arXiv:2408.02778v2 [quant-ph] 24 Nov 2025

Polynomial-Time Classical Simulation of Hidden Shift Circuits via Confluent Rewriting of Symbolic Sums

Matthew Amy and Lucas Shigeru Stinchcombe

School of Computing Science, Simon Fraser University, Canada

Implementations of Roetteler's shifted bent function algorithm have in recent years been used to test and benchmark both classical simulation algorithms and quantum hardware. These circuits have many favorable properties, including a tunable amount of non-Clifford resources and a deterministic output, and moreover do not belong to any class of quantum circuits that is known to be efficiently simulable. We show that this family of circuits can in fact be simulated in polynomial time via symbolic path integrals. We do so by endowing symbolic sums with a confluent rewriting system and show that this rewriting system suffices to reduce the circuit's path integral to the hidden shift in polynomial time. We hence resolve an open conjecture about the efficient simulability of this class of circuits.

1 Introduction

The classical simulation of quantum circuits is an important problem, both for practical issues of testing and benchmarking quantum circuits, devices, and algorithms, as well as for quantifying and elucidating the nature of quantum speedup. One of the most celebrated results in the classical simulation of quantum circuits showed that circuits comprised of only Clifford gates are polynomial-time simulable. On the other hand, given circuits over a universal gate set such as the well-known Clifford+T basis, an efficient, generic classical simulation algorithm would imply $\mathbf{BQP} = \mathbf{P}$, carrying with it enormous impacts on both classical and quantum computation.

In a recent series of work [7–9], it was shown that given a small number of non-Clifford resources, classical simulation of a quantum circuit can be performed in polynomial time. In this sense, non-Clifford resources can be seen as drivers of quantum advantage. The key observation was that gate teleportation reduces the simulation of a Clifford+T circuit to the simulation of a Clifford circuit with a non-Clifford resource state proportional in size to the number of non-Clifford gates. This resource state can then be decomposed as a sum of k stabilizer states and simulation can then proceed by performing k independent, polynomial-time Clifford simulations. The decomposition of resource states into sums of stabilizer states has since been extensively studied [23, 24, 32, 35], but the best-known upper bounds remain exponential in the number of non-Clifford gates to be implemented by teleportation.

Despite its exponential scaling in the number of non-Clifford resources, stabilizer decompositions have shown favorable performance in practical simulations of large circuits. In [8], stabilizer decomposition-based simulation was performed on a class of simulation benchmarks derived from an oracle algorithm due to Roetteler [36]. Roetteler's algorithm, which computes a bitstring s hidden in a pair of Maiorana-McFarland bent functions provided as oracles to the quantum computer, was used as a benchmark for two reasons; one that it is a deterministic algorithm, and secondly that it allowed for the precise control over the number of non-Clifford gates since the non-oracle part of the circuit is strictly Clifford. To control the number of non-Clifford resources, instances of the algorithm were generated using degree 3 constructions of Maiorana-McFarland bent functions, which correspond directly to implementations over the $\{Z, CZ, CCZ\}$, and hence Clifford+T, gate set. It can be noted that such instances *require* implementation over a universal gate set, and hence do not fall under known classically simulable classes like Clifford [1], CNOT-dihedral [13], or commuting circuits [27].

Following its use in [8], this class of implementations of Roetteler's algorithm has been used extensively to benchmark classical simulation methods [4, 7, 11, 20–22, 29, 33]. Through these works however, empirical evidence that it is not a good choice of benchmark in general began to surface. In [4], instances of the

benchmark were simulated deterministically in a fraction of the time by rewriting the circuit's (symbolic) path integral or *path sum*. Benchmark parameters which took hours to simulate probabilistically using stabilizer decompositions were reduced to the hidden shift in seconds via rewriting. Likewise, in [22], which used tensor contraction methods in the closely related ZX-calculus [12], observed that it would contract completely without any need for stabilizer decomposition. A year earlier the same effect was observed in [11], and was conjectured to be polynomial time simulable.

In this paper, we show that the family of hidden shift circuits from [8] is indeed polynomial-time simulable. Our methods are based on static rewriting of the circuit's path sum. We show that by restricting the rewriting rules of [4] to control the degree of polynomials of the circuit path sum gives a *confluent* rewrite system which reduces the path sum of the hidden shift circuit family to the shift in provably polynomial-time. Combining simplification of the path sum with explicit evaluation yields a complete simulation method, thereby answering the conjecture by Codsi in [11] in the affirmative. We further show that simulation remains polynomial-time for a larger class of circuit instances based on Maiorana-McFarland bent functions of bounded degree. Conversely, our simulation methods fail to be tractable for functions of unbounded degree, or for implementations of the oracle which are exponentially smaller than the algebraic (polynomial) normal form of the given bent functions.

The rest of the paper is as follows: Section 2 introduces Roetteler's shifted bent function algorithm, along with the class of concrete circuit implementations we are interested in. In Section 3, we introduce path sums, a rewrite system, and accompanying simulation algorithms. In Section 4, we show that the rewrite system is confluent, and in Section 5 we give sufficient conditions for when the simulation algorithms are efficient, and show that hidden shift circuits satisfy this condition. Finally, Section 6 concludes the paper.

2 Hidden Shift Circuits and the Shifted Bent Function Problem

We begin with an overview of Roetteler's algorithm for the shifted bent function problem, devised in [36] as a problem that gives oracle separations between the complexity classes BQP and P. Briefly, the problem involves an oracle which gives access to two Boolean functions f, g such that $g(x) = f(x + s)$. This oracle is said to "hide" the bitstring s, and the problem is to compute the hidden bitstring by querying f and g. It was shown that for bent functions taken from a family of Boolean functions known as the *Maiorana-McFarland* family, exponentially many classical queries but only linearly many quantum queries are required to compute the shift. Roetteler further showed that if the oracle also gives access to the *dual* of f, a weaker separation of $O(n)$ classical to $O(1)$ quantum queries is obtained, this time by a *deterministic* quantum algorithm. Standard methods [17] then suffice to extend this to a super-polynomial separation. The hidden shift circuit family used to benchmark simulation algorithms is an implementation of the deterministic hidden shift algorithm, where the oracle is "de-oraclized" by giving a concrete circuit implementation. We will briefly introduce the algorithm as presented in [36], and give a construction of the benchmark which captures and generalizes those used in [7, 11, 20–22, 29, 33].

