

Catalysing Completeness and Universality

Aleks Kissinger

University of Oxford

aleks.kissinger@ox.ac.uk

Neil J. Ross

Dalhousie University

neil.jr.ross@dal.ca

John van de Wetering

University of Amsterdam

john@vdwetering.name

A catalysis state is a quantum state that is used to make some desired operation possible or more efficient, while not being consumed in the process. Recent years have seen catalysis used in state-of-the-art protocols for implementing magic state distillation or small angle phase rotations. In this paper we will see that we can also use catalysis to prove that certain gate sets are computationally universal, and to extend completeness results of graphical languages to larger fragments. In particular, we give a simple proof of the computational universality of the CS+Hadamard gate set using the catalysis of a T gate using a CS gate, which sidesteps the more complicated analytic arguments of the original proof by Kitaev. This then also gives us a simple self-contained proof of the computational universality of Toffoli+Hadamard. Additionally, we show that the phase-free ZH-calculus can be extended to a larger complete fragment, just by using a single catalysis rule (and one scalar rule).

In chemistry, a *catalyst* is a substance that facilitates a reaction without being modified by it. In analogy, a quantum state is said to be a *catalyst* or a *catalysis state*, if it can be used to make a desired operation more efficient, or even possible, while not being consumed in the process.

The idea of catalysis has recently found several applications in quantum computation. Catalytic methods were used to reason about resource conversions in fault-tolerant quantum computing, and to derive lower bounds on the cost of certain important computational tasks [6]. Catalytic methods were also used to improve the cost of unitary approximations over restricted gate sets such as the Clifford+ T gate set [2], and to establish number-theoretic characterizations for important extensions of the Clifford gate set [3]. A construction based on catalysis is also currently the leading candidate for the fault-tolerant implementation of small-angle rotations [11], and one of the most promising magic state distillation protocols [12].

In this paper we extend the uses of catalysis in quantum computing in two new directions: establishing the computational universality of gate sets and proving the completeness of graphical calculi.

In the first direction we provide a novel proof of the computational universality of the CS+Hadamard gate set by leveraging the fact that any Clifford+ T circuit can be reduced to a CS+Hadamard circuit using CS gates to catalyse T gates. This side-steps the complicated analytical arguments of the original proof [14], and also establishes that CS+Hadamard circuits only incurs a linear overhead in the number of samples needed compared to Clifford+ T circuits. In addition, this also leads to a simple and self-contained proof of the computational universality of Toffoli+Hadamard circuits.

In the second direction we show that the phase-free ZH-calculus [5, 21] can be extended to a complete calculus for the Clifford+ T fragment by simply including a rule encoding a catalytic equation, along with a scalar cancellation rule. The proof of completeness then follows simply, and this approach works for any tower of quadratic ring extensions

$$\mathbb{D} \subseteq \mathbb{D}[a_1] \subseteq \dots \subseteq \mathbb{D}[a_1, \dots, a_k]$$

where $\mathbb{D} = \mathbb{Z}[\frac{1}{2}]$ and $a_j^2 \in \mathbb{D}[a_1, \dots, a_{j-1}]$. In contrast to previous complete calculi for the Clifford+ T fragment [13, 17], this yields a calculus with rules that are easy to interpret, and relies on a generic proof strategy, rather than one specific to Clifford+ T .

More generally, our results demonstrate that catalytic methods provide powerful means to extend results between different gate sets and graphical calculi.

1 The ZH-calculus and catalysis

The ZH-calculus [4, 5] is a graphical language designed to reason more easily about quantum computing involving controlled unitaries than the earlier ZX-calculus [9, 10]. ZH-diagrams are string diagrams built out of generators representing certain linear maps between qubits that can be composed together either horizontally, corresponding to regular composition of linear maps, and vertically, corresponding to tensor product. The two generators are *Z-spiders* and *H-boxes*. These are represented by circles and squares respectively, and correspond to the following linear maps:

$$\begin{array}{c} \vdots \\ \diagup \quad \diagdown \\ \circ \\ \diagdown \quad \diagup \\ \vdots \end{array} := |0 \dots 0\rangle \langle 0 \dots 0| + |1 \dots 1\rangle \langle 1 \dots 1| \quad (1)$$

$$m \left\{ \begin{array}{c} \vdots \\ \diagup \quad \diagdown \\ \square \\ \diagdown \quad \diagup \\ \vdots \end{array} \right\} n := \sum a^{i_1 \dots i_m j_1 \dots j_n} |j_1 \dots j_n\rangle \langle i_1 \dots i_m| \quad (2)$$

Here the *label* of the H-box a can be any complex number. Both Z-spiders and H-boxes can have any number of inputs or outputs (including zero). If they have n inputs and m outputs, then they correspond to matrices of size $2^n \times 2^m$. The Z-spider matrix consists of all zeroes, except for the top-left and bottom-right corner where there is a 1. The matrix of the H-box is all ones, except for the bottom-right corner where there is an a . If the label of the H-box is -1 , then we usually don't write it. In the special case of 1 input and 1 output the -1 labelled H-box is proportional to the Hadamard. Next to these generators we also have the standard structural generators of compact-closed string-diagrammatic language: identity, swap, cup and cap [20].

We say a graphical calculus is *universal* for a set of matrices when it can represent any matrix in this set using some diagram. When we allow the label of H-boxes to be arbitrary complex numbers, ZH-diagrams are universal for all complex-valued matrices of size $2^n \times 2^m$ [4]. If instead we restrict the labels to some sub-ring R of \mathbb{C} including at least $\mathbb{Z}[\frac{1}{2}]$, then it is universal for matrices over R [5]. In the *phase-free* ZH-calculus we only allow the default label of -1 for the H-boxes, and we augment the calculus with a generator representing the scalar $\frac{1}{2}$ written as a star: \star . The phase-free ZH-calculus is universal for matrices over the ring $\mathbb{Z}[\frac{1}{2}]$ [5].

Let's give some examples of how some useful unitaries are represented as ZH-diagrams. First, We define an *X-spider* and one with a *NOT* applied to it as a derived generator:

$$(X) \quad \begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \circ \\ \diagdown \quad \diagup \\ \dots \end{array} := \star \begin{array}{c} \square \\ \diagup \quad \diagdown \\ \circ \\ \diagdown \quad \diagup \\ \square \end{array} \quad (NOT) \quad \begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \circ \\ \diagdown \quad \diagup \\ \dots \end{array} := \star \begin{array}{c} \dots \\ \diagup \quad \diagdown \\ \circ \\ \diagdown \quad \diagup \\ \dots \end{array} \begin{array}{c} \square \\ \square \end{array} \quad (3)$$

The X-spider allows us to calculate the XORs of computational basis states. Hence, in particular we can use it to represent a CX (i.e. the CNOT) gate:

$$\text{CNOT} = \begin{array}{c} \bullet \\ \text{---} \\ \oplus \end{array} = \begin{array}{c} \circ \\ \text{---} \\ \oplus \end{array} \quad (4)$$

Here we are allowed to write a horizontal wire, because all the (derived) generators of the ZH-calculus are fully symmetric tensors, and hence whether a wire is an input or output does not change the linear map it represents.

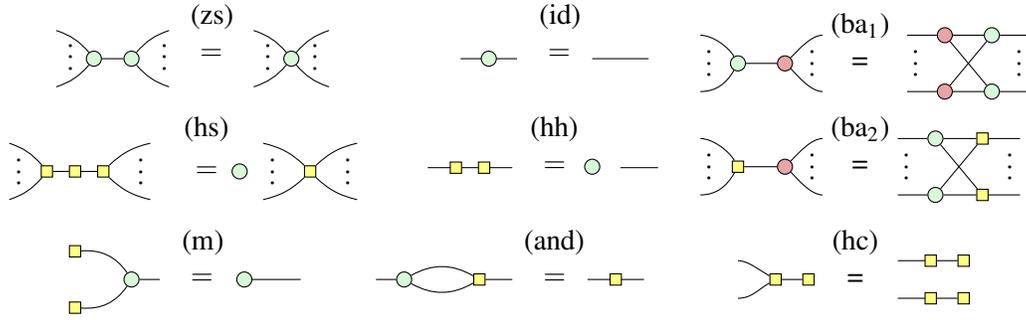
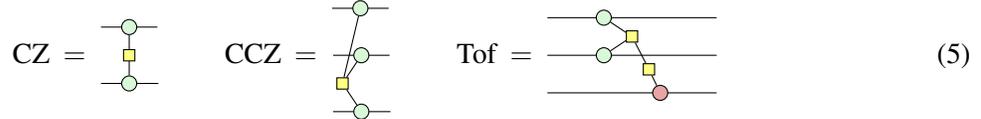


Figure 1: The rules of the phase-free ZH-calculus. The right-hand sides of both *bialgebra* rules (ba_1) and (ba_2) are complete bipartite graphs, with an additional input or output for each vertex. (zs) and (hs) stand respectively for *Z-spider* and *H-spider*; (id) for *identity*; (m) for *multiply*; (and) for the identity involving the AND stating $AND(x,x) = x$; and blafor H-copy.

