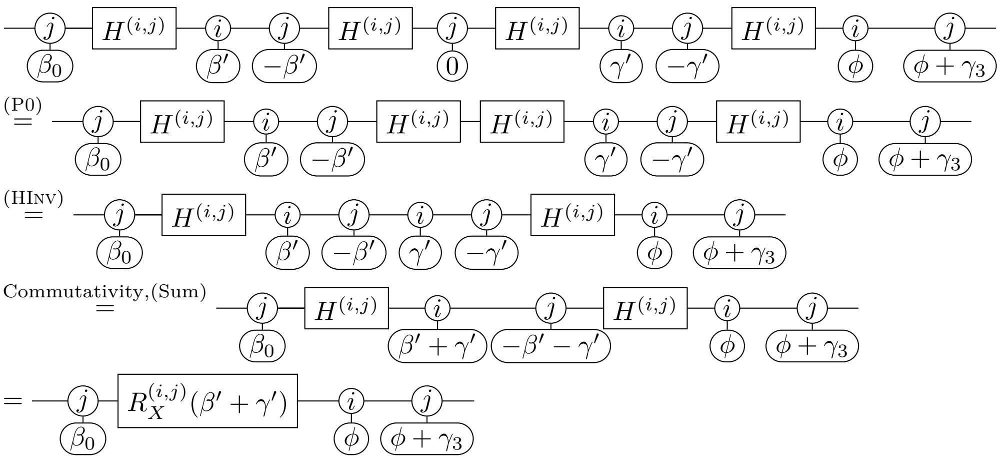
It remains to consider the case $Z(x_0) = 0$. Then at least one of $z_\beta, z'_\beta, z_\gamma, z'_\gamma$ is zero, so at least one of the two (EH) extractions is degenerate. By the admissible degenerate convention of Appendix C, the endpoint phase β_3 in a degenerate first extraction, or the endpoint phase γ_0 in a degenerate second extraction, may be prescribed arbitrarily while staying in the same sound (EH) parameter family. Choose that free endpoint phase so that $\beta_3(x_0) + \gamma_0(x_0) \in \pi\mathbb{Z}$. Reducing modulo 2π gives the claimed dichotomy $\equiv 0$ or $\equiv \pi$. ◀

▶ Remark H.3. The lemma only provides *existence* (not uniqueness) of a suitable x_0, and any such choice is valid for the subsequent diagrammatic derivation. This is the only point in the proof where we appeal to a small analytic argument; once x_0 is fixed, the remainder proceeds purely by the equational rules of QC_d.

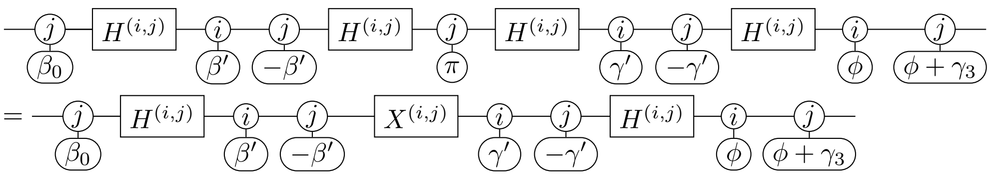
Proof of (EulB).

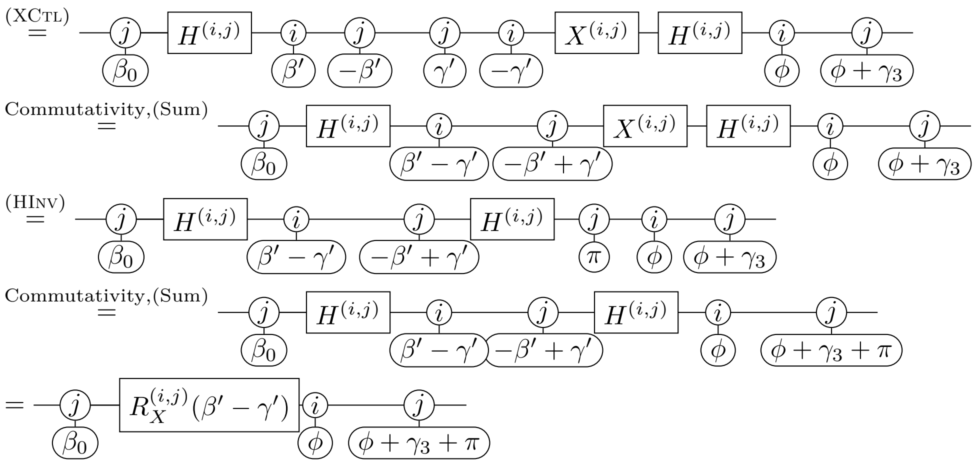


Choose x_0 and the admissible (EH) representatives as in Lemma H.2, so that $\beta_3(x_0) + \gamma_0(x_0) \in \pi\mathbb{Z}$. Working modulo 2π, there are two cases: (i) $\beta_3(x_0) + \gamma_0(x_0) \equiv 0$, and (ii) $\beta_3(x_0) + \gamma_0(x_0) \equiv \pi$. We treat these cases separately below.
Case $\beta_3 + \gamma_0 = 0$.



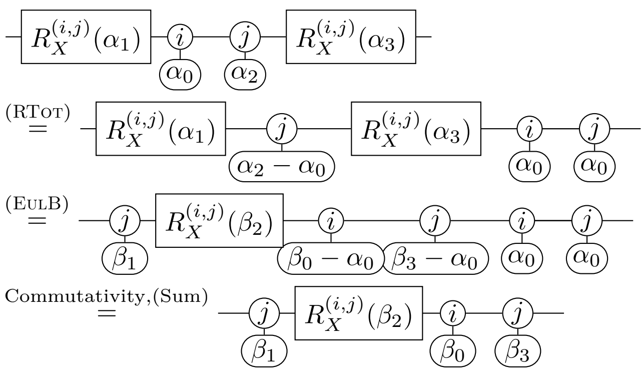
Case $\beta_3 + \gamma_0 = \pi$.





◀

Proof of (Eulr).



◀

I Details of the Gray-code lifts

The encoding of Section 4.2 uses two raw-**LOPP** constructions:

1. a d-mode gadget $B_d^{(i,i+1)}$ whose single-photon semantics is the adjacent two-level Hadamard $H_d^{(i,i+1)}$;
2. the lift operators $\mathrm{Lift}_d^p(-)$ and $\mathrm{Lift}_{d,j}^p(-)$, which replicate a local subcircuit across Gray blocks, mirroring exactly on the reversed branches.

This appendix fixes explicit choices and the block combinatorics used later by decoding.

I.1 A single-qudit network for $H_d^{(i,i+1)}$

Fix $d \geq 2$ and $i \in \{0, \ldots, d-2\}$. We need a d-mode **LOPP** circuit whose single-photon semantics mixes levels i and $i+1$ and fixes the others.

Let $H := \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \in U(2)$ be the usual 2×2 Hadamard, and set $H_d^{(i,i+1)} := I_i \oplus H \oplus I_{d-i-2} \in U(d)$.

▶ **Definition I.1.** *Write $P(\phi) : 1 \to 1$ for the* **LOPP** *phase-shifter generator of angle ϕ, and $B(\theta) : 2 \to 2$ for the* **LOPP** *beam-splitter generator of angle θ (as in Definition 4.2). Set $H_{\mathrm{BS}} := (\mathrm{id}_1 \otimes P(-\pi/2)) \circ B(\pi/4) \circ (\mathrm{id}_1 \otimes P(-\pi/2)) : 2 \to 2$. The d-mode gadget placing this interferometer on modes $i, i+1$ is $B_d^{(i,i+1)} := \mathrm{id}_i \otimes H_{\mathrm{BS}} \otimes \mathrm{id}_{d-i-2} : d \to d$.*

Here $B(\pi/4)$ is the standard 50/50 beam splitter in our phase convention, and the surrounding phase shifters correct the relative phases so that the induced 2×2 matrix is exactly H.

