Algebraic Structure of Controlled States and Operators in the ZXW-calculus

Edwin Agnew
Department of Computer Science
University of Oxford

Lia Yeh
Department of Computer Science
University of Oxford
Quantinuum
17 Beaumont Street
Oxford OX1 2NA, UK
lia.yeh@cs.ox.ac.uk

Richie Yeung
Department of Computer Science
University of Oxford
Quantinuum
17 Beaumont Street
Oxford OX1 2NA, UK
richie.yeung@cs.ox.ac.uk

An important concept for graphical rewriting in the ZXW-calculus is a *controlled diagram* — a diagram extended with an additional input wire for triggering the operation on or off. In this work, we investigate the algebraic structure of controlled diagrams. First, we show that controlled matrices form a ring, yielding powerful rewrite rules for large classes of diagrams; we also show that controlled states form a ring, in this case isomorphic to the ring of multilinear polynomials. Augmenting the ZXW-calculus with controlled square matrices as black-box generators, we prove completeness for polynomials over same-size square matrices. We apply this finding to show that it is unlikely that arbitrary quantum states can be efficiently rewritten to any arithmetic ZXW diagram, as this would imply $\mathsf{RP} = \mathsf{NQP}$. Through formalising the algebraic properties of quantum control, we give rise to powerful new graphical rewrite rules for black-box diagrams.

1 Introduction

Controlling or branching to different possible linear maps, relations, or channels is important across quantum information and quantum computation, and has been studied through many different approaches. In quantum algorithms common techniques are block encodings [12, 20] and linear combination of unitaries [6], while a number of categorical formalisations have included routed quantum circuits [31], the many-worlds calculus [5], categorifying signal flow diagrams [3], and classical and quantum control in quantum modal logic [23].

The question we are interested in is how quantum graphical calculi such as the ZX [7], ZW [8], and ZH [2] calculus can be augmented to support properties of quantum control. An early use of controlled state diagrams was for proving constructive and rational angle ZX calculus completeness [16]. More recently, controlled state and controlled matrix diagrams have been applied to addition and differentiation of ZX diagrams [15], differentiating and integrating ZX diagrams for quantum machine learning [32], Hamiltonian exponentiation and simulation [25], and non-linear optical quantum computing [10]. To sum ZX diagrams, these works have used controlled states along with the W generator from the ZW calculus.

Given how useful controlled diagrams are, a natural question to ask is why they work: What their underlying mathematical structures are, and which equational rewrites they satisfy.

First, we show that the set of all controlled n-partite states defines a commutative ring. We introduce $\boxplus$ which defines an Abelian group and $\boxtimes$ which defines a commutative monoid, and show that $\boxtimes$ distributes over $\boxplus$. The fragment of the qubit ZW calculus corresponding to controlled states, which we

To appear in EPTCS.

call *arithmetic ZXW diagrams* hence defines a ring which we prove is isomorphic to multilinear polynomials $\mathbb{C}[x_1,...,x_n]/({x_1}^2,...,{x_n}^2)$, and prove completeness for. Analogously, we show that the set of all controlled square matrices on n qubits defines a non-commutative ring $(\tilde{M}^n,$  $)$.

We add controlled square matrices and rewrite rules for them to the ZXW-calculus, in which we plug controlled states into each control wire of controlled square matrices. We prove their completeness and that this is isomorphic to multivariate polynomials over same-size square matrices. Commutativity of controlled square matrices holds in the special case that the controls target mutually exclusive sectors, allowing copying of arbitrary controlled diagrams. As a result, we can factor multivariate polynomials over same-size square matrices. This means we now have the ability to factor any Hamiltonian in the ZXW-calculus [25], even with all its terms black-boxed.

Applying these findings, we present a quantum complexity theory result concerning efficient rewriting of quantum states to any arithmetic ZXW diagram, and prove that the higher-order map Ctrl which sends square matrices to controlled square matrices satisfies some important algebraic properties. In sum, these algebraic properties of quantum control give rise to powerful new graphical reasoning capabilities.

2 Preliminaries

This section introduces the ZXW-calculus, and how controlled diagrams are defined in it. The ZXW-calculus is a diagrammatic formalism for qudit computation, unifying the ZX and ZW calculi and synthesising their relative strengths. The ZXW-calculus consists of diagrams built from a small number of generators and equipped with a complete set of rewrite rules which enables all equalities between linear maps to be proven diagrammatically. Diagrams are to be read top to bottom or, in later sections, left to right.

2.1 The ZXW-Calculus

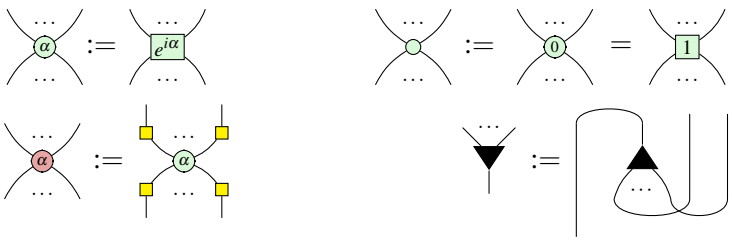
The qubit ZXW-calculus is built from the following generators:

$$\left[\!\left[\ \middle|\ \right]\!\right] = \begin{bmatrix}1&0\\0&1\end{bmatrix} \qquad \left[\!\left[\ \times\ \right]\!\right] = \begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix} \qquad \left[\!\left[\ \cap\ \right]\!\right] = \begin{bmatrix}1\\0\\0\\1\end{bmatrix} \qquad \left[\!\left[\ \cup\ \right]\!\right] = \begin{bmatrix}1&0&0&1\end{bmatrix}$$

$$\left[\!\left[\ \begin{array}{c}n\\ \cdots\\ c\\ \cdots\\ m\end{array}\ \right]\!\right] = |0^m\rangle\langle 0^n| + c|1^m\rangle\langle 1^n|, c \in \mathbb{C} \qquad \left[\!\left[\ \triangle\ \right]\!\right] = |00\rangle\langle 0| + |01\rangle\langle 1| + |10\rangle\langle 1|$$

$$\left[\!\left[\ \begin{array}{c}\\ \blacksquare\\ |\end{array}\ \right]\!\right] = \frac{1}{\sqrt{2}}\begin{bmatrix}1&1\\1&-1\end{bmatrix}$$

For simplicity, we introduce the following additional notation:



Equations in ZXW consist of applying diagrammatic rewrites that prove equalities of the underlying matrices. The complete rule set is given in Appendix A. Several important lemmas are found in Appendix B.

2.2 Controlled Diagrams

Following [25], we cover the definitions of controlled states and controlled square matrices, and arithmetic on them. Note that this is a different definition of controlled states to Ref. [15] in which controlling on $|0\rangle$ is $|+\rangle^{\otimes n}$ instead of $|0\rangle^{\otimes n}$; this choice appears to make a substantial difference in the algebraic properties, which we discuss in Remark 2.

Definition 2.1. For an arbitrary square matrix M, a controlled matrix of M is a diagram $\tilde{M}$ such that:

(2.1)

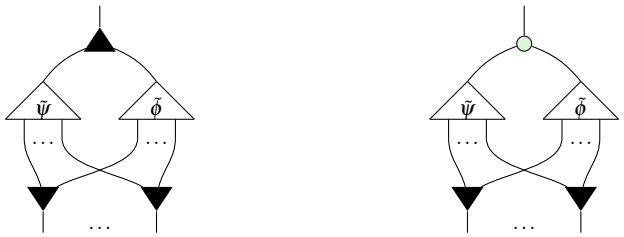
Definition 2.2. For an arbitrary state ψ, a controlled state of ψ is a diagram $\tilde{\psi}$ such that:

(2.2)

Proposition 1 (Propositions 3.3 and 3.4 of [25])**.** Given controlled matrices $\tilde{M}_1, ..., \tilde{M}_k$ and $c_1, ..., c_k \in \mathbb{C}$, the controlled square matrices $\widetilde{\Pi_i M_i}$ and $\widetilde{\Sigma_i c_i M_i}$ are respectively given by

The addition and multiplication of controlled states are defined similarly to controlled matrix arithmetic, except that a layer of  s are appended at the bottom to preserve the number of outputs. The role of  is to *copy* controlled diagrams, as we will show in Section 3.1.

Proposition 2. Given controlled states $\tilde{\psi}$ and $\tilde{\phi}$, we define addition $\tilde{\psi} \boxplus \tilde{\phi}$ and multiplication $\tilde{\psi} \boxtimes \tilde{\phi}$ operations on them to result in the controlled states:



Remark 1. Ref. [25] defined this addition with red spiders at the bottom instead of ▼ s; the linear map is the same, being $\widetilde{\phi + \psi}$. In choosing to use ▼, we will soon define the arithmetic fragment of the ZW-calculus.

Removing the ▼ 's from the bottom of this multiplication gives the controlled diagram for the tensor product in Hilbert space $\widetilde{\psi \otimes \phi}$, as noted in Ref. [15].

Although the interpretation of $\tilde{\psi} \boxtimes \tilde{\phi}$ is not a nice expression in terms of ψ and ϕ, what we will show in this work is that this multiplication is not that of the linear maps, but of multiplication in the ring of *multilinear polynomials* which we will identify with these controlled diagrams.

3 Ring Axioms for Controlled Diagrams

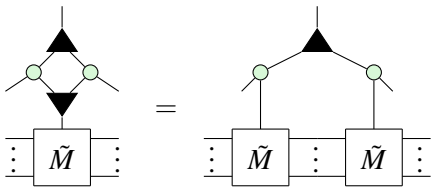
In this section, we reverse-engineer the underlying algebraic properties of controlled state and controlled square matrix diagrams. This builds up to diagrams for the unique normal form for states used for the first proofs of complete axiomatisation for qubit graphical calculi [13, 14].

All proofs in this section can be found in Appendix C.

3.1 Rings

Let $\tilde{M}_n$ be the set of controlled square matrices on n qubits. The goal of this section is to prove that the addition and multiplication operations introduced above induce a ring on $\tilde{M}_n$. By Proposition 1, the addition and multiplication of controlled matrices is just the controlled addition and multiplication of the underlying matrices so the fact that the ring properties hold is not particularly surprising. What is more interesting is how easily these properties can be proven with a small subset of the ZXW rules. Likewise, we show that the set of controlled n-qubit states $\tilde{S}_n$ also forms a ring. The first lemma enables us to copy controlled matrices.

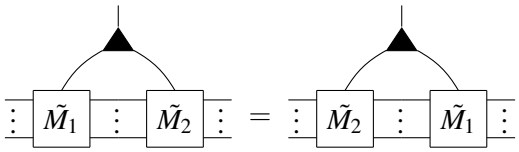
Lemma 3.1. *For any square matrix M,*



(3.1)

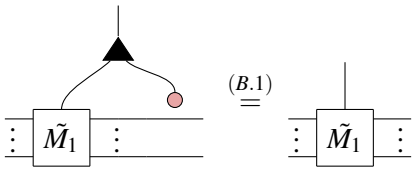
Now we show that controlled matrix addition and multiplication satisfy the ring axioms. Associativity of $+, \times$ follow immediately from (Aso, S1), respectively. Commutativity of addition follows from the commutativity of matrix addition and Proposition 1.

Lemma 3.2. *Let M_1, M_2 be $n \times n$ matrices.*



(3.2)

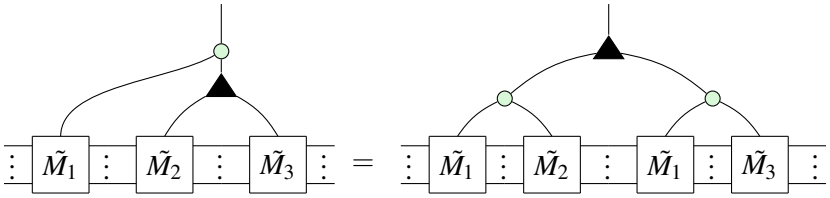
Lemma 3.3. *The additive identity is defined as* $\otimes I_n$*:*



The multiplicative identity is defined very similarly as $\otimes I_n$. The existence of additive inverses relies on the copying lemma from before.