Other useful gates are the CZ, CCZ and Toffoli gate:



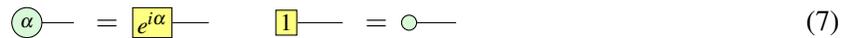
More general controlled-phase gates, can also be represented as ZH-diagrams. In particular, the $CZ(\alpha) := \text{diag}(1, 1, 1, e^{i\alpha})$ gate is represented as follows:



The ZH-calculus is called that because we can actually do calculations with the diagrams. We can treat ZH-diagrams as undirected graphs, because Z-spiders and H-boxes are fully symmetric tensors, and hence the only relevant information in the diagram is which generator is connected to which other. Besides these topological symmetries, we also have a set of *rewrite rules*. We present here the rules for the phase-free ZH-calculus; see Figure 1.

We say a set of rewrite rules is *complete* for a ring R when any two diagrams representing equal linear maps with matrix entries in the ring R can be rewritten into each other using just these rewrite rules. The rule set of Figure 1 is complete for the ring $\mathbb{Z}[\frac{1}{2}]$.

Note that we have the following relations between the spiders of the ZX-calculus [9, 10] and H-boxes:



The following *multiply rule* from the universal ZH-calculus (where H-boxes are labelled by arbitrary complex numbers) will also be useful:



It is this equation that gives the multiply rule (m) its name (take $a = b = -1$).

We can use the ZH-calculus to show the correctness of some simple catalysis equations.

Example 1.1. We can use a single copy of a CS gate and a single $|T\rangle$ state to apply a T gate and get the starting $|T\rangle$ state back:

Here, we used the standard decomposition of a controlled-phase gate $\text{CZ}(\alpha)$ into CNOT gates and $Z(\pm\alpha/2)$ phase gates.

Just using Clifford operations and CS gates, it is not possible to construct a T gate. This is because the number $e^{i\pi/4}$ appearing in the T gate is not part of the ring of entries generated by the matrices of the Clifford and CS gates. However, with Eq. (9) we see that as soon as we have just *one* $|T\rangle$ state available to us, we can use CS gates to apply *any number* of T gates. Indeed, because the catalysis state $|T\rangle$ is not modified by the application of the circuit in Eq. (9), it can be weaved through an arbitrary Clifford+ T circuit in order to replace every T gate by a small Clifford+CS circuit. This catalytic use of $|T\rangle$ states can be contrasted to the process of T gate injection. In the latter process, $|T\rangle$ states are used to perform T gates but can no longer be accessed after the injection has taken place: they have been consumed by the injection.

Example 1.2. We can do something similar with a CCZ gate: we can transform the magic state $|\text{CCZ}\rangle := \text{CCZ}|+++\rangle$ into 3 $|T\rangle$ states using Clifford operations and one T gate:

Here we use some rewrites involving phase gadgets and pushing phases through H-boxes [20].

So again, if we can perform CCZ and Clifford gates and have just a single $|T\rangle$ available, then we can inject as many T gates as we want.

These examples can be captured using the notion of *catalytic embeddings*, which provide a framework for reasoning about certain aspects of catalysis in quantum circuits [2].

Definition 1.3. Let \mathcal{U} and \mathcal{V} be two collection of unitaries. An m -dimensional *catalytic embedding* from \mathcal{U} to \mathcal{V} is a pair $(\phi, |c\rangle)$ of a function $f: \mathcal{U} \rightarrow \mathcal{V}$ and a quantum state $|c\rangle \in \mathbb{C}^m$ such that

$$\phi(U)|\psi\rangle|c\rangle = (U|\psi\rangle)|c\rangle$$

for any unitary $U \in \mathcal{U}$ and any quantum state $|\psi\rangle$. We call the state $|c\rangle$ the *catalyst* of $(\phi, |c\rangle)$.

Definition 1.3 shows that the unitary $\phi(U)$ can be used to apply the unitary U to the state $|\psi\rangle$, provided that the state $|c\rangle$ is available. Intuitively, one should think of the elements of \mathcal{U} as unitaries that are “hard” to implement and of the elements of \mathcal{V} as unitaries that are “easy” to implement. With this in mind, the existence of a catalytic embedding $(\phi, |c\rangle): \mathcal{U} \rightarrow \mathcal{V}$ implies that the hard unitaries can be performed using only the easy ones in the presence of the appropriate catalyst.

As a nice demonstration of the utility of ZH in catalysis, we present in Appendix B a new proof of one of the most useful results using catalysis: how the synthesis of small-angle phase rotations can

be implemented in a fault-tolerant friendly manner, by decomposing it into a series of dyadic angle rotations $Z(\frac{2\pi}{2^n})$, each of which can be implemented using catalysis and an adder gadget [11]. While the end result is not new, our proof of correctness is more elementary, relying only on low level reasoning about catalysis, and does not use special high-level properties concerning phase gradients and Fourier transforms.

2 Catalysing universality

There are several ways in which a gate set can be *universal*. It is common to say that a gate set G is *approximately universal* if any unitary can be approximated up to arbitrary accuracy using circuits over G . Because the fundamental purpose of a quantum computer is to estimate the expectation value of some observable \mathcal{O} , being able to approximately compute such expectation values yields another, weaker, notion of universality: *computational universality*.

We start with some state $|\psi\rangle$, apply some unitary U to it, perform some measurements, and finally post-process these measurements to get an estimate of \mathcal{O} . After many such runs we will get a close approximation of \mathcal{O} . Mathematically we can represent this as

$$\langle \mathcal{O} \rangle = \left\langle \psi \left| \begin{array}{c} \vdots \\ U \\ \mathcal{O} \\ U^\dagger \\ \vdots \end{array} \right| \psi \right\rangle \quad (11)$$

However, when we are trying to estimate this observable, we don't have to do this with just a single quantum circuit we run over and over again. Instead we can have a collection of different quantum circuits V_j (potentially acting on a different number of qubits), input states $|\psi_j\rangle$, and observables \mathcal{O}_j , such that taking a particular weighted average gets us the outcome we are after:

$$\langle \mathcal{O} \rangle = \sum_j \lambda_j \left\langle \psi_j \left| \begin{array}{c} \vdots \\ V_j \\ \mathcal{O}_j \\ V_j^\dagger \\ \vdots \end{array} \right| \psi_j \right\rangle = \sum_j \lambda_j \langle \mathcal{O} \rangle_j, \quad (12)$$

where here we define $\langle \mathcal{O} \rangle_j$ to be the expectation value of \mathcal{O}_j with respect to V_j and $|\psi_j\rangle$. We then see that if we can estimate each of the $\langle \mathcal{O} \rangle_j$, then we can also estimate $\langle \mathcal{O} \rangle$ itself, by just summing our estimates like $\langle \mathcal{O} \rangle = \sum_j \lambda_j \langle \mathcal{O} \rangle_j$.

While we can reduce the calculation of an expectation value to the calculation of a sum of (potentially simpler to calculate) expectation values in this way, there is the important issue of the overhead in the number of samples needed. Generally, we want to determine an error budget for how close we want the estimate to be, and then that determines how many times we need to sample from the quantum computation. Since we are summing together different expectation values, we need to be careful that we aren't blowing up the error in the estimates. Suppose for instance that some $\lambda_k = 100$. Then a small error in our estimate of $\langle \mathcal{O} \rangle_j$ will blow up by a factor of a 100. On the other hand, if $\lambda_k = 1/100$, then any error will also be decreased by a factor of a 100, so that even a large error is not that important. The most efficient strategy is then to sample $\langle \mathcal{O} \rangle_j$ a number of times proportional to $|\lambda_j|$. The total overhead using this sum-of-expectations approach is then $\sum_j |\lambda_j|$ when comparing it to estimating $\langle \mathcal{O} \rangle$ directly with the original circuit.

These sum-of-expectations techniques are used in a variety of subfields in quantum computing. For instance, quasi-probability simulators use such a technique to write a quantum computation as an affine combination of easier to classically simulate quantum computations [7, 16]. They are also used in stochastic compilation in order to better suppress errors, for instance by randomly multiplying a state

by a Pauli in order to get rid of systematic errors [19], or by manipulating the order of Trotter terms when decomposing a Hamiltonian simulation [8]. Here we will see that such a technique can also be used to argue for the computational universality of a gate set, by reducing from a more extensive gate set using catalysis.

2.1 Clifford+CS

Here we will show that we can reduce the calculation of an expectation value involving a Clifford+T circuit to one involving a collection of Clifford+CS circuits with some small overhead. This will prove that the Clifford+CS gate set is computationally universal, as the Clifford+T gate set is as well.