▶ **Lemma I.2.** *For each $d \geq 2$ and $0 \leq i \leq d-2$, the circuit $B_d^{(i,i+1)}$ is a raw* **LOPP** *term built only from phase shifters, beam splitters, and identities, and its single-photon semantics satisfies $\left[\!\left[B_d^{(i,i+1)}\right]\!\right]_{\mathrm{sp}} = H_d^{(i,i+1)} \in U(d)$.*

Proof. Under $[\![-]\!]_{\mathrm{sp}}$, the phase shifter $P(\phi)$ acts as $[e^{\mathrm{i}\phi}]$ on the corresponding mode, so $[\![\mathrm{id}_1 \otimes P(\phi)]\!]_{\mathrm{sp}} = \mathrm{diag}(1, e^{\mathrm{i}\phi})$ on 2 modes. The beam splitter has matrix $[\![B(\theta)]\!]_{\mathrm{sp}} = \left(\begin{smallmatrix} \cos(\theta) & \mathrm{i}\sin(\theta) \\ \mathrm{i}\sin(\theta) & \cos(\theta) \end{smallmatrix}\right)$. For $\theta = \pi/4$ and $\phi = -\pi/2$ we have $\cos(\pi/4) = \sin(\pi/4) = 1/\sqrt{2}$ and $e^{\mathrm{i}\phi} = e^{-\mathrm{i}\pi/2} = -\mathrm{i}$, so $[\![H_{\mathrm{BS}}]\!]_{\mathrm{sp}} = \mathrm{diag}(1, -\mathrm{i}) \cdot \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & \mathrm{i} \\ \mathrm{i} & 1 \end{smallmatrix}\right) \cdot \mathrm{diag}(1, -\mathrm{i}) = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right) = H$. Since parallel composition in **LOPP** is interpreted as block-diagonal direct sum on disjoint mode blocks, we have $[\![\mathrm{id}_i \otimes H_{\mathrm{BS}} \otimes \mathrm{id}_{d-i-2}]\!]_{\mathrm{sp}} = I_i \oplus H \oplus I_{d-i-2} = H_d^{(i,i+1)}$. ◀

I.2 Alternating mirror symmetry on Gray-coded modes

The lift operators of Section 4.2 replicate a local **LOPP** circuit across Gray-code blocks. The only combinatorial point is that for a fixed prefix u, the words $u0, \ldots, u(d-1)$ form one contiguous Gray block, read either forward or backward according to the reflected recursion. The lifts account for this by mirroring precisely on the reversed branches.

▶ **Definition I.3.** *For each $r \geq 0$, let $\rho_r : 1^{\otimes d^r} \longrightarrow 1^{\otimes d^r}$ be the* **LOPP** *permutation circuit that reverses the order of the d^r modes: on the single-photon basis $\{|0\rangle, \ldots, |d^r - 1\rangle\}$ it acts by $\rho_r |k\rangle = |d^r - 1 - k\rangle$.*

For any raw **LOPP** *circuit $C : 1^{\otimes d^r} \to 1^{\otimes d^r}$, define its mirror by $\mathtt{Rev}(C) := \rho_r \circ C \circ \rho_r^{-1}$. Since ρ_r is involutive, $\rho_r^{-1} = \rho_r$.*

For $q \in [d]$, set $\mathtt{Rev}^q(C) := C$ if q is even, and $\mathtt{Rev}^q(C) := \mathtt{Rev}(C)$ if q is odd. This parity test matches the reflected recursion in the Gray code: odd leading digits reverse the traversal of the suffix.

▶ **Definition I.4.** *Fix $d \geq 2$, let $m \geq 0$, and let $C : 1^{\otimes d^m} \to 1^{\otimes d^m}$ be a raw* **LOPP** *circuit.*

- *The ordinary lift is defined by $\mathrm{Lift}_d^0(C) := C$ and $\mathrm{Lift}_d^{p+1}(C) := \bigotimes_{q=0}^{d-1} \mathtt{Rev}^q(\mathrm{Lift}_d^p(C))$. It has arity d^{m+p}.*
- *For $j \in [d]$, set $C^{(q)} := C$ if $q = j$ and $C^{(q)} := \mathrm{id}_{d^m}$ otherwise. The selected lift is defined by $\mathrm{Lift}_{d,j}^0(C) := \bigotimes_{q=0}^{d-1} \mathtt{Rev}^q(C^{(q)})$ and $\mathrm{Lift}_{d,j}^{p+1}(C) := \bigotimes_{q=0}^{d-1} \mathtt{Rev}^q(\mathrm{Lift}_{d,j}^p(C))$. It has arity d^{m+1+p}.*

At level $p+1$, the modes split into d consecutive blocks of size d^{m+p}; the previous lift is placed on each block and mirrored exactly when the new leading digit is odd. The selected lift does the same, but starts from one chosen copy of C and identities elsewhere.

The next lemma isolates the combinatorial feature that makes the lifts correct: fixing a prefix produces a contiguous block of indices, and within that block the last digit runs either forward or backward.

▶ **Lemma I.5.** *Fix $d \geq 2$ and $n \geq 0$. For each word $u \in [d]^n$, there exists an index $s(u)$ with $0 \leq s(u) \leq d^{n+1} - d$ and a sign $\varepsilon(u) \in \{+1, -1\}$ such that $\{\, t \in \{0, \ldots, d^{n+1} - 1\} \mid G_{n+1}^d(t) \text{ has prefix } u \,\} = \{s(u), s(u)+1, \ldots, s(u)+d-1\}$, and for every $r \in \{0, \ldots, d-1\}$, $G_{n+1}^d\big(s(u)+r\big) = u\,r$ if $\varepsilon(u) = +1$, while $G_{n+1}^d\big(s(u)+r\big) = u\,(d-1-r)$ if $\varepsilon(u) = -1$, where $u\,r$ denotes concatenation of the word u with the digit r.*

One may take ε to be the unique function $\varepsilon : [d]^* \to \{\pm 1\}$ defined by $\varepsilon(\epsilon) = +1$ and $\varepsilon(qu) = \varepsilon(u)$ if q is even and $\varepsilon(qu) = -\varepsilon(u)$ if q is odd. In particular, $\varepsilon(u)$ flips sign exactly once for each odd digit encountered when reading u from left to right.

Proof. We prove the block statement and the recursion for ε simultaneously by induction on n.

Base case $n = 0$: we have $u = \epsilon$ and $G_1^d(t) = (t)$ for $0 \le t < d$, so we may take $s(\epsilon) = 0$ and $\varepsilon(\epsilon) = +1$.

Inductive step: let $n \ge 1$ and write $u = qv$ with $q \in [d]$ and $v \in [d]^{n-1}$. Any $t \in \{0, \dots, d^{n+1} - 1\}$ can be written uniquely as $t = q \cdot d^n + r$ with $0 \le r < d^n$.

If q is even, then by Definition 4.4 we have $G_{n+1}^d(qd^n + r) = q \cdot G_n^d(r)$. Thus $G_{n+1}^d(qd^n + r)$ has prefix qv if and only if $G_n^d(r)$ has prefix v. By the induction hypothesis, the set of such r is an interval of length d, hence so is the set of such t, with the same last-digit direction. This corresponds to setting $\varepsilon(qv) = \varepsilon(v)$.

If q is odd, then $G_{n+1}^d(qd^n + r) = q \cdot G_n^d(d^n - 1 - r)$. Put $r' := d^n - 1 - r$. By the induction hypothesis, the set of r' for which $G_n^d(r')$ has prefix v is an interval $[a, a + d - 1]$. Its image under $r = d^n - 1 - r'$ is the interval $[d^n - d - a,\ d^n - 1 - a]$, again of length d. Since the map $r' \mapsto d^n - 1 - r'$ reverses order, the last digit is traversed in the opposite direction on this block; this corresponds to setting $\varepsilon(qv) = -\varepsilon(v)$. ◀

Lemma I.5 is exactly what the lift recursion needs: each fixed prefix determines one contiguous mode block, and the last digit runs through that block either forward or backward. Mirroring on the odd branches restores the same logical orientation in every block.

J Encoding and decoding

Theorem 4.15 states that for every $C : n \to n$ in $\mathbf{CQC}_d$ and every $k, \ell \ge 0$,

$$\mathrm{QC}_d \vdash \mathrm{D}^0_{k+n+\ell}(E^k_\ell(C)) = \mathrm{id}_k \otimes C \otimes \mathrm{id}_\ell.$$

Here E^k_ℓ is the contextual encoding of Definition 4.7, and D^t_n is the contextual decoding of Definition 4.10. Because $\otimes$ is tensor on qudits but direct sum on optics, both translations are context-sensitive and serialise tensor products. The proof first identifies the behaviour of D^t_n on one Gray block, then applies the same calculation to the lift gadgets, and finally concludes by structural induction on $\mathbf{CQC}_d$.

J.1 Decoding a fixed Gray-code block

Fix $a \ge 0$. In reflected d-ary Gray order, the d^{a+1} modes split into d consecutive blocks of length d^a, indexed by the leading Gray digit. On an even block, decoding is just the a-qudit decoding with an outer control; on an odd block, the same holds after mirroring the local circuit.

▶ **Lemma J.1.** *Let C be a* **LOPP** *circuit acting on ℓ consecutive modes, and fix $a \ge 0$ and s with $s + \ell \le d^a$. For every even $j \in \{0, \dots, d-1\}$ we have $\mathrm{D}^{d^a j + s}_{a+1}(C) = \mathtt{C}_j\big(\mathrm{D}^s_a(C)\big)$.*

Proof. We induct on the raw syntax of C. For even j, $G^d_{a+1}(d^a j + t) = j \cdot G^d_a(t)$ throughout the block. Hence the empty diagram and identity decode to $\mathrm{id}_{a+1} = \mathtt{C}_j(\mathrm{id}_a)$, and a phase shifter at local offset t decodes to $\mathtt{C}_j\big(\mathrm{D}^t_a(-\boxed{\theta}-)\big)$. The same factorisation holds for a beam splitter or swap on $t, t+1$: the two Gray labels keep the same leading digit j and differ only in the suffix part, so decoding produces exactly the a-qudit decoding with an extra outer j-control.