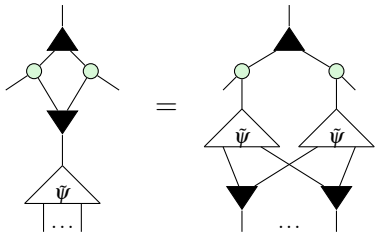
Lemma 3.4. *The additive inverse of $\tilde{M}$ is* $\boxed{-1} \circ \tilde{M}$.

Lemma 3.5. *The addition and multiplication operations of controlled matrices distribute:*



Combining the lemmas of this section shows that controlled matrices form a ring. A similar result can be shown for controlled states. Once again, we start with the ability to copy controlled states.

Lemma 3.6. *For any state ψ,*



(3.3)

Many of the ring axioms follow directly from basic ZXW rules. For example we can show commutativity of addition as follows:

Lemma 3.7. *For n-partite states ψ_1, ψ_2, $\tilde{\psi}_1 \boxplus \tilde{\psi}_2 = \tilde{\psi}_2 \boxplus \tilde{\psi}_1$.*

Associativity of $\boxplus$ follows similarly, using (Aso). Next we have the additive identity.

Lemma 3.8. $\tilde{\psi} \boxplus \tilde{\mathbf{0}} = \tilde{\psi}$.

The additive inverse is defined similarly to the case of controlled matrices.

Lemma 3.9. *For a controlled state $\tilde{\psi}$, its additive inverse is* $\tilde{\psi} \circ \boxed{-1}$.

Associativity and commutativity of $\boxtimes$ follow as before, using (S1) for  . Finally, we must prove distributivity.

Lemma 3.10. $\tilde{\psi}_1 \boxtimes (\tilde{\psi}_2 \boxplus \tilde{\psi}_3) = (\tilde{\psi}_1 \boxtimes \tilde{\psi}_2) \boxplus (\tilde{\psi}_1 \boxtimes \tilde{\psi}_3)$.

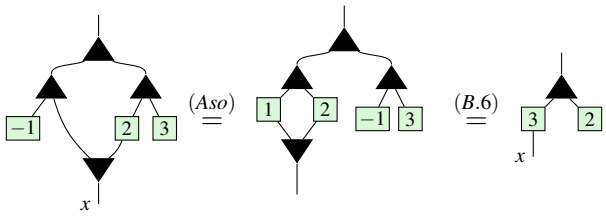
Remark 2. A different addition and multiplication for controlled states was defined in Ref. [16]. There corresponded to entry-wise addition and multiplication of statevectors, while our $\boxplus$ and $\boxtimes$ correspond to addition and multiplication of polynomials in bijective correspondence to controlled states, which we show next.

4 Isomorphism between the Ring of Controlled States and Multilinear Polynomials

Its been known since 2011 that  ,  can be used to add and multiply number states  , respectively [9]. In the previous section we saw that  ,  can moreover be used to copy controlled diagrams. In this section, we explain this connection by demonstrating that controlled states are in fact isomorphic to multilinear polynomials. This being a bijection is a well-known folklore result in the study of entangled states, but to the best of our inquiries we are not aware of a proof. More generally, Ref. [34] presented Cartesian Distributive Categories exemplified by polynomial circuits, which are isomorphic to polynomials over arbitrary commutative semirings or rings; their proof is non-constructive, giving explicit proof only for the case of Boolean circuits [33]. Our proof hinges on a normal form inspired by the recent proof of completeness for the ZXW calculus [19], suggesting that much of the expressive power of the ZXW calculus comes from this algebraic structure.

Firstly, we describe how to interpret certain ZXW diagrams as polynomials. Consider the diagrams:

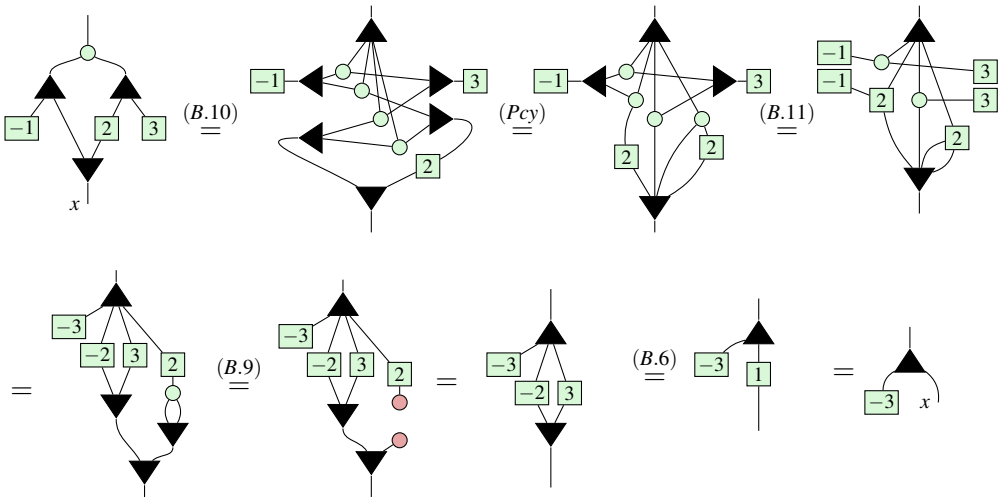
If we treat the bottom wires as an indeterminate x, we can read these bottom-up as computing $x-1$ and $2x+3$, respectively. Moreover, since these diagrams are both controlled states, they can be added together, yield a diagram resembling $3x+2$:



(4.1)

When trying to multiply these diagrams, rather than getting $(x-1)(2x+3) = 2x^2+x-3$, we instead

get $x-3$.



(4.2)

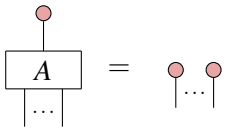
The reason for the missing $2x^2$ term is that Lemma B.9 implies $x^2 = 0$. Other than that, controlled state arithmetic appears to faithfully reflect polynomial arithmetic. To help formalise this correspondence, we introduce the following definition.

Definition 4.1. A ZXW diagram with a single input on top is **arithmetic** if it contains only $\mid$, $\times$ wires, $\wedge^n$, $\curlyvee^n$, $\curlyvee_n$ nodes and $\boxed{a}$ boxes.

Remark 3. This fragment of the ZXW-calculus defines a subcategory; adding $\circ$, this is an instance of a Cartesian Distributive Category as defined in Ref. [34].

To interpret an arithmetic ZXW diagram as an arithmetic expression, read $\wedge^n$ as $+$, $\curlyvee_n$ as $\times$, $\boxed{a}$ as the number a, $\curlyvee^n$ as fanout and output/bottom wires as variables $x_1,...,x_n$ numbered from left to right. The following lemma establishes that all arithmetic diagrams are controlled states:

Lemma 4.1. *For any arithmetic diagram A,*

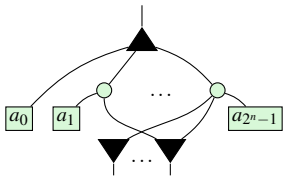


(4.3)

Proof. By definition, other than wires A contains only $\wedge^n$, $\curlyvee_n$, $\curlyvee^n$, and $\boxed{a}$. All $\boxed{a}$'s can be removed with (Ept). Meanwhile all the spiders copy $\circ$ due to (Bs0, K0, B.1) respectively. □

Just as it is typical to represent a polynomial in normal form as a sum of products, it is possible to rewrite every arithmetic diagram into a normal form as a single $\wedge^n$, followed by a layer of $\curlyvee_n$, followed by a layer of $\boxed{a}$, $\curlyvee^n$.

Definition 4.2. An n-output arithmetic diagram is said to be written in **polynormal form** (PNF) if it is of the form:



(4.4)

The ith coefficient a_i is connected to the kth  iff the kth bit in the binary expansion of i is 1.

This normal form is familiar from completeness of all linear maps for qubits [14] and for qudits [19]. The reason we introduce the definition of a PNF is that it is an arithmetic diagram and therefore has a more immediate arithmetic interpretation. The reason for the specific connectivity condition is that it enables a diagram in PNF to directly represent its own matrix.

Proposition 3. The diagram in equation (4.4) equals the matrix $\begin{bmatrix} 1 & a_0 \\ 0 & a_1 \\ ... & ... \\ 0 & a_{2^n-1} \end{bmatrix}$

Proof. See Appendix C. □

Thus, every controlled state can be represented as at least one arithmetic diagram (namely, its PNF). Moreover, we now show that any other arithmetic diagram can always be rewritten to its PNF.

Theorem 4.1. *All arithmetic diagrams can be rewritten into PNF through application of ZXW rules. Therefore, those ZXW-calculus rules applied suffice for completeness for the arithmetic fragment of the ZXW-calculus.*

Proof. An algorithm to rewrite any arithmetic diagram to PNF is given in Appendix C. □

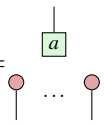
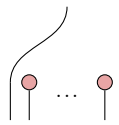
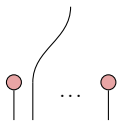
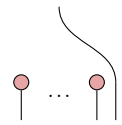
4.1 Isomorphism

At last we can prove the isomorphism. Recall that $\tilde{S}_n$ is the ring of controlled states with n outputs. Throughout, we shall let $\mathscr{P}_n$ denote the multilinear ring $\mathbb{C}[x_1,...,x_n]/(x_1^2,...,x_n^2)$.

Theorem 4.2. *There is an isomorphism $\mathscr{P}_n \simeq \tilde{S}_n$*

First, we shall define the map $\phi_n : \mathscr{P}_n \to \tilde{S}_n$ before proving it induces an isomorphism. ϕ_n is defined to map an arbitrary polynomial $p(x_1,...,x_n) = a_0 + a_1x_n + ... + a_{2^n-1}x_1x_2...x_n$ to the PNF in equation (4.4).

Some important special cases are mapping scalars $a \in \mathbb{C}$ and indeterminates x_i:

$\phi_n(a) =$  $\phi_n(x_1) =$  $, \phi_n(x_2) =$  $, ..., \phi_n(x_n) =$ 

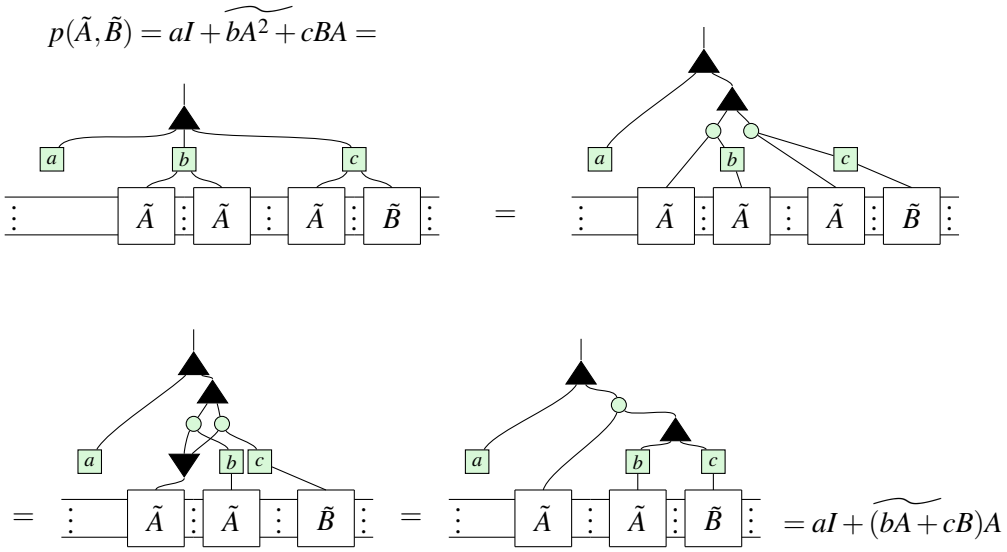
5 Completeness for Factoring Controlled Operators

Instead of the indeterminates being complex numbers represented by $\boxed{a}$'s, we can let them be same-size matrices represented by controlled square matrix diagrams. We then have that:

Theorem 5.1. *ZXW diagrams where the outputs of an arithmetic ZW diagram are each plugged into controls of same-size controlled matrices, are isomorphic to multivariate polynomials over same-size square matrices with complex number coefficients. The rules for their completeness are the same subset of ZXW rules used for completeness for arithmetic diagrams in the ZXW-calculus in Theorem 4.1, plus the controlled square matrix as a generator along with the four rewrite rules for it in Definition 2.1, Lemma 3.1, and Lemma 3.2.*

Proof. The proof is by the same algorithm for rewriting to PNF as Theorem 4.1, modifying step (6) to copy controlled square matrices using Lemma 3.1, using Lemma 3.2 to commute controlled square matrices whose controls act on mutually exclusive sectors. □

As an application, we leverage both our rewrite rules for arithmetic ZW diagrams, and for controlled diagrams, to *factor* them. For example, for same size square matrices I, A, B and $a, b, c \in \mathbb{C}$:



Factoring Hamiltonians is important to optimise quantum algorithms for chemistry and physics simulations. However, previous graphical rewrites for factoring Hamiltonians had only been doable for Hamiltonians with concretely-specified matrix terms [25]. This completeness result guarantees that for any Hamiltonian, even if its matrix terms are black-box, these graphical rewrite rules are capable of deriving any of its possible factorisations.

