Equational Reasoning with Controlled ZXW Diagrams

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

Matthew Wilson
Programming Principles Logic and Verification Group
University College London
matthew.wilson.18@ucl.ac.uk

1 Abstract

asdf

2 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 [9, 17] and linear combination of unitaries [4], while a number of categorical formalisations have included routed quantum circuits [25], the many-worlds calculus [3], categorifying signal flow diagrams [2], and classical and quantum control in quantum modal logic [20].

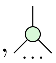
The question we are interested in is how quantum graphical calculi such as the ZX [5], ZW [6], and ZH [1] calculus can be augmented to support properties of quantum control. An early use of controlled state diagrams was for proving constructive and rational angle ZX calculus completeness [13]. More recently, controlled state and controlled matrix diagrams have been applied to addition and differentiation of ZX diagrams [12], differentiating and integrating ZX diagrams for quantum machine learning [26], Hamiltonian exponentiation and simulation [21], and non-linear optical quantum computing [8]. 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.

In this paper, we first introduce a higher-order map Ctrl which sends states to controlled states, and square matrices to controlled square matrices. We prove that Ctrl is a lax monoidal functor on the subcategory of all square matrices. This allows us to use the functorial boxes of Ref. [15] to control ZX diagrams so long as functoriality is only applied when the number of input and output wires are equal. Moreover, AND of the controls is a natural transformation corresponding to nested applications of Ctrl.

Next, we show that the set of all controlled n-partite states defines a commutative ring $(\tilde{S}^n, \boxplus, \boxtimes)$. 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

To appear in EPTCS.

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

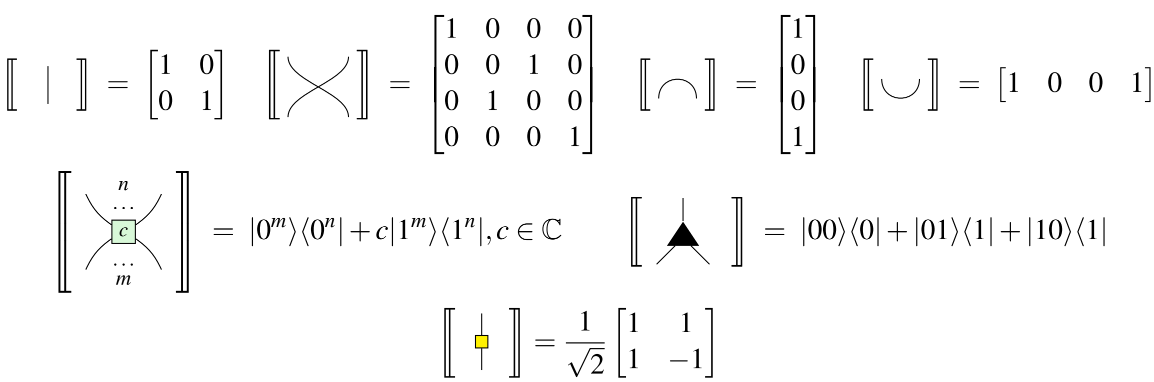
We compose controlled states into each control wire of controlled square matrices to recover multivariate polynomials over same-size square matrices. Commutativity of controlled square matrices holds in the special case that the controls target mutually exclusive sectors, allowing copying of arbitrary controlled diagrams. As a result, we can factor multivariate polynomials over same-size square matrices. This means we can now factor any Hamiltonian in the ZXW calculus [21], even with all its terms black-boxed. In summary, these algebraic properties of quantum control give rise to powerful new graphical rewrite rules for black-box diagrams.

3 ZXW Calculus

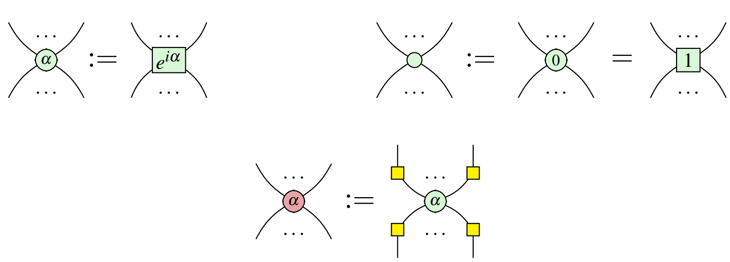
This section introduces the generators of the ZXW calculus. The ZXW calculus is a diagrammatic formalism for qudit computation, unifying the ZX and ZW calculi and synthesising their relative strenghts. The ZXW caluclus consists of diagrams built from a small number of generators and equipped with a complete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

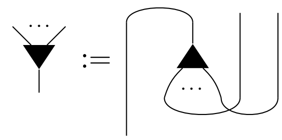
3.1 Generators

The qubit ZXW calculus is built from the following generators:



For simplicity, we introduce the following additional notation:





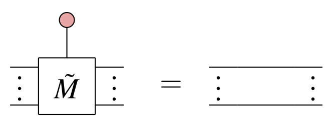
The complete rule set is given in appendix A. Several important lemmas are found in appendix B.

4 Controlled Diagrams

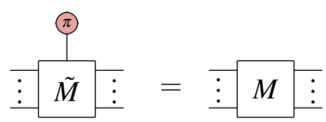
4.1 Definitions

We recall the definition of a controlled diagram from [21],

Definition 4.1. For an arbitrary square matrix M, the controlled matrix of M is the diagram $\tilde{M}$ such that:



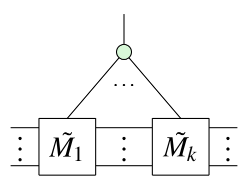
(4.1)



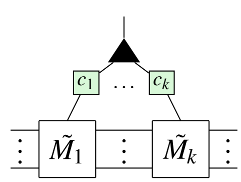
(4.2)

It is possible to perform matrix arithmetic with controlled diagrams.

Proposition 1 ([21])**.** Given controlled matrices $\tilde{M}_1, \ldots, \tilde{M}_k$, the controlled matrix $\widetilde{\Pi_i M_i}$ is given by



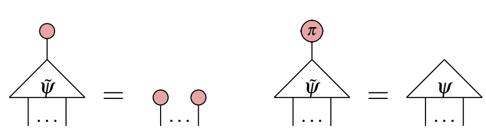
Given controlled matrices $\tilde{M}_1, \ldots, \tilde{M}_k$ and complex numbers $c_1, \ldots, c_k$, the controlled matrix $\widetilde{\Sigma_i c_i M_i}$ is given by



Proof. See propositions 3.3 and 3.4 in [21]. □

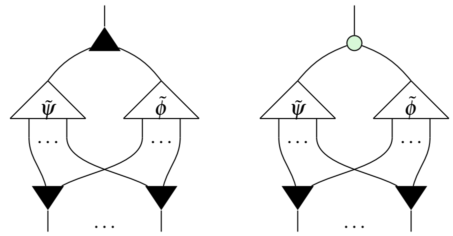
We can also defined the analogue for states.

Definition 4.2. For an arbitrary state ψ, the controlled state of ψ is the diagram $\tilde{\psi}$ such that:



(4.3)

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.

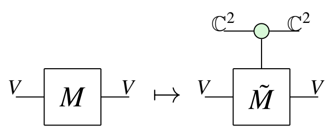


The role of is to *copy* outputs, as shown in section 5.1.

4.2 Control Functor

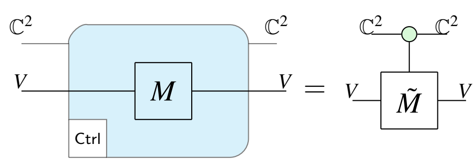
The operation of turning a square matrix to its controlled diagram can be made into a lax monoidal functor. Let $\mathbf{Hilb_{sq}}$ be the subcategory of Hilbert spaces and square matrices. Adding an additional horizontal wire to facilitate composition, $\mathsf{Ctrl} : \mathbf{Hilb_{sq}} \to \mathbf{Hilb}$ is defined as follows for arbitrary $M \in Hom_{Hilb_{sq}}(V,V)$.

Ctrl ::



(4.4)

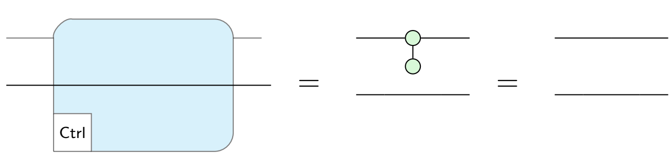
In the functorial box notation of [15], this would be:



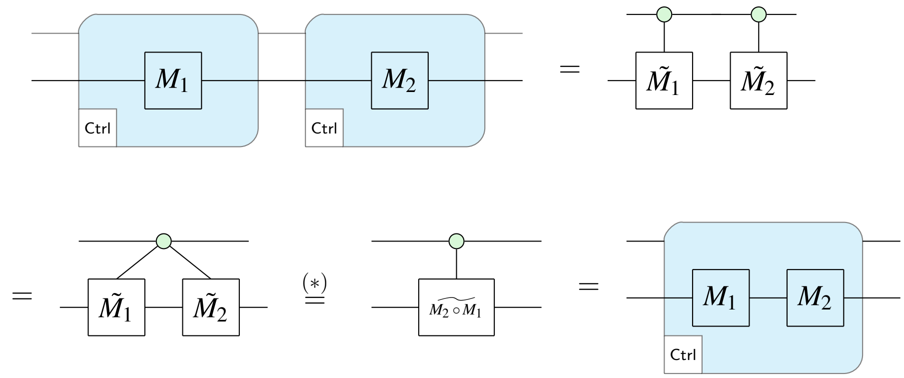
(4.5)

Proposition 2. The map Ctrl defined in (4.4) is a lax monoidal funtor.

