

Equational Reasoning with Controlled ZXW Diagrams

Edwin Agnew

Lia Yeh

Richie Yeung

Matthew Wilson

1 Abstract

asdf

2 Introduction

Considering broadly, controlling or branching to different possible linear maps, relations, or channels has been studied through many different approaches. In quantum algorithms common techniques are block encodings [7, 15] and linear combination of unitaries [3], while a number of categorical formalisations have included routed quantum circuits [23], the many-worlds calculus [2], categorifying signal flow diagrams [1], and classical and quantum control in quantum modal logic [18].

The question we are interested in, is how quantum graphical calculi such as the ZX, ZW, and ZH calculus and their combinations 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 [11]. More recently, controlled state and controlled matrix diagrams have been applied to addition and differentiation of ZX diagrams [10], differentiating and integrating ZX diagrams for quantum machine learning [24], Hamiltonian exponentiation and simulation [19], and non-linear optical quantum computing [6]. To sum ZX diagrams, these works have used controlled states along with the W generator from the ZW calculus [4].

Given how useful controlled diagrams have been, a natural question to ask is why they work: What their underlying mathematical structures are, and which equational rewrites they satisfy. We began exploring this starting from the unique normal form for states used for the first proofs of complete axiomatisation for qubit graphical calculi [8, 9]. We first give a simple proof showing an isomorphism between controlled states and 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. [26] 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 [25].

In this paper, we first define a fragment of the qubit ZW calculus corresponding to controlled states. This defines a subcategory we prove is isomorphic to multilinear polynomials $\mathbb{C}[x_1, \dots, x_n]/(x_1^2, \dots, x_n^2)$.

We then 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 Hilb_{\leq} of linear maps $n \rightarrow m$ such that $n \leq m$. This allows us to use the functorial boxes of Ref. [13] to control ZX diagrams with at least as many output wires as input wires. Furthermore, we apply the ZH calculus to show that under the appropriate circumstances, multiple applications of Ctrl i.e. multiple-controlling is monadic.

Next, we show that the set of all controlled n -partite states defines a commutative ring $(\tilde{\mathcal{S}}^n, \boxplus, \boxtimes)$. We introduce \boxplus which defines an Abelian group and \boxtimes which defines a commutative monoid, and show

that \boxtimes distributes over \boxplus . Analogously, we show that the set of all controlled square matrices on n qubits defines a non-commutative ring $(\tilde{M}^n, \blacktriangle, \circlearrowleft)$. 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 arbitrary qubit Hamiltonians in the ZXW calculus [19], even with all the terms black-boxed.

3 ZXW Calculus

3.1 Generators

The (qubit) ZXW calculus is build from the following generators:

- **Identity wire:**

$$\text{---} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- **Swap:**

$$\text{---} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **Z box:**

$$\text{---} := |0^m\rangle\langle 0^n| + e^{i\alpha}|1^m\rangle\langle 1^n|, \alpha \in \mathbb{C}$$

- **W node:**

$$\blacktriangle := |00\rangle\langle 0| + |01\rangle\langle 1| + |10\rangle\langle 1|$$

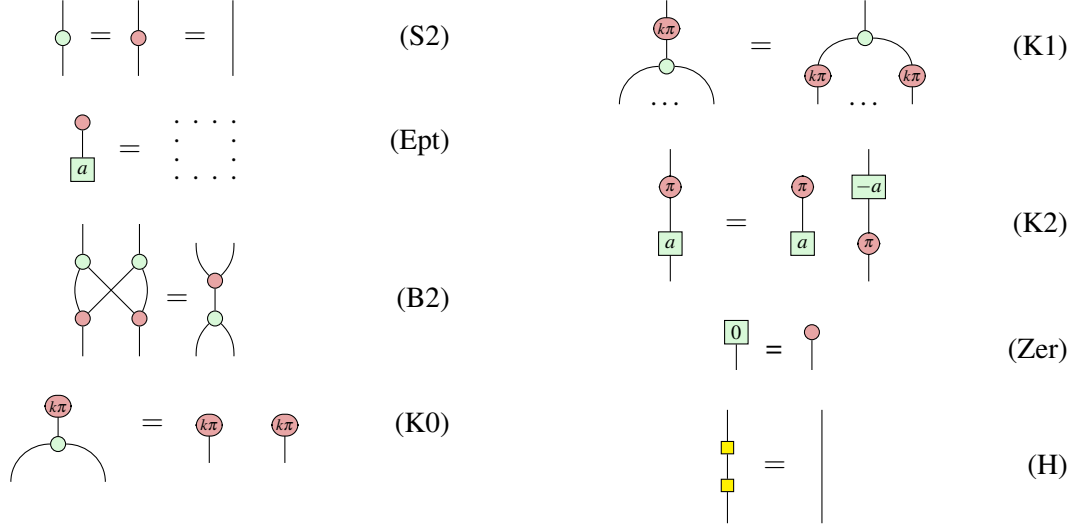
- **H box:**

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

3.2 Rules

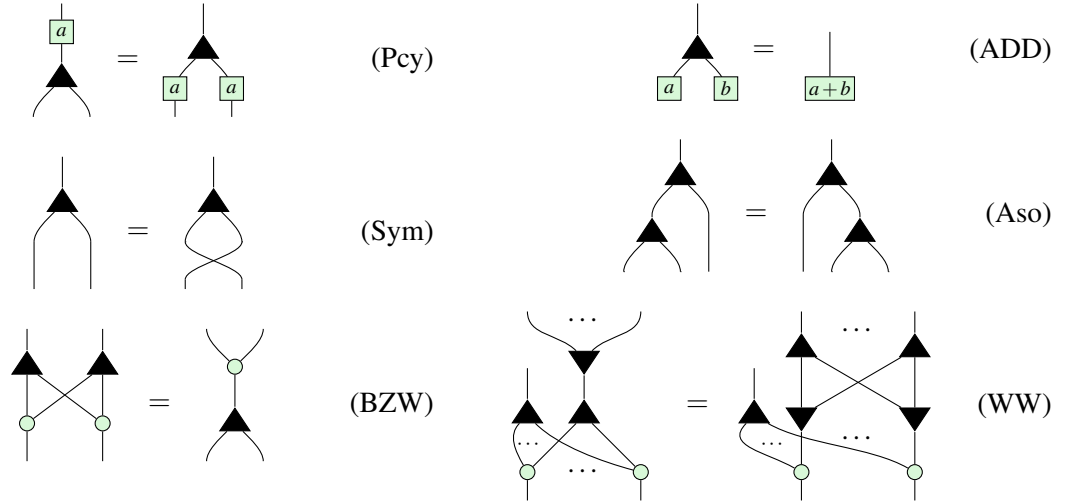
ZX Rules:

$$\begin{array}{c} \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} \quad (\text{S1})$$

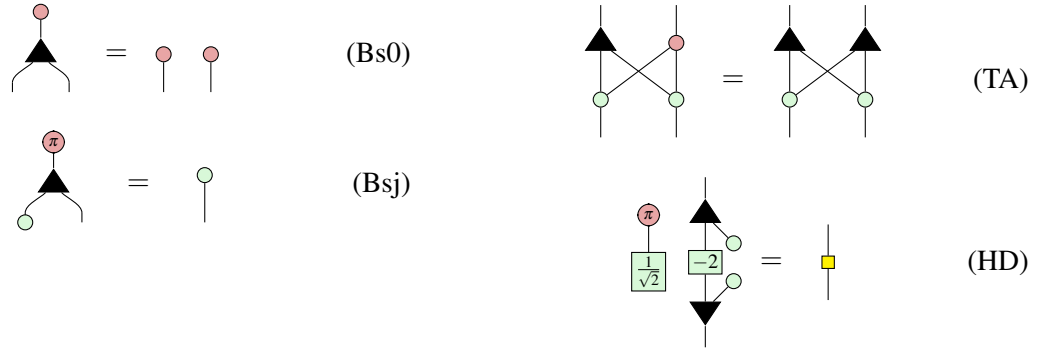


Where $k \in \{0, 1\}$.

ZW Rules:



ZXW Rules:



A number of basic lemmas are found in appendix A.

4 Controlled Diagrams

4.1 Definitions

As defined in [19],

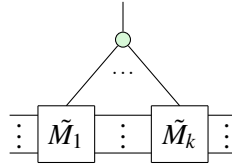
Definition 4.1. For an arbitrary square matrix D , the controlled matrix of D is the diagram \tilde{D} such that:

$$\begin{array}{c} \text{red circle} \\ | \\ \boxed{\tilde{D}} \\ \vdots \end{array} = \begin{array}{c} \vdots \\ \hline \vdots \end{array} \quad (4.1)$$

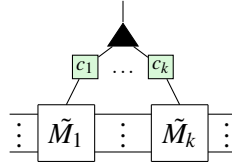
$$\begin{array}{c} \text{red circle with } \pi \\ | \\ \boxed{\tilde{D}} \\ \vdots \end{array} = \begin{array}{c} \vdots \\ \boxed{D} \\ \vdots \end{array} \quad (4.2)$$

It is possible to perform matrix arithmetic with controlled diagrams.

