AND-gates in ZX-calculus:
spider nest identities and QBC-completeness

Anthony Munson
Mathematical Institute
University of Oxford
anthony.munson@queens.ox.ac.uk

Bob Coecke and Quanlong Wang
Department of Computer Science, University of Oxford
Cambridge Quantum Computing Ltd.
bob.coecke, quanlong.wang@cs.ox.ac.uk

In this paper we exploit the utility of the triangle symbol which has a complicated expression in terms of spider diagrams in ZX-calculus, and its role within the ZX-representation of AND-gates in particular. First, we derive spider nest identities which are of key importance to recent developments in quantum circuit optimisation and T-count reduction in particular. Then, using the same rule set, we prove a completeness theorem for quantum Boolean circuits (QBCs) whose rewriting rules can be directly used for a new method of T-count reduction. We give an algorithm based on this method and show that the results of our algorithm outperform the results of all the previous best non-probabilistic algorithms.

1 Introduction

The ZX-calculus [9, 10] is a universal graphical language for qubit theory, which comes equipped with simple rewriting rules that enable one to transform diagrams representing one quantum process into another quantum process. More broadly, it is the work-horse of categorical quantum mechanics, which aims for a high-level formulation of quantum theory [1, 12].

Recently ZX-calculus has been completed by Ng and Wang [25], that is, provided with sufficient additional rules so that any equation between matrices in Hilbert space can be derived in ZX-calculus. This followed earlier completions by Backens for stabiliser theory [3] and one-qubit Clifford+T circuits [4], and by Jeandel, Perdrix and Vilmart for general Clifford+T theory [19].

This paper concerns with the ‘utility’ of ZX-rules. While in principle the rules in [25] are sufficient to any other rule, it is by no means guaranteed that it is in any way intuitive, easy, or even realistic to do so. While the original rules of [9, 10] have been inherited by all ZX-calculi, comparing [25] with two more recent universal completeness results [20, 28] one immediately notices that the additional rules that give completeness are entirely different in each of the papers, and their relationship is by no means obvious.

Consequently, the new game in town is to match rules on their utility, and in some cases we will need to derive new rules. In fact, the new rule of the completeness theorem of [28] was in fact discovered/derived by two of the current authors with the particular purpose of quantum circuit simplification in mind [13], and this utility preceded the completeness result. One ingredient of the axiomatisation of the original universal completeness result [25] is a new primitive of ZX-calculus, the triangle, which is one of the key components to make an algebraic axiomatisation of ZX-calculus [30]. Although it seems that the triangle is an extension of the original form of ZX-calculus, it actually can be expressed in terms of original generators such as green and red spiders [19, 25].

We will explore one utility of the triangle, namely the role of standard rules governing the AND-gate when translated as ZX-rules. In its simplest form the ZX-encoding of the AND-gate directly involves

Submitted to:
QPL 2020

the triangle, this fact was firstly found in [26], while its semantically equivalent form was already shown in [12] (see Exercise 12.10 there). We in particular derive three results based on a same rule set:

Derivation of spider nest identities. Recently ZX-calculus has been used to outperform all other methods in the area of circuit optimisation [23, 21, 15, 14]. Key to these for the purpose of T-gate reductions are so-called 'phase gadgets' named by Kissinger and van de Wetering [23], and were independently introduced by de Beaudrap and Wang, summarised in [15]. Decomposing larger $\pi/4$-phase gadgets into smaller ones (less than 4 lines) is vital for reducing T-count further than reduction effect of phase gadget fusion. In this paper, we derive a general decomposition theorem for arbitrary phase gadget in terms of AND-gates. As a consequence, we obtain all the spider nest identities which play a key role in [15, 14]. We note that the spider nest identities were derived in this way before the papers [15, 14] were formed, while this paper was in the process of preparing.

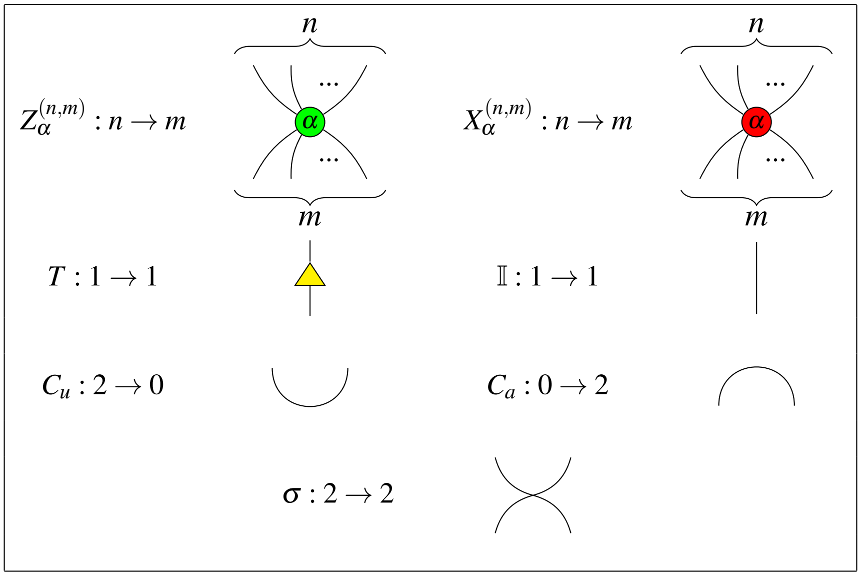
ZX-completeness for quantum Boolean circuits. Using the same rule set, we prove a completeness theorem for quantum Boolean circuits. Circuit relations by Iwama, Kambayashi and Yamashita that achieve this have been known for a while [18], also Cockett and Comfort have proved Iwama et al.'s rules in the symmetric monoidal category, TOF, generated by the Toffoli gate and computational ancillary bits [8]. We obtain our ZX-completeness result by proving Iwama et al.'s rules in the ZX-calculus with the rule set established in this paper, without any resort to having a Toffoli gate as the generator. This work does place results in circuit re-writing all under the umbrella of ZX-calculus. One particular advantage of this is the availability of automation tools which already has proved to be very useful for the above mentioned circuit optimisation results [22].

Algorithm for reducing T-count of quantum circuits. Based on the rules of completeness for quantum Boolean circuits, we give an algorithm for reducing the T-count (number of T-gates) of a family of benchmark circuits called Galois field multipliers [7]. We show in a table that the results of our algorithm outperform the results of all the previous best non-probabilistic algorithms. The technique in our algorithm can be applied to more benchmark circuits beyond Galois field multipliers.

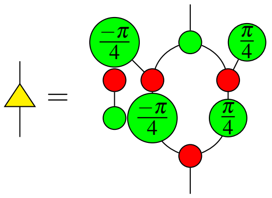
Other work. An early motivation for the GHZ/W-calculus [11], now completed by Hadzihasanovic [16, 17] and known as ZW-calculus, was extended control operations, which is also one of the motivational upshots of the triangle symbol. The triangle symbol entered the ZX-picture when ZW-calculus was translated to the ZX-context. After the ZX-encoding of the AND-gate with triangles was given in [26], there came a very recent ZX-alike calculus called ZH-calculus [5], which also allows easy representation of the AND-gate composed of fewer nodes than that of the ZX-encoding, however, it is not just about the denotation of the AND-gate, all the rules about the triangle in this paper are also important for the utility of the AND-gate. For now, ZX-calculus is (still) the umbrella under which quantum circuit optimisation is outperforming all competition, and were most completeness theorems have been stated, so a study of the AND-gate within this context is more than justified.

2 ZX-calculus generators

The ZX-calculus lives in a compact closed category whose objects are the natural numbers $\mathbb{N}$ and whose monoidal product is addition, $a \otimes b = a + b$. A general morphism $k \to l$ in this category is simply a k-input, l-output diagram generated, via finite sequential composition with outputs from one diagram connecting to inputs form another diagram and tensor composition with diagrams in parallel, by the following elementary diagrams:



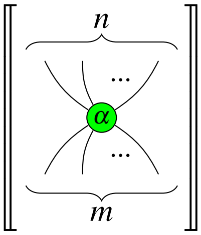
where $\alpha \in [0, 2\pi)$. Throughout this paper, all the diagrams are read from top to bottom, and non-zero scalars are ignored. Note that the generator triangle has a form merely composed of green and red notes as given in [12]:



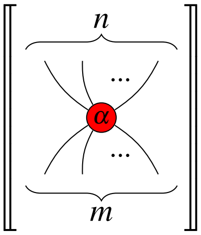
(1)

This means the triangle is not totally external to original ZX-calculus, but we won't use (1) as a rule in this paper.

Each generating diagram $D : k \to l$ above has a standard interpretation as a linear map between Hilbert spaces, $[\![D]\!] : (\mathbb{C}^2)^{\otimes k} \to (\mathbb{C}^2)^{\otimes l}$. By endowing each tensor factor $\mathbb{C}^2$ with its standard inner product and by identifying its elements $(1,0)^T$ and $(0,1)^T$ as the qubit Z-basis states $|0\rangle$ and $|1\rangle$, we can present each map $[\![D]\!]$ as a matrix:

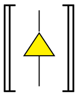


$= |0\rangle^{\otimes m}\langle 0|^{\otimes n} + e^{i\alpha}|1\rangle^{\otimes m}\langle 1|^{\otimes n}$;

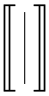


$= |+\rangle^{\otimes m}\langle +|^{\otimes n} + e^{i\alpha}|-\rangle^{\otimes m}\langle -|^{\otimes n}$,

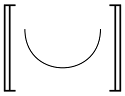
where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ are the qubit X-basis states:



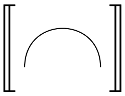
$= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$;



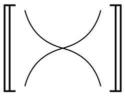
$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$;



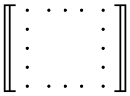
$= \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$;



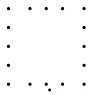
$= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$;



$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$;

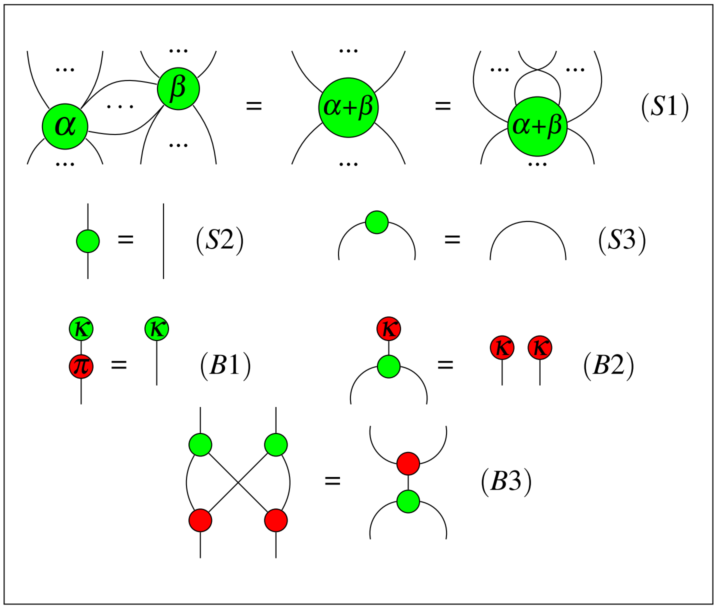


$= 1$,

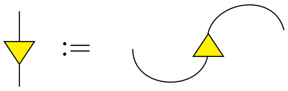
where  represents an empty diagram. We can extend these matrix interpretations to general ZX-diagrams simply by demanding that the composition operations for ZX-diagrams are compatible with those for matrices, namely that $[\![D_1 \circ D_2]\!] = [\![D_1]\!] \circ [\![D_2]\!]$ and $[\![D_1 \otimes D_2]\!] = [\![D_1]\!] \otimes [\![D_2]\!]$ for every suitable pair of ZX-diagrams D_1 and D_2.