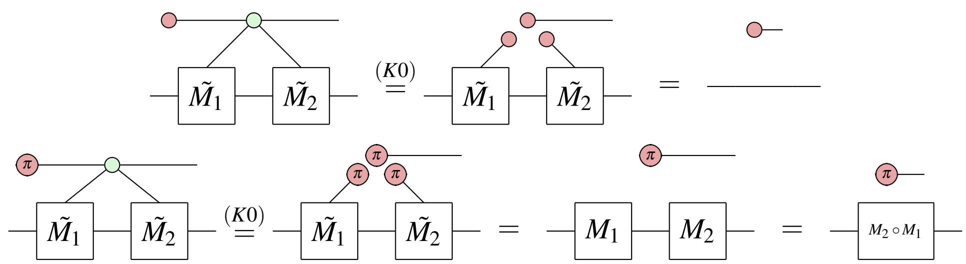
Proof. On $id_V : V \to V$:

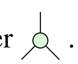


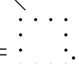
Let $D_1 : V \to V$, $D_2 : V \to V$. Then composing $\mathsf{Ctrl}(M_2) \circ \mathsf{Ctrl}(M_1)$:

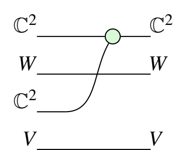


Where $(*)$ follows from:

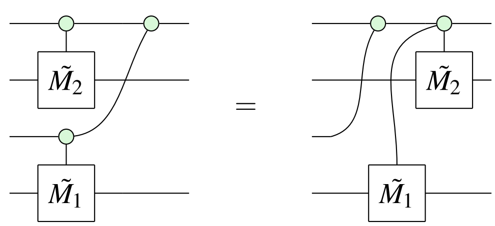


Which implies that controlled diagrams can fuse under  .

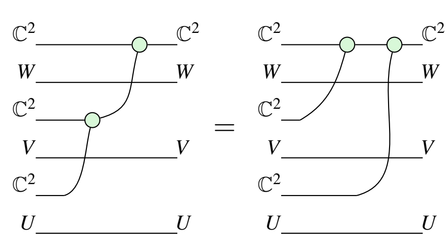
Ctrl preserves the monoidal unit since $\mathbf{1}_{\mathbf{Hilb_{sq}}} = \mathbf{1}_{\mathbf{Hilb}} =$  Ctrl is lax thanks to the following morphism: $\phi_{V,W} : \mathsf{Ctrl}(V) \otimes \mathsf{Ctrl}(W) \to \mathsf{Ctrl}(V \otimes W)$:

$\phi_{V,W} =$ 

ϕ is natural since for any $M_1 : V \to V$, $M_2 : W \to W$, we have:



Finally, ϕ satisfies the coherence condition since for any U, V, W:

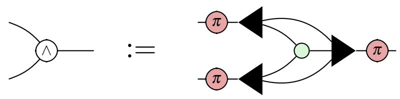


□

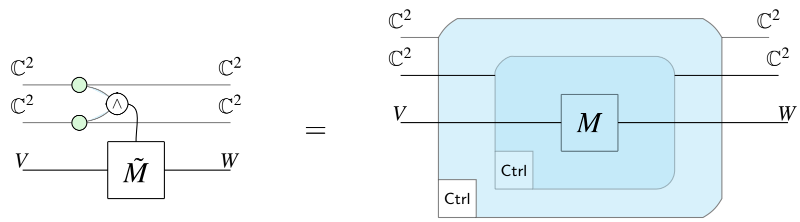
4.3 Multiple Control

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.

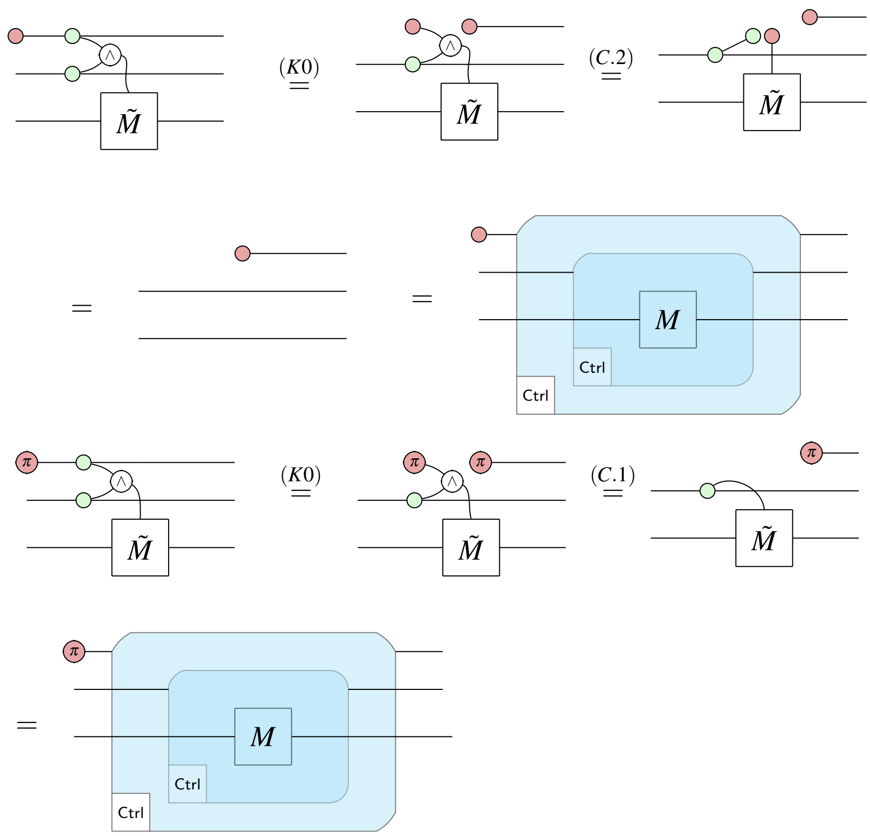
Proposition 3.



Proposition 4.



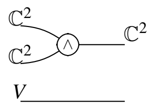
Proof. Plugging basis states:



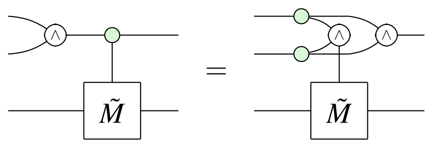
□

Since AND is a monoid, it is reasonable to expect that multiple control induces a monad. Unfortunately, this does not appear to go through. Although AND does define a natural transformation (in the relevant category), defining the unit of the monad necessitates defining the controlled diagram of an arbitrary matrix, which breaks functoriality. Nevertheless we prove that AND is a natural transformation.

Proposition 5. $\mu : \mathsf{Ctrl}^2 \to \mathsf{Ctrl}$ is a natural transformation with components $\mu_V : \mathsf{Ctrl}^2 V \to \mathsf{Ctrl} V$ defined as follows:



Proof.



Due to the $Z-H$ bialgebra in the ZH calculus. □

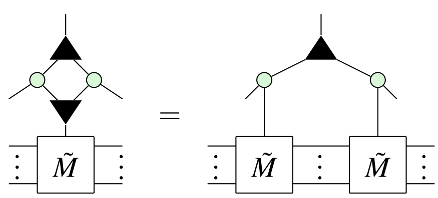
5 Polynomials

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 [10, 11].

5.1 Rings

Let $\tilde{E_n}$ 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 $\tilde{E_n}$. Likewise, we show that the set of controlled n-qubit states $\tilde{S_n}$ also forms a ring. Before doing so, we prove a few important lemmas. The first lemma enables us to copy controlled matrices. Note that all proofs can be found in the appendix.

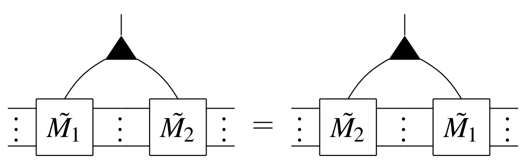
Lemma 5.1. *For any square matrix M,*



(5.1)

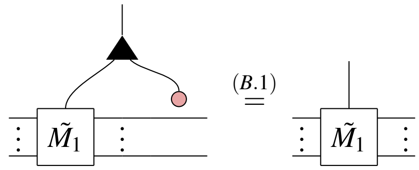
Now we show that controlled matrix addition and multiplication satisfy the ring axioms. Associativity of $+, \times$ follow immediately from (Aso, S1), respectively. Commutativity of addition follows from the commutativity of matrix addition.