We first recall some theory from the Fourier analysis of Boolean functions. We will use both Boolean groups $(\mathbb{F}_2, +)$ and $(\{1, -1\}, \cdot)$ as convenient. They are isomorphic by the mappings

$$(\mathbb{F}_2, +) \underset{(-1)^{(\cdot)}}{\overset{\frac{1-(\cdot)}{2}}{\rightleftarrows}} (\{1, -1\}, \cdot)$$

Recall also that the set of pseudo-Boolean functions $\{f : \mathbb{F}_2^n \to \mathbb{R}\}$ forms a 2^n-dimensional real vector space under the usual notion of function addition. We can endow the vector space with an inner product defined as follows

$$\langle f, g \rangle = \mathbb{E}[fg] = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} f(x)g(x)$$

Observe that the functions $\{\chi_s \mid s \in \mathbb{F}_2^n\}$ where $\chi_s(x) = (-1)^{s \cdot x}$ form a basis of the pseudo-Boolean functions, which we call the parity basis. Indeed it can be seen to be a basis since they are orthonormal and 2^n-many. The Fourier transform of a pseudo-Boolean function f is then defined as

$$\widehat{f}(s) = \langle f, \chi_s \rangle$$

so that f is represented over $\{\chi_s\}$ by

$$f(x) = \sum_{s \in \mathbb{F}_2^n} \widehat{f}(s)\chi_s$$

and the values $\widehat{f}(s)$ are called the Fourier coefficients of f. Now let $f : \mathbb{F}_2^n \to \{1, -1\}$ be a Boolean function. The function f is called a *bent* function if its Fourier coefficients have the same magnitude.

Definition 1. *Let $f : \mathbb{F}_2^n \to \{1, -1\}$ be a Boolean function. Then f is bent if for all $s \in \mathbb{F}_2^n$, $|\widehat{f}(s)| = \frac{1}{\sqrt{2^n}}$.*

Definition 1 essentially states that the Fourier transform $\widehat{f}$ is also a Boolean function up to a scaling. For a bent function f, its *dual* is a Boolean function defined as $\overline{f}(x) = \sqrt{2^n}\widehat{f}(x)$. Note that the dual of $\overline{f}$ is again f. The results of [36] pertain to a construction of bent functions called the Maiorana-McFarland bent functions whose definition we reproduce here.

Proposition 1 (Maiorana-McFarland Bent function)**.** *Let $f : \mathbb{F}_2^{2n} \to \mathbb{F}_2$ be a Boolean function such that*

$$\begin{aligned} f(x, y) =& \langle x, \pi(y)\rangle + f_0(y) \\ & \textit{where } f_0 : \mathbb{F}_2^n \to \mathbb{F}_2 \textit{ and } \pi \in \mathcal{S}_{2^n} \textit{ is a permutation.} \end{aligned}$$

Then f is bent and has the dual function $\overline{f}(x, y) = \langle \pi^{-1}(x), y\rangle + f_0(\pi^{-1}(x))$. We call f a Maiorana-McFarland bent function, and the set of all such functions $\mathcal{M}$.

Proof. By direct computation of the Fourier coefficient

$$\begin{aligned} \widehat{f}(s, t) &= \frac{1}{2^{2n}} \sum_{x,y \in \mathbb{F}_2^n} (-1)^{\langle x, \pi(y)+s\rangle + f_0(y) + \langle y, t\rangle} \\ &= \frac{1}{2^{2n}} \sum_{x,y \in \mathbb{F}_2^n} (-1)^{\langle x, y+s\rangle + f_0(\pi^{-1}(y)) + \langle \pi^{-1}(y), t\rangle} \\ &= \frac{1}{2^n} (-1)^{f_0(\pi^{-1}(s)) + \langle \pi^{-1}(s), t\rangle} \\ &= \frac{1}{2^n} (-1)^{\langle s, \pi(t)\rangle + f_0(\pi^{-1}(s))} \end{aligned}$$

where the step from line 2 to line 3 follows from the fact that the summands where $y \neq s$, occur with both positive and negative factor 2^{n-1} times each and hence cancel, while the summands where $y = s$ occur with positive factor exactly 2^n times. Thus we have $|\widehat{f}(s, t)| = \frac{1}{2^n}$, and its dual is as desired. □

Note that the permutation $\pi \in S_{2^n}$ is any bijection $\pi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ which will be restricted when we discuss the hidden shift circuit family when considered in the context of benchmarking classical simulations. Now we can formally define the shifted bent function problem, which is the task of computing a shift $s \in \mathbb{F}_2^n$ hidden by a bent function.

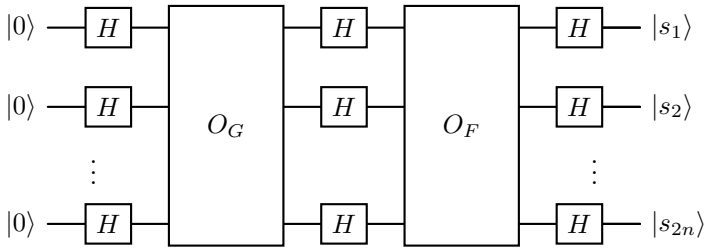
Definition 2 (Shifted Bent Function Problem)**.** *For a given $s \in \mathbb{F}_2^n$, and a bent function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, set $g : x \mapsto f(x + s)$. With oracle access to f, g, compute the shift s where oracles are implemented by following unitaries*

$$O_G : |x\rangle \mapsto (-1)^{g(x)} |x\rangle , \quad O_F : |x\rangle \mapsto (-1)^{\overline{f}(x)} |x\rangle$$

The phase oracles above can equivalently be constructed from oracles computing in the computational basis with the usual method of phase kickback via the state $|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$.