3 ZX-calculus rules

Here is the fragment of stabilizer-style ZX-calculus rules that we will make use of in this paper, where $\alpha, \beta \in [0,\ 2\pi), \kappa \in \{0, \pi\}$. All of these rules also hold when the colours red and green swapped. We also assume a meta rule that “only topology/connectedness matters” [10].

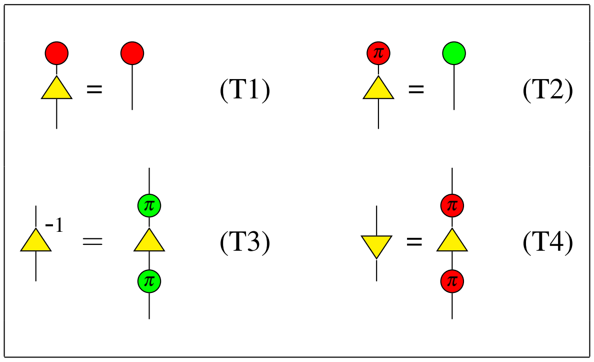


Noting that we haven’t mentioned adjoints, we represent the transpose as follows:

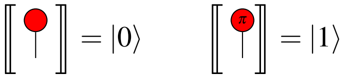


(2)

We will make use of the following simple triangle rules of the universal complete rules of [25]:



These rules represent the fact that the triangle breaks down of unitarity, or equivalently, the non-preservation of unitarity, more specifically, given that the red spider states $X_\alpha^{(0,1)}$ are the qubit Z-basis states:



(3)

we have:

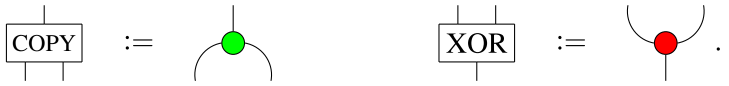
- Rules (T1) and (T2) show that the triangle turns orthonormal states into unbiased states.
- Rules (T3) and (T4) show that the inverse of the triangle is not equal to its adjoint.

This breakdown of unitarity greatly extends expressiveness, for example, for control operations where a basis can be used to switch between unbiased vectors.

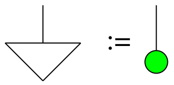
4 ZX rules on AND-gates

In this section, we demonstrate how to obtain ZX rules on AND-gates. To save space, we have placed all the Propositions in the appendix.

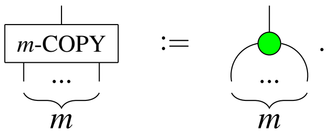
The interpretation (3) also enables us to view each red spider state as a diagrammatic representation of a classical bit value. As such, the standard ZX-rules (S1) and (B2) easily demonstrate how the classical processes COPY and XOR are represented as ZX-diagrams:



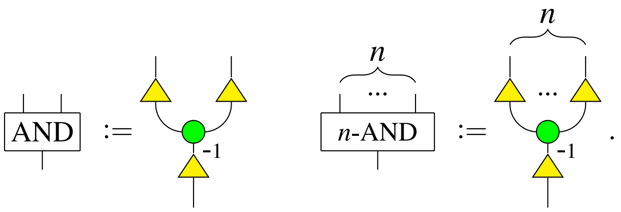
Also, the unit of the copy process can be seen as a classical deleting process:



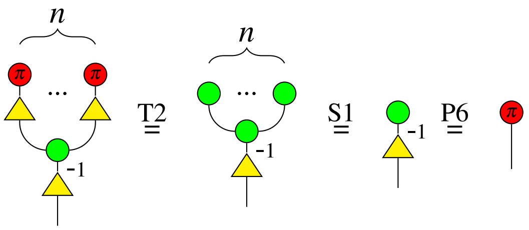
Proposition 1 shows how the green spider $X_0^{(1,m)}$ defines the m-output generalized COPY for any $m \geq 1$:



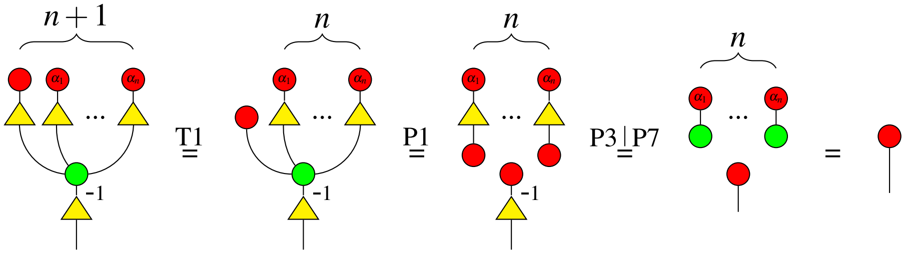
As illustrated in [29], we can represent the AND process and its generalization, the n-AND:



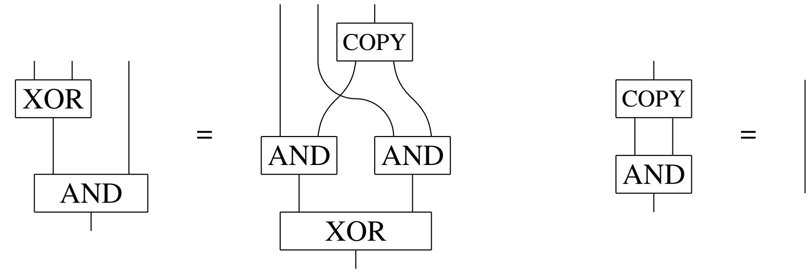
It is straightforward to see that these diagrams give a proper representation of the n-input generalized AND. For, by Proposition 6, the 0-AND process simply returns $|1\rangle$; and, for all $n \geq 0$, the $(n+1)$-AND process returns $|1\rangle$ whenever its input bit string consists solely of $|1\rangle$ states:



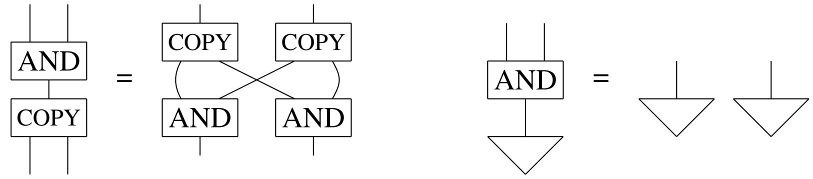
and returns $|0\rangle$ whenever its input bit string contains at least one $|0\rangle$ state:



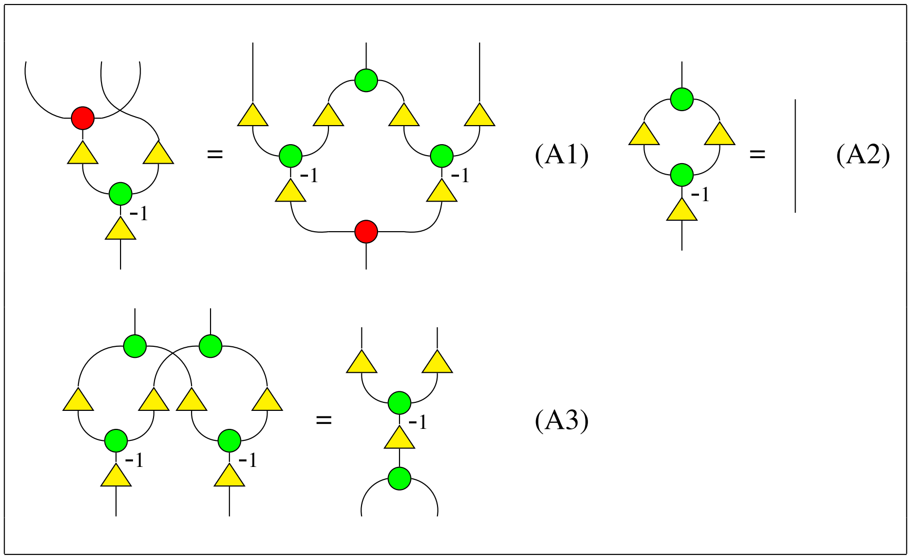
where $\alpha_i \in \{0,\pi\}, i = 1,\cdots,n$. By identifying the COPY, XOR, and AND in ZX-diagrams, we have demonstrated that the ZX-calculus has all the machinery necessary to represent Boolean algebra. In light of this fact, we expect that the Boolean identities $(p \oplus q)\cdot r = (p\cdot r)\oplus(q\cdot r)$ and $p\cdot p = p$ to hold:



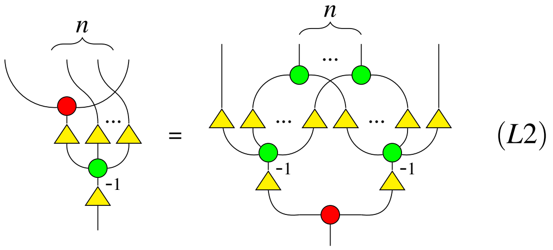
Furthermore, AND is actually a function map, so a homomorphism for COPY and its unit (see e.g., [12]):