Lemma 5.2. *Let M_1, M_2 be $n \times n$ matrices.*



(5.2)

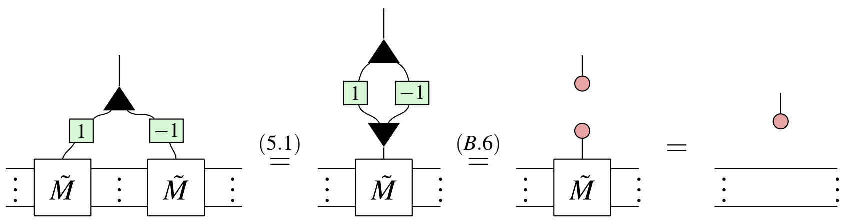
The additive identity is defined as $\otimes I_n$:



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

Lemma 5.3. *The additive inverse of $\tilde{M}$ is $\boxed{-1} \circ \tilde{M}$*

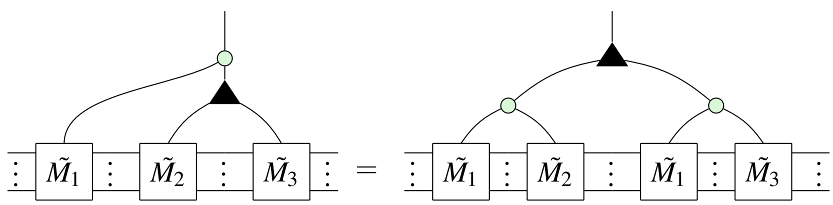
Proof.



□

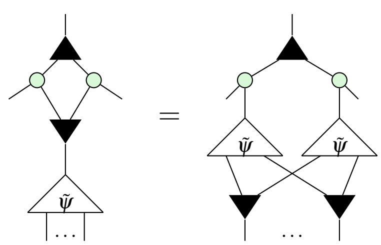
Finally, we prove distributivity.

Lemma 5.4.



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 5.5. *For any state ψ,*



(5.3)

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

Lemma 5.6. *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 5.7. $\tilde{\psi} \boxplus \tilde{\mathbf{0}} = \tilde{\psi}$

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

Lemma 5.8. *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.

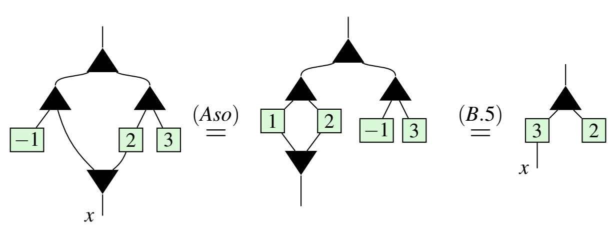
Lemma 5.9. $\tilde{\psi}_1 \boxtimes (\tilde{\psi}_2 \boxplus \tilde{\psi}_3) = (\tilde{\psi}_1 \boxtimes \tilde{\psi}_2) \boxplus (\tilde{\psi}_1 \boxtimes \tilde{\psi}_3)$

5.2 Arithmetic

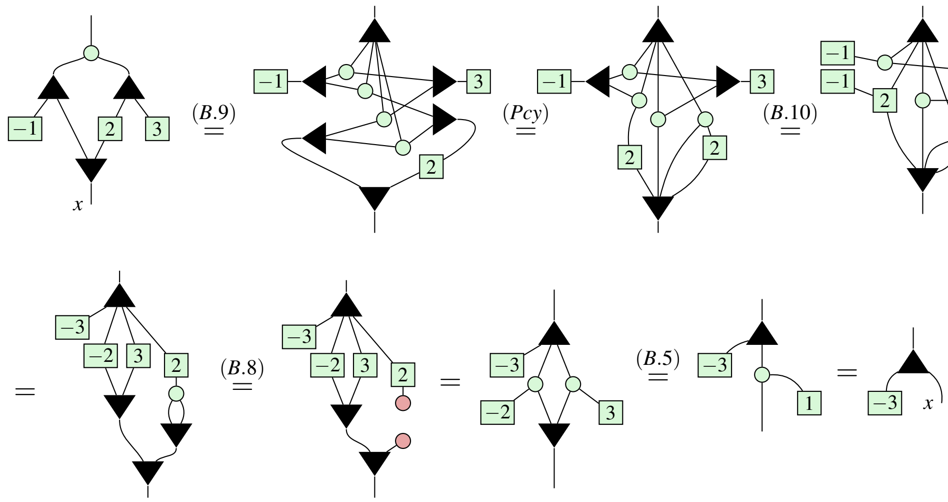
Its been known since 2011 that , can be used to add and multiply number states $\boxed{a}$, respectively [7]. In the previous section we saw that , can moreover be used to copy controlled diagrams. In this section, we explain this connection by demonstrating that controlled states are in fact isomorphic 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. [28] 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 [27].

Firstly, we describe how to interpret certain ZXW diagrams as polynomials. Consider the following 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 $3x+2$:



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

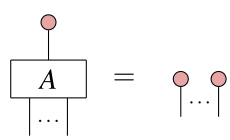


The reason for the missing $2x^2$ term is that (B.8) implies $x^2 = 0$. Other than that, controlled state arithmetic appears to faithfully reflect polynomial arithmetic. To help formalise this correspondence, we introduce the following definition.

Definition 5.1. A ZXW diagram with a single input on top is **arithmetic** if it contains only , wires, , , nodes and a boxes.

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

Lemma 5.10. *For any arithmetic diagram A,*

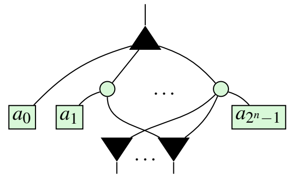


(5.4)

Proof. By definition, other than wires A contains only , , , and a. All a's can be removed with (Ept). Meanwhile all the spiders copy 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 , followed by a layer of , followed by a layer of a, .

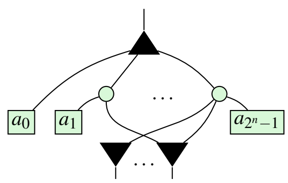
Definition 5.2. An n-output arithmetic diagram is said to be written in **polynormal form** (PNF) if it looks like:



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 very closely related to the completeness normal form (see [16]). 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 PNF to directly represent its own matrix.

Proposition 6.



$$= \begin{bmatrix} 1 & a_0 \\ 0 & a_1 \\ ... & ... \\ 0 & a_{2^n-1} \end{bmatrix} \tag{5.5}$$

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.

Proposition 7. All arithmetic diagrams can be written into PNF

Proof. See appendix C □

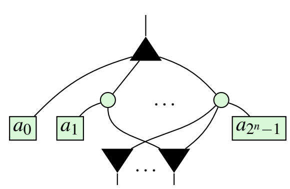
5.3 Isomorphism

At last we can prove the isomorphism. Throughout we shall let $\mathscr{P}_n$ denote the ring $\mathbb{C}[x_1,...,x_n]/(x_1^2,...,x_n^2)$.

Theorem 5.1. *There is an isomorphism $\mathscr{P}_n \simeq \tilde{S}_n$*

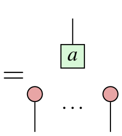
First, we shall define the map $\phi_n : \mathscr{P}_n \to \tilde{S}_n$ before proving it induces an isomorphism. ϕ_n is defined to map an arbitrary polynomial $p(x_1,...,x_n) = a_0 + a_1 x_n + ... + a_{2^n-1} x_1 x_2 ... x_n$ to the following PNF:

$\phi_n(p) =$

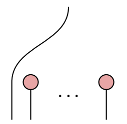
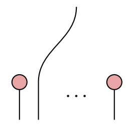
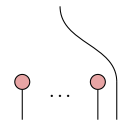


Some important special cases are mapping scalars $a \in \mathbb{C}$:

$\phi_n(a) =$



And mapping indeterminates x_i:

$\phi_n(x_1) =$  $, \phi_n(x_2) =$  $, \ldots, \phi_n(x_n) =$ 

The full proof is found in appendix C.

6 Conclusion

To conclude, we first introduced the higher-order map Ctrl and showed it is a lax monoidal functor on all same-size square matrices. This enabled us to add functorial boxes to such ZXW diagrams. Moreover, we find useful rewrite rules for interactions between controlled diagrams and all generators Z, X, W, and H.

We further showed that all controlled n-partite states form a commutative ring isomorphic to multilinear polynomials, and all controlled n-qubit square matrices form a non-commutative ring. Plugging the former into the control wires of the latter, gives multivariate polynomials over same-size square matrices, such as Hamiltonians. When the controls target mutually exclusive sectors, a rewrite rule can be applied to copy any controlled diagram, and thereby factor any Hamiltonian.

The natural next step is to derive extensions of our results for controlled qubit diagrams to qudits. 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 [16]. A starting guess would be that qudit controlled states are isomorphic to polynomials $\mathbb{C}^{d-1}[x_1,\ldots,x_n]/({x_1}^d,\ldots,{x_n}^d)$ 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. [19].

We would like to try sector-preserving channels [24] and scoped effects [14] as approaches to better formulate the monadic nature of multiple-control. We are also curious about reconciling the interpretation of diagrammatic differentiation of our arithmetic polynomial circuits by the approach in Ref. [28], with that of quantum circuits and ZX diagrams in Refs. [23, 26, 12]. Last but not least, these new semantics for quantum controlled states and matrices could be embedded categorically into a host functional programming language like in Ref. [18], or translated to an equational theory for a quantum programming language like in Ref. [22].

7 Acknowledgements

We thank Razin Shaikh, Itai Leigh, Tim Forrer, and Aleks Kissinger for insightful discussions. LY is funded by a Google PhD Fellowship.

References

[1] 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.

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

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

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

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

[6] 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.

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

[8] 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).

[9] 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.