Suppose we have a Clifford+T circuit C applied to the input state $|\psi\rangle$. Then we can transform C into a circuit C' containing just Clifford gates and CS gates using catalysis, so that $C'|\psi\rangle|T\rangle = C|\psi\rangle|T\rangle$. If we were trying to estimate the observable \mathcal{O} we can then check that:

$$\begin{aligned}
 & \frac{1}{\sqrt{2}} \begin{array}{c} \langle \psi | \\ \vdots \\ \langle \psi | \end{array} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] C' \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] \mathcal{O} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] (C')^\dagger \begin{array}{c} \vdots \\ \psi \\ \rangle \end{array} \frac{1}{\sqrt{2}} \begin{array}{c} \langle \psi | \\ \vdots \\ \langle \psi | \end{array} \quad (9) \\
 & = \frac{1}{2} \begin{array}{c} \langle \psi | \\ \vdots \\ \langle \psi | \end{array} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] C \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] \mathcal{O} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] C^\dagger \begin{array}{c} \vdots \\ \psi \\ \rangle \end{array} \quad (13) \\
 & \stackrel{(zs)}{=} \begin{array}{c} \langle \psi | \\ \vdots \\ \langle \psi | \end{array} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] C \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] \mathcal{O} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] C^\dagger \begin{array}{c} \vdots \\ \psi \\ \rangle \end{array}
 \end{aligned}$$

So instead of running the circuit C , we can run the circuit C' , which doesn't contain any T gates. This is then an example of Eq. (12) where the sum is over just one term and we have $\lambda_1 := 1$, $U_1 := C'$, $|\psi_1\rangle := |\psi\rangle \otimes |T\rangle$ and $\mathcal{O}_1 := \mathcal{O} \otimes I$.

But to prepare $|T\rangle$ we still need to use a T gate, so we need to also get rid of this magic state. We can decompose this magic state into a sum of Clifford states. Because each term in the sum needs to retain the form of an expectation value like (11), we can't just decompose $|T\rangle$ into pure states $|\phi_j\rangle$, instead we need to decompose $|T\rangle\langle T|$ into a sum of density matrices $|\phi_j\rangle\langle\phi_j|$. One way to do this is the following:

$$\begin{aligned}
 |T\rangle\langle T| &= \frac{1}{2} \begin{pmatrix} 1 & e^{i\pi/4} \\ e^{-i\pi/4} & 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & \frac{1+i}{\sqrt{2}} \\ \frac{1-i}{\sqrt{2}} & 1 \end{pmatrix} \\
 &= \frac{1}{\sqrt{2}} (|+\rangle\langle +| + |-i\rangle\langle -i|) - \frac{\sqrt{2}-1}{2} (|0\rangle\langle 0| + |1\rangle\langle 1|). \quad (14)
 \end{aligned}$$

Hence, we can decompose $|T\rangle\langle T|$ into four Clifford states $|\phi_1\rangle = |+\rangle$, $|\phi_2\rangle = |-i\rangle$, $|\phi_3\rangle = |0\rangle$ and $|\phi_4\rangle = |1\rangle$ with weights $\lambda_1 = \lambda_2 = \frac{1}{\sqrt{2}}$ and $\lambda_3 = \lambda_4 = -\frac{\sqrt{2}-1}{2}$. Starting with the left-hand side of Eq. (13) we then have:

$$\begin{aligned}
 & \frac{1}{2} \begin{array}{c} \langle \psi | \\ \vdots \\ \langle \psi | \end{array} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] C' \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] \mathcal{O} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] (C')^\dagger \begin{array}{c} \vdots \\ \psi \\ \rangle \end{array} \frac{1}{2} \begin{array}{c} \langle \psi | \\ \vdots \\ \langle \psi | \end{array} \quad (9) \\
 & = \frac{1}{2} \begin{array}{c} \langle \psi | \\ \vdots \\ \langle \psi | \end{array} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] C' \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] \mathcal{O} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] (C')^\dagger \begin{array}{c} \vdots \\ \psi \\ \rangle \end{array} \quad (15) \\
 & \stackrel{(14)}{=} \sum_j \lambda_j \begin{array}{c} \langle \psi | \\ \vdots \\ \langle \psi | \end{array} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] C' \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] \mathcal{O} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] (C')^\dagger \begin{array}{c} \vdots \\ \psi \\ \rangle \end{array} \\
 & \quad \sum_j \lambda_j \begin{array}{c} \langle \phi_j | \\ \vdots \\ \langle \phi_j | \end{array} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] C' \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] \mathcal{O} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \right] (C')^\dagger \begin{array}{c} \vdots \\ \psi \\ \rangle \end{array}
 \end{aligned}$$

We see then that this is a case of Eq. (12) with $|\psi_j\rangle := |\psi\rangle \otimes |\phi_j\rangle$ and $\mathcal{O}_j := \mathcal{O} \otimes I$ and $U_j = C'$ for all $j \in \{1, 2, 3, 4\}$. Furthermore, we can check that the four terms have $\sum_j |\lambda_j| = 2\sqrt{2} - 1 \approx 1.83$. Hence, if we decompose the magic state in this way we need to collect 1.83 samples more than we would have needed to if we did use the magic $|T\rangle$ state directly.

Summarising the full procedure we see then that we can do the following:

1. Start with the Clifford+T computation you want to calculate.
2. Replace all T gates by a CS gate catalysis circuit using a $|T\rangle$.
3. Replace the $|T\rangle$ state needed for all the catalysis by the Clifford states $|\psi_j\rangle$.
4. Run each of the resulting four circuits a number of times proportional to $|\lambda_j|$.
5. Combine the resulting estimates of the observable by scaling by λ_j to get the final outcome.

When we have Clifford gates and CS gates, the gate set is generated by CNOT, Hadamard, S and CS. The CNOT can be constructed using CS and Hadamard, and if we allow states to be prepared in the $|0\rangle$ and $|1\rangle$ (which is necessary to encode different input states), then we can also prepare an S using a CS. Hence, this gate set is equivalent to just the CS and Hadamard gate. We see then that we have proven the following.

Theorem 2.1. The CS+Hadamard gate set is computationally universal. In particular, a Clifford+T computation can be simulated by a CS+Hadamard computation with a linear overhead in the number of samples, qubits and gates needed.

Remark 2.2. Without using the catalysis, we could have also chosen to write each T gate as a magic state injection, and then replace each of the $|T\rangle\langle T|$ states by the Clifford states $|\phi_j\rangle\langle\phi_j|$. When we do this however, we get a number of terms in the decompositions that scales exponentially in the number of T gates, and in particular $\sum_j |\lambda_j|$ scales exponentially, so that the simulation would no longer be efficient. This makes sense, since replacing all the T gates gives us a Clifford circuit, and we don't expect this gate set to be computationally universal.

Remark 2.3. CS+Hadamard is in fact approximately universal, as proven by Kitaev [14, Lemma 4.6 on p. 1213] (note that what he calls S is the Hadamard gate, and K is the S gate). Kitaev proves this using an unproven 'geometric lemma' that adding a gate to a set of gates that stabilises a given state creates a larger-dimensional space of gates. This proof is not constructive. He then proves what is now known as the Solovay-Kitaev algorithm to show how you would do it constructively. Though this result also establishes the computational universality of CS+Hadamard, we remark that that construction would, for instance, give an approximate decomposition of the T gate, meaning the cost of implementing the T would scale with the desired precision, whereas our method has a constant overhead.

2.2 Real-valued unitaries

An interesting example of a computationally universal gate set is the set of real-valued unitaries [1]. That is, where we only allow unitaries where all matrix entries are real numbers. Obviously this gate set is not approximately universal as we can never approximate any complex-valued unitary (like the S gate), but it turns out we can 'simulate' complex-valued unitaries using a real-valued one on a larger set of qubits.

For a (complex-valued) n -qubit unitary U , let $\Re(U)$ be the real part of U . That is: $\Re(U)_{ij} = \Re(U_{ij})$. Similarly define the complex part $\Im(U)$. Then $U = \Re(U) + i\Im(U)$. Now define the $(n+1)$ -qubit unitary \tilde{U} via

$$\begin{aligned}\tilde{U}(|0\rangle \otimes |\psi\rangle) &:= |0\rangle \otimes (\Re(U)|\psi\rangle) + |1\rangle \otimes (\Im(U)|\psi\rangle) \\ \tilde{U}(|1\rangle \otimes |\psi\rangle) &:= -|0\rangle \otimes (\Im(U)|\psi\rangle) + |1\rangle \otimes (\Re(U)|\psi\rangle)\end{aligned}$$

While it is clear that \tilde{U} is real-valued, it is not immediately obvious that it is unitary. We show this, and the fact that $\widetilde{UV} = \tilde{U}\tilde{V}$ in Appendix A.1.