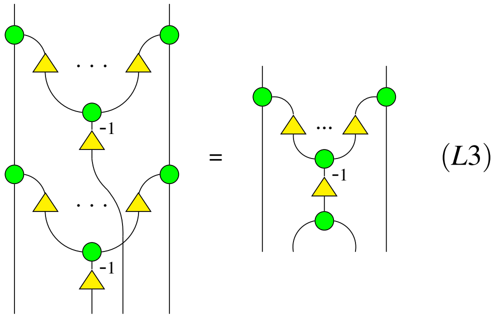


In terms of ZX-diagrams, these equations for AND translate as:



The ZX form of the second equality for the AND process as a function map is not listed as a rule, since it can be derived from other rules (see Proposition 8 in the Appendix). The rules (A1) and (A3) have the following useful generalisations whose proofs can be found in the Appendix:

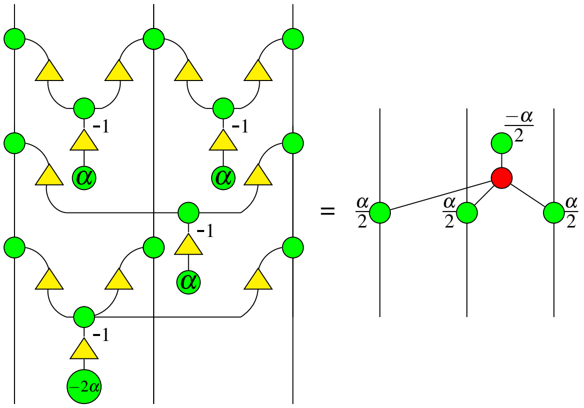




We note that (A1) and (A3) can be derived from another set of rules which give an algebraic axiomatisation of ZX-calculus [30].

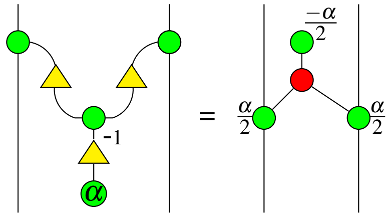
5 AND-gates and spider nest identities

In this section, using the AND-gate representation and rules presented in the previous section, we derive so-called spider nest identities which have been shown to be of vital importance in reducing T-count with ZX-calculus [15, 14]. For this purpose, in addition to the rules listed in the figures of previous section, we assume the following equality with AND-gates:



(4)

where $\alpha \in [0, 2\pi)$. It can be checked that the two diagrams on both sides of (4) have the same standard interpretation up to some scalar. If we plug ● and ● respectively onto the bottom and top of the leftmost lines of both sides of (4), then we get:

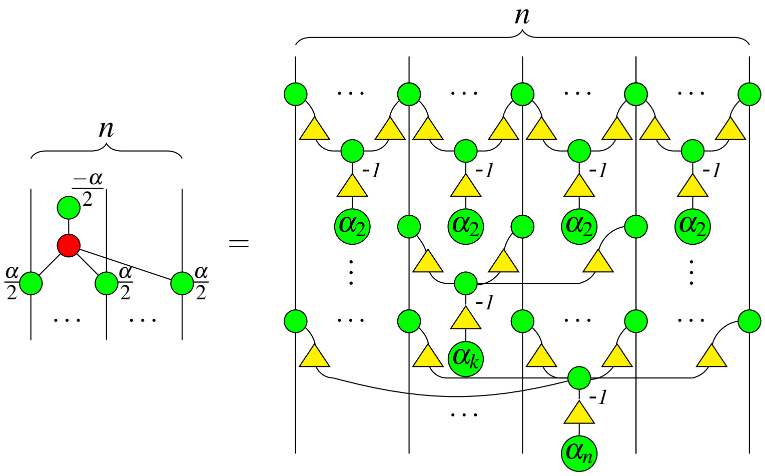


(5)

which represents the controlled phase shift gate ΛZ_α [10] in two different ways.

The following theorem generalises (5) from 2 qubits to n qubits in terms of AND-gates, as is proved in the Appendix.

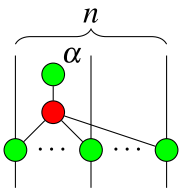
Theorem 5.1. *For any* $\alpha \in [0, 2\pi)$, $n \geq 2$, *we have:*



(6)

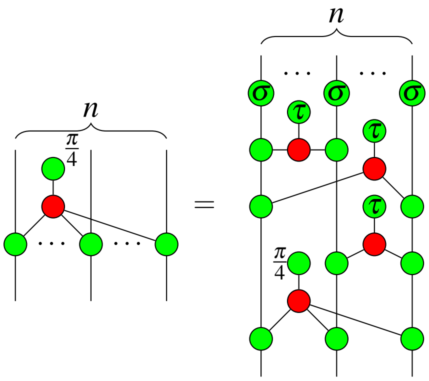
where for each $k \in \{2, \cdots, n\}$ *lines of the RHS of (6), there locates one and only one k-AND gate plugged with a phase with angle* $\alpha_k = (-1)^k 2^{k-2}\alpha$. *Clearly* $\alpha_{k+1} = -2\alpha_k$.

Using the the terminology of [23], we denote the following diagram as a phase n-gadget:



where $n \geq 1$.

Corollary 5.2.



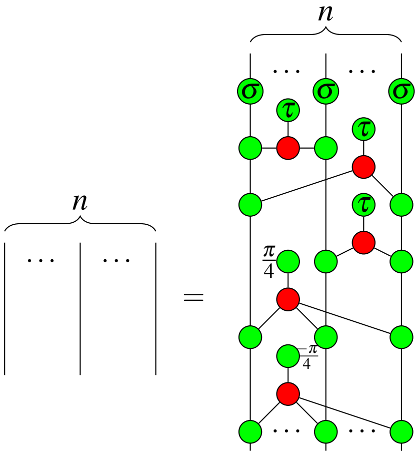
(7)

where $n \geq 3$, for the RHS of (7), on each line, there is a 1*-gadget with phase angle $\sigma = \frac{(n-2)(n-3)\pi}{8}$, for every two of the n lines, there is a* 2*-gadget with phase angle $\tau = \frac{(3-n)\pi}{4}$, and for every three of the n lines, there is a* 3*-gadget with phase angle $\frac{\pi}{4}$.*

Proof. Since the angle of the phase gadget is $\frac{\pi}{4}$, we let $\alpha = -\frac{\pi}{2}$ in (6). Then on the RHS of (6), $\alpha_k = 0, \forall k \geq 4$. Thus all the k-AND gates plugged with phase α_k and $k \geq 4$ become identities, which means we just have phase-plugged 2-AND gates and 3-AND gates left in the RHS of (6). Now if we replace all phase-plugged 2-AND gates and 3-AND gates by the phase gadgets in (5) and (4) respectively, after phase gadget fusion, we then have the equality (7). □

We note that the above proof can be generalised to more general cases if we consider the phase angle on the left of (7) to be $\frac{\pi}{2^k}$ for any $k \geq 0$.

Now if we plug the inverse of the phase gadget on the LHS of (7) to its both sides, then we get the spider nest identities [15, 14] as follows:



6 CNOT-based Quantum Boolean Circuits

In [18], a CNOT gate is defined as an n-bit Toffoli gate [27] denoted by $[t,C]$, where $|x_t\rangle$ is a target bit and $|x_k\rangle$ is a control bit with $k \in C \subseteq \{1,\cdots,n\}$. A quantum Boolean circuit is defined as follows:

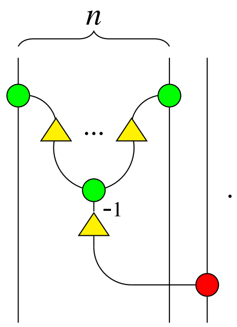
- A *quantum Boolean circuit* of size M over qubits $|x_1\rangle,...,|x_N\rangle$ is a sequence of CNOT gates $[t_1,C_1]\cdots[t_i,C_i]\cdots[t_M,C_M]$ where $1 \leq t_i \leq N$ and $C_i \subseteq \{1,...,N\}$.

A quantum Boolean circuit is said to be *proper* and to compute a Boolean function $f(x_1,\cdots,x_n)$ iff (i) the input state of the circuit $S_0 = |a_1\rangle|a_2\rangle\cdots|a_{n+1}\rangle|0\rangle\cdots|0\rangle$, (ii) the final state of the circuit $S_M = |a_1\rangle|a_2\rangle\cdots|a_{n+1} \oplus f(x_1,\cdots,x_n)\rangle|0\rangle\cdots|0\rangle$. There are six transformation rules introduced in [18] for quantum Boolean circuits:
(1) $[t_1,C_1]\cdot[t_1,C_1] \iff \varepsilon$,
(2) $[t_1,C_1]\cdot[t_2,C_2] \iff [t_2,C_2]\cdot[t_1,C_1]$, if $t_1 \notin C_2$ and $t_2 \notin C_1$,
(3) $[t_1,C_1]\cdot[t_2,C_2] \iff [t_2,C_2]\cdot[t_1,C_1]\cdot[t_1,C_1\cup C_2-\{t_2\}]$ if $t_1 \notin C_2$ and $t_2 \in C_1$,
(4) $[t_1,C_1]\cdot[t_2,C_2] \iff [t_2,C_1\cup C_2-\{t_1\}]\cdot[t_2,C_2]\cdot[t_1,C_1]$ if $t_1 \in C_2$ and $t_2 \notin C_1$,
(5) $[t_1,\{c_1\}]\cdot[t_2,C_2\cup\{c_1\}] \iff [t_1,\{c_1\}]\cdot[t_2,C_2\cup\{t_1\}]$ if $t_1 > n+1$ and no CNOT$_{t_1}$ before $[t_1,\{c_1\}]$,
(6) $[t,C] \iff \varepsilon$ if there is an integer i such that $i \in C$, $i > n+1$, and there is no CNOT$_i$ before $[t,C]$,
where ε represent the 'identity' gate and $\iff$ denote a transformation.

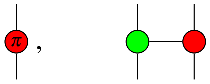
As a main result of [18], this set of transformation rules is proved to be complete for proper quantum Boolean circuits: any two proper quantum Boolean circuits which are equivalent (computing the same Boolean function) can be derived from each other via these transformation rules. This result is claimed by the authors to be "quite non-trivial and theoretically interesting on its own."