6 Complexity

There should not exist any efficient algorithm for rewriting arbitrary ZXW diagrams to polynormal form, as such an algorithm would efficiently compute all matrix elements of any quantum circuit given as a ZX, ZW, or ZXW diagram—for instance any Clifford+T circuit. A natural question is whether it is possible to efficiently rewrite arbitrary ZXW diagrams to *any* mathematically equal arithmetic ZW diagram.

This section examines the significance of translating quantum circuits into polynomials from the perspective of computational complexity. In particular, since there is an efficient (randomised) algorithm for polynomial identity testing, we show it is unlikely there is an efficient algorithm for rewriting an arbitrary ZXW diagram into an arithmetic ZW diagram.

6.1 Algebraic Circuits

While boolean complexity theory studies the number of AND/OR gates required to compute a given function, algebraic circuit complexity studies the number of addition/multiplication operations required to compute a given polynomial. More formally, we have the following definition from [26].

Definition 6.1. An algebraic circuit C over n variables and some field $\mathbb{F}$ is a directed acyclic graph with a single root node, with leaves labelled by constants $c \in \mathbb{F}$ or variables x_i, and internal nodes labelled by $+$ or $\times$ gates. The root of C is identified with the polynomial $p \in \mathbb{F}[x_1,...,x_n]$ that is computed by C.

Definition 6.2. Given an arithmetic circuit C that describes a polynomial $p(x_1,...,x_n) \in \mathbb{F}[x_1,...,x_n]$, the polynomial identity testing problem (PIT) is to decide whether $p = 0$.

PIT can be used to check whether two polynomials are equal since $p = q \iff p - q = 0$. More surprising examples of problems that reduce to PIT are bipartite perfect matching in graphs and primality testing [24]. Thanks to the Schwartz-Zippel Lemma, there is an efficient randomised algorithm for PIT which simply evaluates the polynomial on random inputs and checks whether all of them are zero. More precisely, $PIT \in \mathsf{coRP}$, a complexity class conjectured to equal P.

By Theorem 4.2, we can interpret any quantum state as an algebraic circuit. Therefore, PIT can be used to compare quantum states. The question is how easily this can be done.

6.2 Proof Complexity

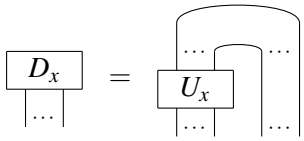
In the original ZW completeness paper for qubits [14] (as well as the original ZXW completeness paper for arbitrary finite dimensions [19]), the proof of completeness centres around a unique normal form equivalent to the one used in Section 4. This approach to proving completeness is by showing that any state can be rewritten to a unique normal form using the ZXW rules. By simply reversing the rules, one can show equality or inequality between any two diagrams. Since the normal form is a direct representation of the diagram's statevector, it may be exponentially larger than the starting diagram; hence, the proof of completeness gives an exponential upper bound for the length of proofs of equality in the ZXW-calculus. A possible strategy for speeding this process up would be to rewrite the starting diagrams into compact arithmetic diagrams, and then use PIT to compare them. We now prove this impossible, under standard complexity assumptions.

Proposition 4. If all quantum states can be rewritten to arithmetic diagrams in polynomial time, then $\mathsf{RP} = \mathsf{NQP}$.

Proof. Suppose that we can efficiently rewrite quantum states to arithmetic diagrams. Then we can use this to reduce an NQP-complete problem to $\overline{PIT} \in \mathsf{RP}$ (which implies $PIT \in \mathsf{coRP}$). Since $\mathsf{RP} \subseteq \mathsf{NP} \subseteq \mathsf{NQP}$, this implies $\mathsf{RP} = \mathsf{NP}$.

The NQP-complete problem we are considering is the exact non-identity problem (ENI) which decides whether some unitary does not compute the identity matrix [28]. Let x be a description of a unitary quantum circuit U_x over $n = poly(|x|)$ qubits. Then by translating each gate of U_x into a constant ZXW

diagram, we can rewrite U_x into a ZXW diagram in polynomial time. Now bend the ZXW diagram into a state D_x, as below.



Now apply the assumption to rewrite D_x into an arithmetic diagram A_x. Interpret A_x as a polynomial p_x over variables $x_1,...,x_n,y_1,...y_n$. The polynomial for the bent identity is represented by the linear sized arithmetic circuit $p^n_\cap := \prod_{i=1}^n (1 + x_i y_{n-i+1})$. So $p_x = p^n_\cap \iff U_x = I$ which means $\overline{PIT}(p_x - p^n_\cap) = 1 \iff ENI(U_x) = 1$. Thus we have reduced ENI to $\overline{PIT}$.

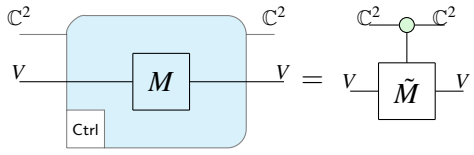
□

Note that under standard derandomisation assumptions, $\mathsf{P} = \mathsf{RP} = \mathsf{coRP}$. Since $\mathsf{NP} \subseteq \mathsf{NQP}$ (an inclusion believed to be strict), then $\mathsf{P} \neq \mathsf{NP}$ implies it is not possible to rewrite arbitrary ZXW diagrams into arithmetic diagrams. An interesting open question is whether there nevertheless exists a small arithmetic circuit for every quantum state. If so, this may have interesting applications for algebraic complexity theory.

7 The Control Higher-Order Map

7.1 Functorial Boxes

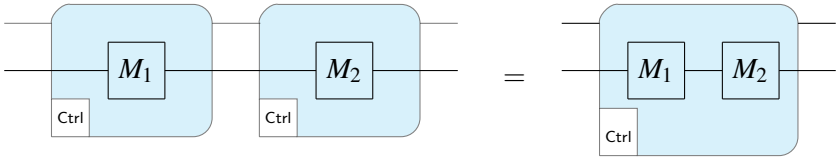
We define the higher-order map Ctrl which takes a square matrix $M : V \to V$ to its controlled square diagram $V \otimes \mathbb{C}^2 \to V \otimes \mathbb{C}^2$. In the functorial box notation of [18], we write:



(7.1)

We prove in appendix C that composing controlled operations behaves nicely.

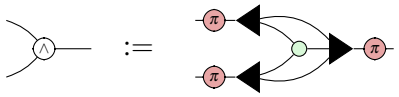
Proposition 5.



7.2 Multiple Control

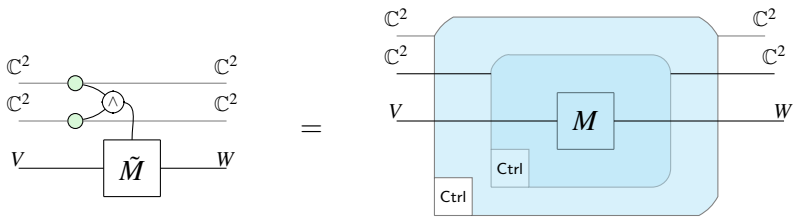
This section proves that controlling a controlled diagram gives the AND of the control wires. First we show how to represent the binary AND gate in the ZXW-calculus, and check it in Appendix C. This construction is by deMorgan's law of a diagram for binary OR from Proposition 3.

Lemma 7.1.



Multiple applications of Ctrl ANDs the controls, proved in Appendix C.

Proposition 6.



8 Conclusion

To conclude, we proved completeness for all controlled n-partite states, which we showed form a commutative ring isomorphic to multilinear polynomials. Also, we showed that all controlled n-qubit square matrices form a non-commutative ring. Furthermore, we have completeness for plugging controlled states into the control wires of controlled diagrams, isomorphic to all multivariate polynomials over same-size square matrices, with application to factoring Hamiltonians. When the controls target mutually exclusive sectors, a rewrite rule can be applied to copy any controlled diagram, and thereby factor any Hamiltonian.

We showed that it is unlikely that arbitrary quantum states or circuits can be efficiently rewritten to any arithmetic ZXW diagram, as this would imply $\mathsf{RP} = \mathsf{NQP}$. This opens up connections between quantum circuit complexity and the far better understood algebraic complexity. We have shown that every (controlled) state computes a polynomial; hence, we can interpret a universal fragment of the ZXW-calculus as corresponding to arithmetic circuits. This generalises Ref. [4], which found an algebraic interpretation of a certain fragment of ZW calculus. This line of reinterpreting quantum circuits as computing polynomials rather than unitary matrices offers a new perspective on quantum computation.

In another direction, we can apply completeness for polynomials isomorphic to controlled states to study entanglement. It can be easily shown diagrammatically that the polynomials corresponding to entangled (non-separable) states are exactly those that cannot be factored into irreducibles containing only variables corresponding to Alice's subsystem or only corresponding to Bob's subsystem. Since there are efficient algorithms for polynomial factorisation [11], this gives rise to a novel entanglement classification algorithm for pure states. Further developing this into a more refined algebraic theory of entanglement, building on the work in Ref. [1], could offer further insights.

The natural next step is extend our results to controlled *qudit* diagrams. While the diagrams being controlled are over qudits, we can consider control in the qubit subspace, as done in the ZXW-calculus completeness proof for any qudit dimension [19]. A starting guess would be that qudit controlled states are isomorphic to polynomials $\mathbb{C}^{d-1}[x_1,...,x_n]/({x_1}^d,...,{x_n}^d)$ due to the Hopf law between Z and W. Qudit multiple-control would likely have more complex structure than the qubit case here, considering the constructions for all prime-dimensional d-ary classical reversible gates built in Ref. [22].

Last, we are interested in exploring how to embed these new semantics for quantum controlled states and matrices into a functional programming language like in Ref. [21], or translated to an equational theory for a quantum programming language like in Ref. [27]. We would like to try sector-preserving channels [30] and scoped effects [17] as approaches to better formulate the semantics of multiple-control. We are also curious about reconciling the interpretation of diagrammatic differentiation of our arithmetic polynomial circuits by the approach in Ref. [34], with that of quantum circuits and ZX diagrams in Refs. [29, 32, 15].

9 Acknowledgements

We are grateful to all the collaboration with Matt Wilson in pointing out the functorial properties of control. We thank Razin Shaikh, Itai Leigh, Tim Forrer, Aleks Kissinger, and Frank (Peng) Fu for insightful discussions. LY is thankful for funding from the Google PhD Fellowship. RY would like to thank Simon Harrison for his generous support via the Wolfson Harrison UKRI Quantum Foundation Scholarship.

References

[1] AGNEW, E. Quantum Polynomials in the ZXW Calculus. Master's thesis, University of Oxford, 2023.