Sequential composition is immediate because D is compositional on $\circ$. For $C = C_1 \otimes C_2$, with C_1 on ℓ_1 modes, the serialisation clause gives

$$\mathrm{D}_{a+1}^{d^a j+s}(C) = \mathrm{D}_{a+1}^{d^a j+s+\ell_1}(C_2) \circ \mathrm{D}_{a+1}^{d^a j+s}(C_1),$$

and the induction hypotheses on C_1 and C_2 turn this into $\mathtt{C}_j(\mathrm{D}_a^s(C))$ by functoriality of $\mathtt{C}_j$. ◀

▶ **Lemma J.2.** *Let C be a circuit of* **LOPP** *acting on ℓ consecutive modes, and fix $a \geq 0$ and s with $s+\ell \leq d^a$. For every odd $j \in \{0, \dots, d-1\}$ we have $\mathrm{D}_{a+1}^{d^a j+s}(C) = \mathtt{C}_j\big(\mathrm{D}_a^{d^a-s-\ell}(\mathrm{Rev}_\ell(C))\big)$, where $\mathrm{Rev}_\ell(C)$ denotes the circuit obtained from C by reversing the order of its ℓ local modes. In the full-block case $\ell = d^a$, this is the mirror $\mathtt{Rev}(C)$ of Definition I.3.*

Proof. For odd j, the block is traversed in reverse: $G_{a+1}^d(d^a j + t) = j \cdot G_a^d(d^a - 1 - t)$. Thus a local interval of length ℓ starting at s is read as the mirrored interval starting at $d^a - s - \ell$. Replacing C by $\mathrm{Rev}_\ell(C)$ compensates for that reversal at the syntactic level, so the generator cases reduce to Lemma J.1 at the mirrored offset. For swaps, the reflected orientation is harmless because $X^{(r,s)} = X^{(s,r)}$. The composition and tensor cases are then identical to the even-block proof. ◀

The two orientation cases can be packaged as one decoding rule.

▶ **Corollary J.3.** *Let C be a* **LOPP** *circuit acting on ℓ consecutive modes, and fix $a \geq 0$, $j \in [d]$, and s with $s+\ell \leq d^a$. Define $s_j(s,\ell)$ to be s if j is even and $d^a - s - \ell$ if j is odd; define $R_j^\ell(C)$ to be C if j is even and $\mathrm{Rev}_\ell(C)$ if j is odd. Then $\mathrm{D}_{a+1}^{d^a j+s}(C) = \mathtt{C}_j\big(\mathrm{D}_a^{s_j(s,\ell)}(R_j^\ell(C))\big)$.*

Proof. If j is even this is Lemma J.1; if j is odd this is Lemma J.2. ◀

▶ **Lemma J.4.** *For every $d \geq 2$ and $0 \leq i \leq d-2$, $\mathrm{D}_1^0\big(B_d^{(i,i+1)}\big) = $* $-\boxed{H^{(r,r+1)}}-$.

Proof. For one qudit, the reflected Gray order is the ordinary order $0, 1, \dots, d-1$. The identity factors in $B_d^{(i,i+1)} = \mathrm{id}_i \otimes H_{\mathrm{BS}} \otimes \mathrm{id}_{d-i-2}$ therefore decode to identities, while the two phase shifters $P(-\pi/2)$ decode to the corresponding level phases on level $i+1$ and the beam splitter $B(\pi/4)$ decodes to the adjacent $R_x^{(i,i+1)}(\pi/4)$. By the definition of H_{BS}, the decoded circuit is exactly the adjacent instance of the Hadamard decomposition proved in (HDEC); hence it is equal to $-\boxed{H^{(r,r+1)}}-$ in QC_d. ◀

J.2 Decoding the lifts

The lift gadgets add Gray digits one layer at a time. Decoding an ordinary lift should therefore produce identity padding on the new qudits, while decoding a selected lift should produce the same padding together with one extra value-control.

▶ **Lemma J.5.** *Let C be a* **LOPP** *circuit acting on d^n modes. Then for every $k \geq 0$ the circuit $\mathrm{Lift}_d^k(C)$ acts on d^{n+k} modes.*

Proof. Induct on k. The case $k = 0$ is immediate. For the step, $\mathrm{Lift}_d^{k+1}(C) = \bigotimes_{q=0}^{d-1} \mathtt{Rev}^q(\mathrm{Lift}_d^k(C))$; each $\mathtt{Rev}^q$ preserves arity, so tensoring d copies of an d^{n+k}-mode circuit yields d^{n+k+1} modes. ◀

▶ **Lemma J.6.** *Let C be a* **LOPP** *circuit acting on d^n modes, and let $m \geq 0$. Then $\mathrm{D}_{n+m}^0\big(\mathrm{Lift}_d^m(C)\big) = \mathrm{id}_m \otimes \mathrm{D}_n^0(C)$ as circuits in* $\mathbf{CQC}_d$.

Proof. Induct on m. The case $m = 0$ is immediate. For the step, set $F := \mathrm{Lift}_d^m(C)$. Then $\mathrm{Lift}_d^{m+1}(C) = \bigotimes_{j=0}^{d-1} \mathtt{Rev}^j(F)$, so decoding serialises to

$$\mathrm{D}^0_{n+m+1}(\mathrm{Lift}_d^{m+1}(C)) = \prod_{j=0}^{d-1} \mathrm{D}^{d^{n+m}j}_{n+m+1}(\mathtt{Rev}^j(F)).$$

Corollary J.3, with $s = 0$ and full-block length $\ell = d^{n+m}$, gives $\mathrm{D}^{d^{n+m}j}_{n+m+1}(\mathtt{Rev}^j(F)) = \mathtt{C}_j(\mathrm{D}^0_{n+m}(F))$ for every j. By commutativity of distinct controls and exhaustivity, the product collapses to $\mathrm{id}_1 \otimes \mathrm{D}^0_{n+m}(F)$. The induction hypothesis then yields

$$\mathrm{D}^0_{n+m+1}(\mathrm{Lift}_d^{m+1}(C)) = \mathrm{id}_1 \otimes (\mathrm{id}_m \otimes \mathrm{D}^0_n(C)) = \mathrm{id}_{m+1} \otimes \mathrm{D}^0_n(C).$$

◀

▶ **Lemma J.7.** *Let C be a* **LOPP** *circuit acting on d^n modes, and let $m \geq 0$. Then $\mathrm{D}^0_{n+1+m}\big(\mathrm{Lift}^m_{d,j}(C)\big) = \mathrm{id}_m \otimes \mathtt{C}_j\big(\mathrm{D}^0_n(C)\big)$ as circuits in $\mathbf{CQC}_d$.*

Proof. Induct on m. For $m = 0$, only the j-th block contains C, and Corollary J.3 turns that block into $\mathtt{C}_j(\mathrm{D}^0_n(C))$, with identities on all other blocks.

For the step, set $F := \mathrm{Lift}^m_{d,j}(C)$. Then $\mathrm{Lift}^{m+1}_{d,j}(C) = \bigotimes_{q=0}^{d-1} \mathtt{Rev}^q(F)$, so

$$\mathrm{D}^0_{n+1+m+1}(\mathrm{Lift}^{m+1}_{d,j}(C)) = \prod_{q=0}^{d-1} \mathrm{D}^{d^{n+1+m}q}_{n+1+m+1}(\mathtt{Rev}^q(F)).$$

Applying Corollary J.3 to each full block gives $\mathrm{D}^{d^{n+1+m}q}_{n+1+m+1}(\mathtt{Rev}^q(F)) = \mathtt{C}_q(\mathrm{D}^0_{n+1+m}(F))$. Commutativity and exhaustivity then collapse the product to $\mathrm{id}_1 \otimes \mathrm{D}^0_{n+1+m}(F)$, and the induction hypothesis yields $\mathrm{id}_{m+1} \otimes \mathtt{C}_j(\mathrm{D}^0_n(C))$. ◀

J.3 Encoding respects the structural quotient

This subsection proves Lemma 4.8. Since the encoding is defined on raw circuits whereas $\mathbf{CQC}_d$ is a quotient by strict PROP coherence and the control-functor laws, we first record the raw semantic invariant used in the case analysis.