This construction from [1] can actually also be seen as an example of catalysis as $\tilde{U}|-i\rangle \otimes |\psi\rangle = |-i\rangle \otimes U|\psi\rangle$. This then might look like we can use the previous sum-of-expectations approach to argue that real-valued unitaries are computationally universal. This does not work however, as we cannot write $|-i\rangle\langle -i|$ as a sum of self-adjoint real matrices, so that we can't represent it directly just using real-valued unitaries and state preparations. We can however still prove computational universality using a slightly different argument.

In fact, we can exactly simulate the probability distribution arising from a complex-valued circuit. To see this, first note that if $|\psi\rangle$ and $|\psi'\rangle$ are real-valued states that then $|\langle\psi|C|\psi'\rangle|^2 = |\langle\psi|\Re(C)|\psi'\rangle|^2 + |\langle\psi|\Im(C)|\psi'\rangle|^2$ (this requires some computation to show). On the other hand, if we input $|0\rangle \otimes |\psi'\rangle$ into \tilde{C} and do a measurement marginalising over the first qubit we also get the probabilities:

$$\sum_{x=0,1} |\langle x, \psi | \tilde{C} | 0, \psi' \rangle|^2 = |\langle \psi | \Re(C) | \psi' \rangle|^2 + |\langle \psi | \Im(C) | \psi' \rangle|^2 = |\langle \psi | C | \psi' \rangle|^2.$$

So the probability distribution we get for C is the same as that for \tilde{C} when we prepare the first qubit in the $|0\rangle$ state and ignore its measurement outcome.

Proposition 2.4. Real-valued unitaries are computationally universal.

2.3 Toffoli+Hadamard

Because we can simulate complex-valued quantum circuits using real-valued unitaries in the direct manner described above, we don't need *all* the real-valued unitaries. Given any computationally universal gate set G we only need \tilde{U} for $U \in G$.

In particular, for CS+Hadamard, we can check that we get a real-valued unitary \widetilde{CS} that is equivalent to a Toffoli up to some swaps. With the Hadamard we just get $\tilde{H} = I \otimes H$. Hence, when we encode the CS+Hadamard gate set, we get the Toffoli+Hadamard gate set [1]. We can hence do the following: starting with a Clifford+T computation, we write it as an ensemble of CS+Hadamard circuits. We then encode each of these circuits into a real-valued Toffoli+Hadamard circuit. By doing this we can efficiently simulate the original Clifford+T circuit. We see then that Toffoli+Hadamard circuits are also computationally universal.

Theorem 2.5. The Toffoli+Hadamard gate set is computationally universal.

Note that we could also do a version of this without first encoding a complex unitary as a real unitary by using the fact that we can catalyse T gates using a CCZ gate. We then get a version of Theorem 2.1 for the Clifford+CCZ gate set. Using this method, we however also need the S gate, which is not necessary using the above approach.

Our proof of the universality of Toffoli+Hadamard follows along the same lines as that of Aharonov [1], namely, by reducing it to CS+Hadamard. However, the original proof of CS+Hadamard universality is non-constructive and relies on a series of non-trivial encodings of unitaries into larger-dimensional spaces, whilst our proof reduces the problem to Clifford+ T universality, which itself simply reduces to universality of CNOT+single-qubit unitaries.

2.4 Catalysing general gate sets

We can generalise these specific statements of computational universality to a general statement about gate sets that have catalytic embeddings. We present the proof in Appendix A.2.

Theorem 2.6. Let \mathcal{U} and \mathcal{V} be gate sets which have a catalytic embedding $(\phi, |c\rangle)$ as in Definition 1.3 and suppose that $|c\rangle\langle c|$ can be written as a sum $\sum_j \lambda_j |\psi_j\rangle\langle\psi_j|$ where each of the $|\psi_i\rangle$ can be prepared by a circuit over \mathcal{V} . Then if the bigger gate set \mathcal{U} is computationally universal, the smaller gate set \mathcal{V} is also computationally universal.

In this result we needed the catalyst $|c\rangle\langle c|$ to be expressible as a sum of states that can be prepared by the smaller gate set. In the examples of catalysis we saw earlier, these states were all stabiliser states, and hence could be expressed as the gate set included all Cliffords. There is however nothing special about a stabiliser decomposition, and any decomposition into expressible states is sufficient. Even an approximate decomposition would suffice, as long as the 1-norm $\sum_j |\lambda_j|$ of the approximate decomposition scales polynomially in the desired error rate ϵ . This condition is needed because the error in the catalysis state needs to be lower if it is used more often, and hence the 1-norm should not increase too rapidly.

3 Catalysing Completeness

We can also use catalysis in order to define extensions of graphical calculi and prove completeness for them. To see how this works, we want to first generalise the T gate catalysis of Eq. (9). First, as our goal will just be to produce states, we can plug $|+\rangle$ into the top wire of Eq. (9). We can then simplify the expression to a more symmetric form:

$$\begin{array}{c} \text{---} \circ \frac{\pi}{4} \text{---} \\ \text{---} \circ \frac{\pi}{4} \text{---} \end{array} \quad (9) \quad \begin{array}{c} \text{---} \circ \text{---} \\ \text{---} \circ \frac{\pi}{4} \text{---} \end{array} \quad \begin{array}{c} \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \end{array} \quad \begin{array}{c} \text{---} \circ \text{---} \\ \text{---} \circ \frac{\pi}{4} \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \end{array} \quad \begin{array}{c} \text{---} \circ \frac{\pi}{4} \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \end{array} \quad \begin{array}{c} \text{---} \circ \frac{\pi}{4} \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \end{array} \quad (16)$$

We can then identify the underlying reason this catalysis works. It is because we have the following 'Euler decomposition' of the H-box with an i phase:

$$\begin{array}{c} \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \end{array} \quad \begin{array}{c} \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \end{array} \quad \begin{array}{c} \text{---} \circ \frac{\pi}{4} \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \frac{\pi}{4} \text{---} \end{array} \quad \begin{array}{c} \text{---} \circ \frac{e^{i\pi/4}}{\text{---}} \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \frac{e^{-i\pi/4}}{\text{---}} \text{---} \end{array} \quad (17)$$

Here we use the fact that $\begin{array}{c} \text{---} \circ \frac{e^{i\alpha}}{\text{---}} \text{---} \\ \text{---} \circ \text{---} \end{array} = \begin{array}{c} \text{---} \circ \alpha \text{---} \\ \text{---} \circ \text{---} \end{array}$ to write the phases as H-boxes. We do this because such a rule doesn't just hold for an H-box with a label that is a complex phase like $e^{i\alpha}$, it in fact holds for any complex $a \neq 0$:

$$\begin{array}{c} \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \end{array} \quad \begin{array}{c} \text{---} \circ \frac{\sqrt{a}}{\text{---}} \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \frac{1/\sqrt{a}}{\text{---}} \text{---} \\ \text{---} \circ \frac{\sqrt{a}}{\text{---}} \text{---} \end{array} \quad (18)$$

This then allows us to write down a generalisation of Eq. (16) to arbitrary H-boxes:

$$\begin{array}{c} \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \end{array} \quad (18) \quad \begin{array}{c} \text{---} \circ \frac{a}{\text{---}} \text{---} \\ \text{---} \circ \frac{1/a}{\text{---}} \text{---} \end{array} \quad \begin{array}{c} \text{---} \circ \frac{a}{\text{---}} \text{---} \\ \text{---} \circ \frac{1/a}{\text{---}} \text{---} \end{array} \quad (8) \quad \begin{array}{c} \text{---} \circ \frac{a}{\text{---}} \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \frac{1}{\text{---}} \text{---} \\ \text{---} \circ \frac{a}{\text{---}} \text{---} \end{array} \quad (7) \quad \begin{array}{c} \text{---} \circ \frac{a}{\text{---}} \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \frac{a}{\text{---}} \text{---} \end{array} \quad = \quad \begin{array}{c} \text{---} \circ \frac{a}{\text{---}} \text{---} \\ \text{---} \circ \frac{a}{\text{---}} \text{---} \end{array} \quad (19)$$

Note that this equation first appeared in [15]. Here we wrote a^2 in the 2-ary H-box instead of a so that we don't have to work with square roots. When we take $a = e^{i\pi/4}$ we get Eq. (16), but this works for any value.

A particularly simple, but still interesting case is when $a = e^{i\frac{\pi}{2}} = i$. Translating this back into circuit form gives us a catalysis of $|i\rangle := |0\rangle + i|1\rangle$ states using a CZ. As a rewrite rule this is essentially equivalent to the Euler decomposition of a Hadamard. While this rule itself is simple, it already demonstrates the power of the catalysis framework in proving completeness.