[2] BACKENS, M., AND KISSINGER, A. ZH: A Complete Graphical Calculus for Quantum Computations Involving Classical Non-linearity. In *Proceedings of the 15th International Conference on Quantum Physics and Logic, Halifax, Canada, 3-7th June 2018* (2019), P. Selinger and G. Chiribella, Eds., vol. 287 of *Electronic Proceedings in Theoretical Computer Science*, Open Publishing Association, pp. 23–42.

[3] BAEZ, J. C., AND ERBELE, J. Categories in control, 2015.

[4] CARETTE, T., MOUTOT, E., PEREZ, T., AND VILMART, R. Compositionality of planar perfect matchings, 2023.

[5] CHARDONNET, K., DE VISME, M., VALIRON, B., AND VILMART, R. The many-worlds calculus, 2023.

[6] CHILDS, A. M., AND WIEBE, N. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Info. Comput. 12*, 11–12 (nov 2012), 901–924.

[7] COECKE, B., AND DUNCAN, R. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics 13* (2011), 043016.

[8] COECKE, B., AND KISSINGER, A. The compositional structure of multipartite quantum entanglement. In *International Colloquium on Automata, Languages, and Programming* (2010), Springer, pp. 297–308.

[9] COECKE, B., KISSINGER, A., MERRY, A., AND ROY, S. The ghz/w-calculus contains rational arithmetic. *arXiv preprint arXiv:1103.2812* (2011).

[10] DE FELICE, G., SHAIKH, R. A., POÓR, B., YEH, L., WANG, Q., AND COECKE, B. Light-matter interaction in the zxw calculus. *arXiv preprint arXiv:2306.02114* (2023).

[11] FORBES, M. A., AND SHPILKA, A. Complexity theory column 88: Challenges in polynomial factorization. *ACM SIGACT News 46*, 4 (2015), 32–49.

[12] GILYÉN, A., SU, Y., LOW, G. H., AND WIEBE, N. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (New York, NY, USA, 2019), STOC 2019, Association for Computing Machinery, p. 193–204.

[13] HADZIHASANOVIC, A. The algebra of entanglement and the geometry of composition, 2017.

[14] HADZIHASANOVIC, A., NG, K. F., AND WANG, Q. Two complete axiomatisations of pure-state qubit quantum computing. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science* (New York, NY, USA, 2018), LICS '18, Association for Computing Machinery, p. 502–511.

[15] JEANDEL, E., PERDRIX, S., AND VESHCHEZEROVA, M. Addition and differentiation of zx-diagrams, 2024.

[16] JEANDEL, E., PERDRIX, S., AND VILMART, R. A generic normal form for zx-diagrams and application to the rational angle completeness, 2018.

[17] LINDLEY, S., MATACHE, C., MOSS, S., STATON, S., WU, N., AND YANG, Z. Scoped effects as parameterized algebraic theories, 2024.

[18] MELLIÈS, P.-A. Functorial boxes in string diagrams. In *International Workshop on Computer Science Logic* (2006), Springer, pp. 1–30.

[19] POÓR, B., WANG, Q., SHAIKH, R. A., YEH, L., YEUNG, R., AND COECKE, B. Completeness for arbitrary finite dimensions of zxw-calculus, a unifying calculus. *arXiv preprint arXiv:2302.12135* (2023).

[20] RALL, P. Quantum algorithms for estimating physical quantities using block encodings. *Phys. Rev. A 102* (Aug 2020), 022408.

[21] RENNELA, M., AND STATON, S. Classical Control, Quantum Circuits and Linear Logic in Enriched Category Theory. *Logical Methods in Computer Science Volume 16, Issue 1* (Mar. 2020).

[22] ROY, P., VAN DE WETERING, J., AND YEH, L. The qudit zh-calculus: Generalised toffoli+hadamard and universality. *Electronic Proceedings in Theoretical Computer Science 384* (Aug. 2023), 142–170.

[23] SATI, H., AND SCHREIBER, U. The quantum monadology, 2023.

[24] SAXENA, N. Progress on polynomial identity testing. *Bull. EATCS 99* (2009), 49–79.

[25] SHAIKH, R. A., WANG, Q., AND YEUNG, R. How to sum and exponentiate hamiltonians in zxw calculus. *arXiv preprint arXiv:2212.04462* (2022).

[26] SHPILKA, A., YEHUDAYOFF, A., ET AL. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends® in Theoretical Computer Science 5*, 3–4 (2010), 207–388.

[27] STATON, S. Algebraic effects, linearity, and quantum programming languages. *SIGPLAN Not. 50*, 1 (jan 2015), 395–406.

[28] TANAKA, Y. Exact non-identity check is nqp-complete. *International Journal of Quantum Information 8*, 05 (2010), 807–819.

[29] TOUMI, A., YEUNG, R., AND DE FELICE, G. Diagrammatic differentiation for quantum machine learning. *Electronic Proceedings in Theoretical Computer Science 343* (Sept. 2021), 132–144.

[30] VANRIETVELDE, A., AND CHIRIBELLA, G. Universal control of quantum processes using sector-preserving channels. *Quantum Information and Computation 21*, 15 & 16 (Nov. 2021), 1320–1352.

[31] VANRIETVELDE, A., KRISTJÁNSSON, H., AND BARRETT, J. Routed quantum circuits. *Quantum 5* (2021), 503.

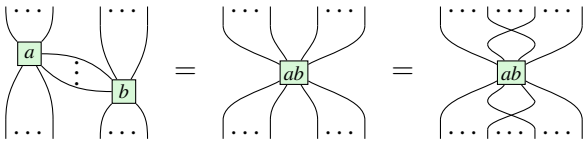
[32] WANG, Q., YEUNG, R., AND KOCH, M. Differentiating and integrating zx diagrams with applications to quantum machine learning, 2022.

[33] WILSON, P., AND ZANASI, F. Reverse derivative ascent: A categorical approach to learning boolean circuits. *Electronic Proceedings in Theoretical Computer Science 333* (Feb. 2021), 247–260.

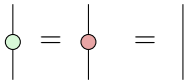
[34] WILSON, P., AND ZANASI, F. An axiomatic approach to differentiation of polynomial circuits. *Journal of Logical and Algebraic Methods in Programming 135* (2023), 100892.

Appendix A ZXW rules

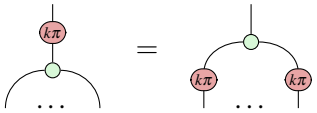
ZX Rules:



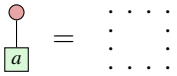
(S1)



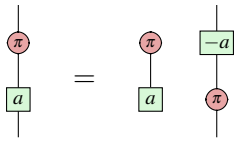
(S2)



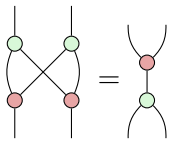
(K1)



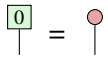
(Ept)



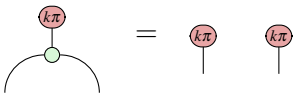
(K2)



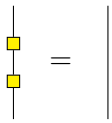
(B2)



(Zer)



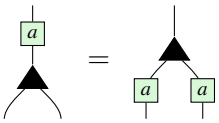
(K0)



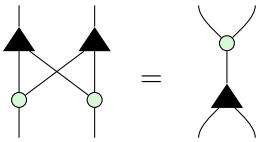
(H)

Where $k \in \{0,1\}$.

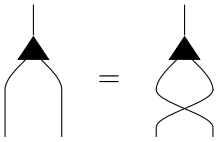
ZW Rules:



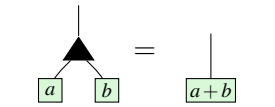
(Pcy)



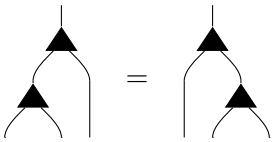
(BZW)



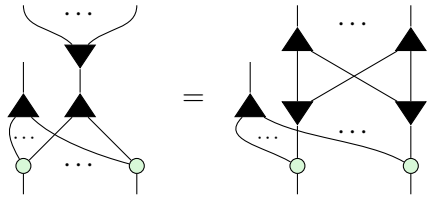
(Sym)



(ADD)

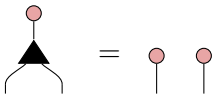


(Aso)

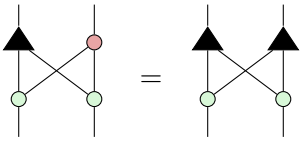


(WW)

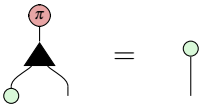
ZXW Rules:



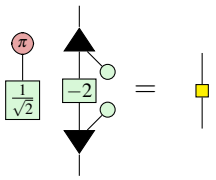
(Bs0)



(TA)



(Bsj)

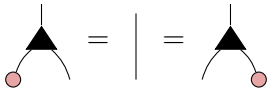


(HD)

Appendix B Basic Lemmas

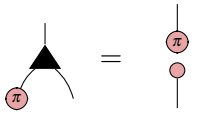
The following two lemmas follow immediately from the bra-ket definition of  :

Lemma B.1.



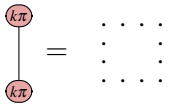
(B.1)

Lemma B.2.



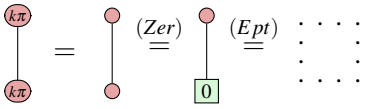
(B.2)

Lemma B.3.



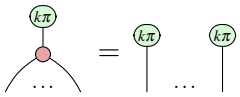
(B.3)

Proof.



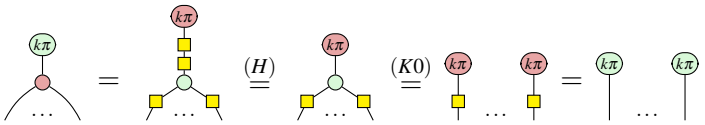
□

Lemma B.4.



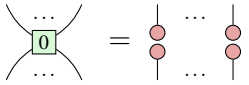
(B.4)

Proof.



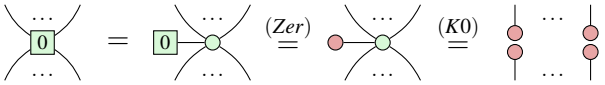
□

Lemma B.5.



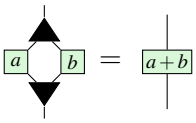
(B.5)

Proof.



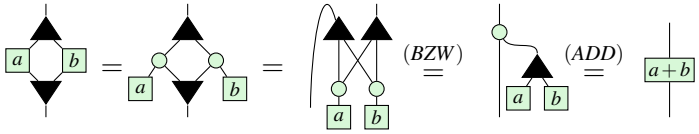
□

Lemma B.6.



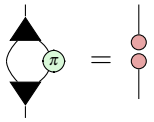
(B.6)

Proof.



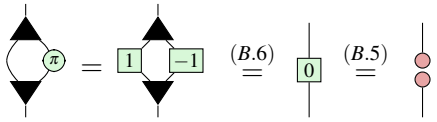
□

Lemma B.7.



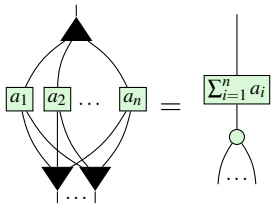
(B.7)

Proof.



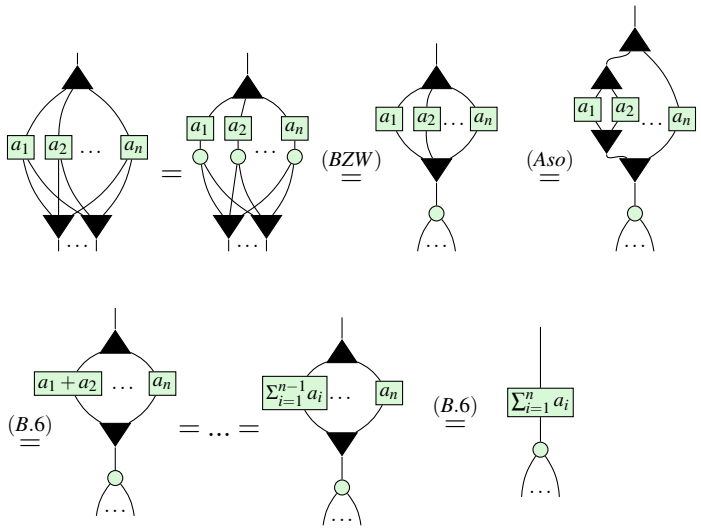
□

Lemma B.8.