▶ **Lemma J.8.** *Let $C : n \to n$ be a raw qudit circuit, interpreted by the same matrix clauses as in Section 2 before quotienting by structural congruence. For all $a, b \geq 0$, $[\![\mathrm{E}^a_b(C)]\!]_{\mathrm{LOPP}} = I_{d^a} \otimes [\![C]\!] \otimes I_{d^b}$.*

Proof. We argue by induction on the raw syntax of C. The clauses for identities, composition, and scalar phases follow immediately from the recursive definition of E^a_b and the corresponding semantic clauses.

For tensor products, write $C_1 : n_1 \to n_1$ and $C_2 : n_2 \to n_2$. By definition, $\mathrm{E}^a_b(C_1 \otimes C_2) = \mathrm{E}^{a+n_1}_b(C_2) \circ \mathrm{E}^a_{b+n_2}(C_1)$. Using the induction hypotheses, the two factors have Gray-ordered semantics $I_{d^{a+n_1}} \otimes [\![C_2]\!] \otimes I_{d^b}$ and $I_{d^a} \otimes [\![C_1]\!] \otimes I_{d^{n_2+b}}$, respectively. Their product is $I_{d^a} \otimes [\![C_1]\!] \otimes [\![C_2]\!] \otimes I_{d^b}$, which is the required semantics of $C_1 \otimes C_2$ in the middle register.

For the adjacent Hadamard generator, Lemma I.2 gives the d-mode Hadamard semantics of $B_d^{(i,i+1)}$. The lift Lift_d^{a+b} duplicates this same action on every branch of the $a + b$ context digits, mirroring exactly on those Gray blocks whose reflected order is reversed. The two block permutations $\sigma^d_{a,b,1}$ and $\sigma^d_{a,1,b}$ move that lifted action from the right end of the context block to the middle qudit position. Hence the total semantics is $I_{d^a} \otimes H^{(i,i+1)} \otimes I_{d^b}$. The swap-generator case is the same calculation with the block-permutation semantics of $\sigma^d_{a,b,c}$: the three permutations in Definition 4.7 realise exactly the qudit symmetry $I_{d^a} \otimes \sigma_{1,1} \otimes I_{d^b}$.

Finally let $C = \mathtt{C}_j(C')$ with $C' : m \to m$. By induction, $\mathrm{E}^0_0(C')$ has Gray-ordered semantics $[\![C']\!]$. The selected lift $\mathrm{Lift}^{a+b}_{d,j}$ places this action on precisely the context branches whose distinguished digit has value j, and places identities on the other branches. The surrounding block permutations put the distinguished digit in front of the encoded target block. The resulting block-diagonal matrix is

$$I_{d^a} \otimes \left(|j\rangle\langle j| \otimes [\![C']\!] + \sum_{\ell \neq j} |\ell\rangle\langle \ell| \otimes I_{d^m} \right) \otimes I_{d^b},$$

which is $I_{d^a} \otimes [\![\mathtt{C}_j(C')]\!] \otimes I_{d^b}$. ◀

▶ **Lemma J.9.** *For all $a, b \geq 0$, if raw qudit circuits $C, C' : n \to n$ satisfy $C \equiv C'$ in the structural congruence of Definition 2.6, then* $\mathrm{LOPP} \vdash \mathrm{E}^a_b(C) = \mathrm{E}^a_b(C')$.

Proof. The congruence $\equiv$ is generated by two kinds of equations: strict symmetric-monoidal coherence for circuits and the control-functor equations for each value-control constructor. By Lemma J.8, it is enough to check that each generator has equal raw qudit semantics, because equal Gray-ordered semantics implies equal single-photon semantics after conjugating by the fixed Gray permutation; Theorem 4.3 then derives the equality in the LOPP equational theory. This use of LOPP completeness is external to the qudit completeness theorem being proved.

Strict PROP coherence is sound for the standard qudit interpretation: units and associativity of $\circ$, units and associativity of $\otimes$, interchange, and the symmetry equations are precisely the matrix equalities for identity matrices, Kronecker products, and permutation matrices. Hence the encodings of both sides have the same optical semantics by Lemma J.8.

It remains to list the control-functor generators. The identity and composition laws follow from the projector formula: $[\![\mathtt{C}_k(\mathrm{id}_n)]\!] = I_{d^{1+n}}$ and $[\![\mathtt{C}_k(G \circ F)]\!] = [\![\mathtt{C}_k(G)]\!]\,[\![\mathtt{C}_k(F)]\!]$. The strength law $\mathtt{C}_k(F \otimes \mathrm{id}_m) = \mathtt{C}_k(F) \otimes \mathrm{id}_m$ says that the same block-diagonal controlled action is tensored with an untouched m-wire identity block. Naturality with respect to a target permutation π is the equality $[\![\mathtt{C}_k(\pi^{-1}F\pi)]\!] = (I_d \otimes P_\pi^{-1})\,[\![\mathtt{C}_k(F)]\!]\,(I_d \otimes P_\pi)$, where P_π is the permutation matrix of π. Finally, the same-control nested-swap law holds because both sides are block diagonal on the two control digits and the only non-identity target block is the branch where both digits are k; swapping the two equal control values leaves that branch unchanged. Thus every generator of $\equiv$ has equal raw qudit semantics, and the encoded circuits are derivably equal in LOPP. ◀

J.4 Encoding and decoding are inverse

The remaining step is a structural induction. The previous lemmas identify the lift gadgets with identity padding or genuine control, and the block permutations with the corresponding qudit-wire symmetries.

Proof. We prove Theorem 4.15 by structural induction on f.

Typing convention. Fix $k, \ell \geq 0$ and a circuit $f : n \to n$ in $\mathbf{CQC}_d$. Then $E^k_\ell(f)$ is a $d^{k+n+\ell}$-mode **LOPP** circuit, and the decoding appearing in the theorem is $\mathrm{D}^0_{k+n+\ell}$. For readability, throughout this proof we write $D(-) := \mathrm{D}^0_{k+n+\ell}(-)$ when the arity is clear from context.

Empty and identity. If $f = \boxed{}$, then by definition $E^k_\ell(f) = \text{---}^{\otimes d^{k+\ell}}$, and hence $D\big(E^k_\ell(f)\big) = D\left(\text{---}^{\otimes d^{k+\ell}}\right) = \mathrm{id}_{k+\ell} = \mathrm{id}_k \otimes \mathrm{id}_0 \otimes \mathrm{id}_\ell$. The case $f = \text{---}$ is similar.

Swap. If $f = \times$, the encoding uses the optical block-permutations $\sigma^d_{k,n,\ell}$: $E^k_\ell(\times) = \sigma^d_{k,\ell,2} \circ \sigma^d_{k+\ell,1,1} \circ \sigma^d_{k,2,\ell}$. These permutations implement the qudit-wire swap by reindexing Gray blocks of modes. By Lemma L.10 from Appendix L we have $D\big(\sigma^d_{k,\ell,2}\big) = \mathrm{id}_k \otimes \sigma_{\ell,2}$, $D\big(\sigma^d_{k+\ell,1,1}\big) = \mathrm{id}_{k+\ell} \otimes \sigma_{1,1}$, and $D\big(\sigma^d_{k,2,\ell}\big) = \mathrm{id}_k \otimes \sigma_{2,\ell}$. Therefore, using strict PROP coherence, $D\big(E^k_\ell(\times)\big) = \mathrm{id}_k \otimes \sigma_{1,1} \otimes \mathrm{id}_\ell = \mathrm{id}_k \otimes \times \otimes \mathrm{id}_\ell$.

Global phase. If $f = \textcircled{\theta}$, then $E^k_\ell(\textcircled{\theta}) = \big(\text{-}\boxed{\theta}\text{-}\big)^{\otimes d^{k+\ell}}$, so decoding produces one basis-controlled phase per Gray basis vector of the context register: $D\big(E^k_\ell(\textcircled{\theta})\big) = \prod_{u \in [d]^{k+\ell}} \mathtt{C}_u\big(\textcircled{\theta}\big)$. By Theorem 3.10 we may reorder these distinct controlled branches into the order used in Corollary G.3. By iterated exhaustivity of control (Corollary G.3), the product over all words u yields $D\big(E^k_\ell(\textcircled{\theta})\big) = \mathrm{id}_{k+\ell} \otimes \textcircled{\theta} = \mathrm{id}_k \otimes \textcircled{\theta} \otimes \mathrm{id}_\ell$.