In this section, we represent each quantum Boolean circuit by ZX diagrams and then obtain the ZX-completeness result for quantum Boolean circuits by proving in ZX-calculus the complete set of six transformation rules presented in [18]. Moreover, we give an algorithm for T-count reduction based on these transformation rules and show its outperformance in contrast with results of state-of-the-art algorithms.

Now we give diagrammatic representation for CNOT gates. The ZX-diagram for a CNOT gate is simply (see e.g., [29]):



As a check to this representation, the reader can verify that in the $n = 0$ and $n = 1$ cases, this representation reduces to the NOT gate and the standard CNOT gate, as expected:



We have now developed the necessary machinery to assert and prove the completeness of our ZX-rules for quantum Boolean circuits. Let $ZX \vdash$ denote what can be inferred from the ZX rules listed in this paper: (S1), (S2), (S3), (B1), (B2), (B3), (T1), (T2), (T3), (T4), (A1), (A2), (A3).

Theorem 6.1. *If D_1 and D_2 are any two CNOT-based circuits expressed as ZX-diagrams such that $[\![D_1]\!] = [\![D_2]\!]$, then $ZX \vdash D_1 = D_2$.*

The proof is given in the appendix.

Remark 6.2. *Theorem 6.1 does not directly follows from the universal completeness of ZX-calculus, because the rules used for the proof of universal completeness are beyond the rules we presented in this paper.*

Below we give an algorithm for reducing T-count (number of T-gates, denoted as #T) of quantum circuits based on the transformation rules. The transformation rules (3) and (4) are all about how to commute two $CNOT_i$ gates when exactly one target meets the other's control bit. Especially, when one of the two gates is the standard CNOT gate and the other an n-bit Toffoli gate, then there will be a new n-bit Toffoli gate produced after their commuting. Based on this fact and the algorithms in [15] and [14], we have the following algorithm for reducing the T-count of a family of benchmark circuits—Galois field multipliers which are composed of standard CNOT gates in the middle and standard Toffoli gates on both left and right ends [7]:

1. Input a circuit of Galois field multiplier.
2. Turn the circuit into a ZX-diagram.
3. Start from the left most standard CNOT gate, commute it through to the left or right end of the circuit (outside of all standard Toffoli gates) with standard Toffoli gates using the transformation rule (3) or (4). The commuting direction (to left or right) depends on the number of Toffoli gates it needs to commute with, the direction with fewer commutations will be chosen; if both directions have the same commuting number, then we choose the left direction.
4. Repeat the above step until all the CNOT gates are moved out.
5. Turn all the Toffoli gates into phase gadgets and do the phase gadget fusion, using the method from [15, 14].
6. Output the circuit form and give the T-count.

The following table presents a comparison of the results of our algorithm with the previous best (non-probabilistic) algorithms for reducing T-count.

Table 1: T-count performance comparison.

Circuit	**Initial** $\#T$	$\#T$ **in** [24]	$\#T$ **in** [23]	$\#T$ **in** [31]	**Ours**
GF(2^4)-mult	112	68	68	68	62
GF(2^5)-mult	175	115	115	115	97
GF(2^6)-mult	252	150	150	150	131
GF(2^7)-mult	343	217	217	217	183
GF(2^8)-mult	448	264	264	264	263
GF(2^9)-mult	567	351	-	351	299
GF(2^{10})-mult	700	410	-	410	361
GF(2^{16})-mult	1792	1040	-	1040	1038

The circuits we obtained by running our algorithms can be found at [6], they have been verified by *feynver* [2].

As one can see from Table 1, our results outperform all the results of previous non-probabilistic algorithms. Note that the spider nest identities can be separately applied with after running the above described algorithm, yet we didn't apply these identities in our algorithm just because we want to see its own performance. We would expect better results if we use those identities. Probabilistic method in addition to our algorithm is in the similar situation as the spider nest identities. We also point out that the technique used in our algorithm can be applied to more benchmark circuits beyond the family of circuits of Galois field multipliers.

Acknowledgements

The authors thank Xiaoning Bian for his kind help on running numerical experiments on the algorithm proposed in this paper. AM thanks the Mellon Mays Undergraduate Fellowship Program for its generous support of his research. BC and QW are grateful for an EPSRC IAA in collaboration with Cambridge Quantum Computing Ltd. QW is supported by AFOSR grant FA2386-18-1-4028.

References

[1] S. Abramsky and B. Coecke. A categorical semantics of quantum protocols. In Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS), pages 415–425, 2004. arXiv:quant-ph/0402130.

[2] Matthew Amy. Towards large-scale functional verification of universal quantum circuits. In Proceedings of QPL 2018, pages 1–21, 2018. [arXiv:1901.09476]; see also https://github.com/meamy/feynman.

[3] M. Backens. The ZX-calculus is complete for stabilizer quantum mechanics. New Journal of Physics, 16:093021, 2014. arXiv:1307.7025.

[4] Miriam Backens. The zx-calculus is complete for the single-qubit clifford+t group. In Bob Coecke, Ichiro Hasuo, and Prakash Panangaden, editors, Proceedings of the 11th workshop on Quantum Physics and Logic, volume 172 of Electronic Proceedings in Theoretical Computer Science, pages 293–303. Open Publishing Association, 2014.

[5] Miriam Backens and Aleks Kissinger. Zh: A complete graphical calculus for quantum computations involving classical non-linearity. In Peter Selinger and Giulio Chiribella, editors, Proceedings of the 15th International Conference on Quantum Physics and Logic,

Halifax, Canada, 3-7th June 2018, volume 287 of Electronic Proceedings in Theoretical Computer Science, pages 23–42. Open Publishing Association, 2019.

[6] Xiaoning Bian. https://github.com/onestruggler/cx-ccx. GitHub.

[7] Donny Cheung, Dmitri Maslov, Jimson Mathew, and Dhiraj K. Pradhan. On the design and optimization of a quantum polynomial-time attack on elliptic curve cryptography. In Yasuhito Kawano and Michele Mosca, editors, Theory of Quantum Computation, Communication, and Cryptography, pages 96–104, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[8] J.R.B. Cockett and Cole Comfort. The category tof. In Peter Selinger and Giulio Chiribella, editors, Proceedings of the 15th International Conference on Quantum Physics and Logic, Halifax, Canada, 3-7th June 2018, volume 287 of Electronic Proceedings in Theoretical Computer Science, pages 67–84. Open Publishing Association, 2019.

[9] B. Coecke and R. Duncan. Interacting quantum observables. In Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP), Lecture Notes in Computer Science, 2008.

[10] B. Coecke and R. Duncan. Interacting quantum observables: categorical algebra and diagrammatics. New Journal of Physics, 13:043016, 2011. arXiv:quant-ph/09064725.

[11] B. Coecke and A. Kissinger. The compositional structure of multipartite quantum entanglement. In Automata, Languages and Programming, Lecture Notes in Computer Science, pages 297–308. Springer, 2010. arXiv:1002.2540.

[12] B. Coecke and A. Kissinger. Picturing Quantum Processes. A First Course in Quantum Theory and Diagrammatic Reasoning. Cambridge University Press, 2017.

[13] B. Coecke and Q. Wang. Zx-rules for 2-qubit Clifford+T quantum circuits. In Jarkko Kari and Irek Ulidowski, editors, Reversible Computation - 10th International Conference, RC 2018, Leicester, UK, September 12-14, 2018, Proceedings, volume 11106 of Lecture Notes in Computer Science, pages 144–161. Springer, 2018.

[14] Niel de Beaudrap, Xiaoning Bian, and Quanlong Wang. Fast and effective techniques for t-count reduction via spider nest identities. to appear in Proceedings 15th Conference on the Theory of Quantum Computation, Communication and Cryptography, 2020. arXiv:2004.05164.

[15] Niel de Beaudrap, Xiaoning Bian, and Quanlong Wang. Techniques to reduce pi/4-parity-phase circuits, motivated by the zx calculus. Proceedings of the16th International Conference on Quantum Physics and Logic 2019, EPTCS, 318:131–149, 2020. arXiv:1911.09039.

[16] A. Hadzihasanovic. A diagrammatic axiomatisation for qubit entanglement. In Proceedings of the 30th Annual IEEE Symposium on Logic in Computer Science (LICS), 2015. arXiv:1501.07082.

[17] A. Hadzihasanovic. The algebra of entanglement and the geometry of composition. PhD thesis, University of Oxford, 2017.

[18] Kazuo Iwama, Yahiko Kambayashi, and Shigeru Yamashita. Transformation rules for designing cnot-based quantum circuits. In Proceedings of the 39th annual Design Automation Conference, pages 419–424. ACM, 2002.

[19] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. A complete axiomatisation of the ZX-calculus for Clifford+T quantum mechanics. In Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '18, pages 559–568, New York, NY, USA, 2018. ACM.

[20] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Diagrammatic reasoning beyond Clifford+T quantum mechanics. In Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '18, pages 569–578, New York, NY, USA, 2018. ACM.

[21] A. Kissinger and A. Meijer-van de Griend. Cnot circuit extraction for topologically-constrained quantum memories. arXiv preprint arXiv:1904.00633, 2019.

[22] A. Kissinger and J. Wetering. Pyzx: Large scale automated diagrammatic reasoning. arXiv preprint arXiv:1904.04735, 2019.

[23] A. Kissinger and J. Wetering. Reducing t-count with the zx-calculus. arXiv preprint arXiv:1903.10477, 2019.

[24] Yunseong Nam, Neil J. Ross, Yuan Su, Andrew M. Childs, and Dmitri Maslov. Automated optimization of large quantum circuits with continuous parameters. npj Quantum Information, 4(1):23, 2018. [arXiv:1710.07345].

[25] Kang Feng Ng and Quanlong Wang. A universal completion of the zx-calculus. arXiv preprint arXiv:1706.09877, 2017.

[26] Kang Feng Ng and Quanlong Wang. Completeness of the zx-calculus for pure qubit clifford+t quantum mechanics. arXiv:1801.07993, 2018.

[27] Tommaso Toffoli. Reversible computing. Technical Report MIT/LCS/TM-151,MIT LCS, Feb., 1980.

[28] Renaud Vilmart. A near-optimal axiomatisation of ZX-calculus for pure qubit quantum mechanics. In Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 2019.