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

[11] 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.

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

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

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

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

[16] 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).

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

[18] 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).

[19] 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.

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

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

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

[23] 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.

[24] 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.

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

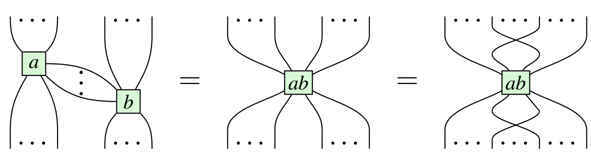
[26] WANG, Q., YEUNG, R., AND KOCH, M. Differentiating and integrating zx diagrams with applications to quantum machine learning, 2022.

[27] 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.

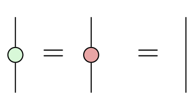
[28] WILSON, P., AND ZANASI, F. An axiomatic approach to differentiation of polynomial circuits. *Journal of Logical and Algebraic Methods in Programming 135* (2023), 100892.

Appendix A ZXW rules

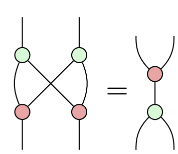
ZX Rules:



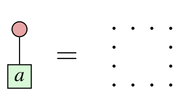
(S1)



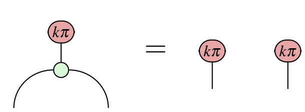
(S2)



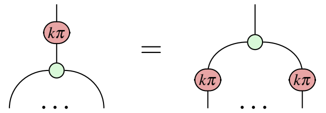
(B2)

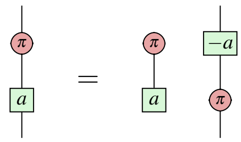


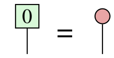
(Ept)

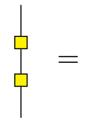


(K0)

 (K1)

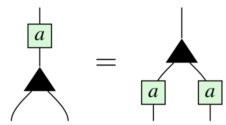
 (K2)

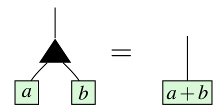
 (Zer)

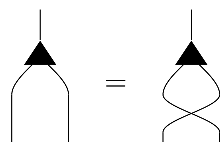
 (H)

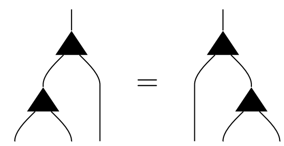
Where $k \in \{0, 1\}$.

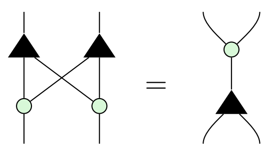
ZW Rules:

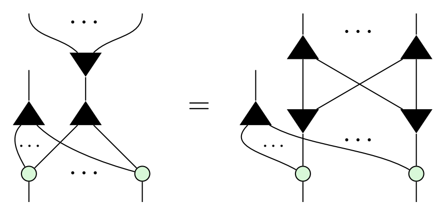
 (Pcy)

 (ADD)

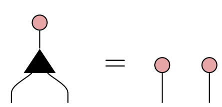
 (Sym)

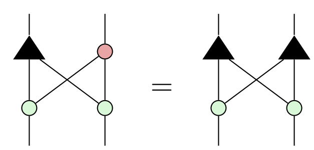
 (Aso)

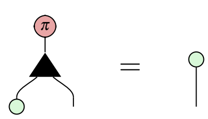
 (BZW)

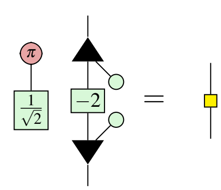
 (WW)

ZXW Rules:

 (Bs0)

 (TA)

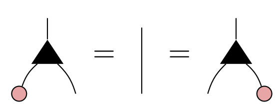
 (Bsj)

 (HD)

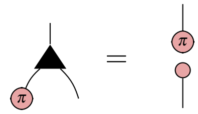
Appendix B Basic Lemmas

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

Lemma B.1.

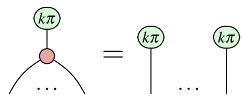
 (B.1)

Lemma B.2.



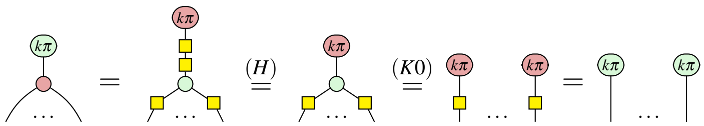
(B.2)

Lemma B.3.



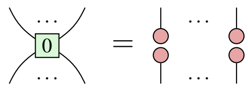
(B.3)

Proof.



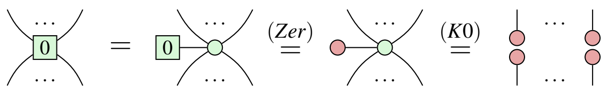
□

Lemma B.4.



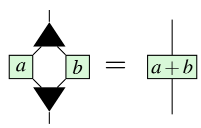
(B.4)

Proof.



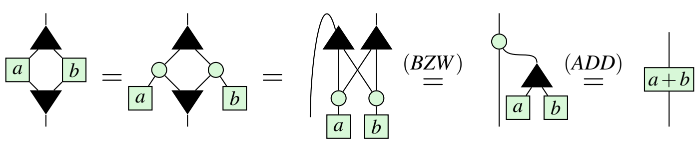
□

Lemma B.5.



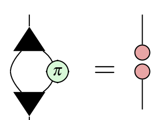
(B.5)

Proof.



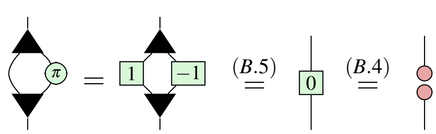
□

Lemma B.6.



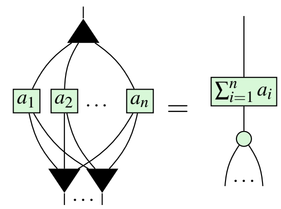
(B.6)

Proof.



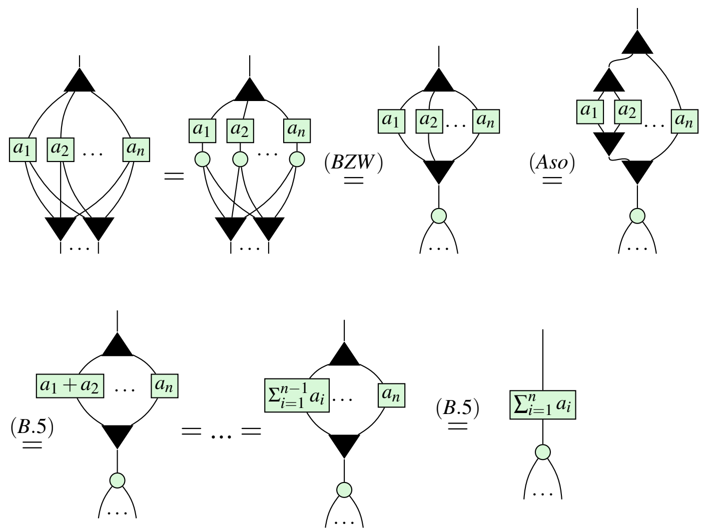
□

Lemma B.7.



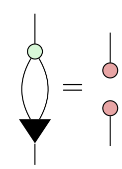
(B.7)

Proof.



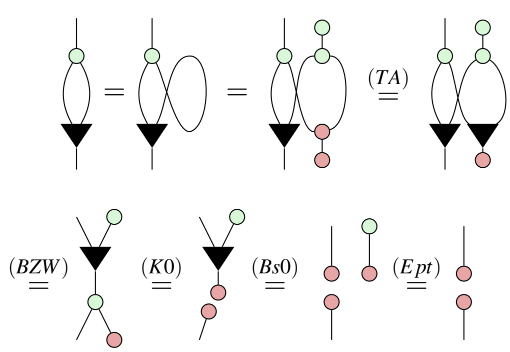
□

Lemma B.8.



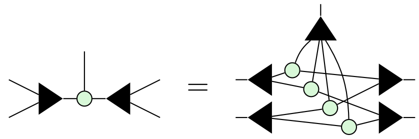
(B.8)

Proof.



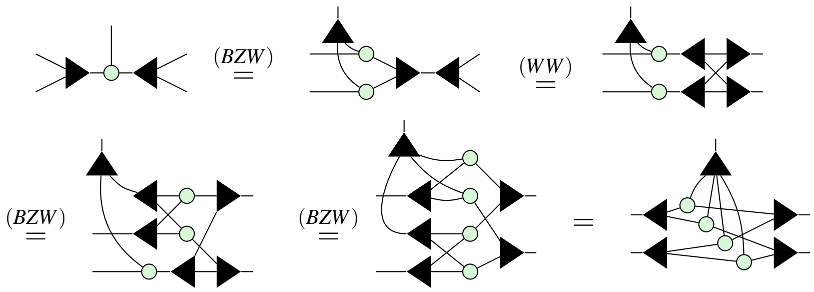
□

Lemma B.9.



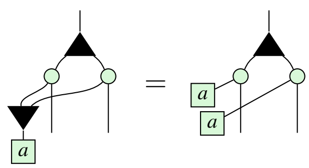
(B.9)

Proof.