Two-level Hadamard. Let $f = \text{-}\boxed{H^{(r,r+1)}}\text{-}$. By Definition 4.7 we have $E^k_\ell(\text{-}\boxed{H^{(r,r+1)}}\text{-}) = \sigma^d_{k,\ell,1} \circ \mathrm{Lift}^{k+\ell}_d\big(B^{(i,i+1)}_d\big) \circ \sigma^d_{k,1,\ell}$. Applying D and functoriality for sequential composition gives $D\big(E^k_\ell(\text{-}\boxed{H^{(r,r+1)}}\text{-})\big) = D\big(\sigma^d_{k,\ell,1}\big) \circ D\Big(\mathrm{Lift}^{k+\ell}_d\big(B^{(i,i+1)}_d\big)\Big) \circ D\big(\sigma^d_{k,1,\ell}\big)$. By Lemma L.10, $D\big(\sigma^d_{k,\ell,1}\big) = \mathrm{id}_k \otimes \sigma_{\ell,1}$ and $D\big(\sigma^d_{k,1,\ell}\big) = \mathrm{id}_k \otimes \sigma_{1,\ell}$. By Lemma J.6, instantiated with $C = B^{(i,i+1)}_d$ and $m = k + \ell$, $D\Big(\mathrm{Lift}^{k+\ell}_d\big(B^{(i,i+1)}_d\big)\Big) = \mathrm{id}_{k+\ell} \otimes \mathrm{D}^0_1\big(B^{(i,i+1)}_d\big)$. Finally, Lemma J.4 gives $\mathrm{D}^0_1\big(B^{(i,i+1)}_d\big) = \text{-}\boxed{H^{(r,r+1)}}\text{-}$. Combining these equalities and simplifying using PROP coherence yields $D\big(E^k_\ell(\text{-}\boxed{H^{(r,r+1)}}\text{-})\big) = \mathrm{id}_k \otimes \text{-}\boxed{H^{(r,r+1)}}\text{-} \otimes \mathrm{id}_\ell$.

Sequential composition. If $f = g \circ h$, then by Definition 4.7, $E^k_\ell(g \circ h) = E^k_\ell(g) \circ E^k_\ell(h)$, so $D\big(E^k_\ell(g \circ h)\big) = D\big(E^k_\ell(g)\big) \circ D\big(E^k_\ell(h)\big)$. The induction hypothesis gives $D(E^k_\ell(g)) = \mathrm{id}_k \otimes g \otimes \mathrm{id}_\ell$ and $D(E^k_\ell(h)) = \mathrm{id}_k \otimes h \otimes \mathrm{id}_\ell$, hence $D\big(E^k_\ell(g \circ h)\big) = \mathrm{id}_k \otimes (g \circ h) \otimes \mathrm{id}_\ell$.

Parallel composition. If $f = g \otimes h$ with $g : n_g \to n_g$ and $h : n_h \to n_h$, then Definition 4.7 serialises the tensor product: $E^k_\ell(g \otimes h) = E^{k+n_g}_\ell(h) \circ E^k_{\ell+n_h}(g)$. Decoding preserves $\circ$, so $D\big(E^k_\ell(g \otimes h)\big) = D\big(E^{k+n_g}_\ell(h)\big) \circ D\big(E^k_{\ell+n_h}(g)\big)$. Applying the induction hypotheses to h at context $(k+n_g, \ell)$ and to g at context $(k, \ell+n_h)$ yields $D\big(E^k_\ell(g \otimes h)\big) = \mathrm{id}_k \otimes (g \otimes h) \otimes \mathrm{id}_\ell$.

Control. Finally, let $f = \mathtt{C}_j(g)$ with $g : m \to m$. By Definition 4.7, $E^k_\ell\big(\mathtt{C}_j(g)\big) = \sigma^d_{k,\ell,m+1} \circ \mathrm{Lift}^{k+\ell}_{d,j}\big(E^0_0(g)\big) \circ \sigma^d_{k,m+1,\ell}$. Decoding and using Lemma L.10 reduces the outer permutations to the corresponding qudit-wire symmetries. For the middle term we use Lemma J.7 to obtain $D\Big(\mathrm{Lift}^{k+\ell}_{d,j}\big(E^0_0(g)\big)\Big) = \mathrm{id}_{k+\ell} \otimes \mathtt{C}_j\big(\mathrm{D}^0_m(E^0_0(g))\big)$. By the induction hypothesis applied to g with $k = \ell = 0$, $\mathrm{D}^0_m(E^0_0(g)) = g$, hence the middle factor is $\mathrm{id}_{k+\ell} \otimes \mathtt{C}_j(g)$, and the surrounding symmetries cancel by PROP coherence. Therefore $D\big(E^k_\ell(\mathtt{C}_j(g))\big) = \mathrm{id}_k \otimes \mathtt{C}_j(g) \otimes \mathrm{id}_\ell$.

This exhausts the constructors of $\mathbf{CQC}_d$, so $\mathrm{QC}_d \vdash \mathrm{D}^0_{k+n+\ell}(E^k_\ell(f)) = \mathrm{id}_k \otimes f \otimes \mathrm{id}_\ell$ for all f, as required. ◀

K Mimicking rules

Theorem 4.14 says that every LOPP derivation decodes to a derivation in QC_d: if LOPP $\vdash C_1 = C_2$ for d^n-mode circuits C_1, C_2, then $\mathrm{QC}_d \vdash D(C_1) = D(C_2)$.

It is enough to check:

1. decoding is compatible with the strict-PROP congruence on raw LOPP terms (i.e. the strict symmetric monoidal/PROP equations);

2. decoding maps each non-structural axiom of the LOPP calculus (Figure 10) to a derivable equation in QC_d.

We write $\equiv$ for the strict-PROP congruence on raw LOPP circuits as in Definition 4.1; Figure 9 reproduces its generating equations.

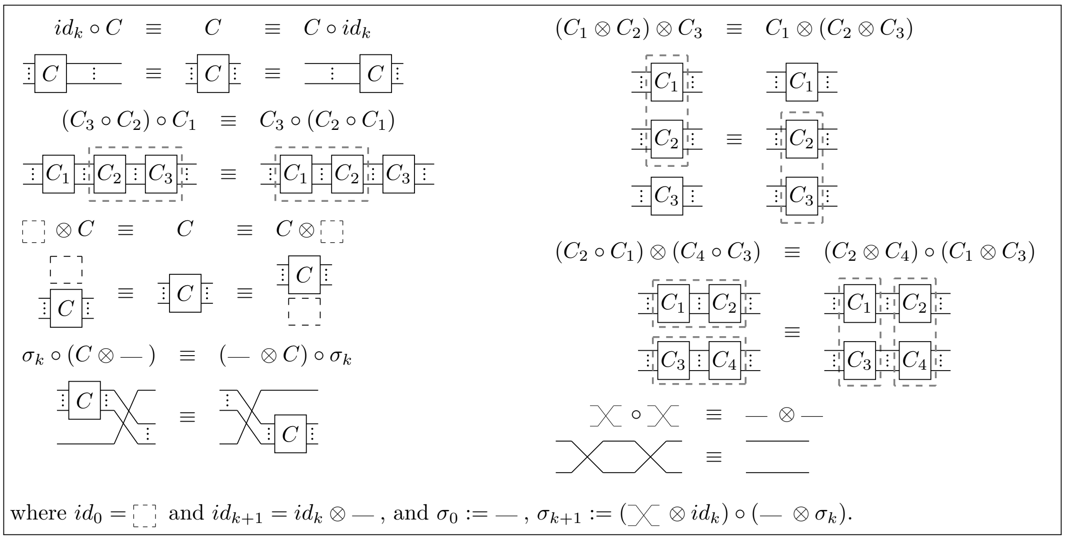


Figure 9 Structural axioms generating the congruence $\equiv$ on raw circuits (either raw quantum circuits or raw optical circuits).

Commutation of decoded circuits on disjoint blocks

The first check is a commuting property for decoded optical subcircuits on disjoint mode intervals.

▶ **Lemma K.1.** *Decoded LOPP generators supported on disjoint mode intervals commute in* QC_d.

Proof. By Lemma 4.11, such a decoded generator is supported on one Gray basis word or on one Gray-neighbouring pair. Disjoint mode intervals therefore give disjoint Gray supports. After applying the SEPARATE normal form of Lemma 3.8, every resulting $\mathcal{G}$-layer has one of two forms relative to a layer from the other decoded generator: either the surrounding control words require incompatible basis values on some qudit, or the active target windows are disjoint. In the first case the pairwise swap is one of the distinct-control commutations used in Theorem 3.10; in the second case it is strict symmetric-monoidal interchange and naturality. Composing these finite pairwise swaps commutes the two decoded generators. ◀

▶ **Lemma K.2.** *Let $C_i : \ell_i \to \ell_i$ be raw optical circuits for $i = 1, 2$. Let $n \geq 0$, and let t_1, t_2 be nonnegative integers satisfying $t_i + \ell_i \leq d^n$ for $i = 1, 2$. If the intervals $[t_1, t_1 + \ell_1)$ and $[t_2, t_2 + \ell_2)$ are disjoint, then* $\mathrm{QC}_d \vdash \mathrm{D}_n^{t_2}(C_2) \circ \mathrm{D}_n^{t_1}(C_1) = \mathrm{D}_n^{t_1}(C_1) \circ \mathrm{D}_n^{t_2}(C_2)$.