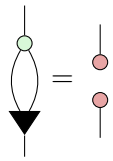
(B.8)

Proof.



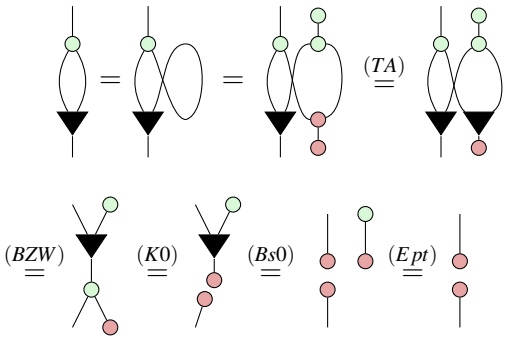
□

Lemma B.9.



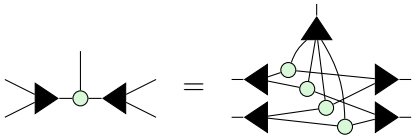
(B.9)

Proof.



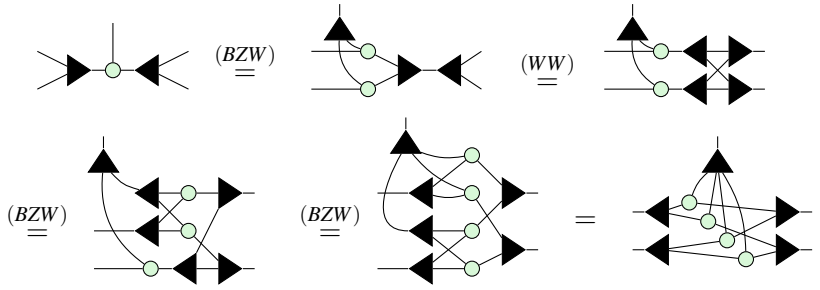
□

Lemma B.10.



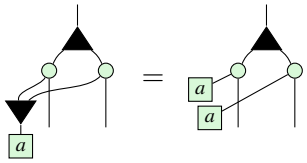
(B.10)

Proof.



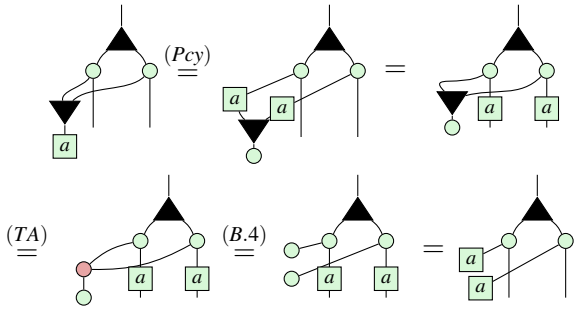
□

Lemma B.11.



(B.11)

Proof.

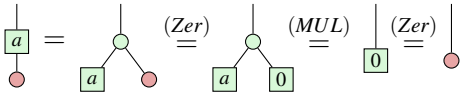


□

Lemma B.12.

(B.12)

Proof.

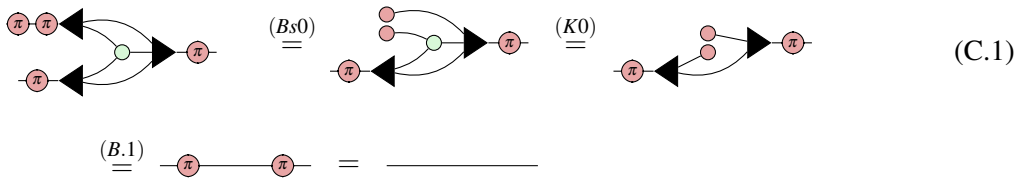


□

Appendix C Main Proofs

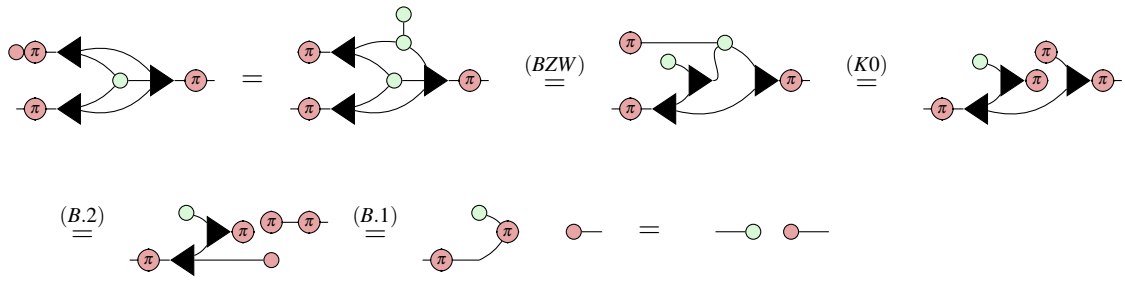
Proof of Lemma 7.1

Proof. We can verify this computes the AND gate by computing on basis states.



(C.1)

Thus $AND(1,x) = x$. Since the diagram is clearly commutative, it remains to check $AND(0,x) = 0$.

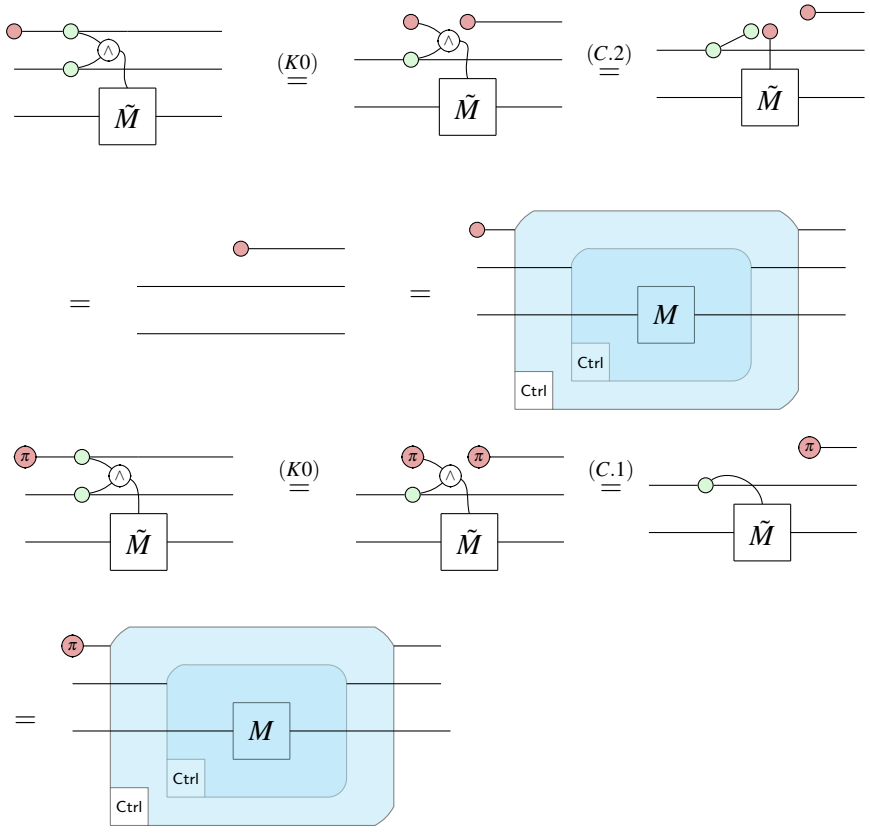


(C.2)

□

Proof of Proposition 6

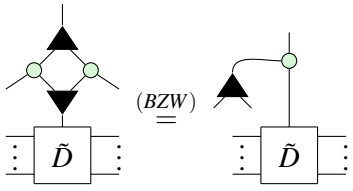
Proof. Plugging basis states:



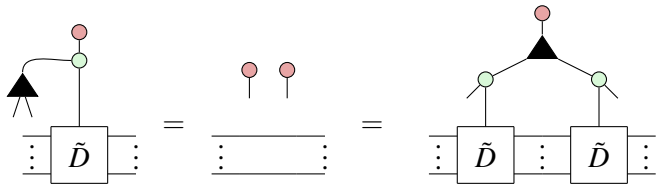
□

Proof of Lemma 3.1

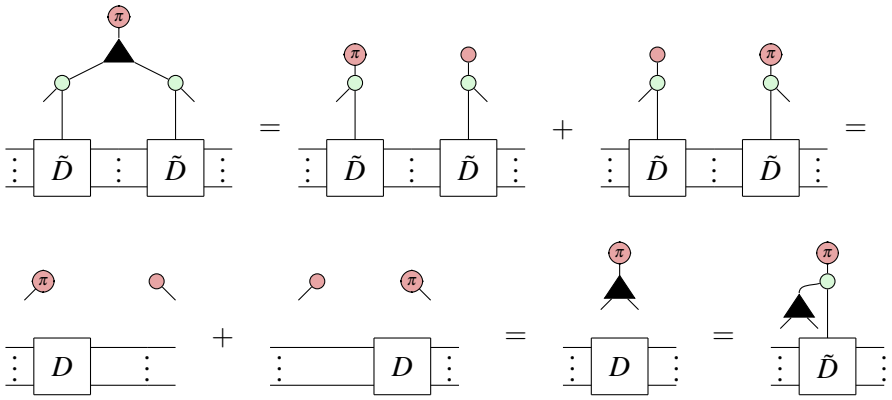
Proof. First of all, using (BZW) we can rewrite the LHS to



Then clearly



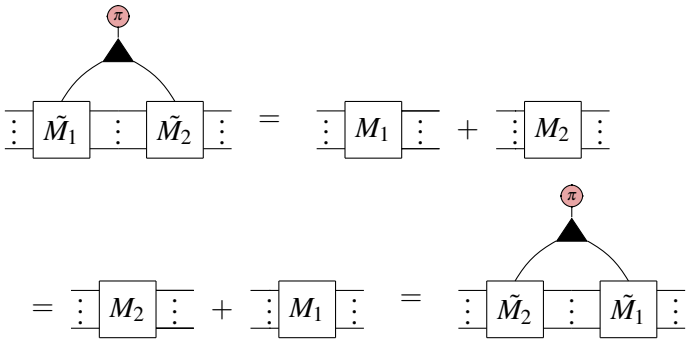
Meanwhile,



Thus the two sides are equal over the Z basis and so are equal as diagrams. □

Proof of Lemma 3.2

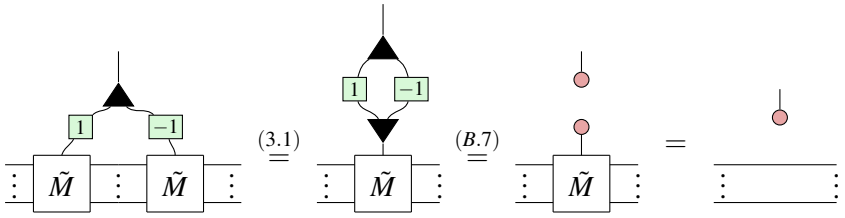
Proof. We prove by plugging red and commutativity of matrix addition. By definition of controlled matrices, plugging  gives I_n on both sides. Meanwhile, plugging  gives:



□

Proof of Lemma 3.4

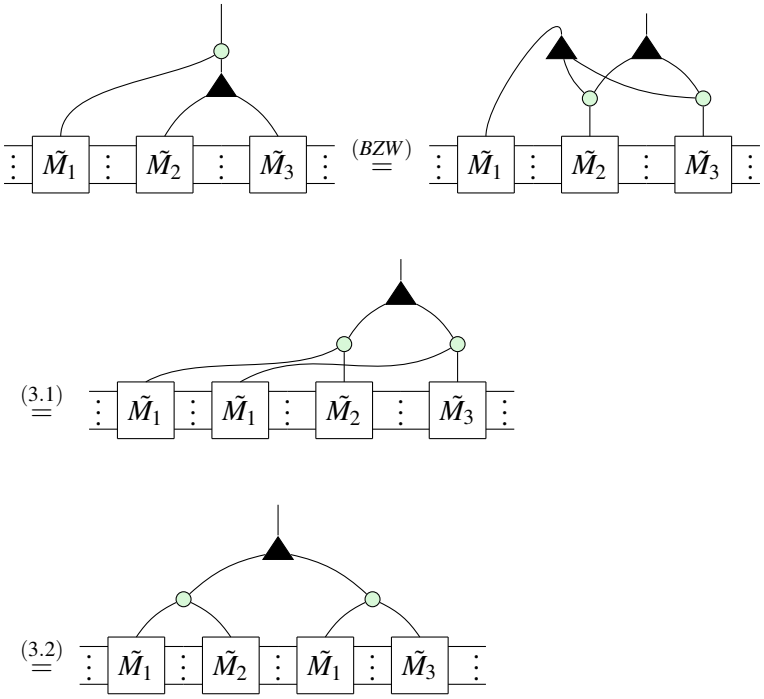
Proof.