The phase-free ZH-calculus is complete for the ring $\mathbb{Z}[\frac{1}{2}]$. By adding the generator \boxed{i} , a single-ary H-box with label $a = i$, we get a universal representation for the ring $\mathbb{Z}[\frac{1}{2}, i]$ [5]. As it turns out, adding the rule Eq. (19) for $a = i$ to the already existing rules for the phase-free fragment is already *almost* enough to get a complete calculus for this bigger fragment $\mathbb{Z}[\frac{1}{2}, i]$ which includes \boxed{i} .

To see this, we first consider what a generic diagram in the $\mathbb{Z}[\frac{1}{2}, i]$ fragment looks like. We added the generator \boxed{i} , so now a diagram consists of generators from the old $\mathbb{Z}[\frac{1}{2}]$ fragment plus this new generator, used an arbitrary number of times. Using Eq. (19) we can however reduce all these separate instances of \boxed{i} into just one of them, reducing the complexity of the diagram. That is, given some diagram D in the $\mathbb{Z}[i]$ fragment, we can rewrite it to a diagram D' containing just generators from the $\mathbb{Z}[\frac{1}{2}]$ fragment such that:

$$\begin{array}{c} \vdots \\ \boxed{D} \\ \vdots \end{array} \stackrel{(19)}{=} \begin{array}{c} \vdots \\ \boxed{D'} \\ \boxed{i} \\ \vdots \end{array} \quad (20)$$

We are for now ignoring the edge case where the original diagram did not contain any instance of \boxed{i} .

As a shorthand, we will write $D'[[\psi]]$ for the diagram we get when we plug $|\psi\rangle$ into the bottom input in Eq. (20). So here we have $D = D'[\boxed{i}]$. Note that $\boxed{i} = |0\rangle + i|1\rangle$. Hence, if we expand it like this we see that D is equal to a sum of two diagrams: D' where we plugged in $|0\rangle$ into the bottom wire, and iD' where we plugged $|1\rangle$ into the bottom wire: $D = D'[[0]] + iD'[[1]]$.

Now suppose we have two diagrams D_1 and D_2 in the $\mathbb{Z}[i]$ fragment and that they implement the same linear map: $D_1 = D_2$. We can both decompose them as described above to get $D'_1[[0]] + iD'_1[[1]] = D'_2[[0]] + iD'_2[[1]]$. Each of these $D'_j[[x]]$ diagrams represents a matrix that is entirely real-valued, so the only way for this equation of complex matrices to hold, is if it holds for the real part and for the complex part separately:

$$D'_1[[0]] = D'_2[[0]] \quad D'_1[[1]] = D'_2[[1]] \quad (21)$$

We then conclude that D'_1 and D'_2 are equal when we input either $|0\rangle$ or $|1\rangle$ into the bottom wire. As these states form a basis, this must then hold for any input. We can then leave this wire open and still have an equality:

$$\begin{array}{c} \vdots \\ \boxed{D'_1} \\ \vdots \end{array} = \begin{array}{c} \vdots \\ \boxed{D'_2} \\ \vdots \end{array} \quad (22)$$

We have this equality as linear maps, but both diagrams are in the $\mathbb{Z}[\frac{1}{2}]$ fragment for which we have completeness. We hence know how to rewrite one into the other using the rules of the phase-free ZH-calculus. This gives us then a path to rewrite the original D_1 into D_2 :

$$\begin{array}{c} \vdots \\ \boxed{D_1} \\ \vdots \end{array} \stackrel{(19)}{=} \begin{array}{c} \vdots \\ \boxed{D'_1} \\ \boxed{i} \\ \vdots \end{array} \stackrel{(*)}{=} \begin{array}{c} \vdots \\ \boxed{D'_2} \\ \boxed{i} \\ \vdots \end{array} \stackrel{(19)}{=} \begin{array}{c} \vdots \\ \boxed{D_2} \\ \vdots \end{array} \quad (23)$$

Here each equality is now a diagrammatic equality, and with $(*)$ we denote we are using rewrites from the original complete calculus for the \mathbb{Z} fragment. This would give us completeness for the fragment $\mathbb{Z}[\frac{1}{2}, i]$, except that we have ignored an edge case. We can only rewrite a diagram in the $\mathbb{Z}[i]$ fragment

as in Eq. (20) if there is at least one generator \boxed{i} — present in the diagram. In fact, we currently haven't assumed any rewrite rule that relate a diagram containing a \boxed{i} — to one that does not contain any \boxed{i} — . This means in particular that our currently considered rule set cannot prove the following true equation:

$$\textcircled{\text{red}}-\boxed{i} = \textcircled{\text{red}}-\textcircled{\text{green}} \quad (24)$$

However, when we also add Eq. (24) as an additional rule, then this problem is solved and it *is* true that we can then always rewrite a diagram in the $\mathbb{Z}[i]$ fragment as in Eq. (20): if the diagram contains at least one \boxed{i} — we can already use Eq. (19) to transform to the form of Eq. (20), and if it does not, we can use Eq. (24) once to introduce one \boxed{i} — , in which case it is also in the form of Eq. (20).

Proposition 3.1. The graphical calculus consisting of the phase-free ZH generators and \boxed{i} — , together with the phase-free rewrite rules of Figure 1 augmented with the catalysis rule Eq. (19) for $a = i$, and the rule $\textcircled{\text{red}}-\boxed{i} = \textcircled{\text{red}}-\textcircled{\text{green}}$ is complete and universal for the ring $\mathbb{Z}[\frac{1}{2}, i]$.

This trick for extending the calculus doesn't just work for i : it works for any complex number $a \neq 0$ such that $a^2 \in \mathbb{Z}$ using a very similar argument. We can also iterate it: once we have a calculus complete and universal for $\mathbb{Z}[\frac{1}{2}, a]$ we can pick any $b \neq 0$ such that $b^2 \in \mathbb{Z}[\frac{1}{2}, a]$ and augment the calculus with the appropriate catalysis and scalar-introduction rule to get a new complete calculus for $\mathbb{Z}[\frac{1}{2}, a, b]$. We present the proof in Appendix A.3.

Theorem 3.2. Let a_1, \dots, a_k be a series of non-zero complex numbers such that $a_j^2 \in \mathbb{Z}[\frac{1}{2}, a_1, \dots, a_{j-1}]$. Then the phase-free ZH-calculus augmented with generators $\boxed{a_j}$ — and the following rules is complete for the ring $\mathbb{Z}[\frac{1}{2}, a_1, \dots, a_k]$:

$$\boxed{a}-\textcircled{\text{red}}-\boxed{a^2} = \begin{array}{c} \boxed{a}- \\ \boxed{a}- \end{array} \quad \text{and} \quad \textcircled{\text{red}}-\boxed{a} = \textcircled{\text{red}}-\textcircled{\text{green}} \quad \text{for all } a = a_j$$

Here the a^2 H-box should be understood as short-hand for some diagram in the smaller fragment representing the matrix for that H-box.

When we take $a_1 = i$ and $a_2 = e^{i\frac{\pi}{4}}$ we get the ring $\mathbb{Z}[\frac{1}{2}, i, e^{i\frac{\pi}{4}}] = \mathbb{Z}[i, \frac{1}{\sqrt{2}}]$ corresponding to Clifford+T computation, and in this case we can simplify the rules a bit more to get a simple axiomatisation of the Clifford+T maps.

Proposition 3.3. The phase-free ZH-calculus augmented with H-boxes with a label of i and $e^{i\frac{\pi}{4}}$ and the following rules is complete for matrices with entries in the ring $\mathbb{Z}[i, \frac{1}{\sqrt{2}}]$:

$$\begin{array}{c} \textcircled{\frac{\pi}{2}} \\ \textcircled{\text{red}} \\ \textcircled{\text{green}} \end{array} = \textcircled{\frac{\pi}{2}} \textcircled{\frac{\pi}{2}} \quad \begin{array}{c} \textcircled{\frac{\pi}{4}} \\ \textcircled{\text{red}} \\ \textcircled{\text{green}} \end{array} \boxed{i} = \textcircled{\frac{\pi}{4}} \textcircled{\frac{\pi}{4}} \quad \textcircled{\frac{\pi}{4}}-\textcircled{\text{red}} = \textcircled{\text{green}}-\textcircled{\text{red}}$$

Proof. These are the additional rules needed by Theorem 3.2 for completeness, except we don't have the scalar introduction rule for the label i . This rule can be derived from the $e^{i\frac{\pi}{4}}$ one, in combination with the catalysis rule for $e^{i\frac{\pi}{4}}$. \square

Here we presented the rules in a slightly different manner to make clear the connection between catalysis for $\frac{\pi}{2}$ and the standard Euler decomposition of the Hadamard. Continuing the division into smaller dyadic rational multiples of π of the form $e^{i\frac{\pi}{2^k}}$, we see that we can get a complete calculus for diagrams corresponding to Clifford-cyclotomic circuits, similar to how exact synthesis for these circuits

was proven in [3]. Independently to our results, completeness for dyadic angles was also shown in the context of the Sum-over-paths formalism in [18] using a technique that is reminiscent of catalysis.