Proposition 1. Given controlled matrices $\tilde{M}_1, \dots, \tilde{M}_k$, the controlled matrix $\widetilde{\prod_i M_i}$ is given by



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



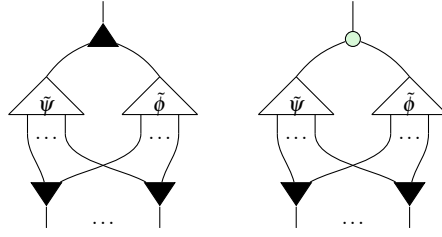
Proof. See propositions 3.3 and 3.4 in [19] □

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:

$$\begin{array}{c} \text{red circle} \\ | \\ \triangleup \tilde{\psi} \\ | \\ \vdots \end{array} = \begin{array}{c} \text{red circle} \\ \vdots \end{array} \quad \begin{array}{c} \text{red circle with } \pi \\ | \\ \triangleup \tilde{\psi} \\ | \\ \vdots \end{array} = \begin{array}{c} \triangleup \psi \\ | \\ \vdots \end{array} \quad (4.3)$$

The addition and multiplication of controlled states are defined similarly to controlled matrix arithmetic, except that a layer of \blacktriangledown s are appended at the bottom to preserve the number of outputs.

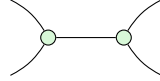


The role of \blacktriangledown is to *copy* inputs, as shown in section ??.

Combining the notions of controlled square matrices and controlled states, we can have a more general notion of controlled matrices.

Definition 4.3. For all $m \leq n$, the controlled matrix of an arbitrary matrix $M \in \mathbb{C}^{m \times n}$ is defined as the diagram \tilde{M} such that:

We focus on matrices with non-decreasing dimension to avoid cases like the following which fail to satisfy functoriality:



4.2 Functor

The operation of turning a non-dimension-decreasing matrix to its controlled diagram can be made into a lax monoidal functor. Let \mathbf{Hilb}_{\leq} be the subcategory of Hilbert spaces and non-dimension-decreasing linear transformations. Adding an additional horizontal wire to facilitate composition, $F : \mathbf{Hilb}_{\leq} \rightarrow \mathbf{Hilb}$ is defined as follows for arbitrary $D \in \text{Hom}_{\mathbf{Hilb}_{\leq}}(V, W)$.

$$F :: V \xrightarrow{\quad} D \xrightarrow{\quad} W \mapsto V \xrightarrow{\quad} \tilde{D} \xrightarrow{\quad} W \quad (4.4)$$

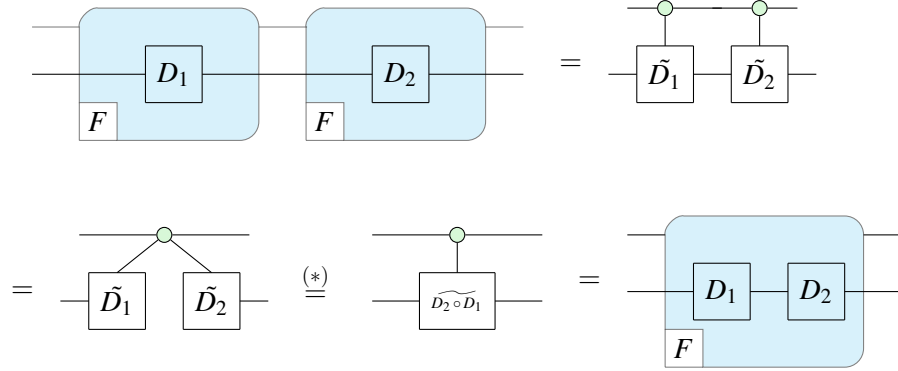
In the functorial box notation of [13], this would be:

$$V \xrightarrow{\quad} D \xrightarrow{\quad} W \xrightarrow{\quad} F \xrightarrow{\quad} W \quad (4.5)$$

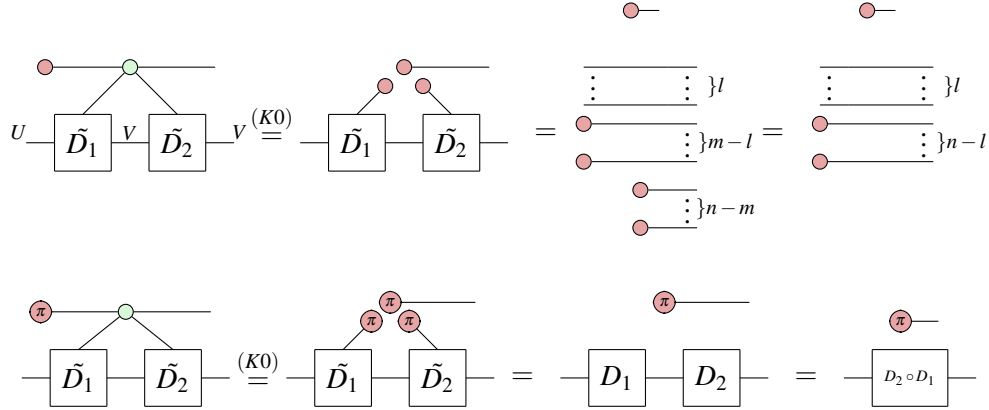
Proposition 2. The map F defined in (4.4) is a lax monoidal functor.

Proof. On $\text{id}_V : V \rightarrow V$:

Let $D_1 : U \rightarrow V$, $D_2 : V \rightarrow W$, where U, V, W have dimensions l, m, n respectively. Then composing $F(D_2) \circ F(D_1)$:

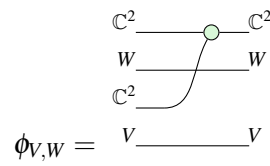


Where $(*)$ follows from

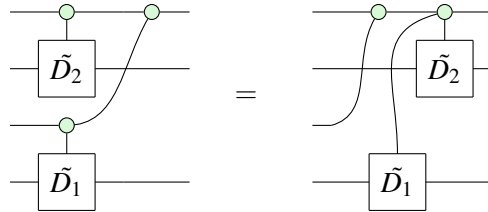


F preserves the monoidal unit since $\mathbf{1}_{\mathbf{Hilb}_\leq} = \mathbf{1}_{\mathbf{Hilb}} = \begin{smallmatrix} \vdots & \vdots \\ \vdots & \vdots \end{smallmatrix}$

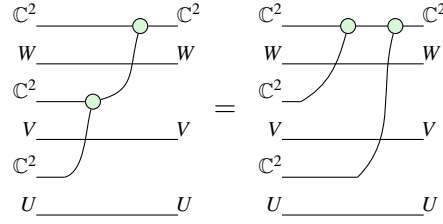
F is lax thanks to the following structure morphism: $\phi_{V,W} : F(V) \otimes F(W) \rightarrow F(V \otimes W)$:



ϕ is natural since for any $D_1 : V \rightarrow V'$, $D_2 : W \rightarrow W'$, we have:



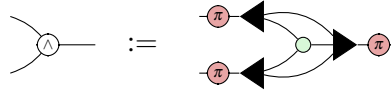
Finally, ϕ satisfies the coherence condition since for any U, V, W :



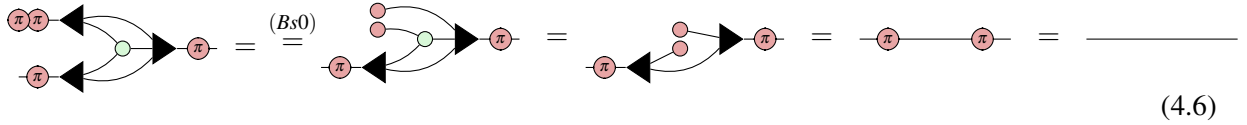
□

4.3 Monad

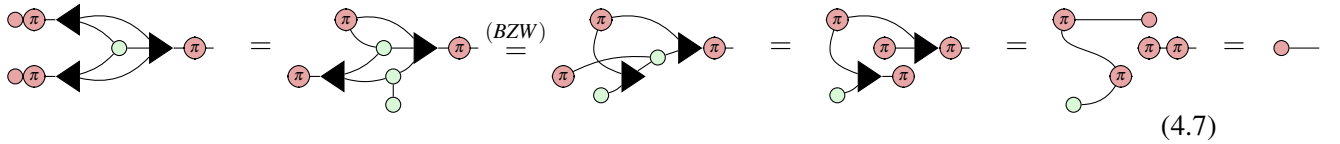
This section proves that controlling a controlled diagram gives the AND of the control wires, thus yielding a monad. First show how to represent the binary AND gate in the ZXW calculus.



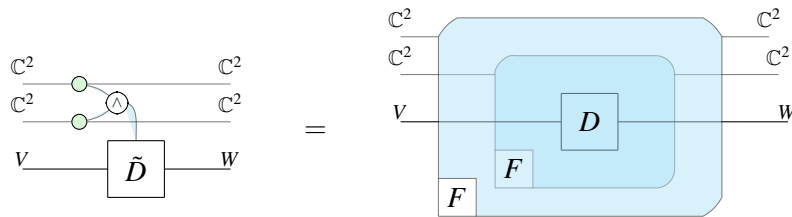
We can verify this computes the AND gate by computing on basis states.