Roetteler [36] showed that for any bent function f, the shifted bent function problem as defined above can be solved deterministically with two oracle calls on a quantum computer. On the other hand, in the classical case, Roetteler also showed that for a Maiorana-McFarland bent function f, classically $\Theta(n)$ queries are necessary and sufficient to compute the shift. The quantum algorithm solving the shifted

bent function problem is given by the following circuit.



(1)

In [7], Bravyi and Gosset benchmarked their simulation algorithm on instances of the quantum circuit 1 for randomly generated Maiorana-McFarland bent functions where the permutation $\pi \in S_{2^n}$ was fixed to be the identity. In addition, the bent functions were of degree at most 3 where the *degree* is the polynomial degree of the Boolean function expressed as a multilinear polynomial. The oracles were implemented over the gate set $\{X, Z, CZ, CCZ\}$, which were then compiled to Clifford+T circuits via $CCZ = (I \otimes I \otimes H)\text{Toff}(I \otimes I \otimes H)$ and a gadget implementing the Toffoli gate Toff over Clifford+T with 4 T gates due to Jones [19]. The choice to use the measurement-based implementation of [19] was motivated by the lower T-count than direct implementations, as their simulation algorithm scales exponentially in the number of T gates. The same construction of the circuit has since been used to benchmark a variety of simulation methods [7, 11, 20–22, 29, 33], as well as benchmarking quantum hardware [25, 39]. More generally, we can define instances of the hidden shift circuit for functions of degree $m + 1$ by extending the gate set to include multiply-controlled Z gates as well as allowing permutations $\pi \in S_n$, which is to say bijections $\pi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ achieved by permuting bits. Note for such permutations, we have the following equality of the inner product function $\langle \cdot, \cdot \rangle : \mathbb{F}_2^{2n} \to \mathbb{F}_2$.

$$\langle \pi^{-1}(x), y \rangle = \langle x, \pi(y) \rangle$$

Definition 3 (Hidden Shift Circuit Family)**.** *Let $\pi \in \mathcal{S}_n$ be a permutation, $f_0 : \mathbb{F}_2^n \to \mathbb{F}_2$ be a Boolean function, and $s = s_1 s_2 \in \mathbb{F}_2^{2n}$ be a bitstring. A hidden shift circuit $C_{(\pi, f_0, s)}$ for the triple (π, f_0, s) is any circuit of the form 1 on $2n$ qubits where oracles*

$$O_G : |x, y\rangle \mapsto (-1)^{\langle x + s_1, \pi(y + s_2)\rangle + f_0(y + s_2)} |x, y\rangle$$
$$O_F : |x, y\rangle \mapsto (-1)^{\langle x, \pi(y)\rangle + f_0(\pi^{-1}(x))} |x, y\rangle$$

are implemented over the gate set $\left\{X, SWAP, Z, CZ, \cdots, C^{(m)}Z\right\}$. In particular this implies that f_0 has degree at most $m + 1$.

The fact that m is a fixed positive integer will be essential in the proof of polynomial-time classical simulation as polynomials of bounded degree have only polynomially-many terms in the number of variables when expressed in their multilinear form. Previous instances of the circuit construction [7, 11, 20–22, 29, 33]were restricted to the Bravyi-Gosset parameters $m = 2$ and $\pi = \text{id}$. Note that $C^{(m)}Z$ is non-Clifford whenever $m \geq 2$, and in particular requires more non-Clifford resources as m increases. When $\deg(f_0) \geq 3$, Roetteler's algorithm moreover *requires* a non-linear Boolean operation, i.e. a CCZ or CCX gate, in order to implement the oracles. As $\{H, CCX\}$ is universal for quantum computing [2], *any* circuit implementation must be over a universal gate set. The circuit family defined above thus constitutes a family of non-trivial circuits which (1) generates an exponential-size superposition of basis states, (2) generates an intermediate state which is highly-entangled, (3) uses interference to produce the correct result, and (4) cannot be implemented over a classically-simulable set of quantum gates. These features are known [28] to be necessary (but not sufficient) conditions for quantum speedup, as together they preclude polynomial-time classical simulation by standard methods.

Correctness of Roetteler's algorithm can be shown by direct calculation, which corresponds to a sequence of simplifications due to interference yielding the final shift s. While simpler proofs are possible, the calculation below is effectively automatable as we show in following sections. Writing g for the s-shifted f which is computed by O_G, $\overline{f}$ again for the dual of f computed by O_F, and letting $U_{(\pi, f_0, s)}$ be

the unitary implemented by circuit $C_{(\pi,f_0,s)}$ we have

$$\begin{aligned}
U_{(\pi,f_0,s)}\left|0\right\rangle &= \frac{1}{2^{3n}} \sum_{x,y,z\in\mathbb{F}_2^{2n}} (-1)^{g(x)+\langle x,y\rangle+\overline{f}(y)+\langle y,z\rangle} \left|z\right\rangle \\
&= \frac{1}{2^{3n}} \sum (-1)^{\langle x_1+s_1,\pi(x_2+s_2)\rangle+f_0(x_2+s_2)+\langle x,y\rangle+\overline{f}(y)+\langle y,z\rangle} \left|z\right\rangle \\
&= \frac{1}{2^{3n}} \sum (-1)^{\langle x_1,\pi(x_2+s_2)+y_1\rangle+\langle s_1,\pi(x_2+s_2)\rangle+f_0(x_2+s_2)+\langle x_2,y_2\rangle+\overline{f}(y)+\langle y,z\rangle} \left|z\right\rangle \\
&= \frac{1}{2^{2n}} \sum (-1)^{\langle s_1,\pi(x_2+s_2)\rangle+f_0(x_2+s_2)+\langle x_2,y_2\rangle+\overline{f}(\pi(x_2+s_2),y_2)+\langle\pi(x_2+s_2),z_1\rangle+\langle y_2,z_2\rangle} \left|z\right\rangle
\end{aligned}$$