[29] Quanlong Wang. Completeness of the ZX-calculus. PhD thesis, University of Oxford, 2018.

[30] Quanlong Wang. An algebraic axiomatisation of zx-calculus. accepted to QPL 2020, 2020. arXiv:1911.06752.

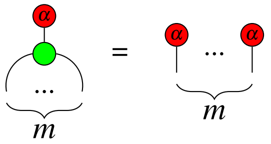
[31] Fang Zhang and Jianxin Chen. Optimizing t gates in clifford+t circuit as $\pi/4$ rotations around paulis. [arXiv:1903.12456], 2019.

Appendix: Propositions, Lemmas and Proofs

In this appendix we give the Propositions, Lemmas and Proofs. Note that in the proofs we just indicate the main rules which are used in derivations, for the sake of simplicity.

Proposition 1

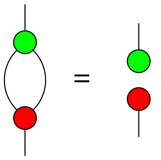
For $m \geq 1$,



(P1)

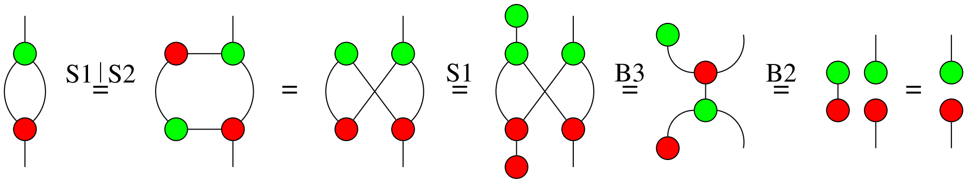
Proof. Using S1 we can decompose an $R_Z^{(1,m)}$ diagram into an $(m-1)$-fold composition of $R_Z^{(1,2)}$ diagrams. Repeatedly applying B2 then yields the desired result.

Proposition 2

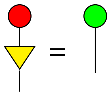


(P2)

Proof.

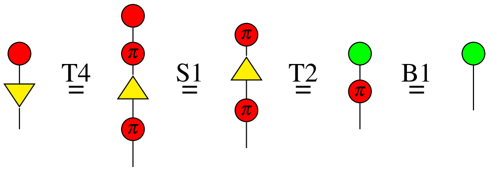


Proposition 3

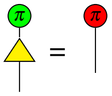


(P3)

Proof.

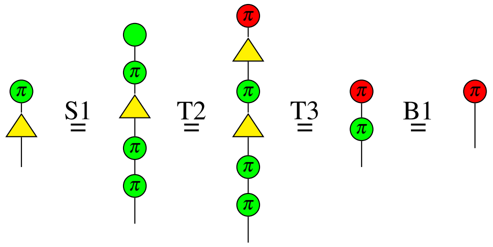


Proposition 4

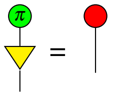


(P4)

Proof.

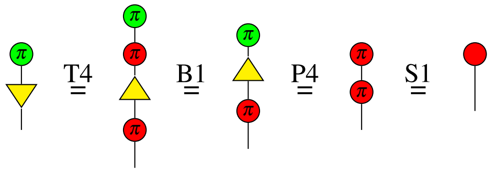


Proposition 5

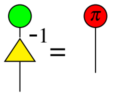


(P5)

Proof.

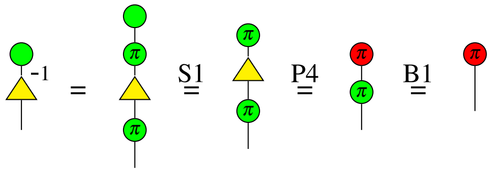


Proposition 6

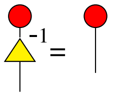


(P6)

Proof.

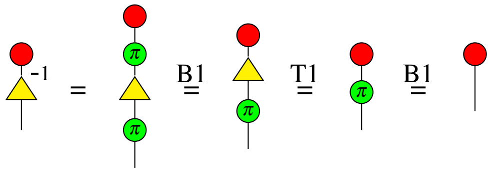


Proposition 7

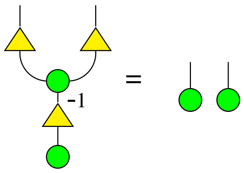


(P7)

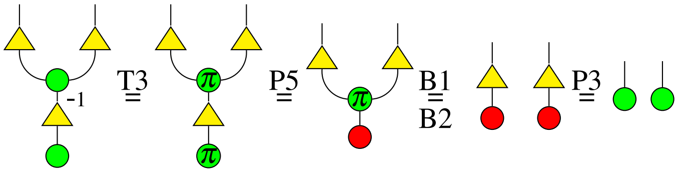
Proof.



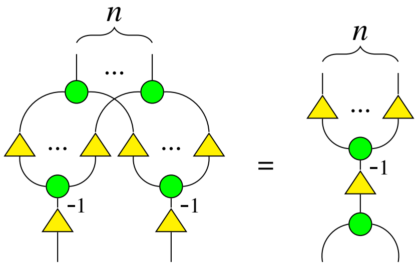
Proposition 8



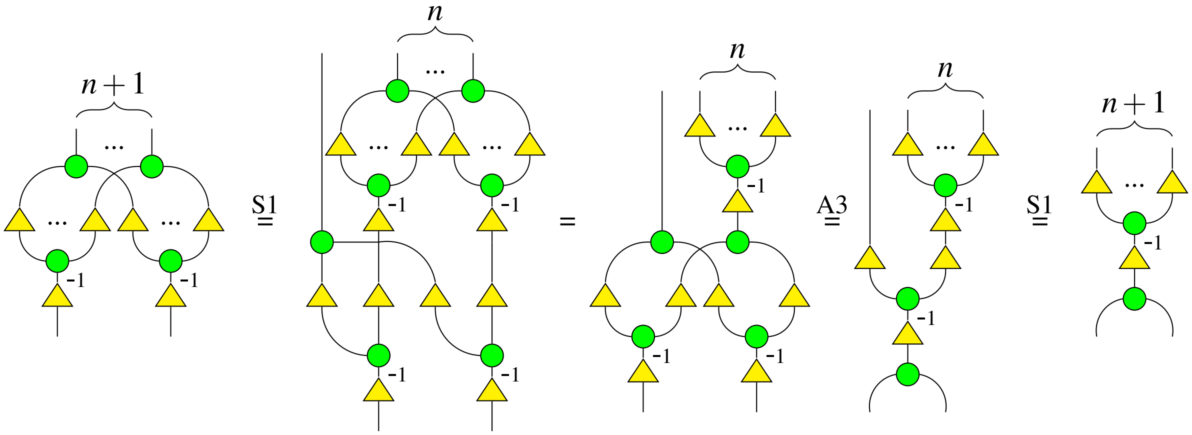
(P8)

Proof. 

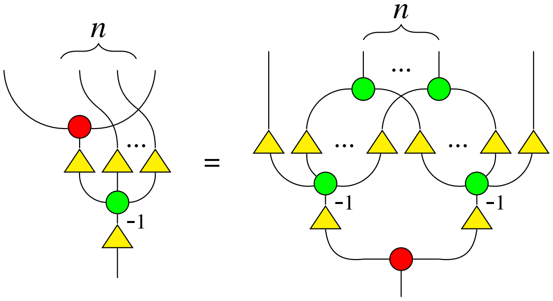
Lemma 1

For all $n \geq 0$,  (L1)

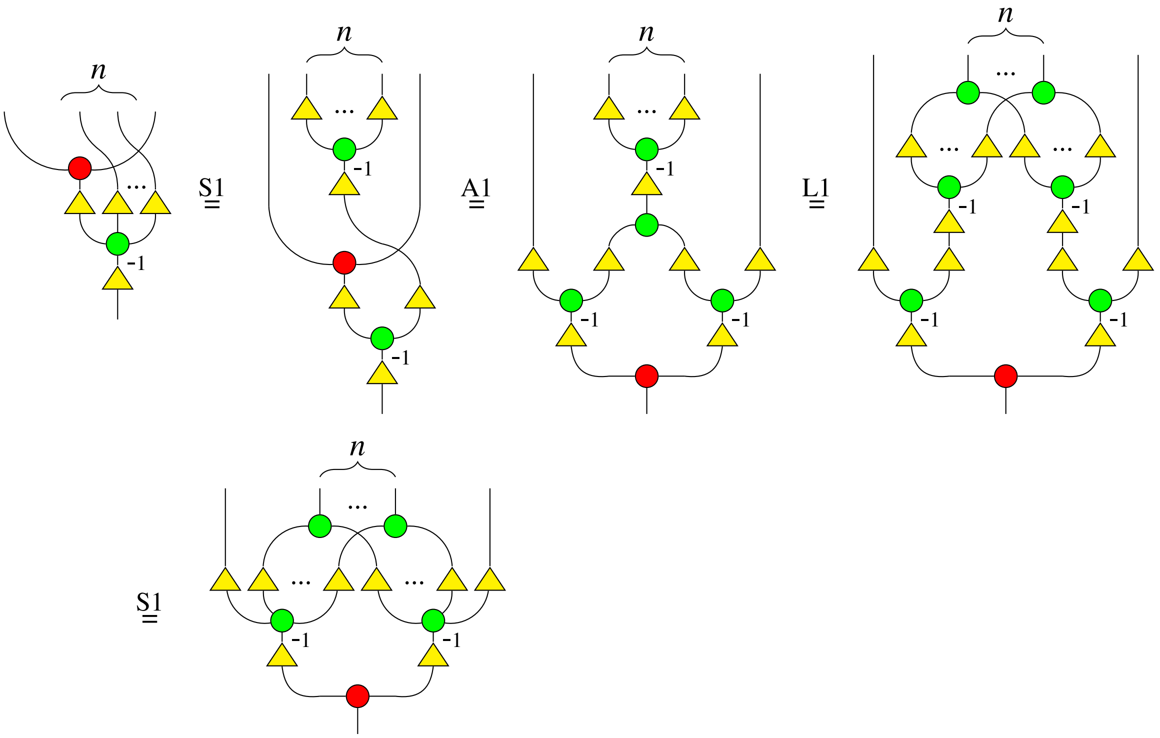
Proof. The $n = 0$ case follows from B2 and P6, the $n = 1$ case follows from (S2) and (T3), and the $n = 2$ case is simply A1. The lemma is true in all other cases, since whenever it holds for some $n \geq 2$, it is also valid for $n+1$:



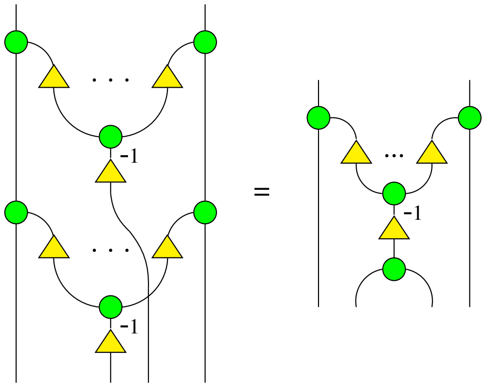
Lemma 2

For all $n \geq 0$,  (L2)

Proof. The $n = 0$ case follows from S2 and T3. We obtain all other cases by applying Lemma 1:

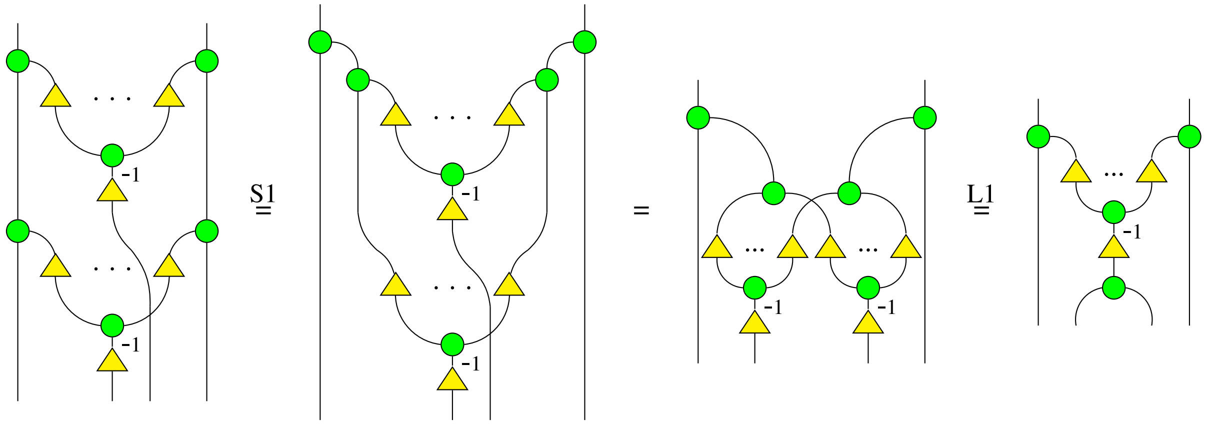


Lemma 3

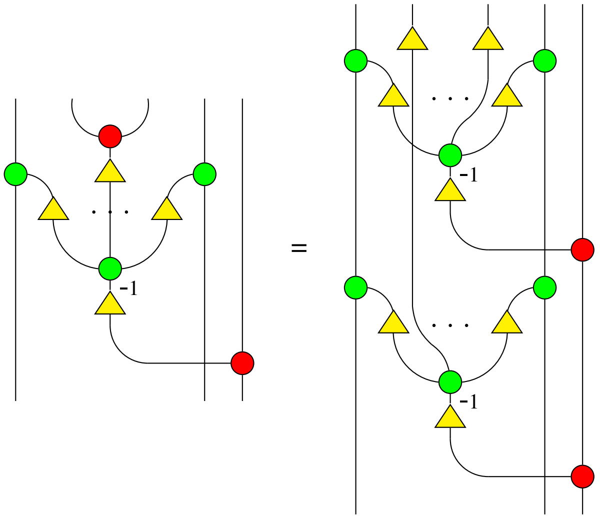


(L3)

Proof.

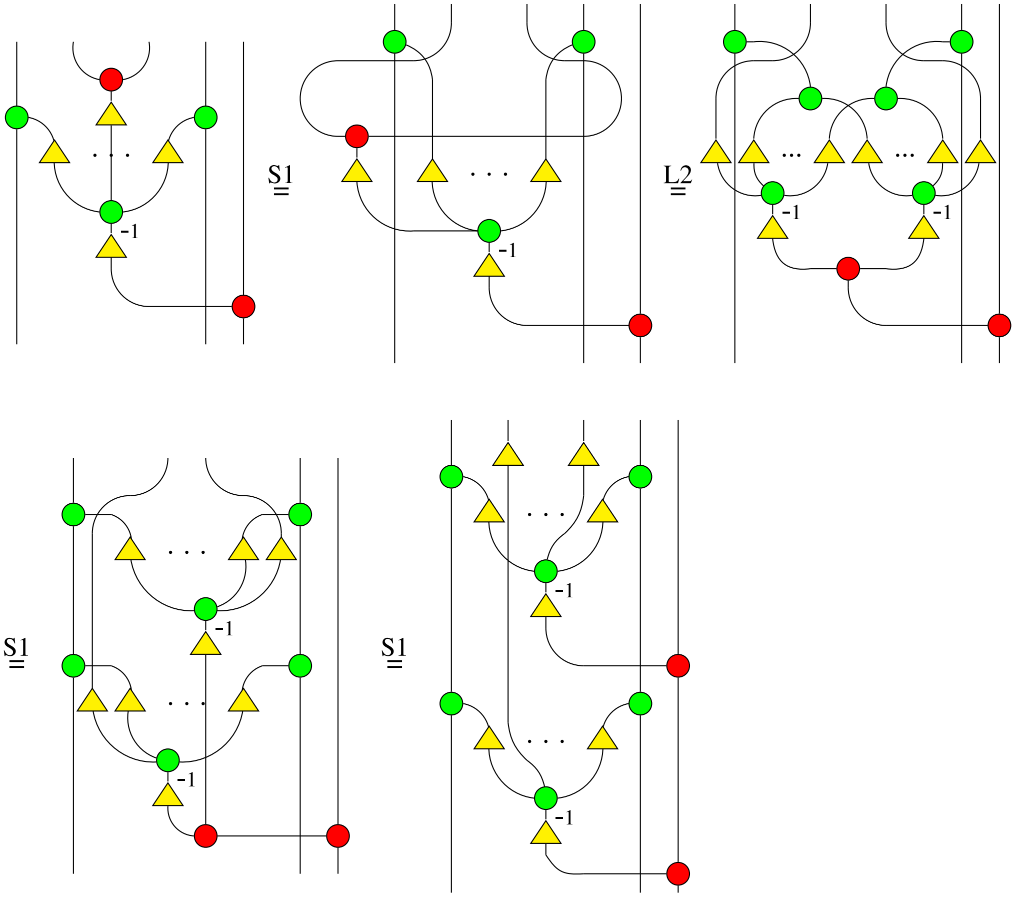


Lemma 4



(L4)

Proof.

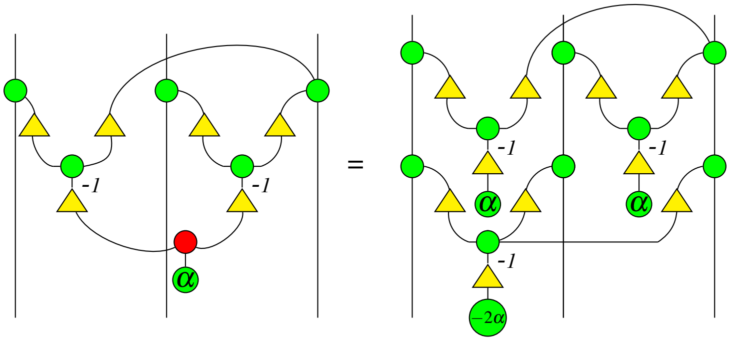


□

Proof of Theorem 5.1

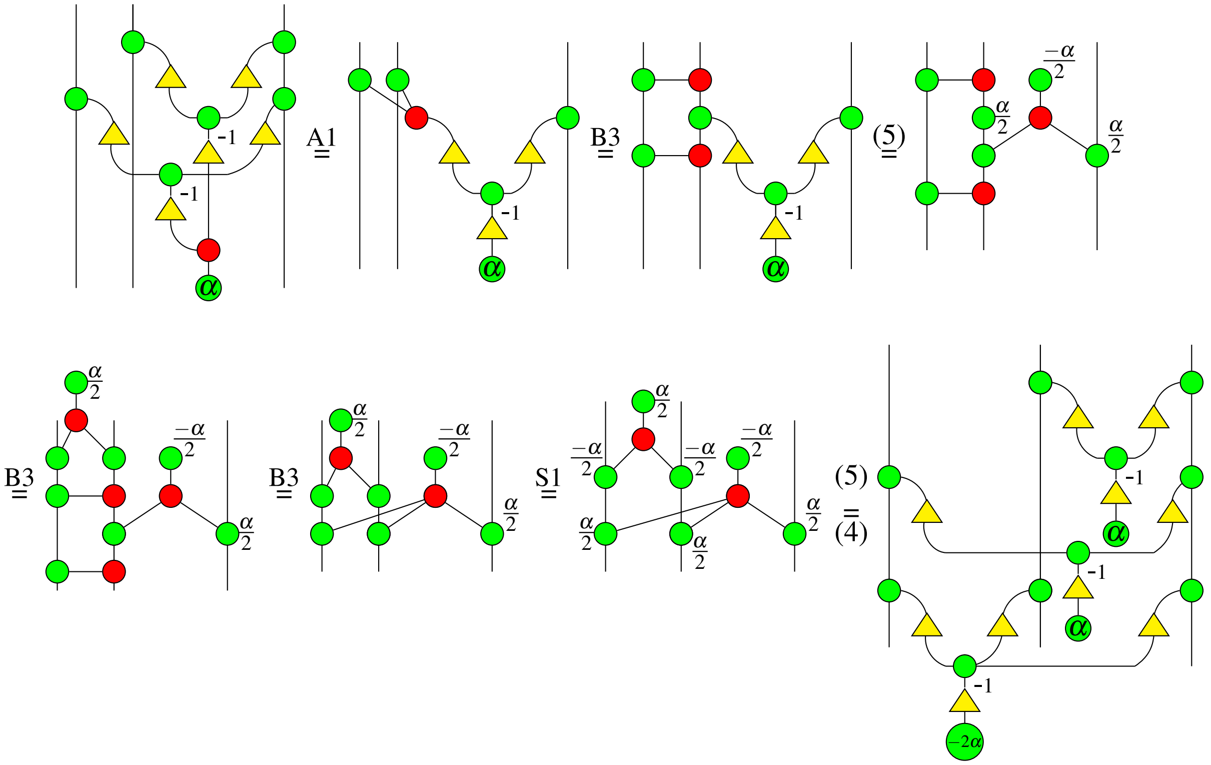
First, we need the following lemmas.