While there have been previous complete graphical calculi for the fragment corresponding to Clifford+ T circuits [13, 17], the result we find here has the double benefit of having easy to interpret axioms, and a generic proof. The axioms consist of the phase-free ZH ones, each of which corresponds to a simple property of the Boolean maps COPY, XOR and AND [5], plus the catalysis rules. This extension is not specific to Clifford+ T and works for any ring extension of the form stated in Theorem 3.2. Note that [5] also describes how the ZH-calculus can be made complete for arbitrary rings, but these require adding three families of rules that are each parametrised over all the elements in the ring, and hence gives a much more complex rule set.

4 Conclusion

We applied the technique of quantum state catalysis to prove generic results in universality and completeness. In particular, we obtained new proofs of the universality of the CS+Hadamard and Toffoli+Hadamard gate sets, as well as a new simple graphical calculus for the Clifford+ T fragment. Our results simplify the original proofs of these statements considerably, and pave the way for further applications of catalysis in these areas. Notably, our completeness result did not use any special property of the ZH-calculus. The fact that it was universal for a large enough fragment to support catalysis was enough to find completeness of an extension. This technique could hence also be applied to study calculi over qudits, where for certain interesting fragments of quantum computing (like qudit Clifford+ T), there is still no complete calculus. In a related direction, one could consider using catalytic methods to study mixed-dimensional calculi, since catalytic constructions such as the ones in [2] can often be made to take advantage of mixed-dimensional ancillae.

References

- [1] Dorit Aharonov (2003): *A simple proof that Toffoli and Hadamard are quantum universal*. *arXiv preprint quant-ph/0301040*.
- [2] Matthew Amy, Matthew Crawford, Andrew N. Glaudell, Melissa L. Macasieb, Samuel S. Mendelson & Neil J. Ross (2023): *Catalytic embeddings of quantum circuits*. Preprint. Available from arXiv:2305.07720.
- [3] Matthew Amy, Andrew N. Glaudell, Shaun Kelso, William Maxwell, Samuel S. Mendelson & Neil J. Ross (2023): *Exact synthesis of multiqubit Clifford-cyclotomic circuits*. Preprint. Available from arXiv:2311.07741.
- [4] Miriam Backens & Aleks Kissinger (2019): *ZH: A Complete Graphical Calculus for Quantum Computations Involving Classical Non-linearity*. In Peter Selinger & Giulio Chiribella, editors: *Proceedings of the 15th International Conference on Quantum Physics and Logic, Halifax, Canada, 3-7th June 2018, Electronic Proceedings in Theoretical Computer Science 287*, Open Publishing Association, pp. 18–34, doi:10.4204/EPTCS.287.2.
- [5] Miriam Backens, Aleks Kissinger, Hector Miller-Bakewell, John van de Wetering & Sal Wolffs (2023): *Completeness of the ZH-calculus*. *Compositionality 5*, doi:10.32408/compositionality-5-5.
- [6] Michael Beverland, Earl Campbell, Mark Howard & Vadym Kliuchnikov (2020): *Lower bounds on the non-Clifford resources for quantum computations*. *Quantum Science and Technology 5*(3), p. 035009, doi:10.1088/2058-9565/ab8963. arXiv:1904.01124.

- [7] Sergey Bravyi & David Gosset (2016): *Improved Classical Simulation of Quantum Circuits Dominated by Clifford Gates*. *Physical Review Letters* 116(25), p. 250501, doi:10.1103/PhysRevLett.116.250501. arXiv:1601.07601.
- [8] Earl Campbell (2019): *Random Compiler for Fast Hamiltonian Simulation*. *Phys. Rev. Lett.* 123, p. 070503, doi:10.1103/PhysRevLett.123.070503. Available at <https://link.aps.org/doi/10.1103/PhysRevLett.123.070503>.
- [9] Bob Coecke & Ross Duncan (2008): *Interacting quantum observables*. In: *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP)*, Lecture Notes in Computer Science, doi:10.1007/978-3-540-70583-3_25.
- [10] Bob Coecke & Ross Duncan (2011): *Interacting quantum observables: categorical algebra and diagrammatics*. *New Journal of Physics* 13, p. 043016, doi:10.1088/1367-2630/13/4/043016.
- [11] Craig Gidney (2018): *Halving the cost of quantum addition*. *Quantum* 2, p. 74, doi:10.22331/q-2018-06-18-74.
- [12] Craig Gidney & Austin G. Fowler (2019): *Efficient magic state factories with a catalyzed $|CCZ\rangle$ to $2|T\rangle$ transformation*. *Quantum* 3, p. 135, doi:10.22331/q-2019-04-30-135.
- [13] Emmanuel Jeandel, Simon Perdrix & Renaud Vilmart (2018): *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*, ACM, pp. 559–568, doi:10.1145/3209108.3209131.
- [14] A Yu Kitaev (1997): *Quantum computations: algorithms and error correction*. *Russian Mathematical Surveys* 52(6), p. 1191.
- [15] Tuomas Laakkonen, Konstantinos Meichanetzidis & John van de Wetering (2023): *Picturing Counting Reductions with the ZH-Calculus*. In Shane Mansfield, Benoit Valiron & Vladimir Zamdzhiev, editors: *Proceedings of the Twentieth International Conference on Quantum Physics and Logic, Paris, France, 17-21st July 2023, Electronic Proceedings in Theoretical Computer Science* 384, Open Publishing Association, pp. 89–113, doi:10.4204/EPTCS.384.6.
- [16] Victor Veitch, Christopher Ferrie, David Gross & Joseph Emerson (2012): *Negative quasi-probability as a resource for quantum computation*. *New Journal of Physics* 14(11), p. 113011, doi:10.1088/1367-2630/14/11/113011.
- [17] Renaud Vilmart (2019): *A ZX-Calculus with Triangles for Toffoli-Hadamard, Clifford+T, and Beyond*. In Peter Selinger & Giulio Chiribella, editors: *Proceedings of the 15th International Conference on Quantum Physics and Logic, Halifax, Canada, 3-7th June 2018, Electronic Proceedings in Theoretical Computer Science* 287, Open Publishing Association, pp. 313–344, doi:10.4204/EPTCS.287.18.
- [18] Renaud Vilmart (2023): *Completeness of Sum-Over-Paths for Toffoli-Hadamard and the Dyadic Fragments of Quantum Computation*. In Bartek Klin & Elaine Pimentel, editors: *31st EACSL Annual Conference on Computer Science Logic (CSL 2023), Leibniz International Proceedings in Informatics (LIPIcs)* 252, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 36:1–36:17, doi:10.4230/LIPIcs.CSL.2023.36.
- [19] Joel J Wallman & Joseph Emerson (2016): *Noise tailoring for scalable quantum computation via randomized compiling*. *Physical Review A* 94(5), p. 052325, doi:10.1103/PhysRevA.94.052325.
- [20] John van de Wetering (2020): *ZX-calculus for the working quantum computer scientist*. arXiv preprint arXiv:2012.13966.
- [21] John van de Wetering & Sal Wolffs (2019): *Completeness of the Phase-free ZH-calculus*. Preprint. arXiv:1904.07545.

A Proofs

A.1 Universality of real-valued unitaries

Lemma A.1. \tilde{U} is indeed unitary for any choice of U .

Proof. We will first show the following claims:

- We can express $\Re(U^\dagger)$ and $\Im(U^\dagger)$ in terms of $\Re(U)$ and $\Im(U)$.
 - We can Express $\Re(UV)$ and $\Im(UV)$ in terms of $\Re(U)$, $\Re(V)$, $\Im(U)$ and $\Im(V)$.
 - We have $(\langle 0| \otimes \langle \psi| \tilde{U}^\dagger)(\tilde{U}(|0\rangle \otimes |\psi'\rangle)) = \langle \psi| \psi'\rangle$.
 - We have $(\langle 0| \otimes \langle \psi| \tilde{U}^\dagger)(\tilde{U}(|1\rangle \otimes |\psi'\rangle)) = 0$.
- $\Re(U^\dagger) = \Re(U)^\dagger$. $\Im(U^\dagger) = -\Im(U)^\dagger$.
 - $\Re(UV) = \Re(U)\Re(V) - \Im(U)\Im(V)$. $\Im(UV) = \Re(U)\Im(V) + \Im(U)\Re(V)$.
 - First note that $(\langle 0| \otimes \langle \psi| \tilde{U}^\dagger = (\tilde{U}(|0\rangle \otimes |\psi\rangle))^\dagger = \langle 0| \otimes (\langle \psi| \Re(U^\dagger)) - \langle 1| \otimes (\langle \psi| \Im(U^\dagger))$. Hence, using $\langle 0|1\rangle = 0$ the inner product reduces to $\langle \psi| \Re(U^\dagger) \Re(U) | \psi'\rangle - \langle \psi| \Im(U^\dagger) \Im(U) | \psi'\rangle = \langle \psi| (\Re(U^\dagger) \Re(U) - \Im(U^\dagger) \Im(U)) | \psi'\rangle = \langle \psi| \Re(U^\dagger U) | \psi'\rangle = \langle \psi| \Re(I) | \psi'\rangle = \langle \psi| \psi'\rangle$.
 - Similar to the above.