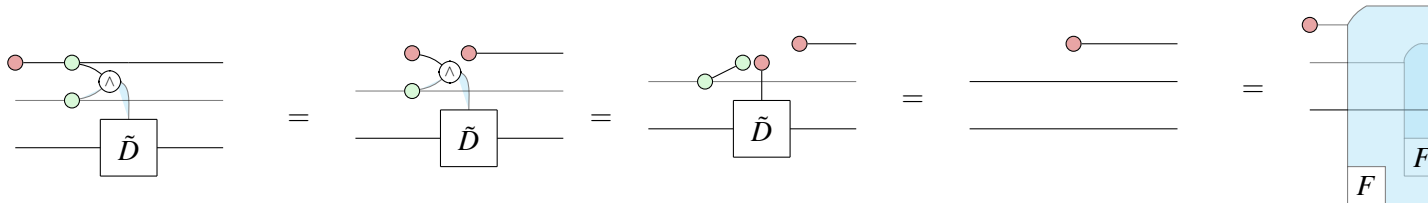
Thus $AND(1, x) = x$. Since the diagram is clearly commutative, it remains to check $AND(0, 0) = 0$.

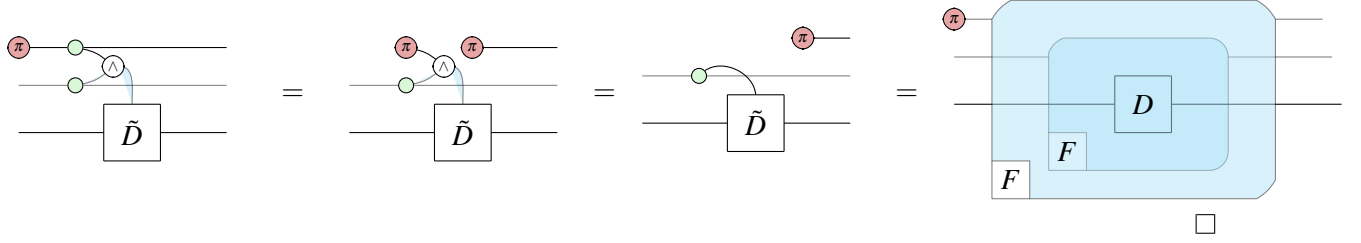


Proposition 3.



Proof. Plugging basis states:





In order to make multiple-control a monad, we first need to make F an endofunctor. Although F does take DND matrices to DND matrices, the definition of a monad requires that the components of the multiplication $\mu_V : F^2V \rightarrow FV$ live in the codomain. However, AND is a dimension decreasing matrix so does not live in \mathbf{Hilb}_\leq . To get around this, we define a new category

5 Polynomials

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 . 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 D ,*

(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.*

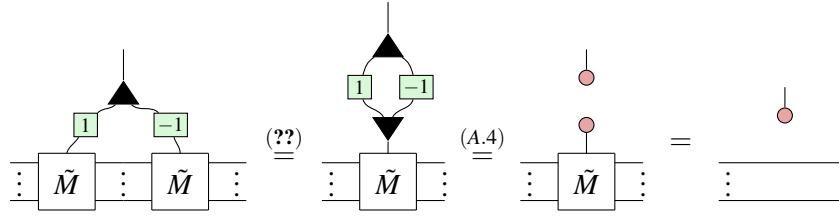
(5.2)

The additive identity is defined as $\text{red circle} \otimes I_n$:

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

Lemma 5.3. The additive inverse of \tilde{M} is $\begin{array}{|c|} \hline -1 \\ \hline \end{array} \circ \tilde{M}$

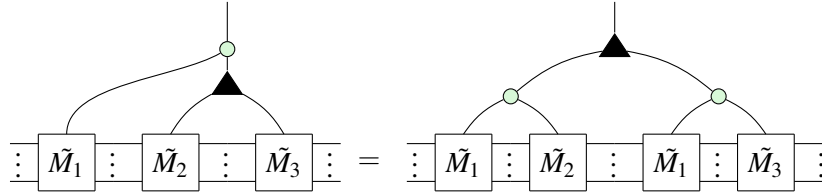
Proof.



□

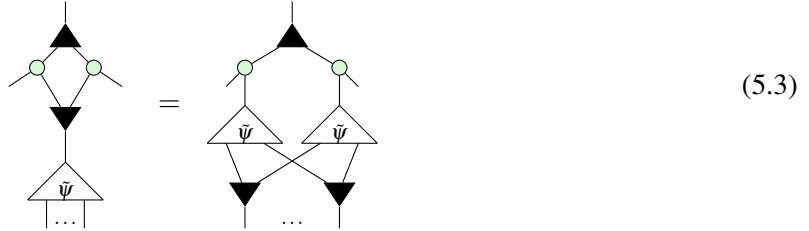
Finally, we prove distributivity.

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. $\tilde{\psi} \boxplus \tilde{0} = \tilde{\psi}$




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 $\tilde{\psi} \circ \begin{array}{|c|} \hline -1 \\ \hline \end{array}$

Associativity and commutativity of \boxtimes follow as before, using (S1) for $\begin{array}{|c|} \hline \text{green circle} \\ \hline \end{array}$. Finally, we must prove distributivity.

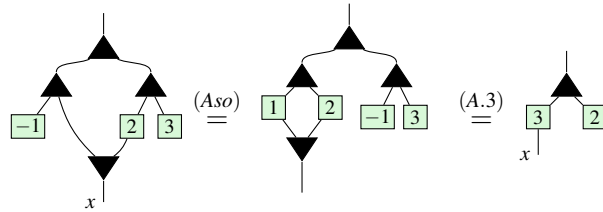
Lemma 5.9. $\tilde{\psi}_1 \boxtimes (\tilde{\psi}_2 \boxplus \tilde{\psi}_3) = (\tilde{\psi}_1 \boxtimes \tilde{\psi}_2) \boxplus (\tilde{\psi}_1 \boxtimes \tilde{\psi}_3)$

5.2 Arithmetic

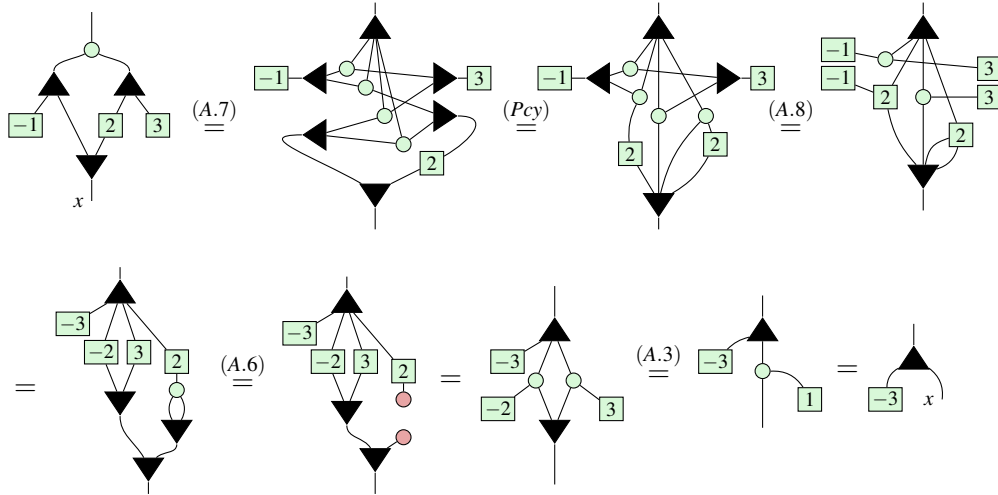
It's been known since 2011 that  can be used to add and multiply numberstates , respectively [5]. In the previous section we saw that  can moreover be used to copy controlled diagrams. In this section, we explain this connection by demonstrating that controlled states are in fact isomorphic to multilinear polynomials. 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 (A.6) 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  boxes.

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

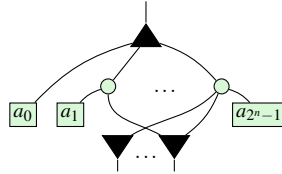
Lemma 5.10. *For any arithmetic diagram A ,*

$$\begin{array}{c} \bullet \\ | \\ \boxed{A} \\ | \\ \dots \end{array} = \begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \dots \end{array} \quad (5.4)$$

Proof. By definition, other than wires A contains only \blacktriangle , \odot , \blacktriangledown , and \boxed{a} . All \boxed{a} 's can be removed with (Ept). Meanwhile all the spiders copy \bullet due to (Bs0, K0, ??) respectively. \square

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 \blacktriangle , followed by a layer of \odot , followed by a layer of \boxed{a} , \blacktriangledown .

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



The i th coefficient a_i is connected to the k th \blacktriangledown iff the k th bit in the binary expansion of i is 1.