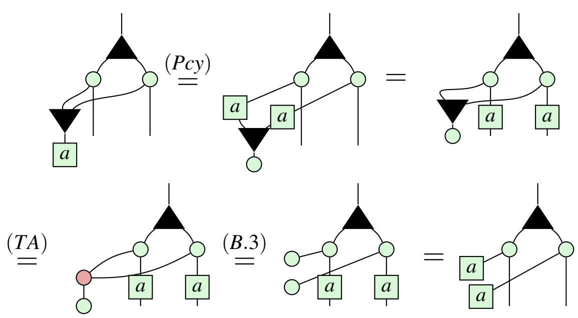
□

Lemma B.10.



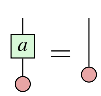
(B.10)

Proof.



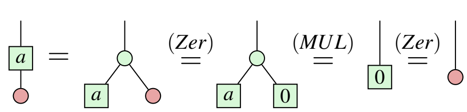
□

Lemma B.11.



(B.11)

Proof.

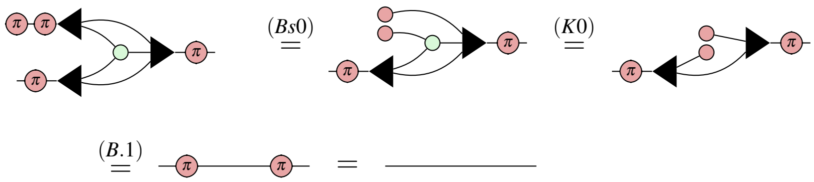


□

Appendix C Main Proofs

Proof of Proposition 3

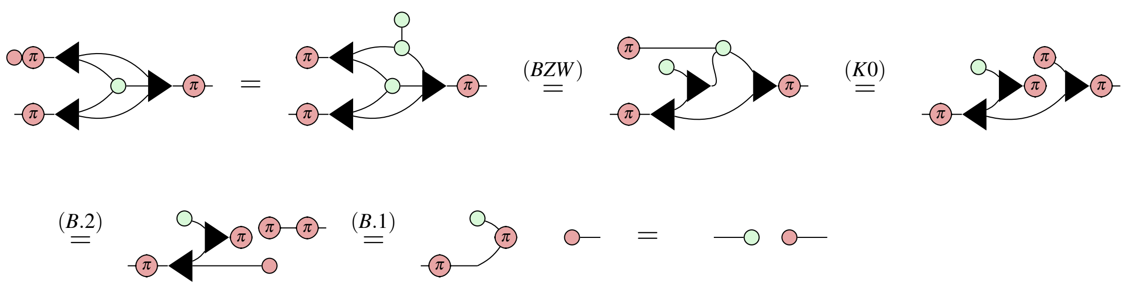
Proof. We can verify this computes the AND gate by computing on basis states.



(C.1)

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

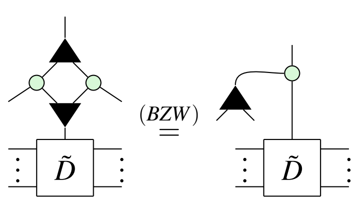
(C.2)



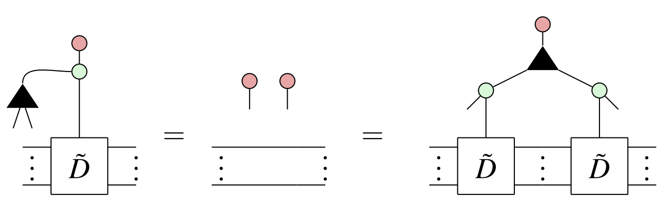
□

Proof of Lemma 5.1

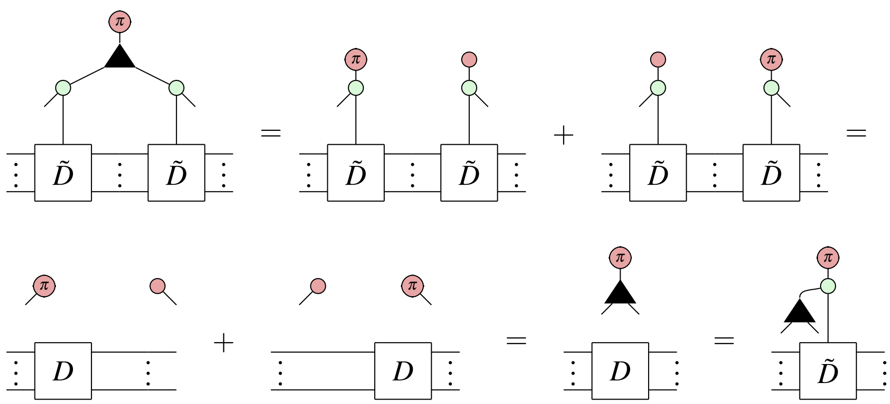
Proof. First of all, using (BZW) we can rewrite the LHS to



Then clearly



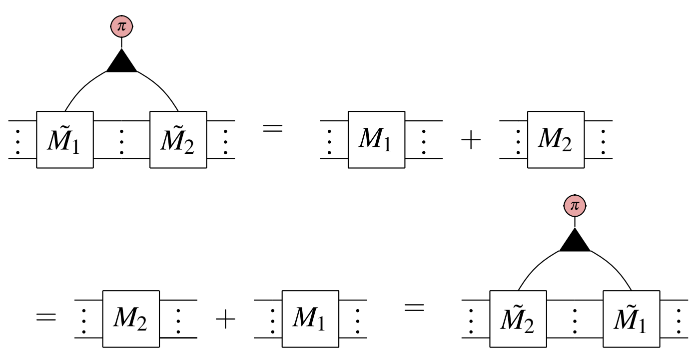
Meanwhile,



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

Proof of Lemma 5.2

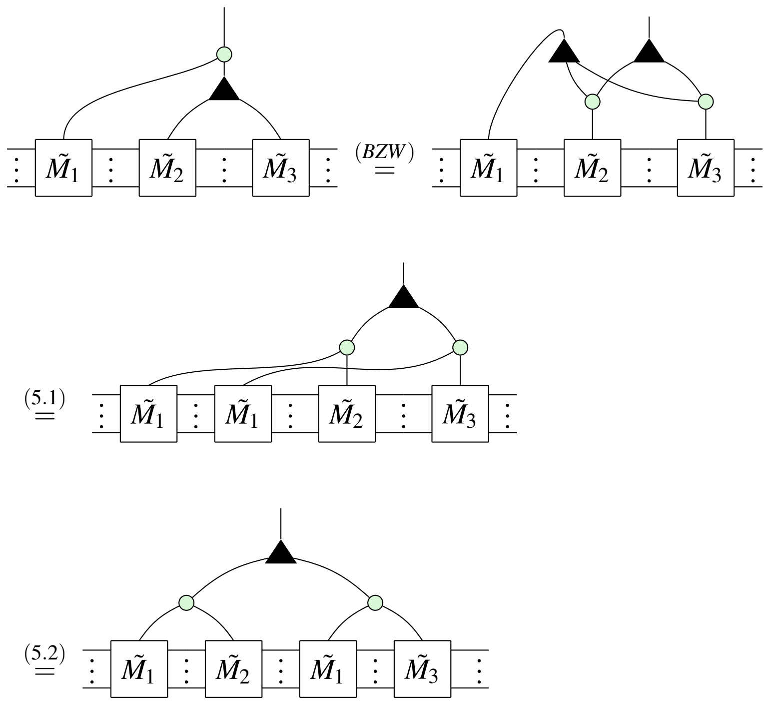
Proof. We prove by plugging red and commutativity of matrix addition. By definition of controlled matrices, plugging gives I_n on both sides. Meanwhile, plugging gives:



□

Proof of Lemma 5.4

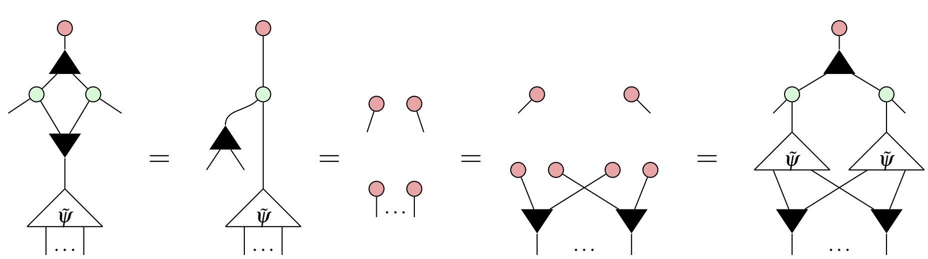
Proof.