Let $|\psi_k\rangle$ form an orthogonal basis of n -qubit state space. The last two points above show that \tilde{U} preserves the orthogonality of $\{|\psi_k\rangle \otimes |0\rangle, |\psi_k\rangle \otimes |1\rangle\}$. Hence, since it sends an orthogonal basis to an orthogonal basis, it is unitary. \square

The encoding into \tilde{U} is compositional, meaning we can apply it iteratively to a sequence of unitaries.

Lemma A.2. $\widetilde{UV} = \tilde{U}\tilde{V}$.

Proof. Proven easily by making a case distinction on input states $|\psi\rangle \otimes |0\rangle$ and $|\psi\rangle \otimes |1\rangle$. \square

A.2 Generic universality through catalysis

Proof of Theorem 2.6. Let C be some circuit over U . Then $C' := \phi(C)$ is a circuit over \mathcal{V} such that $C'|\psi\rangle \otimes |c\rangle = (C|\psi\rangle) \otimes |c\rangle$. By assumption we have $|\psi_j\rangle = C_j|0 \dots 0\rangle$ for some circuit C_j over \mathcal{V} . Hence, for some observable \mathcal{O} we have:

$$\begin{aligned}
 & \left\langle \begin{array}{c} \psi \\ \vdots \\ \psi \end{array} \right| C \mathcal{O} C^\dagger \left| \begin{array}{c} \psi \\ \vdots \\ \psi \end{array} \right\rangle = \left\langle \begin{array}{c} \psi \\ \vdots \\ \psi \\ \hline c \\ \vdots \\ c \end{array} \right| C \mathcal{O} C^\dagger \left| \begin{array}{c} \psi \\ \vdots \\ \psi \\ \hline c \\ \vdots \\ c \end{array} \right\rangle = \left\langle \begin{array}{c} \psi \\ \vdots \\ \psi \\ \hline c \\ \vdots \\ c \end{array} \right| C' \mathcal{O} (C')^\dagger \left| \begin{array}{c} \psi \\ \vdots \\ \psi \\ \hline c \\ \vdots \\ c \end{array} \right\rangle \\
 & = \sum_j \lambda_j \left\langle \begin{array}{c} \psi \\ \vdots \\ \psi \\ \hline \psi_j \\ \vdots \\ \psi_j \end{array} \right| C' \mathcal{O} (C')^\dagger \left| \begin{array}{c} \psi \\ \vdots \\ \psi \\ \hline \psi_j \\ \vdots \\ \psi_j \end{array} \right\rangle = \sum_j \lambda_j \left\langle \begin{array}{c} \psi \\ \vdots \\ \psi \\ \hline 0 \\ \vdots \\ 0 \end{array} \right| C' \mathcal{O} (C')^\dagger \left| \begin{array}{c} \psi \\ \vdots \\ \psi \\ \hline 0 \\ \vdots \\ 0 \end{array} \right\rangle \\
 & \quad \left\langle \begin{array}{c} \psi \\ \vdots \\ \psi \\ \hline 0 \\ \vdots \\ 0 \end{array} \right| C_j \mathcal{O} (C_j)^\dagger \left| \begin{array}{c} \psi \\ \vdots \\ \psi \\ \hline 0 \\ \vdots \\ 0 \end{array} \right\rangle
 \end{aligned}$$

Hence, setting $|\psi'_j\rangle := |\psi\rangle \otimes |0 \dots 0\rangle$, $C'_j := C' \circ (I \otimes C_j)$ and $\mathcal{O}_j := \mathcal{O} \otimes I$, we see that we can simulate the computation of an expectation value over a circuit in \mathcal{U} using a set of circuits in \mathcal{V} . Any computation done in \mathcal{U} can then also be done using circuits in \mathcal{V} . Hence, \mathcal{V} is computationally universal. \square

A.3 Completeness of extensions of the ZH-calculus

Proof of Theorem 3.2. First, note that the catalysis equation is well-typed: By assumption we have $a_i^2 \in \mathbb{Z}[\frac{1}{2}, a_1, \dots, a_{i-1}]$. We know that if we have the generators $\boxed{a_j}$ — for $j \leq i-1$, then the calculus can represent any matrix with entries in $\mathbb{Z}[\frac{1}{2}, a_1, \dots, a_{i-1}]$. So there is some way to represent the matrix of the H-box with a label of a_i^2 . Assuming the calculus over this ring is complete, any such way to represent this matrix is equivalent, and hence would lead to an equivalent catalysis rule.

We will prove by induction on k with base case $k = 1$. We may assume $a_1 \notin \mathbb{Z}[\frac{1}{2}]$, since otherwise the statement is trivial by completeness of the phase-free calculus. In that case any number in $\mathbb{Z}[\frac{1}{2}, a_1]$ can be uniquely written as $z_1 + a_1 z_2$ for $z_1, z_2 \in \mathbb{Z}[\frac{1}{2}]$. We can now do all the steps we described before for $a = i$, translating a diagram D containing an arbitrary number of the H-box with label a_1 into a diagram D' in the $\mathbb{Z}[\frac{1}{2}]$ fragment which just requires a single input of the a_1 H-box: $D = D'[[0\rangle + a_1|1\rangle] = D'[[0\rangle] + a_1 D'[[1\rangle]$. If we then have an equality between two diagrams D_1 and D_2 in the $\mathbb{Z}[\frac{1}{2}, a_1]$ fragment, we get $D'_1[[0\rangle] + a_1 D'_1[[1\rangle] = D'_2[[0\rangle] + a_1 D'_2[[1\rangle]$. Because each of the component diagrams only contains elements from $\mathbb{Z}[\frac{1}{2}]$, and a decomposition $z_1 + a_1 z_2$ is unique, this equation can only hold if each of the two separate components are equal. We hence again get two equalities $D'_1[[0\rangle] = D'_2[[0\rangle]$ and $D'_1[[1\rangle] = D'_2[[1\rangle]$, which allows us to conclude that $D'_1 = D'_2$ with the wire left open. As D'_1 and D'_2 are diagrams in the smaller fragment for which we have completeness, we can rewrite one into the other. Plugging in the H-box with label a_1 then gives us a diagrammatic proof of equality.

The induction step follows similarly: we just observe that if $a_i \notin \mathbb{Z}[\frac{1}{2}, a_1, \dots, a_{i-1}]$ that there is then again a unique way to write a number in the ring $\mathbb{Z}[\frac{1}{2}, a_1, \dots, a_{i-1}, a_i]$ as $z_1 + a_i z_2$ where $z_1, z_2 \in \mathbb{Z}[\frac{1}{2}, a_1, \dots, a_{i-1}]$. We can then again use the catalysis and scalar introduction rule to rewrite the diagram into the form where we can use the completeness of the smaller fragment. \square

B Small angle rotations, adders and catalysis

Catalysis is not just interesting from a theoretical viewpoint, allowing you to prove the universality and completeness of certain gate sets or generators, it is also a *practically* useful tool. In this section we will see how catalysis can be used to derive an efficient way to implement small angle rotations.