Proof. The disjointness assumption gives either $t_1 + \ell_1 \leq t_2$ or $t_2 + \ell_2 \leq t_1$. Since the conclusion is symmetric in the two pairs up to reversing the displayed equality, it suffices to treat the first case. Assume therefore that $t_1 + \ell_1 \leq t_2$.

We argue by structural induction on the pair (C_1, C_2).

If either circuit is a sequential composite, the statement follows immediately from the induction hypothesis and the defining clause $\mathrm{D}_n^t(C_2 \circ C_1) = \mathrm{D}_n^t(C_2) \circ \mathrm{D}_n^t(C_1)$.

If either circuit is a tensor composite, we use the serialisation clause $\mathrm{D}_n^t(C_1 \otimes C_2) = \mathrm{D}_n^{t+\ell_1}(C_2) \circ \mathrm{D}_n^t(C_1)$ (Definition 4.10) together with associativity of $\circ$ to reduce to commuting decoded subcircuits whose acted-on mode intervals remain disjoint; the required commutations are then provided by the induction hypothesis.

The induction therefore reduces to the case where both C_1 and C_2 are LOPP generators (phase shifter, beam splitter, or swap). This is Lemma K.1. ◀

Contexts and compatibility with structural congruence

▶ **Definition K.3.** *A (one-hole) raw context $\mathbb{C}[\cdot]_i$ is a term over the LOPP signature built from $\circ$ and $\otimes$ with a distinguished placeholder of arity i. If its outer arity is N, we call it an N-mode context. For any i-mode raw circuit C, write $\mathbb{C}[C]$ for the circuit obtained by filling the hole with C.*

▶ **Lemma K.4.** *Let $\mathbb{C}[\cdot]_i$ be an N-mode raw context with a hole of arity i, let $C_1, C_2 : i \to i$ be raw optical circuits with $C_1 \equiv C_2$, and let t, n satisfy $t + N \leq d^n$. Then $\mathrm{QC}_d \vdash \mathrm{D}_n^t(\mathbb{C}[C_1]) = \mathrm{D}_n^t(\mathbb{C}[C_2])$. In particular, if C_1, C_2 are full d^n-mode circuits and $C_1 \equiv C_2$, then $\mathrm{QC}_d \vdash D(C_1) = D(C_2)$.*

Proof. By induction on the context structure it suffices to check preservation of the generating axioms of Figure 9. Identities and tensor units are sent to id_n by Definition 4.10. Associativity of $\circ$ is preserved because D_n^t is defined compositionally on $\circ$. Associativity of $\otimes$ is preserved because D_n^t serialises $\otimes$ into a fixed sequential composite, and the two bracketings decode to the same composite up to associativity of $\circ$. For the mixed-product/interchange axiom in Figure 9, expanding both sides via the serialisation clause yields composites of decoded subcircuits acting on disjoint mode blocks; Lemma K.2 provides the required commutations. It remains to check the two structural equations involving raw mode swaps. An adjacent optical swap on modes $r, r+1$ decodes, by Definition 4.10 and Lemma 4.5, to $\Lambda_w^u(X^{(v,v+\varepsilon)})$, i.e. to the controlled adjacent two-level transposition of the two Gray basis words labelling those modes. Its involutivity is therefore the controlled instance of (XInv), tensored with identities and conjugated by the placement permutation $\sigma_{|u|,|w|,1}$.

For naturality, first factor the raw structural permutation σ_k into adjacent optical swaps, as in Figure 9. Decoding this factorisation gives the corresponding product of adjacent controlled two-level transpositions on Gray basis words. The relations (XInv), (XX1), and (XX2) are precisely the involutive and braid moves needed to make this decoded permutation independent of the chosen adjacent-swap factorisation. Conjugating a decoded generator by one adjacent decoded swap only relabels its Gray support: for phases this is the phase-through-X calculation (CXP); for decoded swaps it is the same adjacent-transposition calculus; and for beam splitters it is the identical calculation with X replaced by the adjacent R_x-gate, using (RSym) for orientation reversal. The statement for an arbitrary raw circuit C then follows by induction on C, using the already proved composition and tensor clauses. Hence the decoded forms of raw swap naturality and swap involutivity are derivable in QC_d. ◀

Mimicking the LOPP equational theory

Recall the non-structural axioms of LOPP from [18] (Figure 10).

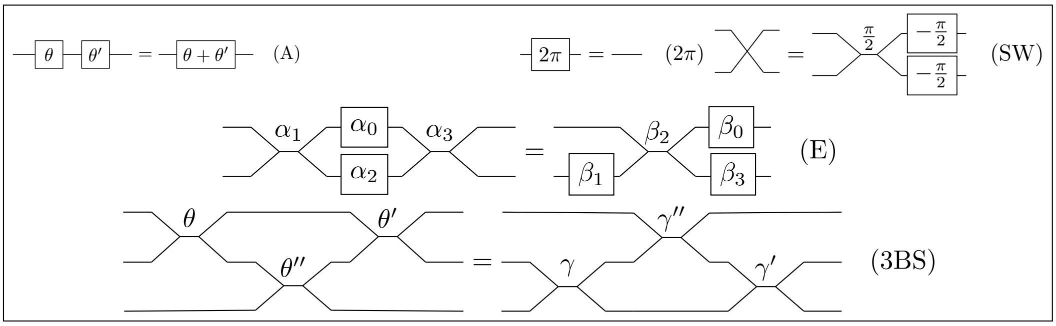


Figure 10 Non-structural axioms of the LOPP calculus: phase addition, 2π-periodicity, swap decomposition, and Euler/three-beamsplitter identities.

▶ **Lemma K.5.** *Let a source equational theory be the congruence generated by a structural congruence $\equiv$ and a set of non-structural axiom schemata. Suppose a decoding into a target theory respects $\equiv$ in every raw context and that every contextual instance of each non-structural source axiom decodes to a derivable target equation. Then every source derivation decodes to a target derivation.*

Proof. The source derivability relation is the least congruence containing the structural congruence and the listed non-structural axioms. The two hypotheses give the result on those generators, and closure under contexts, composition, tensor, symmetry, and transitivity follows by the congruence rules of the target theory. ◀

Proof of Theorem 4.14. By Lemma K.5 and Lemma K.4, the remaining check is that each non-structural axiom of Figure 10 is preserved by decoding at an arbitrary offset t.

Let t, n be such that the relevant local LOPP circuit sits inside d^n modes. When $G^d_n(t) = uiv$ and $G^d_n(t+1) = ujv$ with $|i - j| = 1$, write $\Lambda^u_v(G^{(i,j)})$ for the qudit circuit obtained by placing the one-qudit gate $G^{(i,j)}$ on the changed coordinate and controlling on the surrounding word uv. This is only notation for the decoding clause in Definition 4.10; if the Gray edge is traversed in the opposite orientation, the same notation is read with the two levels exchanged, using (RSym) and $X^{(i,j)} = X^{(j,i)}$.

- *Phase addition* (A). Decoding a phase shifter on mode t yields $\mathsf{C}_{G^d_n(t)}(\textcircled{\theta})$ (Definition 4.10). The decoded equation is therefore an instance of the QC_d phase arithmetic axiom (Sum), closed under iterated control.
- *2π-periodicity* (2π). Similarly, decoding $-\boxed{2\pi}-$ yields a multi-controlled 2π-phase, which is equal to id_n by the QC_d axiom (2π) (again closed under iterated control). The right-hand side decodes to id_n.
- *Swap decomposition* (SW). This is a two-mode local identity. If $G^d_n(t) = uiv$ and $G^d_n(t+1) = ujv$, then the left-hand side decodes to $\Lambda^u_v(X^{(i,j)})$. The right-hand side is the optical phase–beam-splitter–phase decomposition of the same two-mode swap; after decoding, the beam splitter is $\Lambda^u_v(R_x^{(i,j)}(\pi/2))$ and the phases are the corresponding controlled level phases. Expanding $R_x^{(i,j)}$ and $X^{(i,j)}$ by Appendix B, and using (Sum), (2π), (HTot), and (HInv) under the surrounding controls, gives the same controlled swap $\Lambda^u_v(X^{(i,j)})$. Thus the decoded swap axiom is a controlled instance of the two-level swap decomposition already available in QC_d; the global block-permutation version used for contexts is proved separately in Appendix L.
- *Euler identity* (E). Again let the two modes have Gray labels uiv and ujv. Phase shifters on either optical mode decode to level phases on i or j, and beam splitters decode to $R_x^{(i,j)}$

on the same two-level subspace, all under the common surrounding control uv. After this replacement, the decoded optical Euler axiom is precisely the surrounding-control instance Λ_v^u((EH)), with the deterministic parameter convention fixed in Appendix C. The only orientation choice is whether the edge is read as $i \to j$ or $j \to i$; (RSYM) and the symmetry of the derived $H^{(i,j)}$ notation reduce the reflected case to the displayed one.