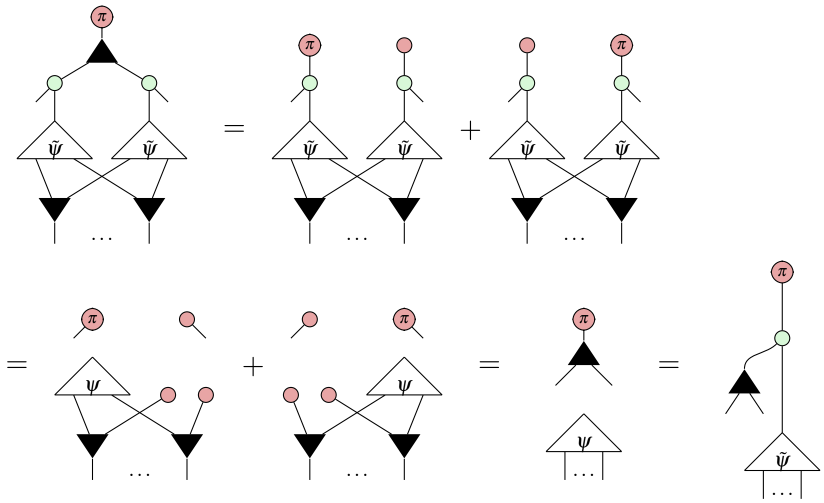
□

Proof of Lemma 5.5

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

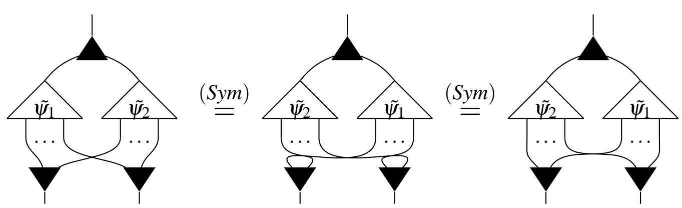


Meanwhile, plugging $|1\rangle$ gives



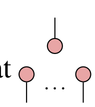
Completing the proof □

Proof of Lemma 5.6

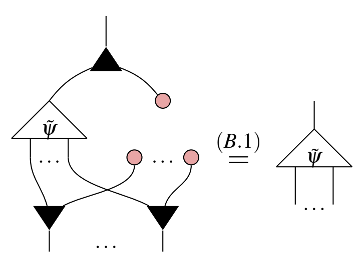
Proof. 

□

Proof of Lemma 5.7

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

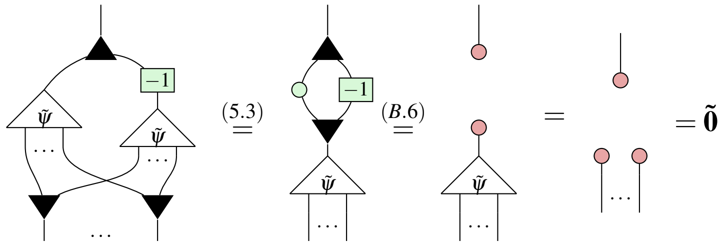
Then we have:



□

Proof of Lemma 5.8

Proof. $\tilde{\psi} \circ \boxed{-1}$ is still a controlled state since $\boxed{-1}$ does nothing to . Then $\tilde{\psi} \circ \boxed{-1}$ inverts $\tilde{\psi}$ since:

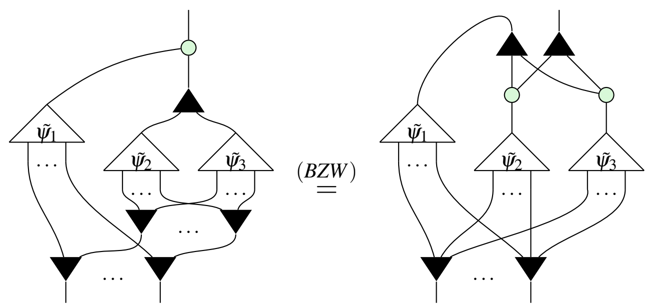


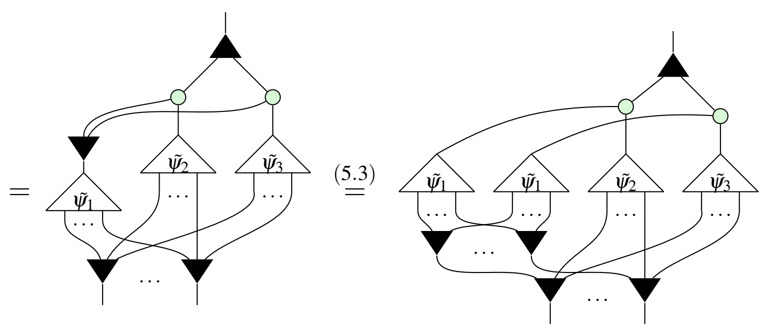
□

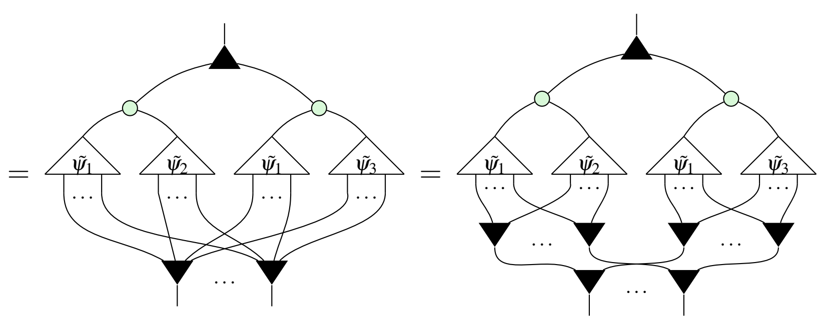
Proof of Lemma 5.8

Proof.

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







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

□

Proof of Proposition 6

Proof. We prove by induction on n.

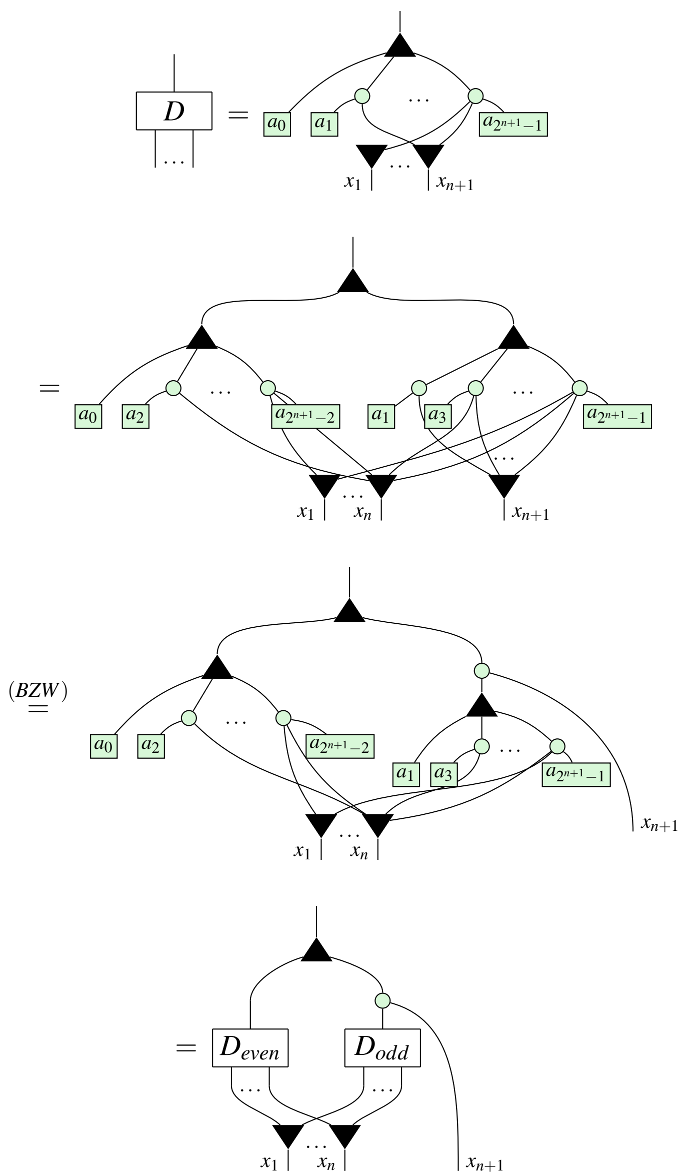
For the base case, $n = 0$. The only PNF with no outputs is a number so we have:

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

as desired.

For inductive hypothesis, we assume that (5.5) holds for every PNF on n outputs. We use this hypothesis to extend it to PNFs with $n+1$ outputs.

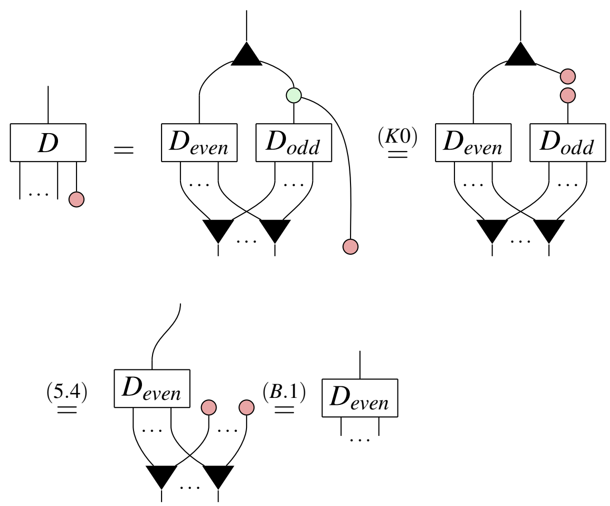
Let D be an arbitrary PNF with $n+1$ outputs. Firstly, observe that x_{n+1} is connected to only the odd coefficients $\{a_{2k+1}\}$ since these are exactly the indices with 1 in the least significant bit. Thus we can rewrite:



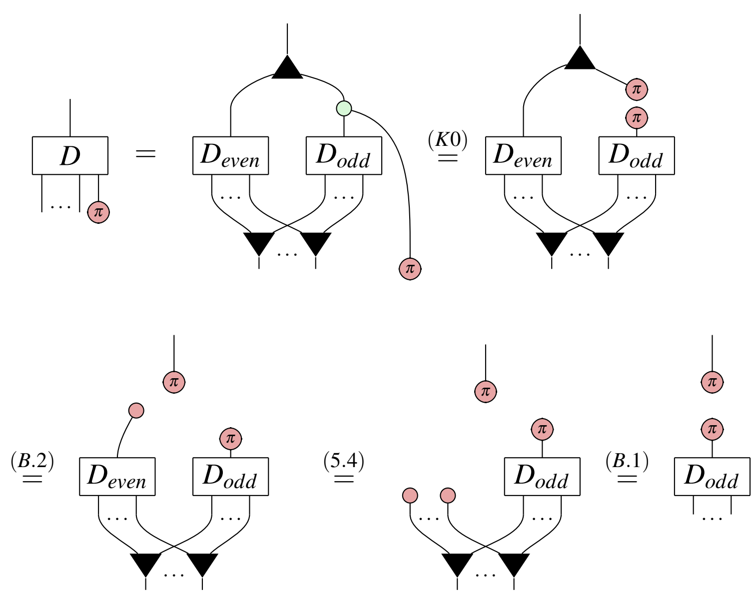
Where D_{even}, D_{odd} are PNF diagrams. Since they are over n variables, we can apply the inductive hypothesis and obtain:

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

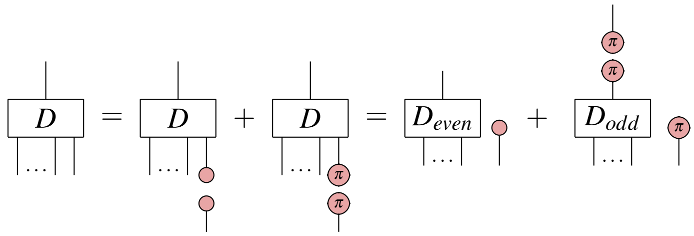
Next, plugging red we observe:



Meanwhile,



Summing these together,



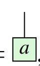
$$= (D_{even} \otimes |0\rangle) + (D_{odd}|1\rangle\langle 1| \otimes |1\rangle)$$

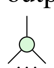
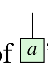
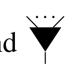
$$\overset{(*)}{=} \begin{bmatrix} 1 & a_0 \\ 0 & a_2 \\ \dots & \dots \\ 0 & a_{2^{n+1}-2} \end{bmatrix} \otimes |0\rangle + \begin{bmatrix} 0 & a_1 \\ 0 & a_3 \\ \dots & \dots \\ 0 & a_{2^{n+1}-1} \end{bmatrix} \otimes |1\rangle$$

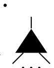
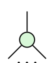
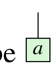
$$= \begin{bmatrix} 1 & a_0 \\ 0 & 0 \\ 0 & a_2 \\ 0 & 0 \\ \dots & \dots \\ 0 & a_{2^{n+1}-2} \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & a_1 \\ 0 & 0 \\ 0 & a_3 \\ \dots & \dots \\ 0 & 0 \\ 0 & a_{2^{n+1}-1} \end{bmatrix} = \begin{bmatrix} 1 & a_0 \\ 0 & a_1 \\ 0 & a_2 \\ 0 & a_3 \\ \dots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

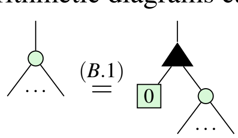
Completing the inductive step. □

Proof of Propsotion 7

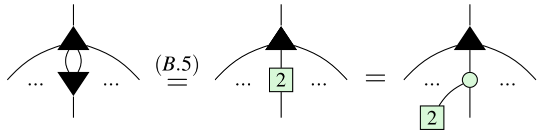
Proof. Let A be an arithmetic diagram. If $A =$ , we are done.

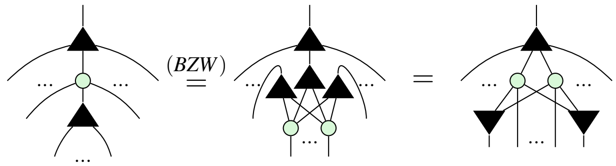
Otherwise, A has at least one output. First, we shall rewrite A into three layers, consisting of: (1) a single W at the top, (2) a layer of  and (3) a layer of 's and 's. Then we shall collect terms and order the boxes to produce a PNF.

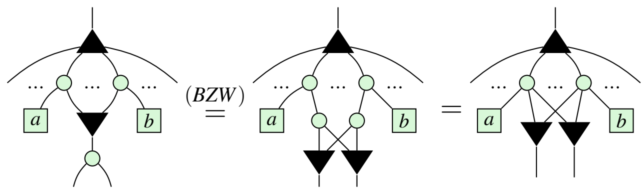
If the top of A is not already , it must be . It cannot be  since the remaining arithmetic diagram would then have no inputs which is impossible. It cannot be  since there is only one input and arithmetic diagrams cannot contain . Thus we can rewrite:

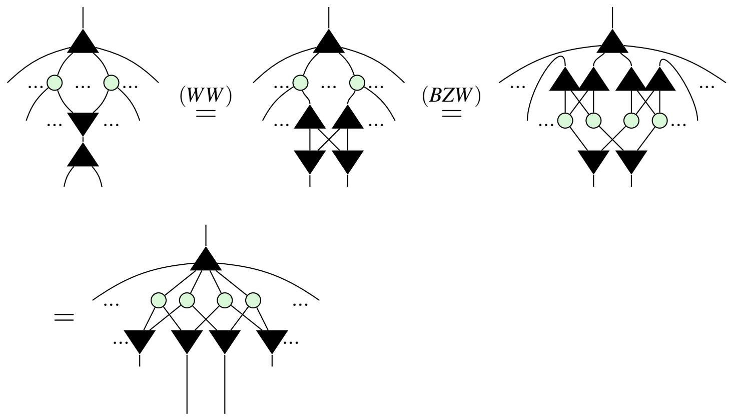
(1) 

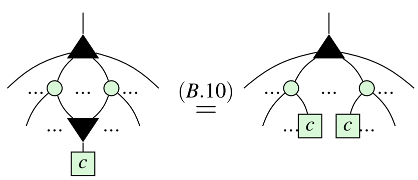
(1) guarantees there is a W at the top. We shall now repeatedly apply rewrites underneath the W until there are exactly three layers. Assume that fusion is applied as much as possible between each stage and (B.8) is applied and simplified with (K0) to remove  whenever possible. Then for as long as there are at least 4 layers, we can apply one of the following rewrites:

(2) 

(3) 

(4) 

(5) 

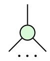
(6) 

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 '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 's, use (B.7) to fuse them together.

(8) Pad: use (B.4) 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  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.

□

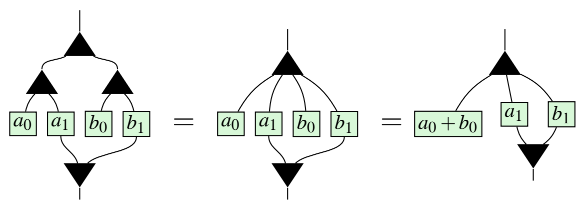
Proof of Theorem 5.1

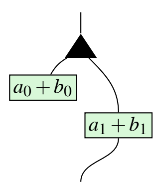
Proof. First, we show ϕ_n is a homomorphism, i.e.

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

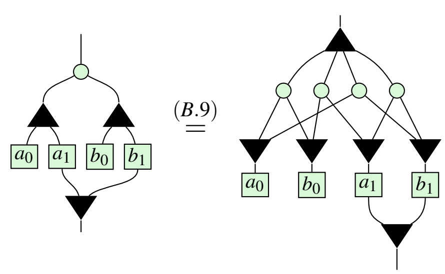
The strategy for the proof will be an induction on n.

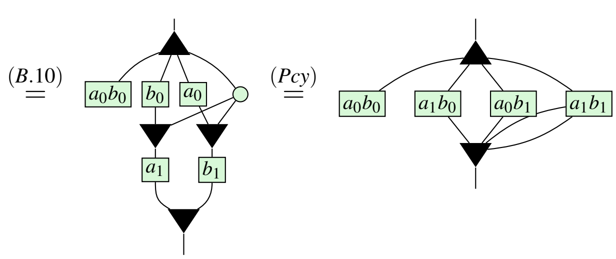
Base case: We have not defined controlled states for $n=0$, so the base case begins with $n=1$. Let $p,q \in \mathscr{P}_1$. Write as $p(x_1) = a_0 + a_1x_1, q(x_1) = b_0 + b_1x_1$, where $a_0,a_1,b_0,b_1 \in \mathbb{C}$. Then since $p+q = a_0+b_0+(a_1+b_1)x_1$,

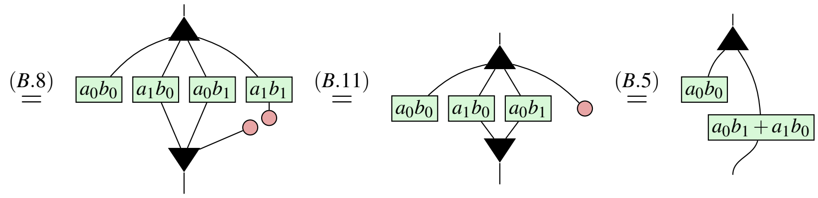
$\phi_1(p) \boxplus \phi_1(q) =$ 

$\overset{(B.5)}{=}$  $= \phi_1(p+q)$

Meanwhile, since $p \times q = a_0a_1 + (a_0b_1 + a_1b_0)x_1$,

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





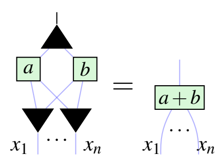
$$= \phi_1(p \times q)$$

Completing the base case.