where the last equality follows because $\sum_{x_1,y_1\in\mathbb{F}_2^n}(-1)^{\langle x_1,\pi(x_2+s_2)+y_1\rangle} = 2^n$ if $y_1 = \pi(x_2+s_2)$, and 0 otherwise, and we leave the variables summed over implicit. Expanding $\overline{f}(\pi(x_2+s_2),y_2)$ we get $\langle x_2+s_2,y_2\rangle + f_0(x_2+s_2)$ and in particular the two $f_0(x_2+s_2)$ terms cancel. The remaining calculation follows below:

$$\begin{aligned}
U_{(\pi,f_0,s)}\left|0\right\rangle &= \frac{1}{2^{2n}} \sum (-1)^{\langle s_1,\pi(x_2+s_2)\rangle+\langle x_2,y_2\rangle+\langle x_2+s_2,y_2\rangle+\langle\pi(x_2+s_2),z_1\rangle+\langle y_2,z_2\rangle} \left|z\right\rangle \\
&= \frac{1}{2^{2n}} \sum (-1)^{\langle s_1+z_1,\pi(x_2+s_2)\rangle+\langle y_2,z_2+s_2\rangle} \left|z\right\rangle \\
&= \left|s_1,s_2\right\rangle
\end{aligned}$$

where again the last equality follows by noting that $\sum_{y_1,z_2\in\mathbb{F}_2^n}(-1)^{\langle y_2,z_2+s_2\rangle}$ destructively interferes whenever $z_2 \neq s_2$, and likewise $\sum_{x_2,z_1\in\mathbb{F}_2^n}(-1)^{\langle s_1+z_1,\pi(x_2+s_2)\rangle}$ destructively interferes when $z_1 \neq s_1$.

It is worth noting that for the specific constructions of hidden shift circuits used in [8], there is a similar polynomial length proof of correctness by circuit equalities, which we give in Appendix B. The circuit equality proof relies not only on the structure of the construction, but also on a particular sequence of circuit rewrites which does not in general correspond to an effective simulation — or even circuit simplification — method. Instead, we codify the above proof using path sums, where we find that the algebraic equalities employed in the above proof correspond to a *confluent* rewrite system, in that every sequence of rewrites results in the same normal form. Thus the above proof will witness that a simulation algorithm which arbitrarily simplifies the codified expression using these algebraic equalities is guaranteed to yield the hidden shift, including for more general cases where the hidden shift circuit is not implemented literally as constructed in [8].

3 Path Sums

We now turn our attention to path sums, which form the basis of our simulation methods. Path sums, corresponding to a discretization of Feynman's path integral, are symbolic representations of linear operators as finite sums of parameterized linear operators over variables ranging in $\mathbb{F}_2$. The analysis of circuits via discretized path integrals, colloquially termed the *sum-over-paths* technique, has seen application over the years in a great deal of contexts, notably proving classical complexity results [14,26]. In [4] the sum-over-paths technique was formalized as a mathematical object — called a *path sum* — and equipped with a rewriting system which allowed it to be statically reduced without explicitly expanding the sum. The path sum has since been extended in various ways [4–6,37,38]; these various incarnations as rewrite systems have been shown to be complete for the Clifford [37] and Toffoli-Hadamard [38] fragments, as well as general $\mathcal{R}$-linear operators with *unbalanced* amplitudes [5]. Beyond completeness results, path sums parameterized on classical data such as natural numbers, called *higher order path sums*, were introduced in [10] to verify families of circuits by deductive program verification.

Path sums are represented concretely as collections of polynomials giving the phase, inputs, and outputs along a *path* — a particular assignment to the variables over which the sum is defined. We restrict our attention to polynomials over $\mathbb{F}_2$, corresponding to operators over the universal Toffoli-Hadamard gate set, as this simplification has the benefit that all polynomials are of the same polynomial ring over $\mathbb{F}_2$. This fragment was previously studied and denoted $\mathbf{SOP}[\frac{1}{2}]$ in [38], where a rewrite system was given which was *complete* when viewed as an equational theory, but not *confluent*.

Definition 4 (Boolean Path Sum)**.** *A (Boolean) path sum is an expression of the form*

$$s\sum_{V}(-1)^{P}\left|O_1,\ldots,O_m\right\rangle\left\langle I_1,\ldots,I_n\right|$$

where

1. $s\in\mathbb{C}$ *is a complex scalar*
2. V *is a set of indeterminates over* $\mathbb{F}_2$
3. $P, O_1,\ldots,O_m, I_1,\ldots,I_n$ *are multivariate polynomials in* $\mathbb{F}_2[V]/I_V$

where I_V *is the ideal generated by* v^2-v *for all* $v\in V$*.*

Recall that taking the quotient by the ideal generated by v^2-v expresses the fact that v^2 (and hence all positive powers of v) are identified with v. While we use the notation $s\sum_V(-1)^P\left|O_1,\ldots,O_m\right\rangle\left\langle I_1,\ldots,I_n\right|$ to express the intuition of a path sum as a linear operator, a path sum is equally well-presented by the tuple

$$(V, s, P, O := (O_1,\ldots,O_m), I := (I_1,\ldots,I_n)).$$

The set of all path sums we will call **PS**, and polynomials which are elements of $\mathbb{F}_2[V]/I_V$ we will refer to as *Boolean polynomials*. Note that Boolean polynomials may be represented uniquely as *multilinear polynomials* — that is polynomials which are degree ≤ 1 in any given variable. For an expression $A\in\mathbf{PS}$, we will use the convention that the various components of the expression are labeled as follows.

$$A = s_A\sum_{V_A}(-1)^{P_A}\left|O_A\right\rangle\left\langle I_A\right|$$

Furthermore, we will call $A:|I_A|\to|O_A|$, the signature of A. All path sums will have an interpretation as a linear operator in the following way by computing the sum over the variables V_A as they range in $\mathbb{F}_2$. In particular for $|V_A|=k$, $|I_A|=n$, and $|O_A|=m$ we may view P_A, I_A, O_A as functions on $v\in\mathbb{F}_2^k$ where

$$\begin{aligned} P_A &: \mathbb{F}_2^k\to\mathbb{F}_2\\ I_A &: \mathbb{F}_2^k\to\mathbb{F}_2^n\\ O_A &: \mathbb{F}_2^k\to\mathbb{F}_2^m \end{aligned}$$

respectively, the corresponding linear operator is $eval(A):\mathbb{C}^{2^n}\to\mathbb{C}^{2^m}$ defined by

$$eval(A) = s_A\sum_{v\in\mathbb{F}_2^k}(-1)^{P_A(v)}\left|O_A(v)\right\rangle\left\langle I_A(v)\right| \tag{2}$$

As path sums correspond to symbolic representations of linear operators, we can define composition and the tensor product of path sum expressions in the following way.