- *Three-beamsplitter identity* (3BS). Put $w_r = G_n^d(t+r)$ for $r = 0, 1, 2$. By Lemma 4.5, w_0 and w_1 differ in one digit, and w_1 and w_2 differ in one digit. There are only two cases. If the same digit changes twice, then, after fixing the surrounding word, the three labels are $uiv, u(i+\varepsilon)v, u(i+2\varepsilon)v$ for $\varepsilon = \pm 1$. The decoded equation is the controlled instance of the one-qudit three-rotation axiom (3Rx), with the angle convention of Appendix C and with (RSYM) used when the reflected block gives the decreasing orientation.
  If the two changed digits are different, the three labels form a corner of a two-coordinate square: after a wire permutation and renaming of adjacent levels they have the shape $u\,i\,a\,v,\ u\,j\,a\,v,\ u\,j\,b\,v$, with $|i-j| = |a-b| = 1$. The first and third rotations act on different qudit coordinates, while the middle one carries the fixed value of the other coordinate as a value-control. This is exactly the surrounding-control instance of the derived mixed rule (3CRX), whose parameters are the same as for (3Rx).

Decoding preserves each generator of the LOPP equational theory and is stable under contexts, so LOPP $\vdash C_1 = C_2$ implies $\mathrm{QC}_d \vdash D(C_1) = D(C_2)$. ◀

L Explicit encoding and decoding of swaps

This appendix shows that the optical block-swaps used by the encoding decode to the intended qudit-level swaps in $\mathbf{CQC}_d$.

L.1 Gray-code indices for words

A mode in the d-ary Gray code for n qudits is labelled by a word $w \in [d]^n$. The inverse map below returns its position in the Gray ordering.

▶ **Definition L.1.** *For a base-d word $w \in [d]^n$ we write $G_d^{-1}(w)$ for its index in the d-ary Gray ordering. It is defined recursively as follows.*

- *For the empty word ϵ we set $G_d^{-1}(\epsilon) = 0$.*
- *For a non-empty word $w = h \cdot t$ of length $n+1$, with $h \in \{0, \dots, d-1\}$ the first digit and $t \in [d]^n$ the suffix, we put*

$$G_d^{-1}(h \cdot t) = \begin{cases} h\, d^n + G_d^{-1}(t) & \textit{if } h \textit{ is even,} \\ (h+1)\, d^n - 1 - G_d^{-1}(t) & \textit{if } h \textit{ is odd.} \end{cases}$$

L.2 Elementary LOPP swap gadgets

The elementary gadget swaps two basis states.

▶ **Definition L.2.** *Let $w_1, w_2 \in [d]^n$ be two distinct words of the same length. The notation C_{d,w_1,w_2} denotes the raw* **LOPP** *word-swap constructed in Definition L.4. Its intended action is the mode transposition that swaps the two modes labelled by w_1 and w_2, and acts as the identity on all other modes.*

We also use this notation in structured instances such as $C_{d,\beta\cdot q_1\cdot\alpha,\ \beta\cdot p_1\cdot\alpha}$, where β and α are fixed prefixes and suffixes. Decoding then produces the expected two-level transposition with surrounding controls determined by β and α. The saturated version below ranges over all such prefixes and suffixes so that commutativity and exhaustivity can later remove those extra controls.

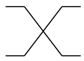
▶ **Definition L.3.** *For integers $k, \ell \geq 0$ and digits $q_1, q_2, p_1, p_2 \in \{0, \ldots, d-1\}$ we define*

$$C^{\star}_{d,k,\ell,[q_1,q_2],[p_1,p_2]} = \prod_{\beta\in[d]^k} \prod_{\alpha\in[d]^\ell} C_{d,\ \beta\cdot q_1\cdot q_2\cdot\alpha,\ \beta\cdot p_1\cdot p_2\cdot\alpha}.$$

Here β and α range over words of lengths k and ℓ, respectively, both in lexicographic increasing order with β as the outer index; for $k = 0$ or $\ell = 0$ the corresponding word is empty.

$C^{\star}$ applies the same three-CNOT pattern for every choice of prefix β and suffix α. The decoding calculation below first identifies one two-position factor and then uses commutativity and exhaustivity of controls to remove the saturated prefix and suffix controls.

L.3 Adjacent construction of word-swaps

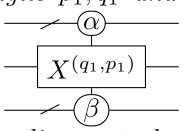
For $n \geq 1$ and $0 \leq r < d^n - 1$, write $A^n_r := id_r \otimes$  $\otimes\, id_{d^n-r-2}$ for the adjacent optical swap of modes r and $r+1$.

▶ **Definition L.4.** *Let $w_1, w_2 \in [d]^n$ be distinct. Put $i = G_d^{-1}(w_1)$ and $j = G_d^{-1}(w_2)$. If $i > j$, define $C_{d,w_1,w_2} := C_{d,w_2,w_1}$. If $i < j$, let $r_1, \ldots, r_{2(j-i)-1}$ be the concatenation of $i, i+1, \ldots, j-1$ and $j-2, j-3, \ldots, i$, with the second list empty when $j = i+1$, and set $C_{d,w_1,w_2} := A^n_{r_{2(j-i)-1}} \circ \cdots \circ A^n_{r_1}$.*

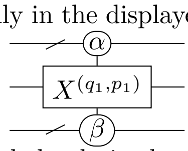
This finite product is well-typed at arity d^n. As a permutation of optical modes, the first $j - i$ adjacent swaps move mode i to mode j, and the remaining adjacent swaps return every intermediate mode to its original position. Thus C_{d,w_1,w_2} implements exactly the transposition of the two modes labelled by w_1 and w_2.

L.4 Decoding of the adjacent swap construction

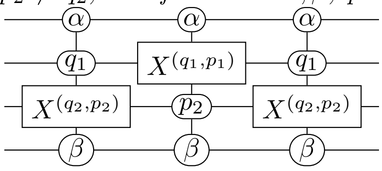
The next lemmas record the effect of decoding these gadgets.

▶ **Lemma L.5.** *For any distinct digits p_1, q_1 and words α, β, put $w_q = \beta\cdot q_1\cdot\alpha$ and $w_p = \beta\cdot p_1\cdot\alpha$. Then $D(C_{d,w_q,w_p}) =$* 

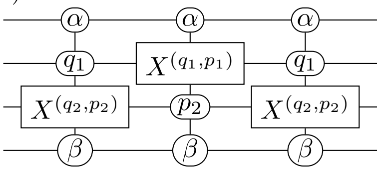
. This is a single two-level $X^{(p_1,q_1)}$ on the appropriate qudit, with the surrounding controls determined by β and α.

Proof. By Definition L.4, C_{d,w_q,w_p} is a factorisation of the optical transposition between the two Gray modes w_q and w_p into adjacent optical swaps. Definition 4.10 and Lemma 4.5 decode each adjacent optical swap as the controlled adjacent two-level transposition on the corresponding neighbouring Gray words. The relations (XInv), (XX1), and (XX2) give the usual adjacent-transposition calculus, so the decoded palindromic factorisation reduces to the transposition of w_q and w_p and fixes all other Gray basis words. Since these two words differ only in the displayed digit, the resulting qudit circuit is exactly the controlled $X^{(p_1,q_1)}$ shown as 

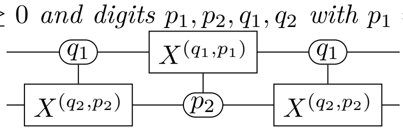
. This is an internal QC_d calculation using the decoded adjacent swaps and the derived transposition rules; it does not invoke the final completeness theorem. ◀

▶ **Lemma L.6.** *For digits p_1, p_2, q_1, q_2 with $p_1 \neq q_1$ and $p_2 \neq q_2$, and for words α, β, put $w_q = \beta \cdot q_1 \cdot q_2 \cdot \alpha$ and $w_p = \beta \cdot p_1 \cdot p_2 \cdot \alpha$. Then $D(C_{d,w_q,w_p}) =$* 

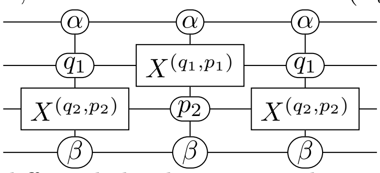
.

Proof. By Definition L.4 and the same decoded adjacent-transposition calculation used in Lemma L.5, $D(C_{d,w_q,w_p})$ is the basis-state transposition exchanging w_q and w_p. Let $w_m := \beta \cdot q_1 \cdot p_2 \cdot \alpha$. The three single-position transpositions $(w_q\, w_m)$, $(w_m\, w_p)$, and $(w_q\, w_m)$ compose to $(w_q\, w_p)$; in QC_d this is the controlled instance of the transposition rules (XX2) and (XInv). Lemma L.5 identifies these three single-position transpositions with the three factors in 

, giving the claimed decoding. ◀

▶ **Lemma L.7.** *For all $k, \ell \geq 0$ and digits p_1, p_2, q_1, q_2 with $p_1 \neq q_1$ and $p_2 \neq q_2$, we have $D\big(C^{\star}_{d,k,\ell,[q_1,q_2],[p_1,p_2]}\big) = id_k \otimes$* 

$\otimes\, id_\ell$.