□

Proof of Lemma 3.5

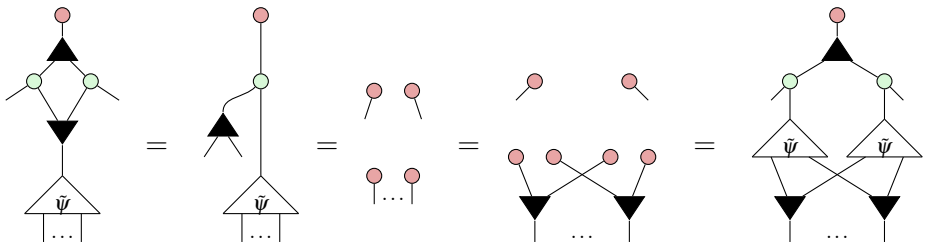
Proof.



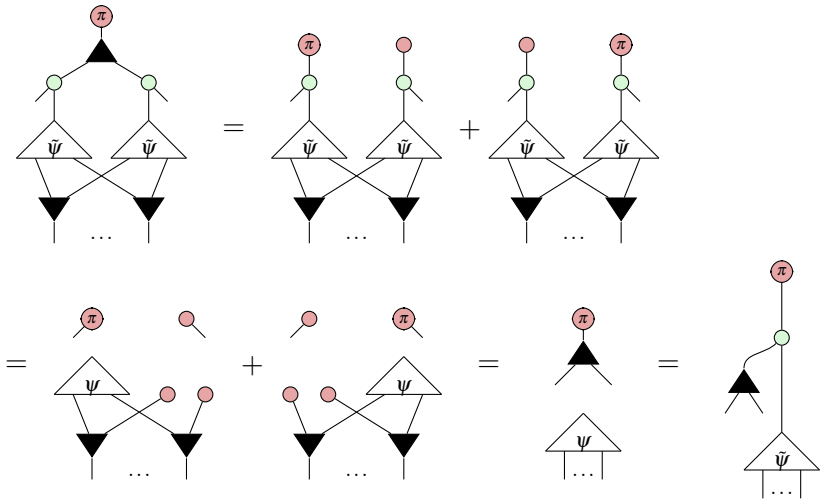
□

Proof of Lemma 3.6

Proof. As before, plugging $|0\rangle$ gives

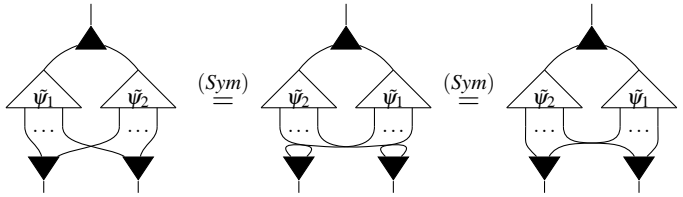


Meanwhile, plugging $|1\rangle$ gives



Completing the proof □

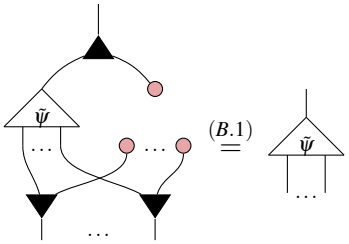
Proof of Lemma 3.7

Proof.  □

Proof of Lemma 3.8

Proof. It is clear that  is the controlled state $\tilde{\mathbf{0}}$.

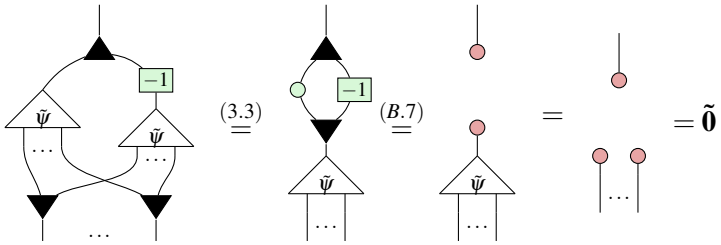
Then we have:



□

Proof of Lemma 3.9

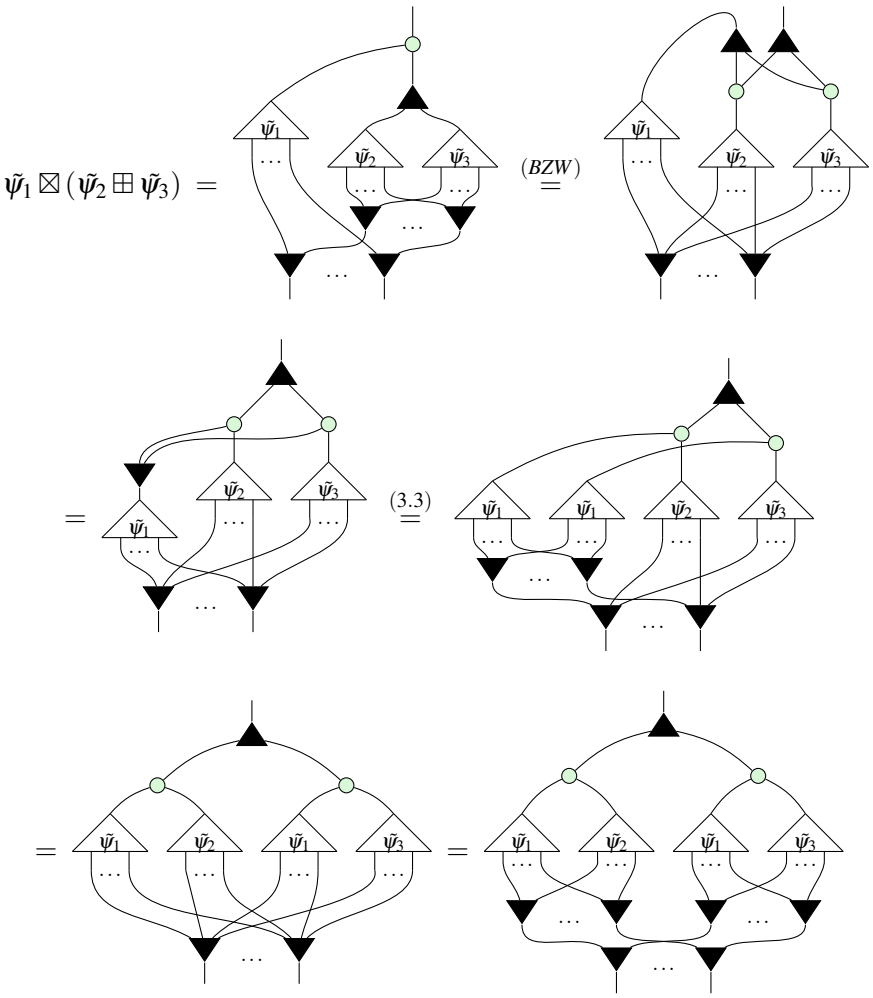
Proof. $\tilde{\psi} \circ$  is still a controlled state since  does nothing to  . Then $\tilde{\psi} \circ$  inverts $\tilde{\psi}$ since:



□

Proof of Lemma 3.9

Proof.



□

Proof of Proposition 3

Proof. We prove by induction on n.

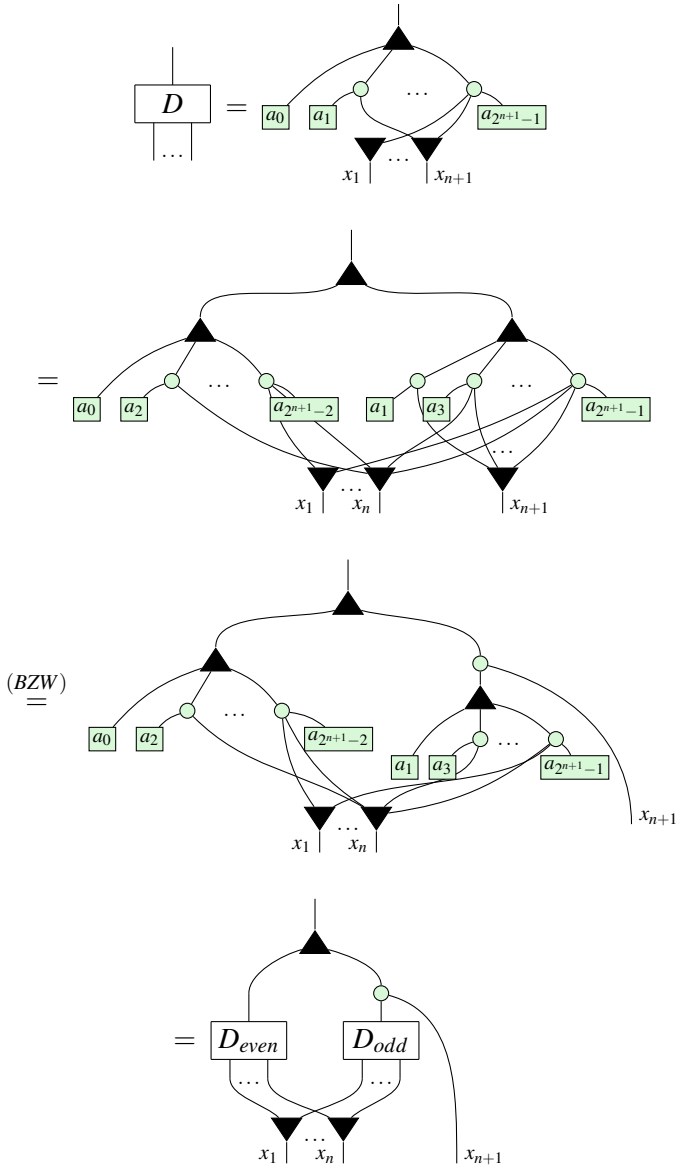
For the base case, $n = 0$. The only PNF with no outputs is a number so we have:

$$\boxed{a_0} = \begin{bmatrix} 1 & a_0 \end{bmatrix}$$

as desired.

For inductive hypothesis, we assume that Proposition 3 holds for every PNF on n outputs. We use this hypothesis to extend it to PNFs with $n+1$ outputs.