Definition 5 (Composition, Tensor product, Identity, Zero, Adjoint)**.** *Without loss of generality we assume in the following that for any* $A, B\in\mathbf{PS}$ *that* $Var(A)\cap Var(B)=\emptyset$*, and that* $y_i\notin Var(A)\cup Var(B)$ *for any* i*. We define*

1. *for* $A:m\to n$*,* $B:k\to m\in\mathbf{PS}$*, composition* $A\circ B:k\to n$

$$A\circ B = \frac{s_A s_B}{2^m}\sum_{V_A\cup V_B\cup\{y_1,\ldots,y_m\}}(-1)^{P_A+P_B+\sum_{i=1}^m y_i(O_{B,i}+I_{A,i})}\left|O_A\right\rangle\left\langle I_B\right|$$

2. *for* $A:m\to n, B:k\to l\in\mathbf{PS}$ *the tensor product* $A\otimes B:m+k\to n+l$

$$A\otimes B = s_A s_B\sum_{V_A\cup V_B}(-1)^{P_A+P_B}\left|O_A O_B\right\rangle\left\langle I_A I_B\right|$$

3. for each $n \in \mathbb{Z}_{\geq 0}$ the identity, I_n

$$I_n = \sum_{\{y_1,\ldots,y_n\}} (-1)^0 \left|y_1,\ldots,y_n\right\rangle \left\langle y_1,\ldots,y_n\right|$$

4. for each $m, n \in \mathbb{Z}_{\geq 0}$ the zero, $0_{m,n}$ with signature $m \to n$.

$$0_{m,n} = 0 \sum_{\emptyset} (-1)^0 \left|\underbrace{0,\ldots,0}_{n}\right\rangle \left\langle\underbrace{0,\ldots,0}_{m}\right|$$

5. for $A : m \to n \in \boldsymbol{PS}$ the adjoint $A^\dagger : n \to m$

$$A^\dagger = s_A^* \sum_{V_A} (-1)^{P_A} \left|I_A\right\rangle \left\langle O_A\right|$$

The composition, tensor products, and adjoint on path sums can be seen to be sound with respect to *eval*, in the sense that for arbitrary $A, B \in \mathbf{PS}$

$$\begin{aligned} eval(A \otimes B) &= eval(A) \otimes eval(B) \\ eval(A \circ B) &= eval(A) \circ eval(B) \\ eval(A^\dagger) &= eval(A)^\dagger \end{aligned}$$

whenever they are well-defined with respect to their signatures. Indeed, for composition notice that the term $y_i(O_{B,i} + I_{A,i})$ asserts that $O_{B,i} = I_{A,i}$ for all i, since whenever $O_{B,i} + I_{A,i} = 1$, summing over y_i destructively interferes, and constructively interferes when $O_{B,i} + I_{A,i} = 0$. Note also that a sum over the empty set corresponds to the absence of a sum, i.e.

$$eval(s \sum_{\emptyset} (-1)^c \left|d_1,\ldots,d_n\right\rangle \left\langle b_1,\ldots,b_m\right|) = s(-1)^c \left|d_1,\ldots,d_n\right\rangle \left\langle b_1,\ldots,b_m\right|$$

where $c, d_i, b_j \in \mathbb{F}_2$. Having equipped path sums with identities and parallel and sequential composition, we may interpret circuits $C \in (\mathcal{G}, I, \otimes, \cdot)$ over a gate set $\mathcal{G}$ by giving an interpretation of the gates in $\mathcal{G}$. Given a mapping $[\![\cdot]\!] : \mathcal{G} \to \mathbf{PS}$ such that for $g \in \mathcal{G}$ with signatures $g : m \to n$ we say that $[\![\cdot]\!]$ is well-formed if $[\![g]\!] : m \to n$, i.e. the signatures of $[\![g]\!]$ match. We extend well-formed interpretations $[\![\cdot]\!]$ to circuits $C \in (\mathcal{G}, I, \otimes, \cdot)$ in the obvious way:

$$\begin{aligned} [\![I_n]\!] &= I_n \\ [\![g]\!] &= [\![g]\!] \\ [\![g_2 \cdot g_1]\!] &= [\![g_2]\!] \circ [\![g_1]\!] \\ [\![g_1 \otimes g_2]\!] &= [\![g_1]\!] \otimes [\![g_2]\!] \end{aligned}$$

Given linear operators $U_g : \mathbb{C}^{2^m} \to \mathbb{C}^{2^n}$ for each $g : m \to n \in \mathcal{G}$, if $eval([\![g]\!]) = U_g$ for every $g \in \mathcal{G}$, it follows that $eval([\![C]\!]) = U_C$ for any circuit C over the gate set $\mathcal{G}$ — that is, $[\![C]\!]$ is a sound interpretation of C. We define a sound interpretation of $\mathcal{G} = \left\{H, X, \mathrm{SWAP}, Z, CZ, \cdots, C^{(m)}Z\right\}$ in **PS** as follows

$$\begin{aligned} [\![H]\!] &= \frac{1}{\sqrt{2}} \sum_{\{x,y\}} (-1)^{xy} \left|y\right\rangle \left\langle x\right| \\ [\![X]\!] &= \sum_{\{x\}} \left|x+1\right\rangle \left\langle x\right| \\ [\![C^{(m)}Z]\!] &= \sum_{\{x_1,\ldots x_m,y\}} (-1)^{y \prod_{i=1}^m x_i} \left|x_1,\ldots x_m,y\right\rangle \left\langle x_1,\ldots x_m,y\right| \\ [\![SWAP]\!] &= \sum_{\{x,y\}} \left|y,x\right\rangle \left\langle x,y\right| \end{aligned}$$