Proof. By definition, $D(C^{\star}_{d,k,\ell,[q_1,q_2],[p_1,p_2]})$ is the product, over all prefixes β and suffixes α, of the decoded factors $D(C_{d,\beta \cdot q_1 \cdot q_2 \cdot \alpha,\, \beta \cdot p_1 \cdot p_2 \cdot \alpha})$. Lemma L.6 identifies each factor as 

. These factors act on the same pair of target levels and differ only by their surrounding control words; by commutation of distinct controls (Thm. 3.10) we may group the first CNOT-style component of every factor, then the middle component of every factor, and finally the last component of every factor. For each of these three groups, the product ranges over every prefix $\beta \in [d]^k$ and suffix $\alpha \in [d]^\ell$, so iterated exhaustivity (Corollary G.3) removes all surrounding controls. The three collapsed groups are exactly the displayed three-CNOT pattern, tensored with identities on the untouched wires. This yields the claimed form. ◀

L.5 Decoding block-swaps

Block-swaps are assembled from these saturated gadgets.

▶ **Definition L.8.** *For integers $k, \ell \geq 0$ we define the LOPP block-swap $\sigma^d_{k,\ell,1}$ that exchanges a block of ℓ qudit wires with a single qudit wire to its right by*

$$\sigma^d_{k,\ell,1} = \prod_{j=0}^{\ell-1} \prod_{x=0}^{d-2} \prod_{y=x+1}^{d-1} C^{\star}_{d,k+j,\ell-j-1,\,[x,y],[y,x]}.$$

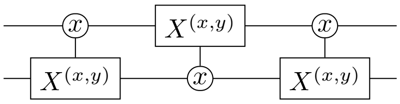
The products are read in lexicographic increasing order of the displayed indices j, x, y, with j outermost.

▶ **Lemma L.9.** *We have $D\left(\sigma^d_{k,\ell,1}\right) = id_k \otimes \sigma_{\ell,1}$, where $\sigma_{\ell,1}$ is the usual PROP symmetry swapping an ℓ-wire block with a single wire.*

Proof. Using Lemma L.7 we obtain

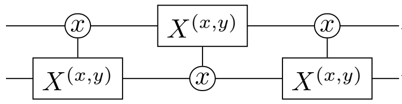
$$D\left(\sigma^d_{k,\ell,1}\right) = \prod_{j=0}^{\ell-1}\prod_{x=0}^{d-2}\prod_{y=x+1}^{d-1} D\left(C^\star_{d,k+j,\ell-j-1,[x,y],[y,x]}\right)$$

$$= \prod_{j=0}^{\ell-1}\prod_{x=0}^{d-2}\prod_{y=x+1}^{d-1}\Bigg(id_{k+j} \otimes$$



$$\otimes\, id_{\ell-j-1}\Bigg)$$

$$= \prod_{j=0}^{\ell-1}\Bigg(id_{k+j} \otimes \Bigg(\prod_{x<y}$$



$$\Bigg)\otimes id_{\ell-j-1}\Bigg).$$

(the bracketed product over $x<y$ is marked $\overset{(S)}{=} \sigma_{1,1}$)

The product over $x < y$ yields exactly the adjacent swap $\sigma_{1,1}$ by the swap decomposition rule (S). Factoring out the leftmost k wires, we obtain

$$D\left(\sigma^d_{k,\ell,1}\right) = id_k \otimes \left(\prod_{j=0}^{\ell-1}\left(id_j \otimes \sigma_{1,1} \otimes id_{\ell-j-1}\right)\right)$$
$$= id_k \otimes \sigma_{\ell,1},$$

since the product on the right is precisely the standard factorisation of a block-swap into ℓ adjacent transpositions. ◀

Finally we pass from swapping one wire with an ℓ-block to swapping two arbitrary blocks.

▶ **Lemma L.10.** *For $c = 1$, take $\sigma^d_{a,b,1}$ as defined above. For $c > 1$, define inductively $\sigma^d_{a,b,c} := \sigma^d_{a,b+c-1,1} \circ \sigma^d_{a,b+1,c-1}$. Then $D\left(\sigma^d_{a,b,c}\right) = id_a \otimes \sigma_{b,c}$.*

Proof. By definition,

$$D\left(\sigma^d_{a,b,c}\right) = D\left(\sigma^d_{a,b+c-1,1} \circ \sigma^d_{a,b+1,c-1}\right)$$
$$= D\left(\sigma^d_{a,b+c-1,1}\right) \circ D\left(\sigma^d_{a,b+1,c-1}\right)$$
$$\overset{Lemma\ L.9}{=} \left(id_a \otimes \sigma_{b+c-1,1}\right) \circ D\left(\sigma^d_{a,b+1,c-1}\right).$$

By the induction hypothesis on c we have $D(\sigma^d_{a,b+1,c-1}) = id_a \otimes \sigma_{b+1,c-1}$, so

$$D\left(\sigma^d_{a,b,c}\right) = id_a \otimes \left(\sigma_{b+c-1,1} \circ \sigma_{b+1,c-1}\right)$$
$$= id_a \otimes \sigma_{b,c},$$

since $\sigma_{b,c}$ is given by first moving the rightmost wire across the block of size $b + c-1$ and then swapping the resulting $(b + 1)$-block with the remaining $c-1$ wires. This is exactly the composition $\sigma_{b+c-1,1} \circ \sigma_{b+1,c-1}$. ◀

M Axiom soundness checks

This appendix records the semantic verification used in the soundness direction of Theorem 4.17. The detailed case-by-case checks are collected in Tables 1 and 2: the first concerns the core gate-algebra schemata of Figure 2, and the second the control-support schemata of Figure 3. Each row now records only the key semantic reason that the displayed schema is sound. Closure of QC_d under sequential composition, tensor product, structural symmetries, and value-control then preserves soundness for every contextual instance of the schemata.

Schema	Essential soundness check
(Sum), (2π)	Immediate from $e^{i\alpha}e^{i\beta} = e^{i(\alpha+\beta)}$ and $e^{2\pi i} = 1$; any value-control only adds orthogonal branch projectors.
(XH)	Since $X^{(i,j)}$ is the transposition of levels i and j, conjugating $H^{(j,k)}$ by $X^{(i,j)}$ relabels its active pair to $H^{(i,k)}$. Hence both sides agree on $\mathrm{span}\{\lvert i\rangle, \lvert j\rangle, \lvert k\rangle\}$ and are identity elsewhere.
(H^2)	The adjacent Hadamard squares to the identity on $\mathrm{span}\{\lvert i\rangle, \lvert i+1\rangle\}$ and is identity on the orthogonal complement.
(EH)	Lemma C.1 proves the required 2×2 matrix identity for the admissible angle parameters; both sides are identity on the orthogonal complement.
(3Rx)	Lemma C.2 proves the corresponding 2×2 identity, including the degenerate angle cases; outside the selected two-level subspace both sides are identity.
(CXC)	Both sides induce the same permutation of computational basis states.
(S)	On $\lvert p\rangle \lvert q\rangle$, every factor is identity when $p = q$, while for $p \neq q$ exactly the factor indexed by $(\min\{p, q\}, \max\{p, q\})$ acts and sends $\lvert p\rangle \lvert q\rangle$ to $\lvert q\rangle \lvert p\rangle$.

Table 1 Semantic verification of the core gate-algebra schemata of Figure 2.

Schema	Essential soundness check
(ExP), (ExH)	If U denotes the scalar phase or the adjacent Hadamard, the left-hand side is $\bigoplus_{k\in[d]} U = I_d \otimes U$, exactly the denotation of the right-hand side.
(SHH), (SHP), (SHπ)	The side conditions in Figure 3 ensure that the nontrivial basis states moved by the two gates are disjoint. Proposition 3.2 then implies that the corresponding block-diagonal operators commute.
(BPP), (BPπ), (B$\pi\pi$d), (BHπ), (BHH), (BHP), (B$\pi\pi$s)	Writing the two denotations with control projectors P_k and P_ℓ for $k \neq \ell$, their non-identity blocks are supported on orthogonal direct summands because $P_k P_\ell = 0$. Therefore the operators commute.
(CoP), (Coπ)	Decompose with respect to the basis $\lvert a\rangle \lvert b\rangle$ of the two control wires. On every block except the designated branch pair, both sides are identity; on that block, after the swap of the control wires, both sides apply the same target unitary. Hence the block-diagonal matrices coincide.

Table 2 Semantic verification of the control-support schemata of Figure 3.