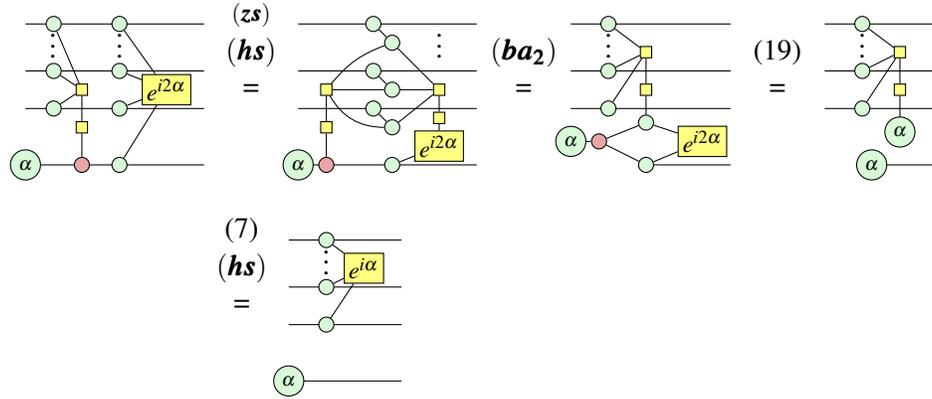
To do that we first need to generalise Eq. (9) to allow us to implement *controlled*-phase gates. To see how this works it will be helpful to first write Eq. (9) in circuit notation:

$$\begin{array}{c} \boxed{Z(\alpha)} \\ |Z(\alpha)\rangle \text{---} \end{array} = \begin{array}{c} \bullet \quad \bullet \\ |Z(\alpha)\rangle \text{---} \oplus \boxed{Z(2\alpha)} \end{array} \quad (25)$$

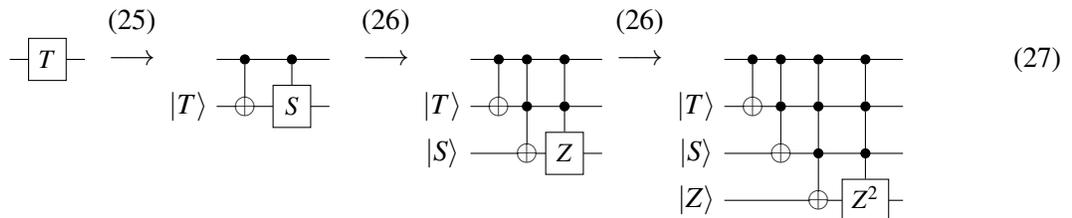
Here we wrote a slightly more general circuit where we replace the T and controlled- S gates with $Z(\alpha)$ and controlled- $Z(2\alpha)$ gates. As a shorthand we write $|Z(\alpha)\rangle := Z(\alpha)|+\rangle$ as a generalisation of $|T\rangle = T|+\rangle$. Since this is a circuit equality that holds on the nose (with a correct global phase), it should continue to hold when we add additional control wires:

$$\begin{array}{c} \bullet \\ \vdots \\ \bullet \\ \boxed{Z(\alpha)} \\ |Z(\alpha)\rangle \text{---} \end{array} = \begin{array}{c} \bullet \quad \bullet \\ \vdots \\ \bullet \quad \bullet \\ |Z(\alpha)\rangle \text{---} \oplus \boxed{Z(2\alpha)} \end{array} \quad (26)$$

We can prove this is correct using ZH:



Because we can apply catalysis equally well to controlled phases, we can start iterating the procedure producing bigger and bigger controlled-phase gates, where the phase being controlled is also increasingly large. For instance, if we want to implement a T gate, we can do the following:



Here in the last step we are left with a controlled Z^2 operation. But since $Z^2 = \text{id}$ this does not do anything and we can remove it. So at this point we can stop the iteration of the catalysis. We see then that we can implement a T gate just using multiple-controlled Toffoli gates, if we have the right catalysis states lying around. This procedure works to implement any $Z(2\pi/2^k)$ gate: we then get a ladder of k Toffoli gates. This implements a controlled-decrementer circuit that decreases the value of an n -bit number by 1, controlled on the top wire. By making a ladder of these controlled-decrementers we implement a subtraction circuit that maps $|a, b\rangle \mapsto |a, b - a\rangle$ for some n -bit numbers a and b . For this reason, when we apply a subtraction circuit to a collection of catalysis states, this implements phase gates on on the

top qubits:

$$\begin{array}{c}
 |T\rangle \\
 |S\rangle \\
 |Z\rangle
 \end{array}
 \text{Sub}
 \quad := \quad
 \begin{array}{c}
 \bullet \\
 \bullet \\
 \bullet
 \end{array}
 \begin{array}{c}
 |T\rangle \\
 |S\rangle \\
 |Z\rangle
 \end{array}
 \begin{array}{c}
 -1 \\
 -1 \\
 -1
 \end{array}
 \quad \stackrel{(27)}{=} \quad
 \begin{array}{c}
 T \\
 \bullet \\
 \bullet
 \end{array}
 \begin{array}{c}
 |T\rangle \\
 |S\rangle \\
 |Z\rangle
 \end{array}
 \begin{array}{c}
 -1 \\
 -1
 \end{array}
 \quad \stackrel{(27)}{=} \quad
 \begin{array}{c}
 T \\
 S \\
 Z \\
 -1
 \end{array}
 \quad \stackrel{(27)}{=} \quad
 \begin{array}{c}
 T \\
 S \\
 Z
 \end{array}
 \quad (28)$$

An adder can be implemented quite efficiently, so we transform Eq. (28) slightly, so that it uses an adder instead of a subtracter. By taking Eq. (28) and composing both sides on the right by $\text{Add} = \text{Sub}^\dagger$, and on the left by $(T \otimes S \otimes Z)^\dagger$. After cancelling with the adjoints we are then left with the following equation:

$$\begin{array}{c}
 T^\dagger \\
 S^\dagger \\
 Z^\dagger \\
 |T\rangle \\
 |S\rangle \\
 |Z\rangle
 \end{array}
 \quad = \quad
 \begin{array}{c}
 \text{Add} \\
 |T\rangle \\
 |S\rangle \\
 |Z\rangle
 \end{array}
 \quad (29)$$

We showed the construction here for 3 bits, but this works for any number of bits n , in which case the smallest phase we implement is $Z(2\pi/2^n)$.

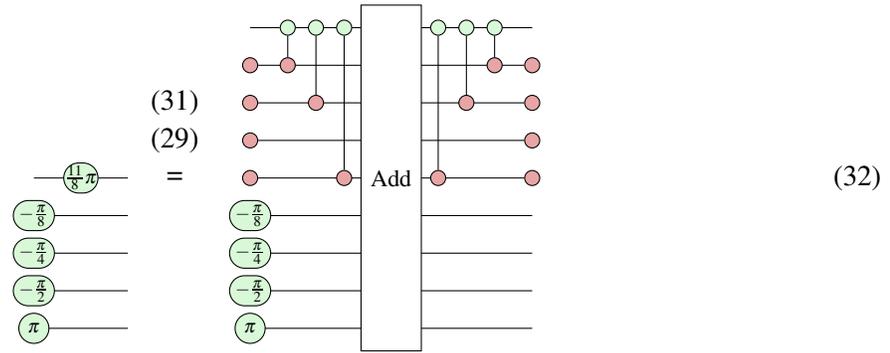
Such a series of parallel phases is not that useful, but by using ancillae we can make them work on the same qubit. First, we can transfer the application of a phase gate to a zeroed ancilla:

$$\begin{array}{c}
 \bullet \\
 \bullet
 \end{array}
 \begin{array}{c}
 \alpha \\
 \alpha
 \end{array}
 \quad \stackrel{(zs)}{=} \quad
 \begin{array}{c}
 \bullet \\
 \bullet
 \end{array}
 \begin{array}{c}
 \alpha \\
 \alpha
 \end{array}
 \quad \stackrel{(id)}{=} \quad
 \begin{array}{c}
 \bullet \\
 \bullet
 \end{array}
 \begin{array}{c}
 \alpha \\
 \alpha
 \end{array}
 \quad \stackrel{(zs)}{=} \quad
 \begin{array}{c}
 \alpha
 \end{array}
 \quad (30)$$

Now when we have a complicated phase, we can decompose it into its components, and put each of these onto its own ancilla. Suppose for instance we want to implement the phase $Z(\frac{11}{8}\pi)$. We can then write 11 bitwise as 1011 so that $Z(\frac{11}{8}\pi) = Z(2\pi/2^4(2^3 + 2^1 + 2^0))$. We can then put each of these component phases onto their own ancilla to get:

$$\begin{array}{c}
 \frac{11}{8}\pi
 \end{array}
 \quad = \quad
 \begin{array}{c}
 \pi \\
 \frac{\pi}{4} \\
 \frac{\pi}{8}
 \end{array}
 \quad = \quad
 \begin{array}{c}
 \bullet \\
 \bullet \\
 \bullet \\
 \bullet
 \end{array}
 \begin{array}{c}
 \frac{\pi}{8} \\
 \frac{\pi}{4} \\
 \frac{\pi}{2} \\
 \pi
 \end{array}
 \quad (31)$$

We have here also added a zeroed ancilla that gets a $Z(\frac{\pi}{2})$ applied that does nothing. We need this qubit to complete the pattern: we see then that we get the right shape needed to use Eq. (29). However, note that Eq. (29) has adjoint phases, instead of the actual phases we need. There are multiple ways we can deal with this. One way is to realise that for phase gates, the adjoint is the conjugate: $T^\dagger = \overline{T}$. Hence, if we take the conjugate of both sides of Eq. (29) we do get the right phases. Since the Adder is a real matrix, this stays the same, but the states needed for the catalysis also flip: $|\overline{T}\rangle = |T^\dagger\rangle$. We then have everything we need to produce the circuit we are after:



This is the construction that is presented in [11]. There it was proven correct by arguing about the interaction between the quantum Fourier transform and addition. In comparison, the construction we present here is more bottom-up and only uses elementary facts about quantum circuits and catalysis.