This normal form is very closely related to the completeness normal form (see [14]). Simply applying (TA) to the \blacktriangledown s at the bottom of a PNF and fusing the number boxes gives a CoNF diagram. 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 4.

$$\begin{array}{c} \bullet \\ | \\ \boxed{A} \\ | \\ \dots \end{array} = \begin{bmatrix} 1 & a_0 \\ 0 & a_1 \\ \dots & \dots \\ 0 & a_{2^n-1} \end{bmatrix} \quad (5.5)$$

Proof. See appendix B \square

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 5. All arithmetic diagrams can be written into PNF

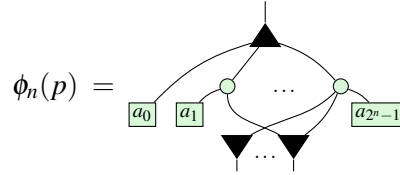
Proof. See appendix B \square

5.3 Isomorphism

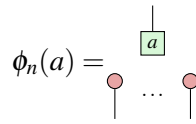
At last we can prove the isomorphism. Throughout we shall let \mathcal{P}_n denote the ring $\mathbb{C}[x_1, \dots, x_n]/(x_1^2, \dots, x_n^2)$.

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

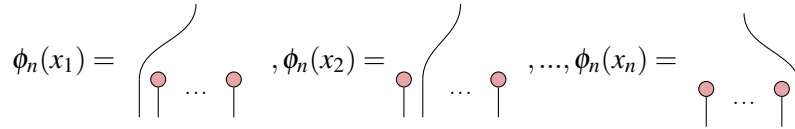
First, we shall define the map $\phi_n : \mathcal{P}_n \rightarrow \tilde{\mathcal{S}}_n$ before proving it induces an isomorphism. ϕ_n is defined to map an arbitrary polynomial $p(x_1, \dots, x_n) = a_0 + a_1 x_1 + \dots + a_{2^n-1} x_1 x_2 \dots x_n$ to the following PNF:



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



And mapping indeterminates x_i :



The full proof is found in appendix B.

6 Applications

7 Conclusion

In summary, we first proved an isomorphism between controlled states in the ZXW calculus, and multilinear polynomials $\mathbb{C}[x_1, \dots, x_n]/(x_1^2, \dots, x_n^2)$. We then showed that the higher-order map Ctrl is a lax monoidal functor on all linear maps $n \rightarrow m$ such that $n \leq m$, letting us add functorial boxes to such ZXW diagrams. We moreover find that multiple-control is monadic.

We further showed that all controlled n -partite states form a commutative ring, 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 thus 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 [14]. A starting guess would be that qudit controlled states are isomorphic to polynomials $\mathbb{C}[x_1, \dots, x_n]/(x_1^d, \dots, x_n^d)$ due to the Hopf law between Z and W. Qudit multiple-control would likely have more complex structure than the monadic case here for qubits, considering the ingredients of all prime-dimensional d -ary classical reversible gates built in Ref. [17].

We would like to try sector-preserving channels [22] and scoped effects [12] 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. [26], with that of quantum circuits and ZX diagrams in Refs. [21, 24, 10]. 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. [16], or translated to an equational theory for a quantum programming language like in Ref. [20].

8 Acknowledgements

We thank Razin Shaikh and Itai Leigh for insightful discussions. LY is funded by a Google PhD Fellowship.

References

- [1] BAEZ, J. C., AND ERBELE, J. Categories in control, 2015.
- [2] CHARDONNET, K., DE VISME, M., VALIRON, B., AND VILMART, R. The many-worlds calculus, 2023.
- [3] CHILDS, A. M., AND WIEBE, N. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Info. Comput.* 12, 11–12 (nov 2012), 901–924.
- [4] 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.
- [5] COECKE, B., KISSINGER, A., MERRY, A., AND ROY, S. The ghz/w-calculus contains rational arithmetic. *arXiv preprint arXiv:1103.2812* (2011).
- [6] 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).
- [7] 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.
- [8] HADZIHASANOVIC, A. The algebra of entanglement and the geometry of composition, 2017.
- [9] 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.
- [10] JEANDEL, E., PERDRIX, S., AND VESHCHEROVA, M. Addition and differentiation of zx-diagrams, 2024.
- [11] JEANDEL, E., PERDRIX, S., AND VILMART, R. A generic normal form for zx-diagrams and application to the rational angle completeness, 2018.

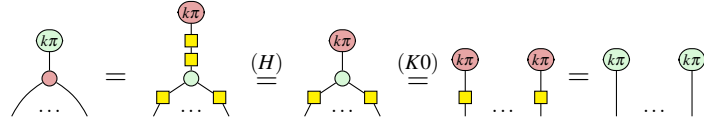
- [12] LINDLEY, S., MATACHE, C., MOSS, S., STATON, S., WU, N., AND YANG, Z. Scoped effects as parameterized algebraic theories, 2024.
- [13] MELLIÈS, P.-A. Functorial boxes in string diagrams. In *International Workshop on Computer Science Logic* (2006), Springer, pp. 1–30.
- [14] POÓR, B., WANG, Q., 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).
- [15] RALL, P. Quantum algorithms for estimating physical quantities using block encodings. *Phys. Rev. A* 102 (Aug 2020), 022408.
- [16] 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).
- [17] 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.
- [18] SATI, H., AND SCHREIBER, U. The quantum monadology, 2023.
- [19] SHAIKH, R. A., WANG, Q., AND YEUNG, R. How to sum and exponentiate hamiltonians in zxw calculus. *arXiv preprint arXiv:2212.04462* (2022).
- [20] STATON, S. Algebraic effects, linearity, and quantum programming languages. *SIGPLAN Not.* 50, 1 (jan 2015), 395–406.
- [21] 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.
- [22] 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.
- [23] VANRIETVELDE, A., KRISTJÁNSSON, H., AND BARRETT, J. Routed quantum circuits. *Quantum* 5 (2021), 503.
- [24] WANG, Q., YEUNG, R., AND KOCH, M. Differentiating and integrating zx diagrams with applications to quantum machine learning, 2022.
- [25] 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.
- [26] 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 Basic Lemmas

Lemma A.1.

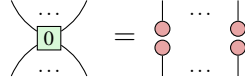
$$\begin{array}{c} \textcircled{k\pi} \\ | \\ \text{---} \end{array} = \begin{array}{c} \textcircled{k\pi} \\ | \\ \text{---} \end{array} \quad \begin{array}{c} \textcircled{k\pi} \\ | \\ \text{---} \end{array} \quad (A.1)$$

Proof.



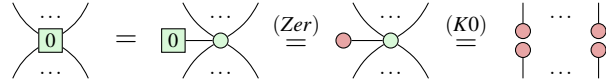
□

Lemma A.2.



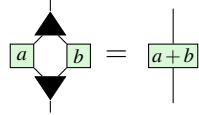
(A.2)

Proof.



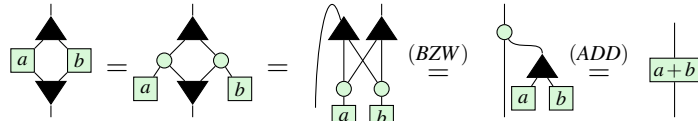
□

Lemma A.3.



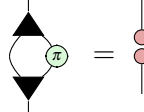
(A.3)

Proof.



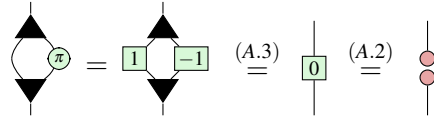
□

Lemma A.4.



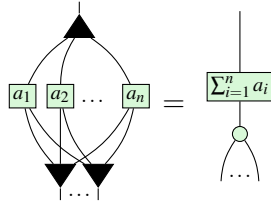
(A.4)

Proof.



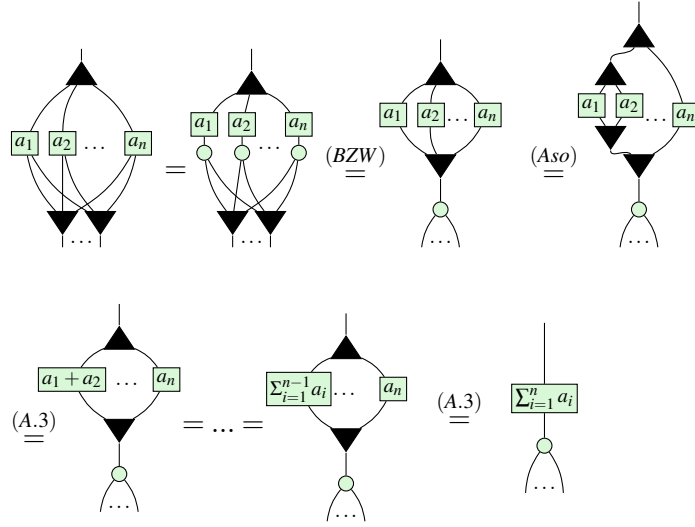
□

Lemma A.5.



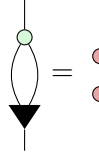
(A.5)

Proof.