Example 2. *The following path sum interpretations for two Hadamard gates in sequence and in parallel illustrate the path sum composition and tensor rules respectively and can be seen to evaluate to their standard matrix interpretations.*

$$\llbracket \text{—}\boxed{H}\text{—}\boxed{H}\text{—} \rrbracket = \left(\frac{1}{\sqrt{2}}\sum_{\{x_1,z_1\}}(-1)^{x_1z_1}\left|x_1\right\rangle\left\langle z_1\right|\right)\circ\left(\frac{1}{\sqrt{2}}\sum_{\{x_2,z_2\}}(-1)^{x_2z_2}\left|x_2\right\rangle\left\langle z_2\right|\right)$$

$$= \frac{1}{4}\sum_{\{x_1,x_2,z_1,z_2,y\}}(-1)^{x_1z_1+x_2z_2+y(z_1+x_2)}\left|x_1\right\rangle\left\langle z_2\right|$$

$$\xmapsto{eval} \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

$$\left\llbracket \begin{array}{l}\text{—}\boxed{H}\text{—}\\ \text{—}\boxed{H}\text{—}\end{array} \right\rrbracket = \left(\frac{1}{\sqrt{2}}\sum_{\{x_1,z_1\}}(-1)^{x_1z_1}\left|x_1\right\rangle\left\langle z_1\right|\right)\otimes\left(\frac{1}{\sqrt{2}}\sum_{\{x_2,z_2\}}(-1)^{x_2z_2}\left|x_2\right\rangle\left\langle z_2\right|\right)$$

$$= \frac{1}{2}\sum_{\{x_1,x_2,z_1,z_2\}}(-1)^{x_1z_1+x_2z_2}\left|x_1,x_2\right\rangle\left\langle z_1,z_2\right|$$

$$\xmapsto{eval} \frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}$$

For any circuit C with n qubit lines and k gates, we will denote its *volume* by $|C|$ which is the product of the number of qubit lines and number of gates, *i.e.* $|C| = nk$. Note that sometimes the depth d of a circuit is used in place of the number of gates to define volume. Since $d \leq k \leq kn$, these notions are polynomially related.

Proposition 3. *Let $\mathcal{G} = \{H, X, SWAP, Z, CZ, \cdots, C^{(m)}Z\}$ for some fixed m. For any circuit C over $\mathcal{G}$, $\llbracket C \rrbracket$ can be computed in time and has size polynomial in the volume $|C|$ of C. Furthermore, $\llbracket C \rrbracket$ has at most $O(|C|)$ variables.*

Proof. Let C have n qubit lines and k gates and notice that the size of any path sum expression $A \in \mathbf{PS}$ over $\mathcal{G}$ is at most polynomial in $|V_A|$. Indeed, the Boolean polynomials of A have degree at most $m+1$. Then the number of terms of a polynomial in $\mathbb{F}_2[V_A]/I_{V_A}$ of degree $\leq m+1$ is bounded by

$$\binom{|V_A|}{0} + \binom{|V_A|}{1} + \cdots + \binom{|V_A|}{m+1} = O(|V_A|^{m+1})$$

which bounds the size of A by $O(n|V_A|^{m+1})$. Each composition $C = A \circ B$ has number of variables $|V_C| = |V_A| + |V_B| + O(n)$ while tensor products preserve the number of variables. Thus at the end of the procedure we have applied at most k compositions introducing at most $O(kn)$ new variables. Since each $g \in \mathcal{G}$ has at most $m+1$ variables, we have variable count of the final expression of $\leq O(kn) + (m+1)k = O(kn)$ giving a final size of

$$O(n(kn)^{m+1})$$

which takes time $k \cdot O(n(kn)^{m+1}) = O((kn)^{m+2})$ for computing $\leq k$ intermediate expressions. □

3.1 Affine equivalence of path sums

A simple observation of path sum evaluation 2 reveals that for an invertible affine map $M : \mathbb{F}_2^k \to \mathbb{F}_2^k$, the evaluation of A is preserved by a transformation Mv on the points $v \in \mathbb{F}_2^k$

$$eval(A) = s_A \sum_{v \in \mathbb{F}_2^k}(-1)^{P_A(Mv)}\left|O_A(Mv)\right\rangle\left\langle I_A(Mv)\right| \tag{3}$$

since the sum is commutative and M is invertible. However, we can also choose to view the transformation of points instead as a transformation on the polynomials of our path sum A, which is to say a transformation of path sum A yielding path sum B with components such that

$$\begin{aligned} P_B(v) &= P_A(Mv) \\ O_B(v) &= O_A(Mv) \\ I_B(v) &= I_A(Mv) \end{aligned}$$

When $Mv = Lv + b$ is the representation of M where L is a linear map and $b \in \mathbb{F}_2^k$ is a translation, this is achieved by an isomorphism on the polynomials of A which sends $v_i \mapsto L_i v + b_i$ where L_i, b_i are the i-th rows of L, b respectively.

Viewed in this way, path sums which only differ by an affine translation of the evaluation points $v \in \mathbb{F}_2^k$ can be captured by a ring isomorphism applied uniformly on all constituent polynomials of the expression. The observation shows that we need only consider *degree-preserving* isomorphisms ϕ which is to say the total degree $\deg(\phi(v)) = 1$ for all $v \in V$. In order to be precise about what degree-preserving means in the context of Boolean rings $\mathbb{F}_2[V]/I_V$ in which the polynomials of our path sums reside, we will make use of a functor which will allow us to carry over these notions which are defined unambiguously in $\mathbb{F}_2[V]$.