Lemma 6.3. *For any $\alpha \in [0, 2\pi)$, we have:*



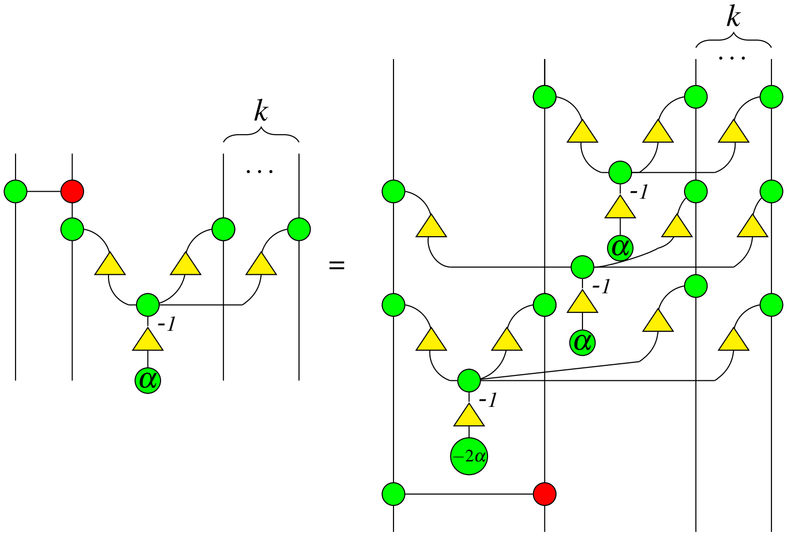
(8)

Proof.



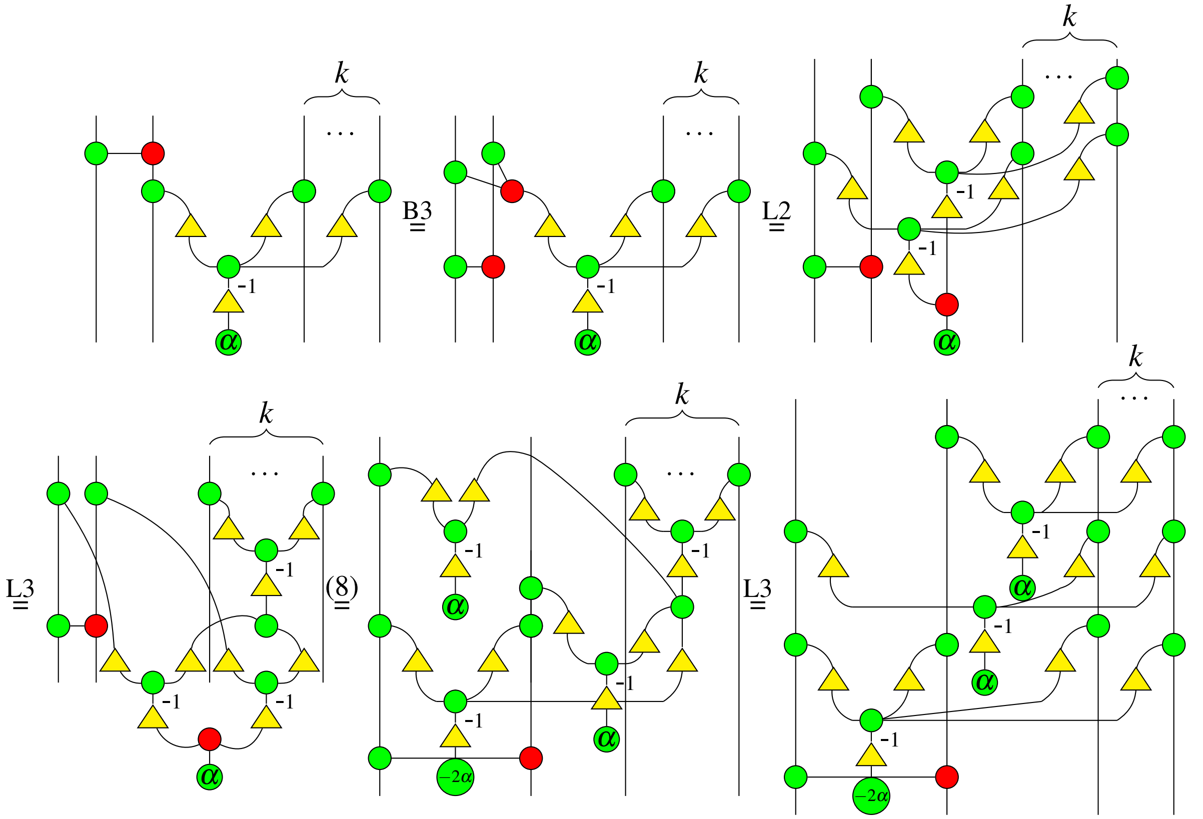
□

Lemma 6.4. *For any* $\alpha \in [0, 2\pi)$, $k \geq 1$, *we have*



(9)

Proof.

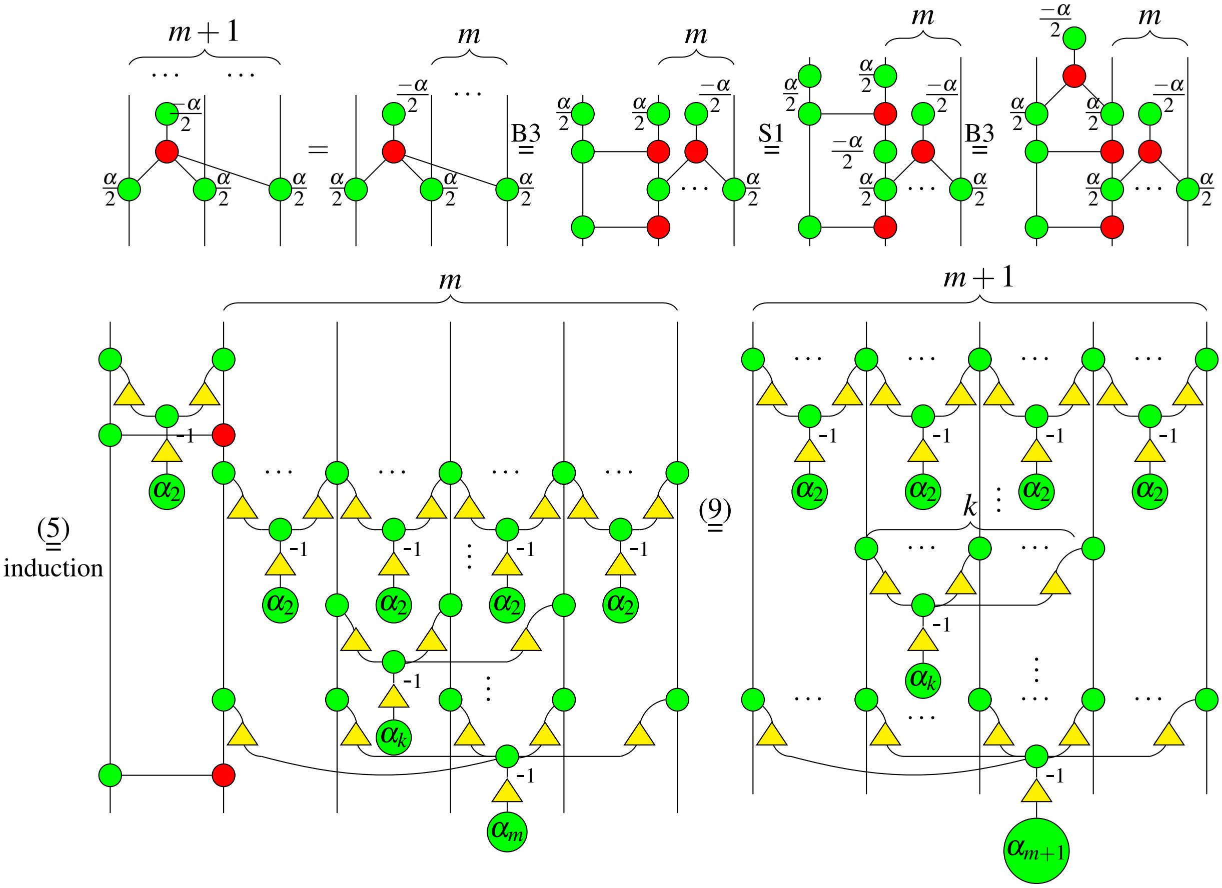


□

Now we are ready to prove Theorem 5.1.

Proof. We prove by induction on n. If $n = 2$, it is just the equality (5) which can be derived from (4).

Assume that (6) holds for $n = m$. Then for $n = m+1$, we have

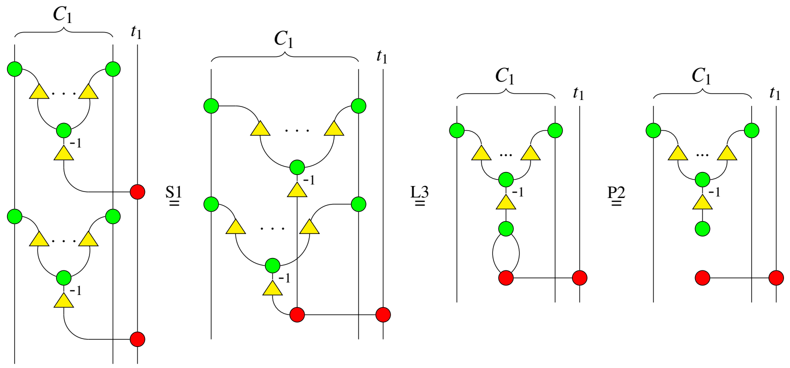


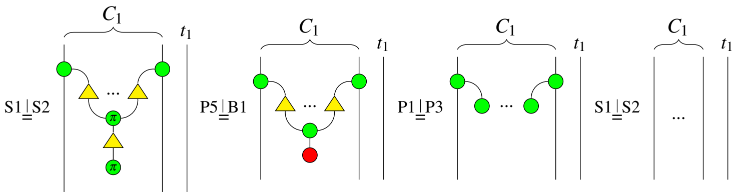
Therefore, (6) holds for $n = m+1$. This completes the proof. □

Proof of Theorem 6.1

Proof. It suffices to prove that each of the six transformation rules, expressed as ZX-diagrams, are derivable using only the rule set given in the tables above.