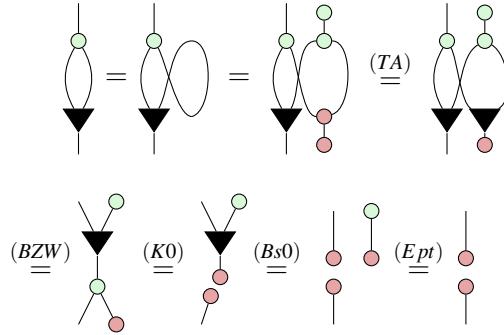
□

Lemma A.6.



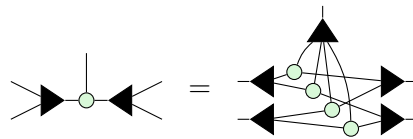
(A.6)

Proof.



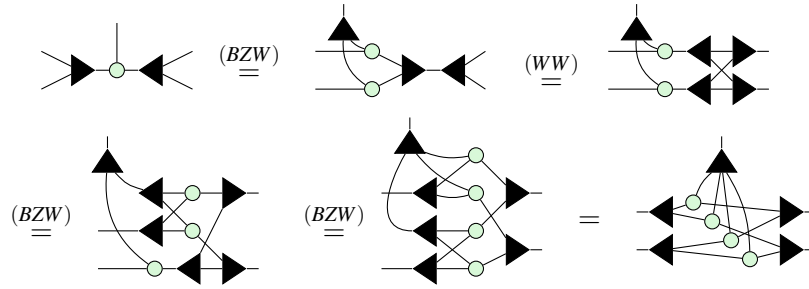
□

Lemma A.7.



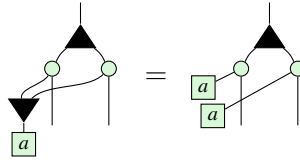
(A.7)

Proof.



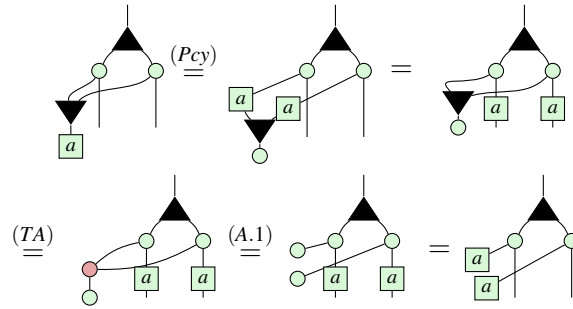
□

Lemma A.8.



(A.8)

Proof.



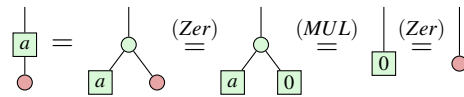
□

Lemma A.9.



(A.9)

Proof.

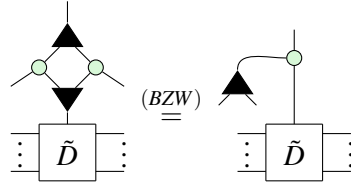


□

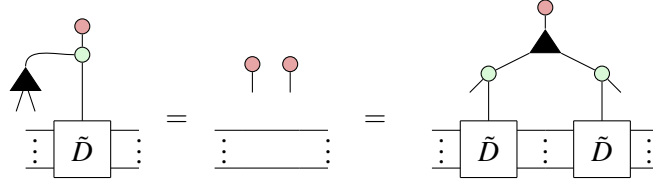
Appendix B Proofs

Proof of lemma 5.1

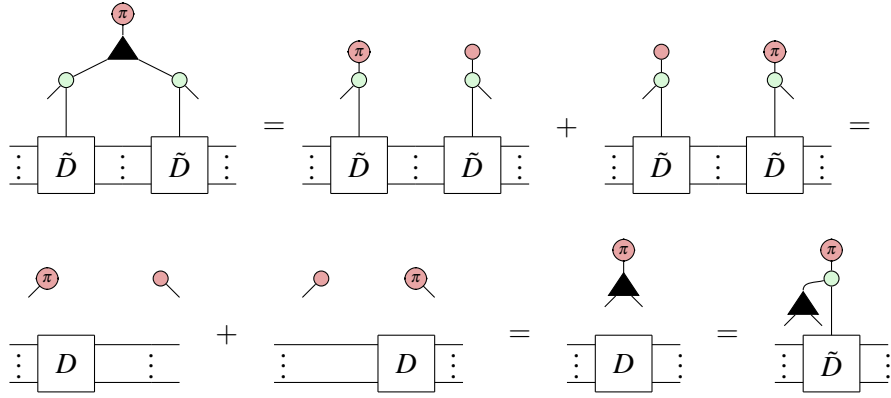
Proof. First of all, using (BZW) we can rewrite the LHS to



Then clearly



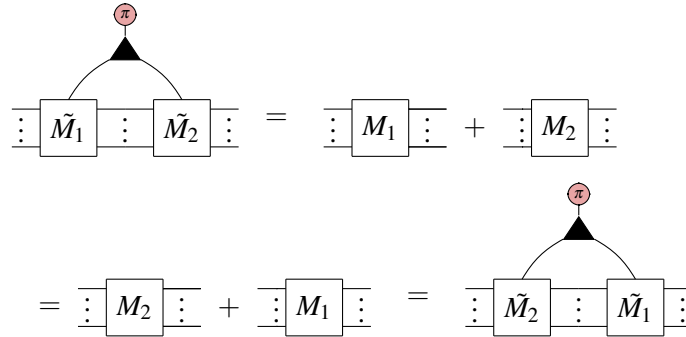
Meanwhile,



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

Proof of lemma 5.2

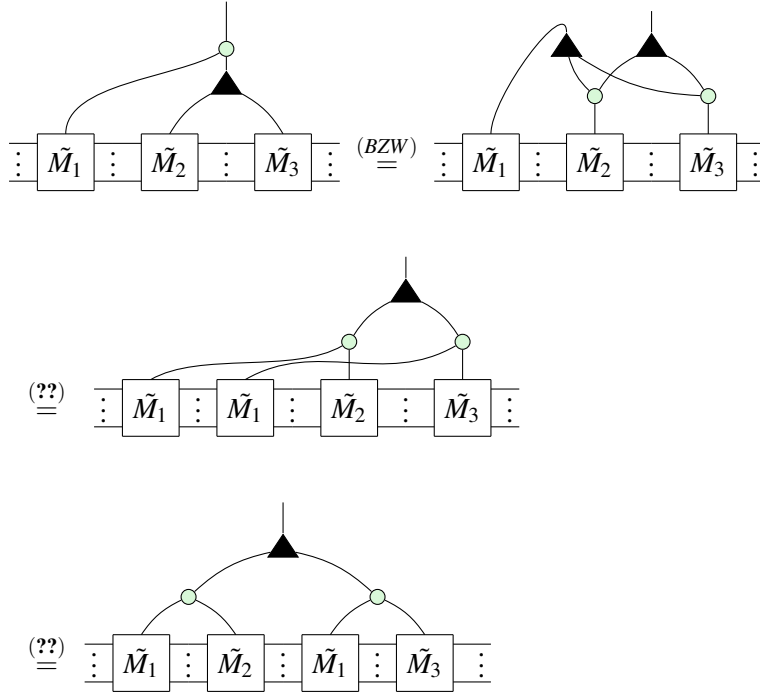
Proof. We prove by plugging red and commutativity of matrix addition. By definition of controlled matrices, plugging $\begin{smallmatrix} \text{red circle} \\ | \end{smallmatrix}$ gives I_n on both sides. Meanwhile, plugging $\begin{smallmatrix} \pi \\ | \end{smallmatrix}$ gives:



\square

Proof of lemma ??

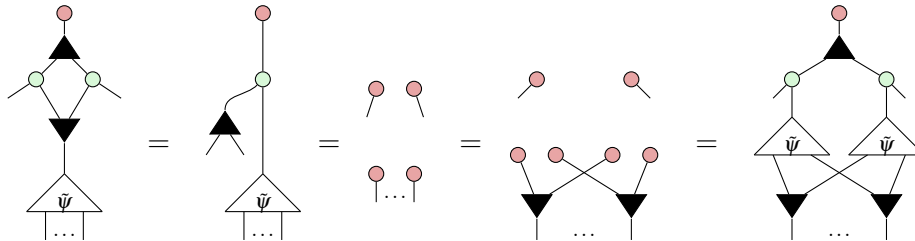
Proof.