For a homomorphism $\phi : \mathbb{F}_2[V] \to \mathbb{F}_2[W]$ which is *degree-nonincreasing*, that is, the total degree $\deg(\phi(v)) \leq 1$ for all indeterminates $v \in V$, ϕ induces a unique homomorphism $\phi' : \mathbb{F}_2[V]/I_V \to \mathbb{F}_2[W]/I_W$ as described in the following diagram which preserves composition, identities and inverses.

$$\begin{array}{ccc} \mathbb{F}_2[V] & \xrightarrow{\quad\phi\quad} & \mathbb{F}_2[W] \\ \downarrow{\scriptstyle \pi} & & \downarrow{\scriptstyle \pi} \\ \mathbb{F}_2[V]/I_V & \xrightarrow[\phi']{} & \mathbb{F}_2[W]/I_W \end{array} \tag{4}$$

In an abuse of notation we will refer to both ϕ and the uniquely induced ϕ' as simply ϕ. With these technical considerations out of the way, we can now formally define affine transformations of path sums.

Definition 6 (Affine Transformation of Path Sums)**.** *Let $A \in \boldsymbol{PS}$ be an arbitrary expression, and $\phi : \mathbb{F}_2[V_A] \to \mathbb{F}_2[W]$ be a degree-preserving ring isomorphism such that $|V_A| = |W|$. We call ϕ an affine transformation of path sums and $\phi(A)$ the affine transformation of A by ϕ, defined by*

$$\phi(A) = s_A \sum_W (-1)^{\phi(P)} \left|\phi(O_A)\right\rangle \left\langle\phi(I_A)\right|$$

where $\phi(O_A) = (\phi(O_{A,1}), \ldots, \phi(O_{A,m}))$ and likewise for $\phi(I_A)$.

In this work, we are concerned with path sums where all relevant polynomials are over $\mathbb{F}_2$. However, affine transformations can be extended to more elaborate path sums, by a Boolean lifting described in [4]. For instance, for the Clifford path sums in [37], where path sums had the form

$$s \sum_V e^{2i\pi \frac{1}{8} P^{(0)} + \frac{1}{4} P^{(1)} + \frac{1}{2} P^{(2)}} \left|O\right\rangle \left\langle I\right| . \tag{5}$$

where $P^{(k)} \in \mathbb{F}_2[V]$ is a Boolean polynomial limited to degree at most k, we see that an affine transformation preserves the form 5. As shown for linear substitutions in [4], this is a consequence of the fact that more generally for fixed m, affine transformations preserve expressions of the form

$$s \sum_V e^{2i\pi (\frac{1}{2^m} P^{(0)} + \cdots + \frac{1}{2^k} P^{(m-k)} + \cdots + \frac{1}{2} P^{(m-1)})} \left|O\right\rangle \left\langle I\right| .$$

And thus a bound on the time complexity of evaluating the path sum based on the degrees of the polynomials involved holds for the entire equivalence class of expressions that differ only by an affine transformation.

It will often be necessary to refer to the variables of a polynomial which we define formally as follows.

Definition 7. *Let K be a field and $f \in K[\mathcal{V}]$. Then $Var(f)$ is defined as the minimal subset $V \subseteq \mathcal{V}$ such that $f \in K[V]$. If $f_1, \ldots, f_n$ are polynomials (in potentially different polynomial rings), we define*

$$Var(f_1, \ldots, f_n) = \bigcup_{i=1}^{n} Var(f_i).$$

Although many of our lemmas will be stated in general for affine transformations, the main result of this work will be with respect to a simpler notion of equivalence which will correspond to when L of $Mv = Lv + b$ is a permutation matrix. This can be thought of as renaming variables, potentially with an affine translation by a scalar, and can be restated in terms of polynomial isomorphisms. Whenever a degree-nonincreasing homomorphism $\phi : \mathbb{F}_2[V] \to \mathbb{F}_2[W]$ is such that $|\mathrm{Var}(\phi(v))| \leq 1$, for all indeterminates $v \in V$, we say that ϕ is *simple*.

Definition 8 (Simple Transformation of Path Sums)**.** *Let $A \in \boldsymbol{PS}$ be an arbitrary expression, and $\phi : \mathbb{F}_2[V_A] \to \mathbb{F}_2[W]$ be a simple degree-preserving isomorphism such that $|V_A| = |W|$. Then we call ϕ a simple transformation and $\phi(A)$ the simple transformation of A by ϕ.*

We will make frequent use of the fact that for a simple transformation $\phi(A) = B$, for any $x \in V_A$, the restriction of the domain of ϕ to $\mathbb{F}_2[V_A \setminus \{x\}]$ has image contained in $\mathbb{F}_2[V_B \setminus \{z\}]$ for some $z \in V_B$. Simple transformations naturally yield the equivalence relation $\sim$ which we will use throughout this work.

Definition 9 (Simple Equivalence)**.** *For $A, B \in \boldsymbol{PS}$, we say that $A \sim B$ if and only if there exists a simple transformation ϕ such that $\phi(A) = B$.*

Example 4. *Consider the path sum interpretation of the quantum circuit consisting of two Hadamard gates in sequence. The simple transformation ϕ defined by*

$$\phi : u \mapsto u + 1, \quad w \mapsto w, \quad x \mapsto z + 1, \quad y \mapsto y + 1, \quad z \mapsto x + 1,$$

witnesses the following simple equivalence

$$\begin{aligned} [\![\text{—}\boxed{H}\text{—}\boxed{H}\text{—}]\!] &= \frac{1}{4} \sum_{\{u,w,x,y,z\}} (-1)^{zw+xy+uw+uy} |z\rangle \langle x| \\ &\sim \frac{1}{4} \sum_{\{u,w,x,y,z\}} (-1)^{xw+zy+z+uw+uy+u} |x+1\rangle \langle z+1| \end{aligned}$$

The simple transformation corresponds to the translation of evaluation points M as in eq. (3) defined by the following with respect to the ordering u, w, x, y, z.

$$M : v \mapsto \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} v + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Simple equivalence is comparable to α-equivalence in symbolic logic, with an additional equivalence made between a sum over a variable $x \in \mathbb{F}_2$ and the sum over its negation $1 + x \in \mathbb{F}_2$, and in this sense can be thought of as an *internal* equivalence of path sums. More generally, affine equivalence can be viewed as an equivalence relation for Boolean path sums which was enforced in [37] by the (bra) and (ket) rules. There, by virtue of the restriction to path sums of $\mathbf{SOP}_{Clif}$, it was shown that all path sums can be rewritten by the rewrite system $\to_{Clif+}$ to a form where all variables occurring in (non-zero) path sums occur in the O_i, I_j state polynomials. Then by the fact that the degrees of polynomials O_i, I_j for path sums of $\mathbf{SOP}_{Clif}$ are always 1, a canonical choice of the equivalence class was made by applying affine transformations allowed by the (bra) and (ket) rules using the ordering of polynomials $O_1 \prec \cdots \prec O_m \prec I_1 \prec \cdots \prec I_n$.