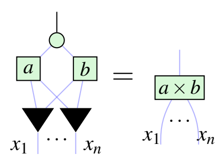
Inductive step:

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 $p(x_1,\ldots,x_{n+1}) = a_0 + a_1x_{n+1} + \ldots + a_{2^{n+1}-1}x_1x_2\ldots x_{n+1} \in \mathscr{P}_{n+1}$ 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 $x_1,\ldots,x_n$ will be drawn in light blue, while the x_{n+1} wires will be drawn in black. Thus the inductive hypothesis states that:



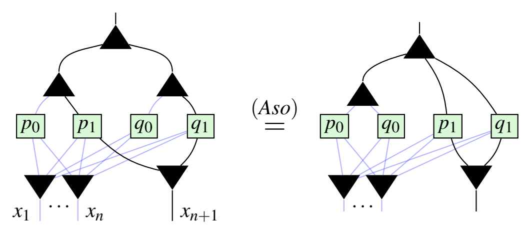
(IH1)

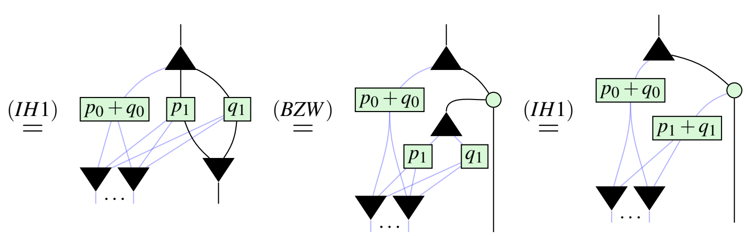


(IH2)

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:

$$\phi_{n+1}(p) \boxplus \phi_{n+1}(q) =$$

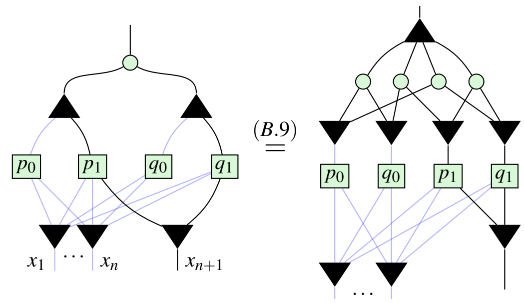


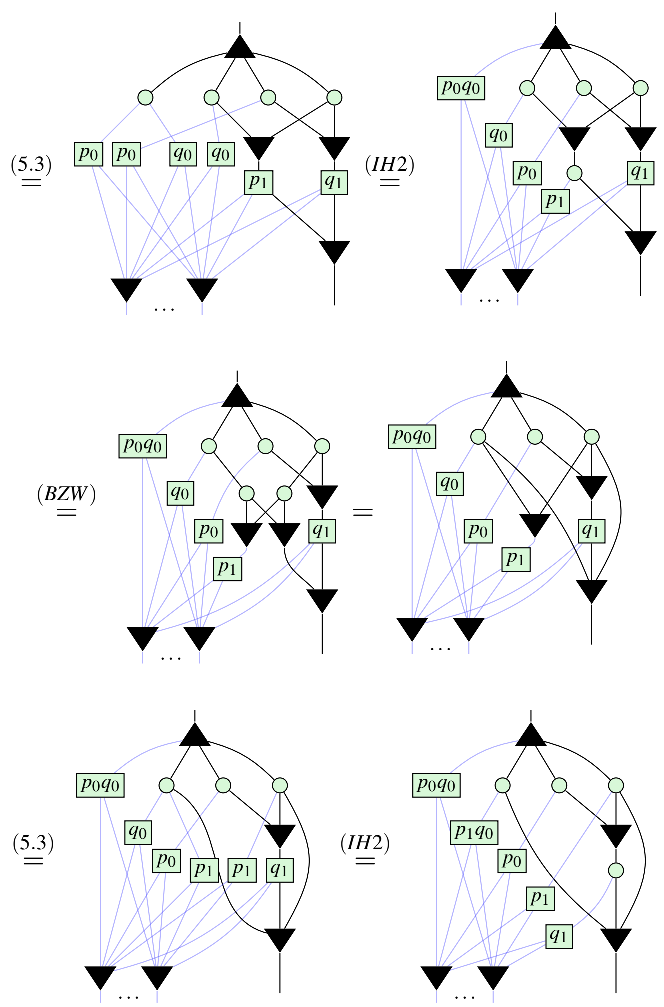


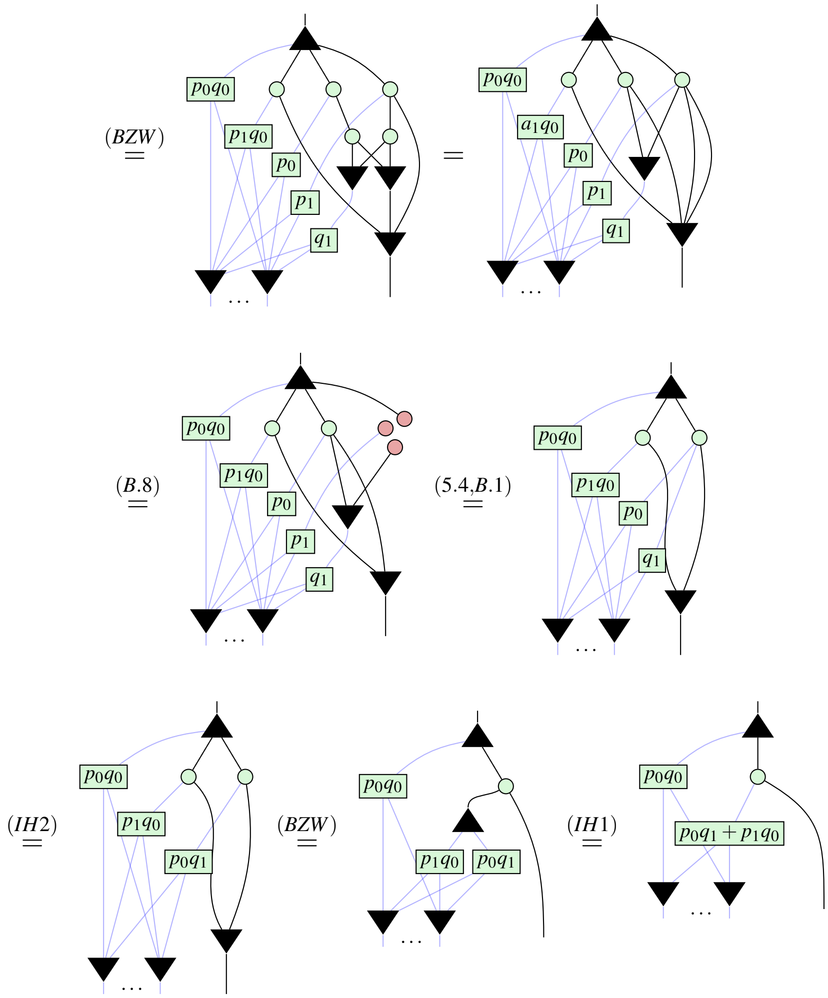
$$= \phi_{n+1}(p_0 + q_0 + (p_1 + q_1)x_{n+1}) = \phi_{n+1}(p+q)$$

Similarly, for multiplication:

$$\phi_{n+1}(p) \boxtimes \phi_{n+1}(q) =$$





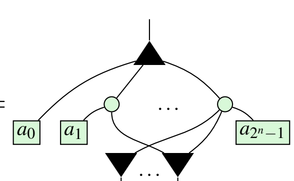


$$= \ \phi_{n+1}(p_0q_0 + (p_0q_1 + p_1q_0)x_{n+1}) \ = \ \phi_{n+1}(p \times q)$$

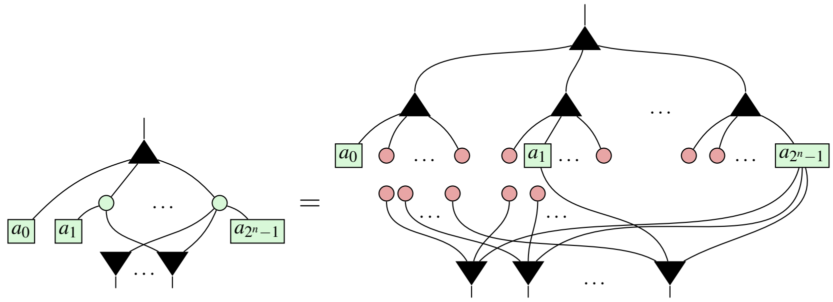
This completes the inductive step, proving that $\forall n > 1$, ϕ_n is a homomorphism.

Finally, to see ϕ_n is an isomorphism, we use proposition 7 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:



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

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

□