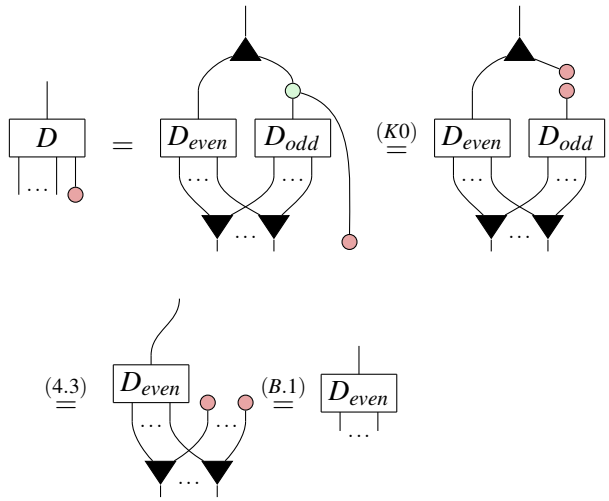
Let D be an arbitrary PNF with $n+1$ outputs. Firstly, observe that x_{n+1} is connected to only the odd coefficients $\{a_{2k+1}\}$ since these are exactly the indices with 1 in the least significant bit. Thus we can rewrite:



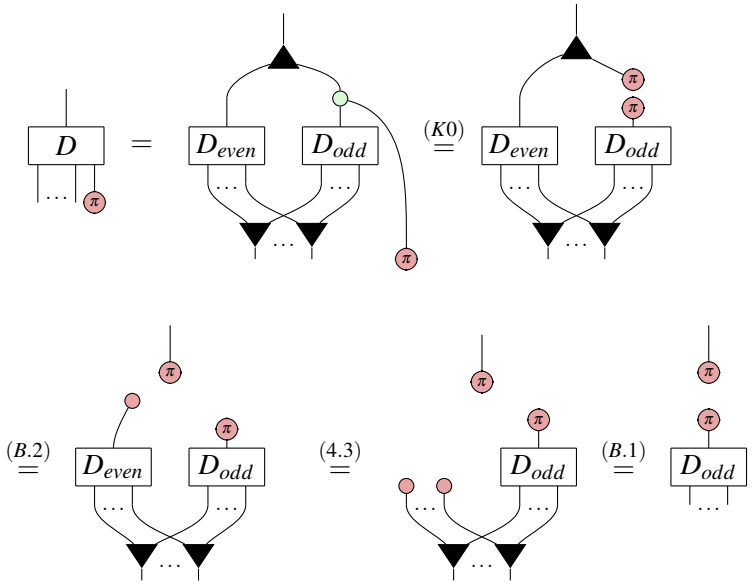
Where D_{even}, D_{odd} are PNF diagrams. Since they are over n variables, we can apply the inductive hypothesis and obtain:

$$D_{even} = \begin{bmatrix} 1 & a_0 \\ 0 & a_2 \\ \dots & \dots \\ 0 & a_{2^{n+1}-2} \end{bmatrix}, D_{odd} = \begin{bmatrix} 1 & a_1 \\ 0 & a_3 \\ \dots & \dots \\ 0 & a_{2^{n+1}-1} \end{bmatrix} \tag{*}$$

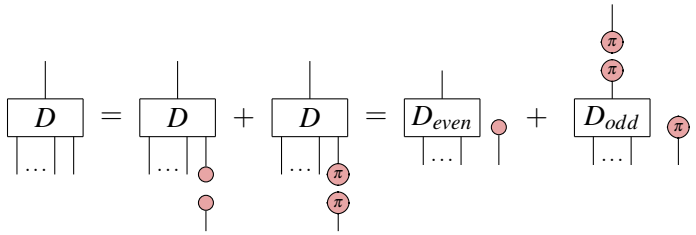
Next, plugging red we observe:



Meanwhile,



Summing these together,



$$= (D_{even} \otimes |0\rangle) + (D_{odd}|1\rangle\langle 1| \otimes |1\rangle)$$

$$\overset{(*)}{=} \begin{bmatrix} 1 & a_0 \\ 0 & a_2 \\ \dots & \dots \\ 0 & a_{2^{n+1}-2} \end{bmatrix} \otimes |0\rangle + \begin{bmatrix} 0 & a_1 \\ 0 & a_3 \\ \dots & \dots \\ 0 & a_{2^{n+1}-1} \end{bmatrix} \otimes |1\rangle$$

$$= \begin{bmatrix} 1 & a_0 \\ 0 & 0 \\ 0 & a_2 \\ 0 & 0 \\ \dots & \dots \\ 0 & a_{2^{n+1}-2} \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & a_1 \\ 0 & 0 \\ 0 & a_3 \\ \dots & \dots \\ 0 & 0 \\ 0 & a_{2^{n+1}-1} \end{bmatrix} = \begin{bmatrix} 1 & a_0 \\ 0 & a_1 \\ 0 & a_2 \\ 0 & a_3 \\ \dots & \dots \\ 0 & a_{2^{n+1}-2} \\ 0 & a_{2^{n+1}-1} \end{bmatrix}$$

Completing the inductive step. □

Proof of Theorem 4.1

Proof. Let A be an arithmetic diagram. If $A =$ 

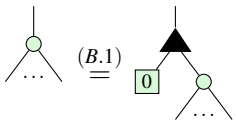
, we are done.

Otherwise, A has at least one output. First, we shall rewrite A into three layers, consisting of: (1) a single W at the top, (2) a layer of  and (3) a layer of 

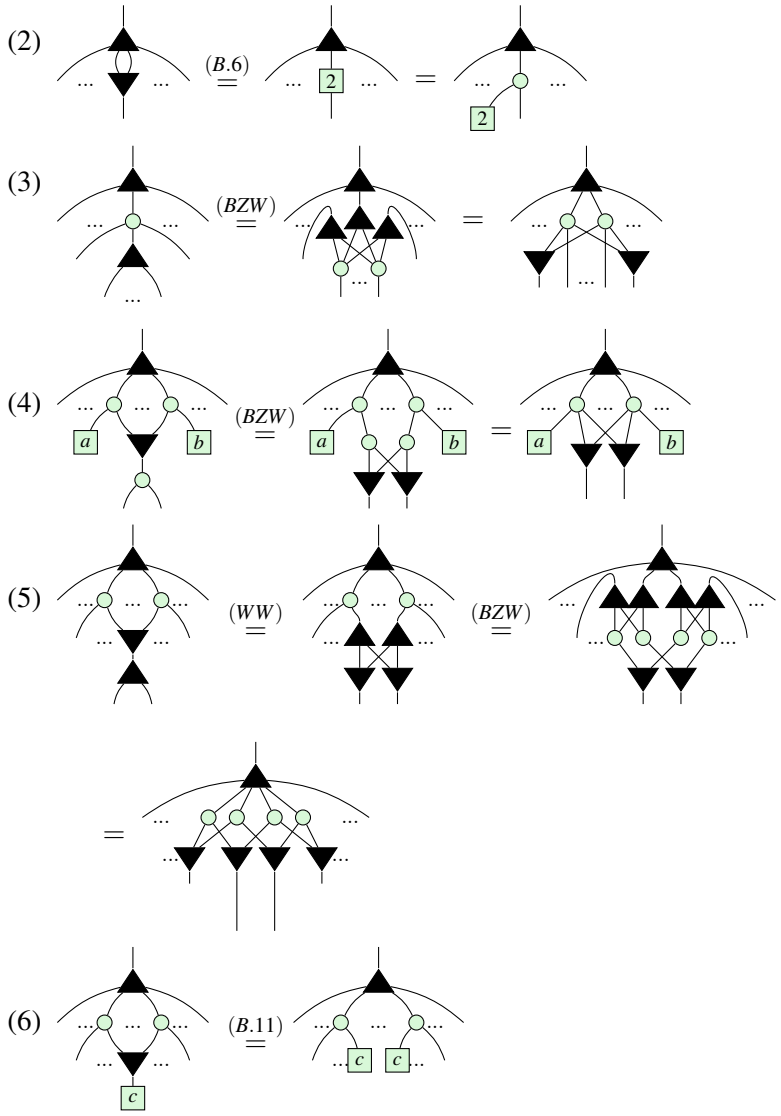
's and 's. Then we shall collect terms and order the boxes to produce a PNF.

If the top of A is not already  , it must be  . It cannot be 

since the remaining arithmetic diagram would then have no inputs which is impossible. It cannot be  since there is only one input and arithmetic diagrams cannot contain  . Thus we can rewrite:

(1) 

(1) guarantees there is a W at the top. We shall now repeatedly apply rewrites underneath the W until there are exactly three layers. Assume that fusion is applied as much as possible between each stage and (B.9) is applied and simplified with (K0) to remove  whenever possible. Then for as long as there are at least 4 layers, we can apply one of the following rewrites:



Clearly, we can only stop applying these rules once A is a sum of products of copies. Steps (2) and (3) ensure the top of A has such a structure and steps (4) - (6) ensure that there is nothing beneath the $\overset{\cdots}{\blacktriangledown}$'s . To see that this will always terminate, observe that (2) and (3) preserve the depth of A while (4), (5), (6) all decrease it. (2) and (3) can only be applied a finite number of times before another simplification must be used. So repeatedly applying these rewrites must eventually shrink the depth down to 3, as desired. Finally, to put A in PNF we must:

(7) Collect terms: whenever there are two boxes connected to exactly the same set of $\overset{\cdots}{\blacktriangledown}$'s, use (B.8) to fuse them together.

(8) Pad: use (B.5) to insert $\boxed{0}$ for any connectivities that do not exist in A.

(9) Reorder: use (Sym) to reorder coefficients into the canonical order.

Step (7) ensures that every $\overset{\cdots}{\curlyvee}$ has unique connectivity. Step (8) ensures there are exactly 2^n coefficients so that step (9) can order them in the appropriate way.

Thus A has been written in PNF, completing the proof.

□

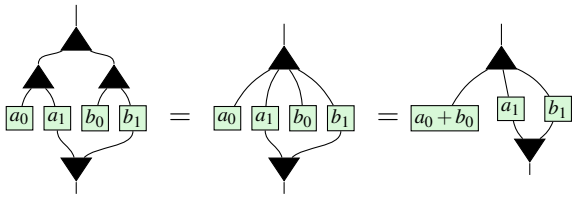
Proof of Theorem 4.2

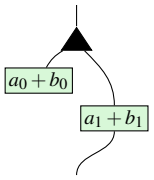
Proof. First, we show ϕ_n is a homomorphism, i.e.

$$\forall p,q \in \mathscr{P}_n, \phi_n(p+q) = \phi_n(p) \boxplus \phi_n(q), \quad \phi_n(p \times q) = \phi_n(p) \boxtimes \phi_n(q)$$

The strategy for the proof will be an induction on n.

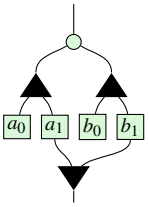
Base case: We have not defined controlled states for $n = 0$, so the base case begins with $n = 1$. Let $p,q \in \mathscr{P}_1$. Write as $p(x_1) = a_0 + a_1 x_1, q(x_1) = b_0 + b_1 x_1$, where $a_0, a_1, b_0, b_1 \in \mathbb{C}$. Then since $p+q = a_0 + b_0 + (a_1 + b_1)x_1$,

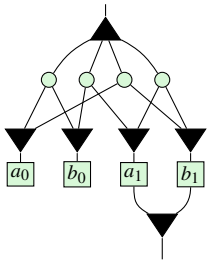
$\phi_1(p) \boxplus \phi_1(q) =$ 

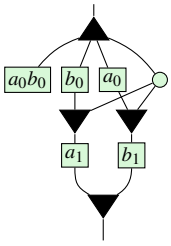
$\overset{(B.6)}{=}$ 

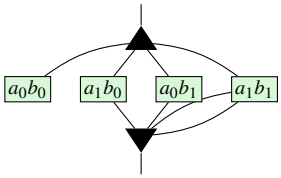
$= \phi_1(p+q)$

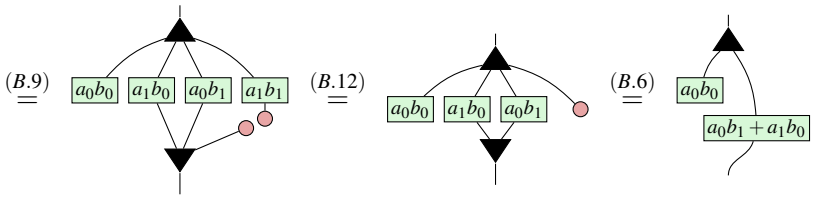
Meanwhile, since $p \times q = a_0 a_1 + (a_0 b_1 + a_1 b_0)x_1$,

$\phi_1(p) \boxtimes \phi_1(q) =$ 

$\overset{(B.10)}{=}$ 

$\overset{(B.11)}{=}$ 

$\overset{(Pcy)}{=}$ 



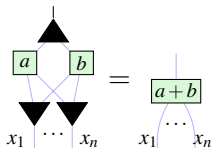
$$= \phi_1(p \times q)$$

Completing the base case.