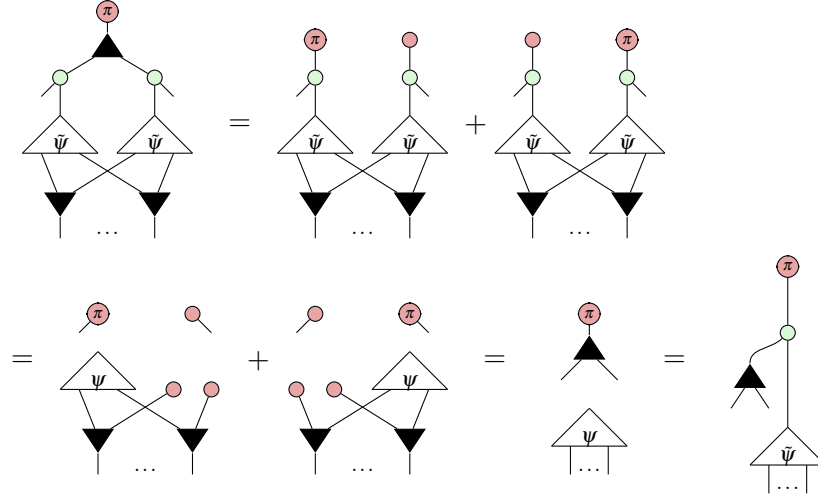
□

Proof of proposition 5.5

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

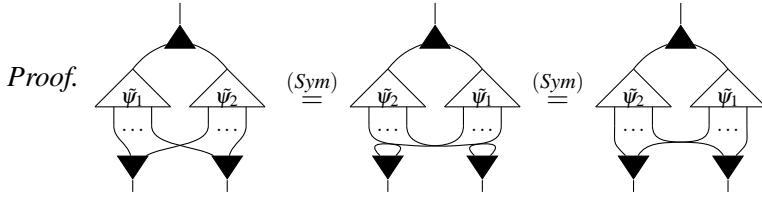


Meanwhile, plugging $|1\rangle$ gives



Completing the proof □

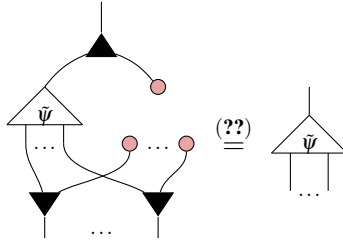
Proof of lemma 5.6



Proof of lemma 5.7

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

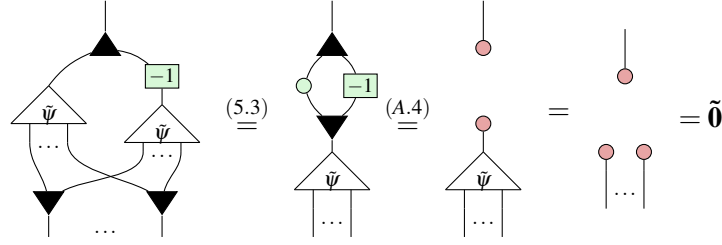
Then we have:



□

Proof of lemma 5.8

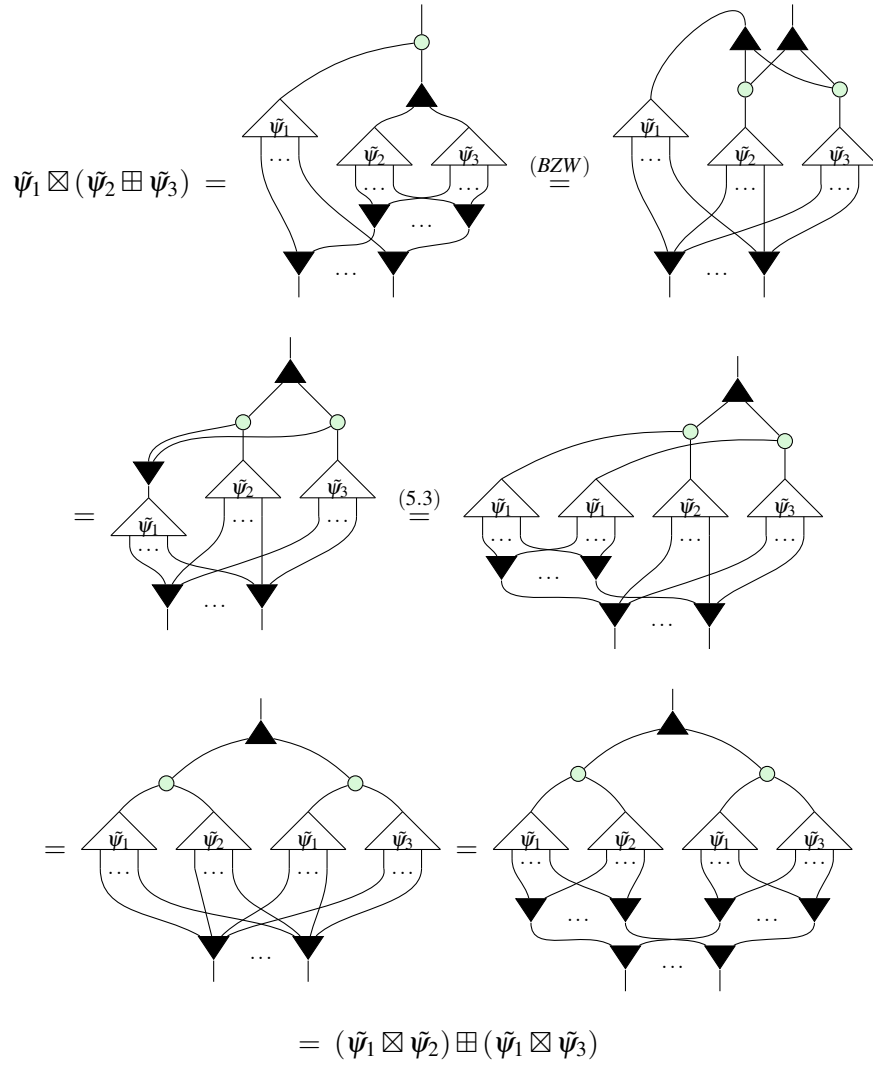
Proof. $\tilde{\psi} \circ \boxed{-1}$ is still a controlled state since $\boxed{-1}$ does nothing to \bullet . Then $\tilde{\psi} \circ \boxed{-1}$ inverts $\tilde{\psi}$ since:



□

Proof of lemma 5.8

Proof.



□

Proof of proposition 4

Proof. We prove by induction on n .

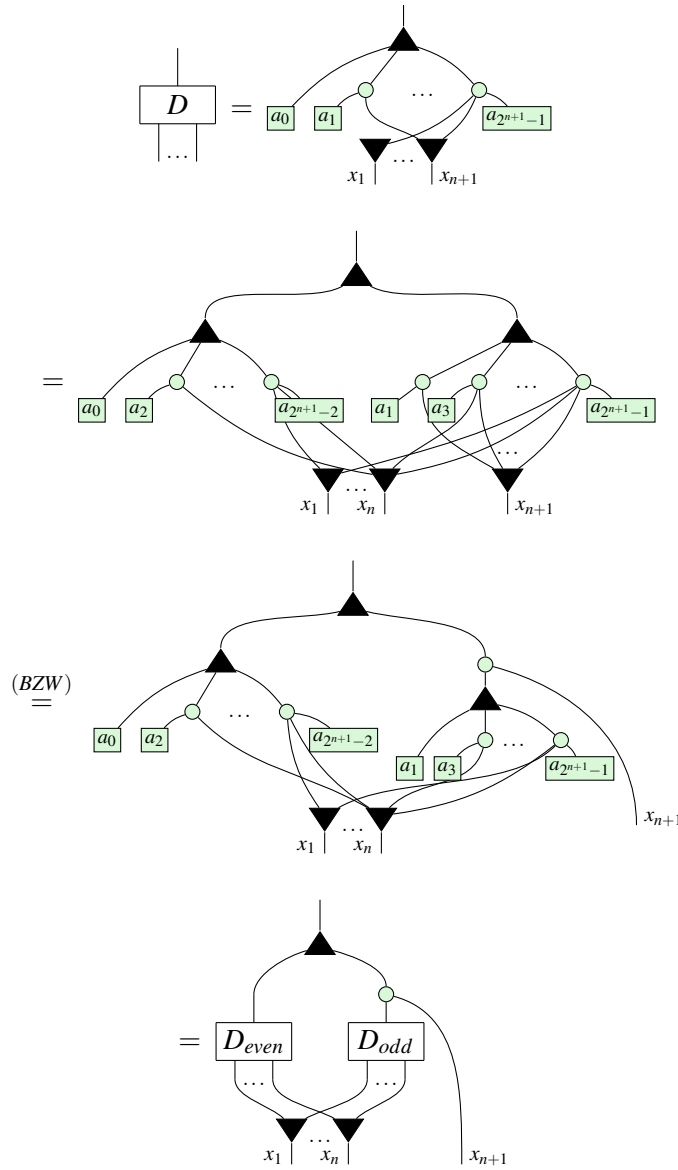
For the base case, $n = 0$. The only PNF with no outputs is a number so we have:

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

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.