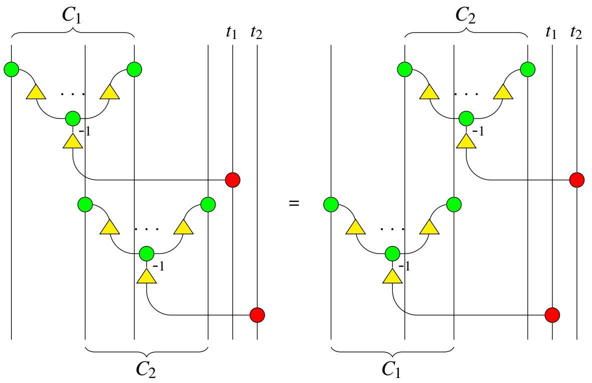
(1) $[t_1, C_1] \cdot [t_1, C_1] \iff \varepsilon$



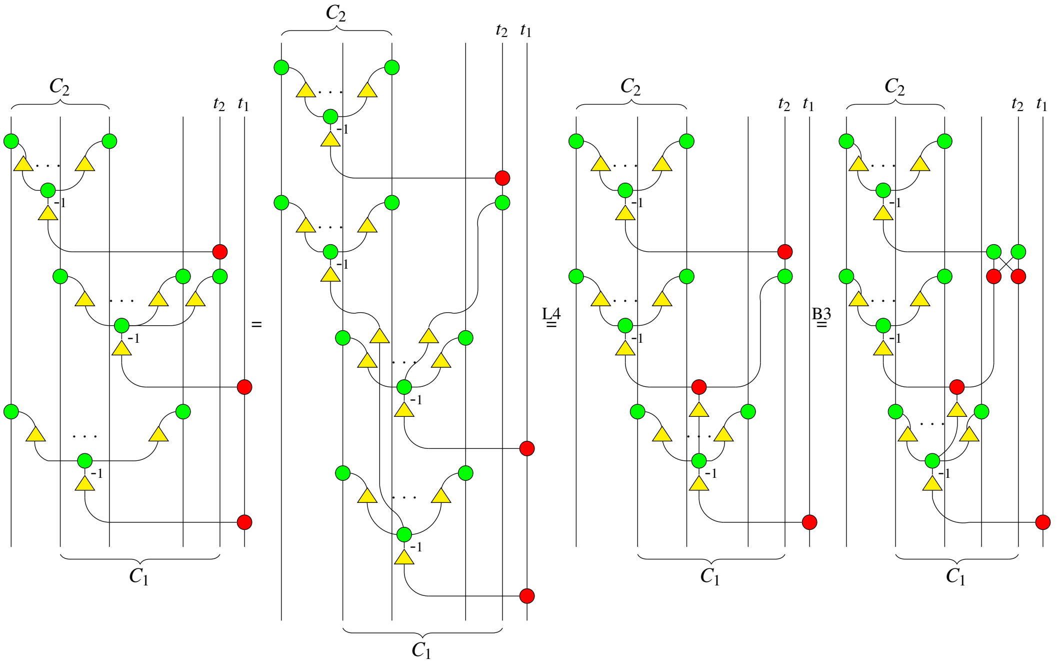


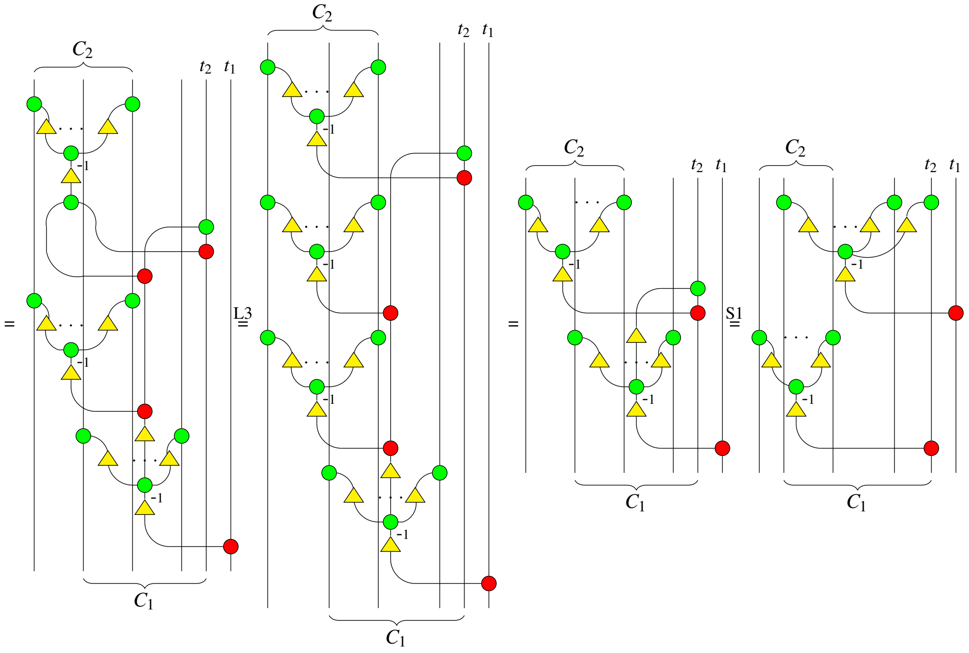
(2) $[t_1,C_1]\cdot[t_2,C_2] \iff [t_2,C_2]\cdot[t_1,C_1]$, if $t_1 \notin C_2$ and $t_2 \notin C_1$

This is a simple consequence of (S1). The case in which $t_1 \neq t_2$ is shown below.



(3) $[t_1,C_1]\cdot[t_2,C_2] \iff [t_2,C_2]\cdot[t_1,C_1]\cdot[t_1,C_1\cup C_2-\{t_2\}]$ if $t_1 \notin C_2$ and $t_2 \in C_1$

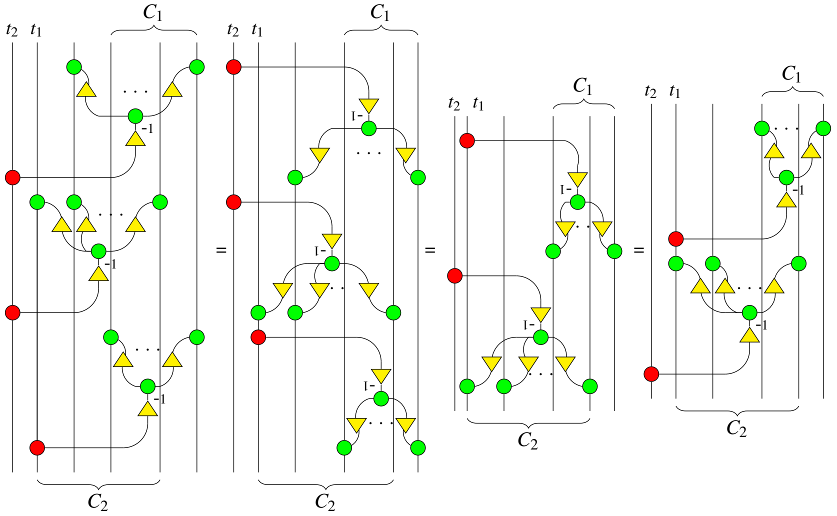




The second-to-last equality comes from rule 1.

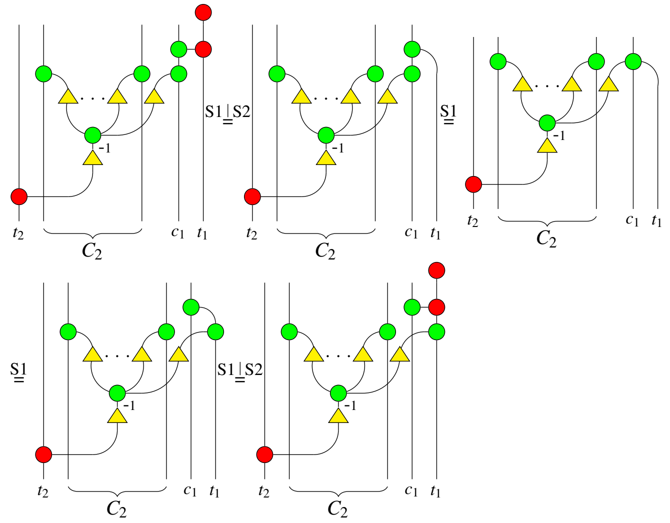
(4) $[t_1, C_1] \cdot [t_2, C_2] \iff [t_2, C_1 \cup C_2 - \{t_1\}] \cdot [t_2, C_2] \cdot [t_1, C_1]$ if $t_1 \in C_2$ and $t_2 \notin C_1$

This follows immediately from rule 3:

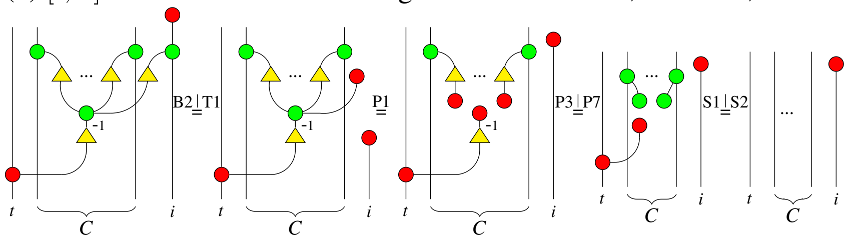


(5) $[t_1, \{c_1\}] \cdot [t_2, C_2 \cup \{c_1\}] \iff [t_1, \{c_1\}] \cdot [t_2, C_2 \cup \{t_1\}]$ if $t_1 > n+1$ and no CNOT$_{t_1}$ before $[t_1, \{c_1\}]$

Note that we must assume $t_1 \neq t_2$ for the generalized CNOT gate $[t_2, C_2 \cup \{t_1\}]$ to be well-defined.



(6) $[t,C] \iff \varepsilon$ if there is an integer i such that $i \in C$, $i > n+1$, and there is no CNOT$_i$ before $[t,C]$



□