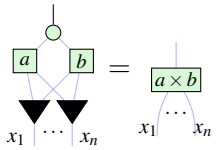
Inductive step:

Let $Hom(n)$ assert than ϕ_n is a homomorphism. Then for the inductive step we wish to prove that $\forall n, Hom(n) \implies Hom(n+1)$.

The proof relies on the recursive definition of $R[x_1,x_2] = R[x_1][x_2]$, for any ring R, to rewrite an arbitrary polynomial $p(x_1,...,x_{n+1}) = a_0 + a_1x_{n+1} + ... + a_{2^{n+1}-1}x_1x_2...x_{n+1} \in \mathscr{P}_{n+1}$ as $p(x_{n+1}) = p_0 + p_1x_{n+1}$, where $p_0, p_1 \in \mathscr{P}_n$. This allows the p_i to be treated similarly to the scalars in the base case. To emphasise this, they will be drawn in green boxes. To help distinguish when an operation is covered by the inductive hypothesis, the wires for variables $x_1,...,x_n$ will be drawn in light blue, while the x_{n+1} wires will be drawn in black. Thus the inductive hypothesis states that:

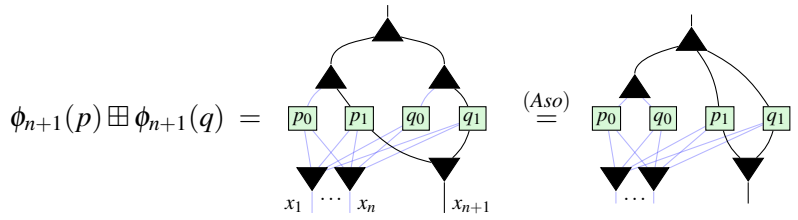


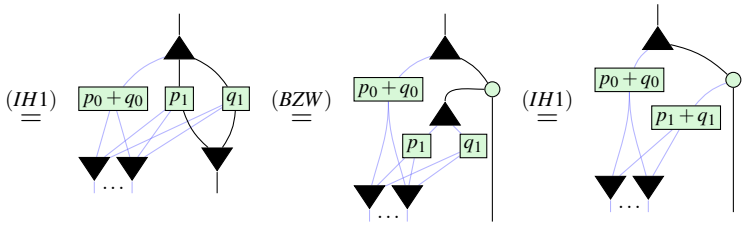
(IH1)



(IH2)

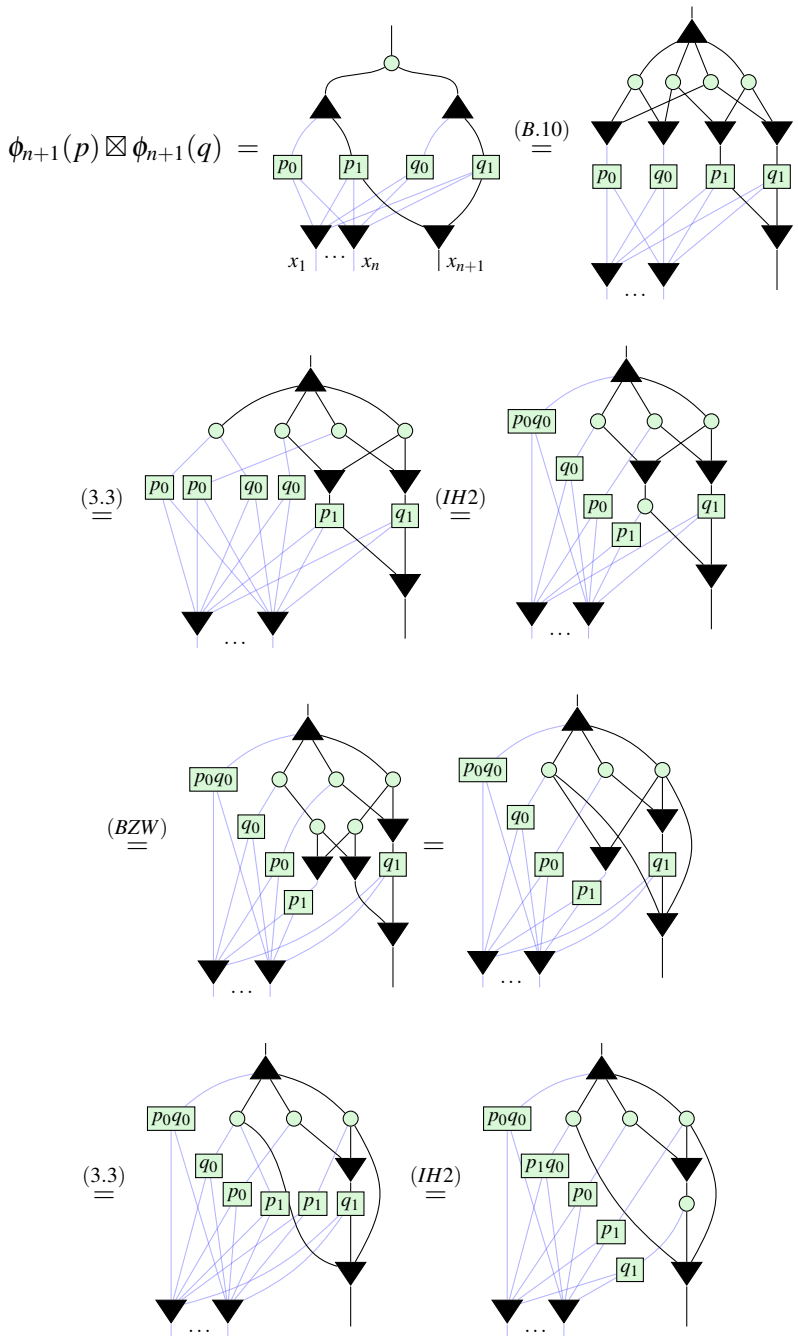
Let $p(x_{n+1}) = p_0 + p_1x_{n+1}, q(x_{n+1}) = q_0 + q_1x_{n+1}$, where $p_0, p_1, q_0, q_1 \in \mathscr{P}_n$. Then for addition:

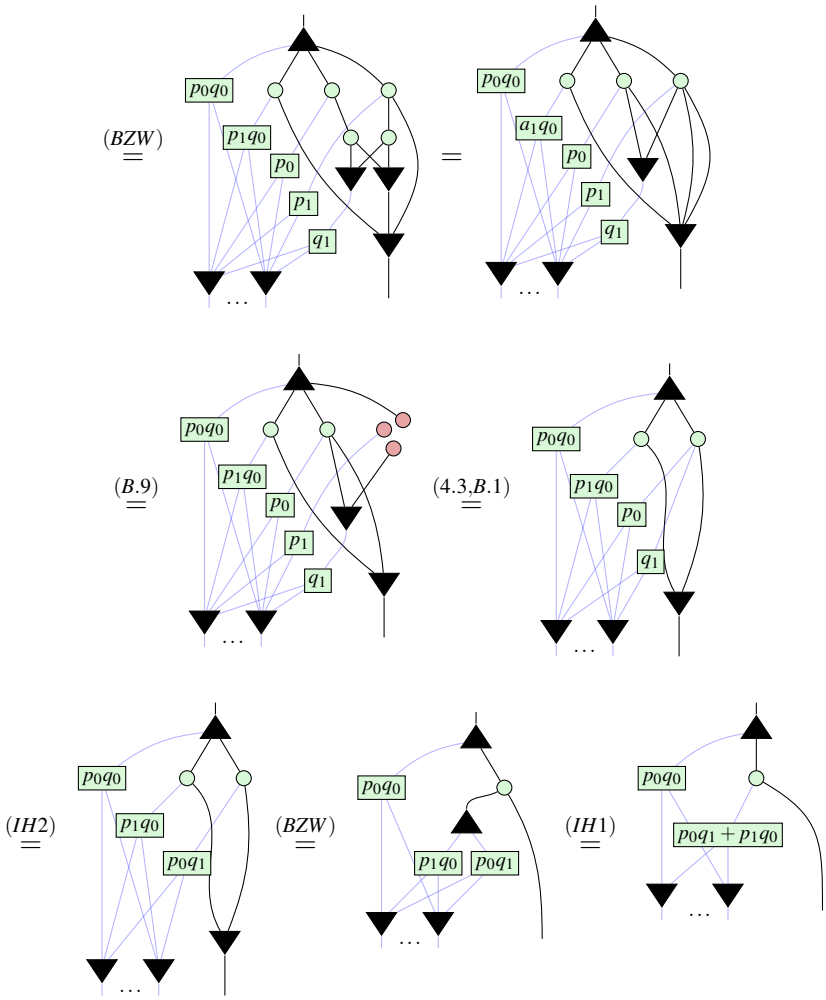




$$= \phi_{n+1}(p_0 + q_0 + (p_1 + q_1)x_{n+1}) = \phi_{n+1}(p+q)$$

Similarly, for multiplication:



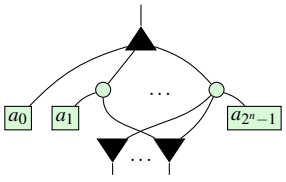


$$= \phi_{n+1}(p_0q_0 + (p_0q_1 + p_1q_0)x_{n+1}) = \phi_{n+1}(p \times q)$$

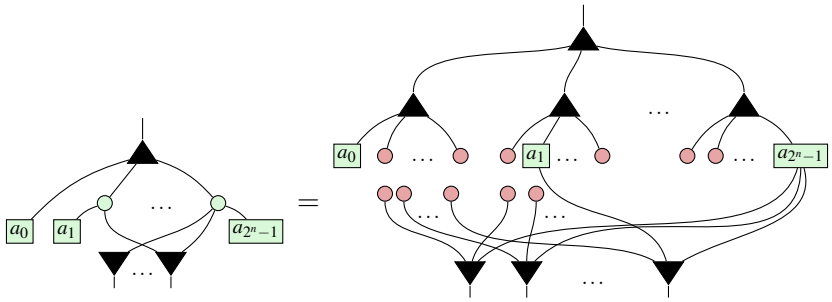
This completes the inductive step, proving that $\forall n > 1$, ϕ_n is a homomorphism.

Finally, to see ϕ_n is an isomorphism, we use Theorem 4.1 to write an arbitrary controlled state in PNF:

$$\begin{bmatrix} 1 & a_0 \\ 0 & a_1 \\ \dots & .. \\ 0 & a_{2^n-1} \end{bmatrix} =$$



Then all we have to do is interpret it as the image of a polynomial:



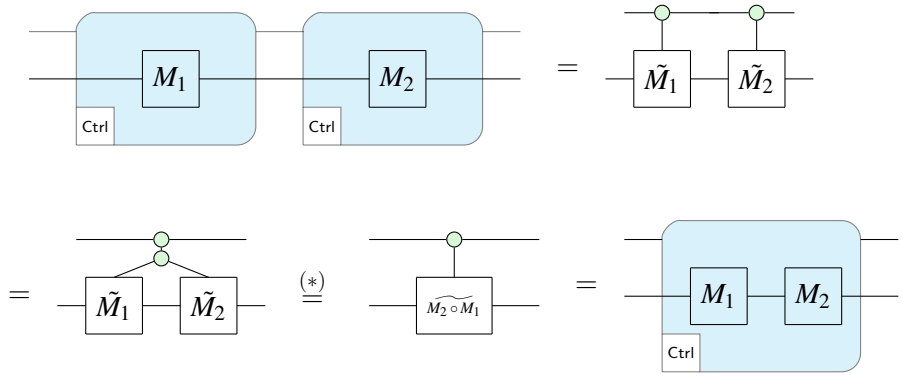
$$= \phi_n(a_0) + \phi_n(a_1 x_n) + ... + \phi_n(a_{2^n-1} x_1 x_2 ... x_n)$$

$$= \phi_n(a_0 + a_1 x_n + ... + a_{2^n-1} x_1 x_2 ... x_n)$$

□

Proof of Proposition 5

Proof.



Where $(*)$ follows from Proposition 1.

□