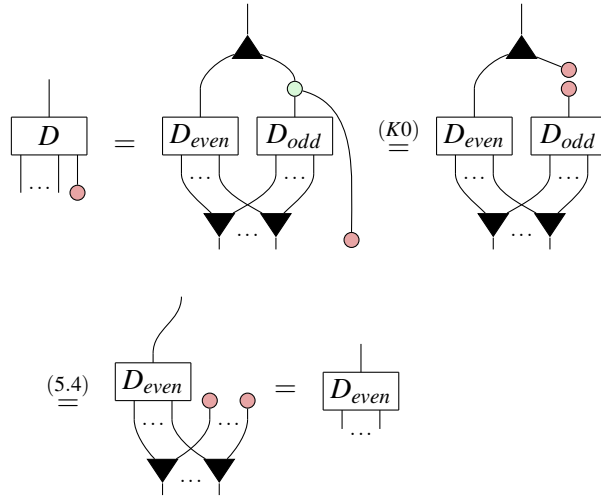
Let D be an arbitrary PNF with $n + 1$ outputs. Firstly, observe that x_{n+1} is connected to only the odd coefficients $\{a_{2k+1}\}$ since these are exactly the indices with 1 in the least significant bit. Thus we can rewrite:



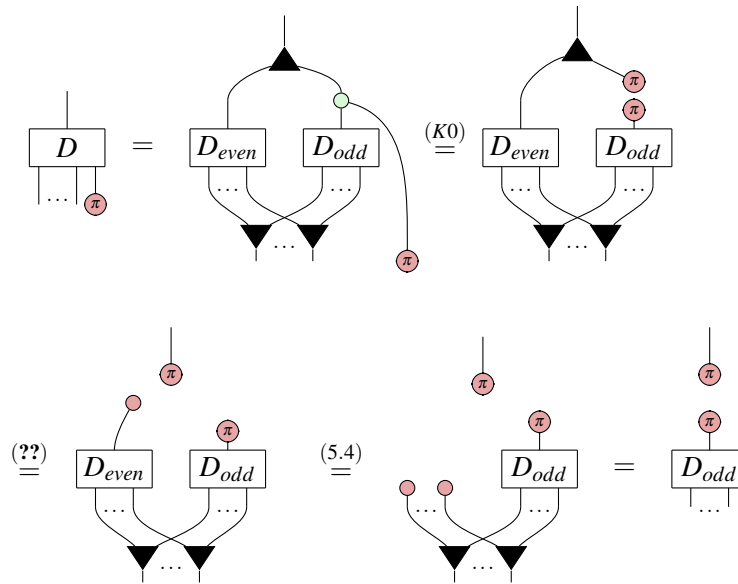
Where D_{even}, D_{odd} are PNF diagrams. Since they are over n variables, we can apply the inductive hypothesis and obtain:

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

Next, plugging red we observe:



Meanwhile,



Summing these together,

$$\begin{aligned}
 \begin{array}{c} | \\ \boxed{D} \\ \dots \end{array} &= \begin{array}{c} | \\ \boxed{D} \\ \dots \end{array} + \begin{array}{c} | \\ \boxed{D} \\ \dots \end{array} = \begin{array}{c} | \\ \boxed{D_{\text{even}}} \\ \dots \end{array} + \begin{array}{c} | \\ \boxed{D_{\text{odd}}} \\ \dots \end{array} \\
 &= (D_{\text{even}} \otimes |0\rangle) + (D_{\text{odd}} |1\rangle \langle 1| \otimes |1\rangle) \\
 &\stackrel{(*)}{=} \begin{bmatrix} 1 & a_0 \\ 0 & a_2 \\ \dots & \dots \\ 0 & a_{2^{n+1}-2} \end{bmatrix} \otimes |0\rangle + \begin{bmatrix} 0 & a_1 \\ 0 & a_3 \\ \dots & \dots \\ 0 & a_{2^{n+1}-1} \end{bmatrix} \otimes |1\rangle \\
 &= \begin{bmatrix} 1 & a_0 \\ 0 & 0 \\ 0 & a_2 \\ 0 & 0 \\ \dots & \dots \\ 0 & a_{2^{n+1}-2} \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & a_1 \\ 0 & 0 \\ 0 & a_3 \\ \dots & \dots \\ 0 & 0 \\ 0 & a_{2^{n+1}-1} \end{bmatrix} = \begin{bmatrix} 1 & a_0 \\ 0 & a_1 \\ 0 & a_2 \\ 0 & a_3 \\ \dots & \dots \\ 0 & a_{2^{n+1}-2} \\ 0 & a_{2^{n+1}-1} \end{bmatrix}
 \end{aligned}$$

Completing the inductive step. □

Proof of proposition 5

Proof. Let A be an arithmetic diagram. If $A = \boxed{a}$, we are done.

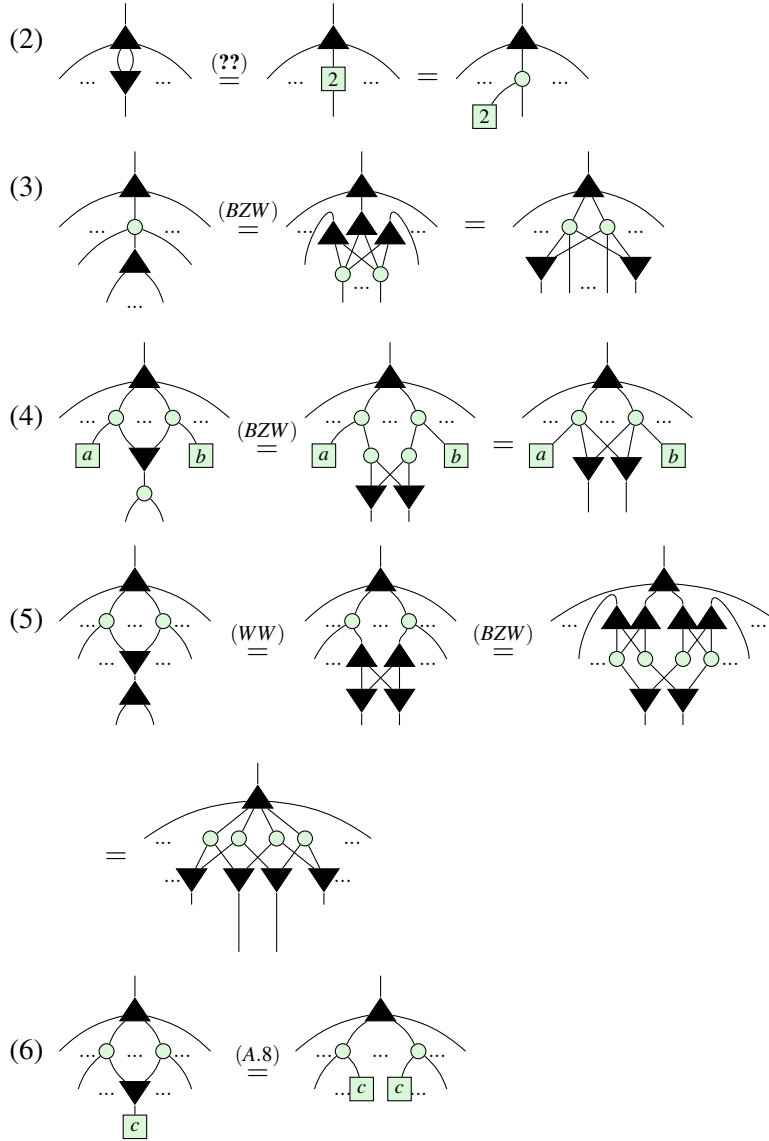
Otherwise, A has at least one output. First, we shall rewrite A into three layers, consisting of: (1) a single W at the top, (2) a layer of $\begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \dots \end{array}$ and (3) a layer of \boxed{a} 's and $\begin{array}{c} \blacktriangledown \\ \diagup \quad \diagdown \\ \dots \end{array}$'s. Then we shall collect terms and order the boxes to produce a PNF.

If the top of A is not already $\begin{array}{c} \blacktriangle \\ \diagup \quad \diagdown \\ \dots \end{array}$, it must be $\begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \dots \end{array}$. It cannot be \boxed{a} since the remaining arithmetic diagram would then have no inputs which is impossible. It cannot be $\begin{array}{c} \blacktriangledown \\ \diagup \quad \diagdown \\ \dots \end{array}$ since there is only one input and arithmetic diagrams cannot contain \cap . Thus we can rewrite:

$$(1) \quad \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \dots \end{array} \stackrel{??}{=} \begin{array}{c} \blacktriangle \\ \diagup \quad \diagdown \\ \dots \end{array} \begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \dots \end{array}$$


(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

(A.6) is applied and simplified with (K0) to remove $\begin{array}{c} \circ \\ \diagup \quad \diagdown \\ \dots \end{array}$ 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 \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 \blacktriangledown 's, use (A.5) to fuse them together.
- (8) Pad: use (A.2) 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. □

B.1 Isomorphism

Proof of Theorem 5.1

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

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

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 $p, q \in \mathcal{P}_1$. 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 $p+q = a_0+b_0 + (a_1+b_1)x_1$,

$$\begin{aligned} \phi_1(p) \boxplus \phi_1(q) &= \begin{array}{c} \text{Diagram 1: A triangle with a green circle at the top. The bottom edge has four boxes: } a_0, a_1, b_0, b_1. \text{ Arrows connect } a_0 \text{ to } a_1, a_1 \text{ to } b_0, b_0 \text{ to } b_1, \text{ and } a_0 \text{ to } b_1. \end{array} \\ &= \begin{array}{c} \text{Diagram 2: A triangle with a green circle at the top. The bottom edge has four boxes: } a_0, a_1, b_0, b_1. \text{ Arrows connect } a_0 \text{ to } a_1, a_1 \text{ to } b_0, b_0 \text{ to } b_1, \text{ and } a_0 \text{ to } b_1. \end{array} \\ &= \begin{array}{c} \text{Diagram 3: A triangle with a green circle at the top. The bottom edge has two boxes: } a_0+b_0 \text{ and } a_1+b_1. \end{array} \\ &\stackrel{(A.3)}{=} \begin{array}{c} \text{Diagram 4: A triangle with a green circle at the top. The bottom edge has two boxes: } a_0+b_0 \text{ and } a_1+b_1. \end{array} = \phi_1(p+q) \end{aligned}$$

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

$$\begin{aligned} \phi_1(p) \boxtimes \phi_1(q) &= \begin{array}{c} \text{Diagram 1: A triangle with a green circle at the top. The bottom edge has four boxes: } a_0, a_1, b_0, b_1. \end{array} \\ &\stackrel{(A.7)}{=} \begin{array}{c} \text{Diagram 2: A triangle with a green circle at the top. The bottom edge has four boxes: } a_0, b_0, a_1, b_1. \end{array} \\ &\stackrel{(A.8)}{=} \begin{array}{c} \text{Diagram 3: A triangle with a green circle at the top. The bottom edge has four boxes: } a_0b_0, b_0, a_0, a_1, b_1. \end{array} \\ &\stackrel{(Pcy)}{=} \begin{array}{c} \text{Diagram 4: A triangle with a green circle at the top. The bottom edge has four boxes: } a_0b_0, a_1b_0, a_0b_1, a_1b_1. \end{array} \end{aligned}$$

$$\begin{aligned}
& \stackrel{(A.6)}{=} \text{Diagram 1} \stackrel{(A.9)}{=} \text{Diagram 2} \stackrel{(A.3)}{=} \text{Diagram 3} \\
& = \phi_1(p \times q)
\end{aligned}$$

Completing the base case.

Inductive step:

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

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 $p(x_1, \dots, x_{n+1}) = a_0 + a_1x_{n+1} + \dots + a_{2^{n+1}-1}x_1x_2\dots x_{n+1} \in \mathcal{P}_{n+1}$ as $p(x_{n+1}) = p_0 + p_1x_{n+1}$, where $p_0, p_1 \in \mathcal{P}_n$. 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, \dots, 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:

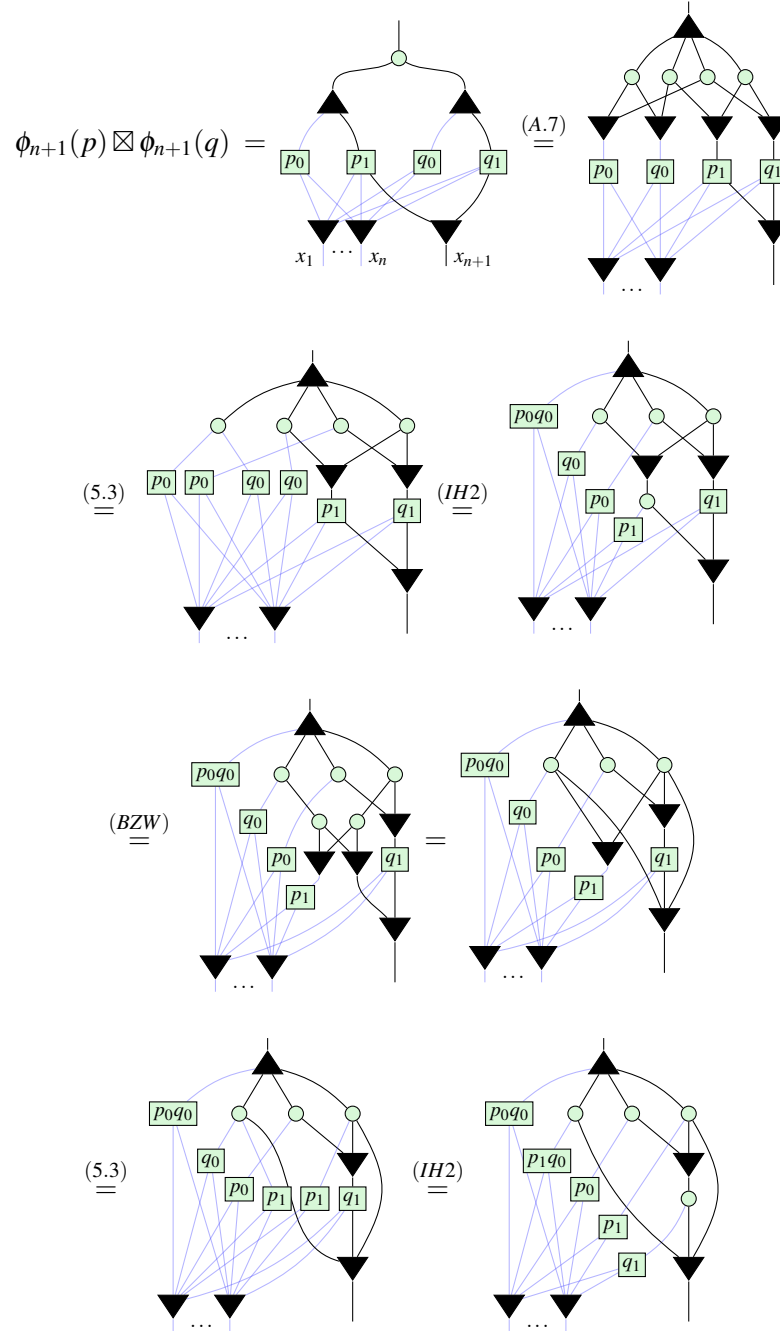
$$\begin{array}{c} \text{Diagram 1} \end{array} \stackrel{(IH1)}{=} \begin{array}{c} \text{Diagram 2} \end{array}$$

$$\begin{array}{c} \text{Diagram 1} \end{array} \stackrel{(IH2)}{=} \begin{array}{c} \text{Diagram 2} \end{array}$$

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

$$\begin{aligned}
\phi_{n+1}(p) \boxplus \phi_{n+1}(q) &= \text{Diagram 1} \stackrel{(Aso)}{=} \text{Diagram 2} \\
&\stackrel{(IH1)}{=} \text{Diagram 3} \stackrel{(BZW)}{=} \text{Diagram 4} \stackrel{(IH1)}{=} \text{Diagram 5} \\
&= \phi_{n+1}(p_0 + q_0 + (p_1 + q_1)x_{n+1}) = \phi_{n+1}(p + q)
\end{aligned}$$

Similarly, for multiplication:



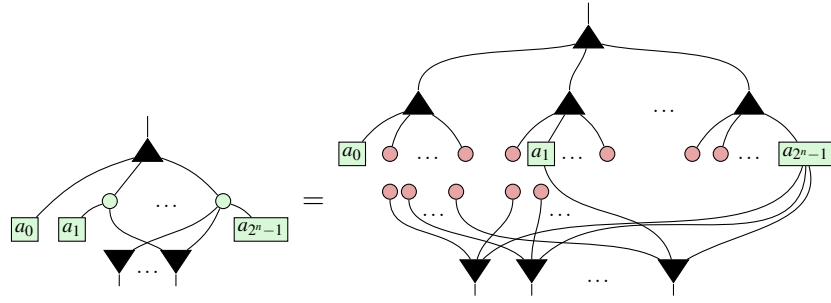
$$\begin{array}{c}
\begin{array}{ccc}
\begin{array}{c} \text{(BZW)} \\ \equiv \end{array} & \begin{array}{c} \text{Diagram 1} \end{array} & = & \begin{array}{c} \text{Diagram 2} \end{array} \\
\begin{array}{c} \text{(A.6)} \\ \equiv \end{array} & \begin{array}{c} \text{Diagram 3} \end{array} & \equiv & \begin{array}{c} \text{Diagram 4} \end{array} \\
\begin{array}{c} \text{(IH2)} \\ \equiv \end{array} & \begin{array}{c} \text{Diagram 5} \end{array} & \equiv & \begin{array}{c} \text{Diagram 6} \end{array} & \equiv & \begin{array}{c} \text{Diagram 7} \end{array} \\
& \text{(BZW)} & & \text{(IH1)} & & \\
& \equiv & & \equiv & & \\
& \begin{array}{c} \text{Diagram 8} \end{array} & & \begin{array}{c} \text{Diagram 9} \end{array} & & \\
& = \phi_{n+1}(p_0q_0 + (p_0q_1 + p_1q_0)x_{n+1}) = \phi_{n+1}(p \times q)
\end{array}$$

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 5 to write an arbitrary controlled state in PNF:

$$\begin{bmatrix} 1 & a_0 \\ 0 & a_1 \\ \vdots & \vdots \\ 0 & a_{2^n-1} \end{bmatrix} = \begin{array}{c} \text{Diagram 10} \end{array}$$

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



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

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

□