It can be seen that deciding simple equivalence for path sums in general is GI-hard. Indeed, there is a standard reduction from the graph isomorphism problem to deciding whether two quadratics are equivalent up to a permutation of variables. We can embed this quadratic into a path sum, and with

the help of an additional polynomial, we can force simple equivalence to be a permutation. Specifically, we can construct a path sum S_G for a graph $G = (V, E)$ with vertices $V = \{v_1, \ldots, v_n\}$ viewed as indeterminates as follows.

$$S_G = \sum_V (-1)^{\prod_i^n v_i} |\sum_{(i,j)\in E \,|\, i<j} v_i v_j\rangle$$

Then for undirected graphs G, H without loops on n vertices, $S_G \sim S_H$ if and only if there is a graph isomorphism $G \cong H$.

More generally for deciding affine equivalence, the problem has in fact been studied under the name, the Isomorphism of Polynomials with One Secret (IP1S) problem [15, 34] where its *two secret* variant, known simply as the Isomorphism of Polynomials (IP) problem, has been used as a basis for a public key encryption scheme called the Hidden Field Equations (HFE) [30]. We describe the IP1S problem in brief. In the following, let $\mathbb{F}_q$ be a finite field where q is a prime power, and $\mathrm{GA}_n(\mathbb{F}_q)$ the general affine group on the vector space $\mathbb{F}_q^n$. The IP1S problem is to compute, given two lists of polynomials $a = (a_1, \ldots, a_m)$, $b = (b_1, \ldots, b_m) \in \mathbb{F}_q[X]^m$ on n indeterminates, an affine map $S \in \mathrm{GA}_n(\mathbb{F}_q)$, such that

$$(b_1(x), \ldots, b_m(x)) = (a_1(Sx), \ldots, a_m(Sx))$$

as functions $\mathbb{F}_q^n \to \mathbb{F}_q^m$, if such an S exists. The problem of deciding IP1S is known to be GI-hard [31] by a reduction from the graph isomorphism problem to the IP1S decision problem for $q = 2$. Since deciding affine equivalence of path sums is the IP1S decision problem for $q = 2$ and the list of path sum polynomials $(P, O := (O_1, \ldots, O_m), I := (I_1, \ldots, I_n))$, it follows that deciding affine equivalence of path sums is also GI-hard in general.

3.2 Rewrite rules

Having interpreted a circuit C as a path sum $[\![C]\!] \in \mathbf{PS}$, simulation will be done by simplifying $[\![C]\!]$ by a rewrite system $\to$ which is formally a binary relation on $\mathbf{PS}$. If simple equivalence, or more generally affine equivalence is *internal*, we can consider rewrites as *external* equivalence of path sums due to interference.

Rewrite systems for path sums have been studied in [4–6, 37, 38]. In [4], a rewrite system which was complete for deciding equivalence of Clifford unitaries was given, and was generalized to all Clifford path sums in [37]. Further, [38] gave a complete rewrite system for the Boolean fragment we consider in this paper. The following rewrite system corresponds to a subset of the Clifford-complete system of [37] restricted to Boolean path sums. For a variable $v \in V$ and $f \in \mathbb{F}_2[V]$, we will denote by $[v \leftarrow f]$ the substitution of v with f.

Definition 10 (Rewrite System)**.** *Let $\to$ be a binary relation on $\boldsymbol{PS}$ which is the union of binary relations $\to_{elim}, \to_z, \to_{hh}$ defined below.*

1. *Whenever $x \notin Var(P, O, I)$,*

$$s\sum_V (-1)^P |O\rangle\langle I| \longrightarrow_{elim} 2s \sum_{V\setminus\{x\}} (-1)^P |O\rangle\langle I| \tag{Elim}$$

2. *Whenever $z \notin Var(R, O, I)$*

$$s\sum_V (-1)^{z+R} |O\rangle\langle I| \longrightarrow_z 0 \sum_{\emptyset} (-1)^0 |0, \ldots, 0\rangle\langle 0, \ldots, 0| \tag{Z}$$

3. *Whenever $x \notin Var(R, O, I), y \notin Var(Q)$ and $|Var(Q)| \leq 1$*

$$s\sum_V (-1)^{x(y+Q)+R} |O\rangle\langle I| \longrightarrow_{hh} \left(s \sum_{V\setminus\{x\}} (-1)^R |O\rangle\langle I|\right)[y \leftarrow Q] \tag{HH}$$

We will call the variable x in the precondition for the HH rule, the *pivot* variable. Note that since polynomials are assumed to be presented as *multilinear* polynomials (polynomials with degree at most 1 in any given variable), we have $x \notin \mathrm{Var}(y+Q)$. Further, $\mathrm{Var}(Q) \leq 1$ implies that $\deg(Q) \leq 1$ since a multilinear polynomial in a single variable is just a linear polynomial, and a constant when there are no variables. Like in the case of affine transformations of path sums which were essentially polynomial isomorphisms, we can view the Z rule and HH rules as inducing polynomial homomorphisms. Indeed, the Z rule induces the zero homomorphism $\mathbb{F}_2[V] \to \mathbb{F}_2$ which sends everything to zero. Likewise, the HH rule induces a substitution homomorphism $\mathbb{F}_2[V] \to \mathbb{F}_2[V \setminus \{x\}]$. While in previous rewrite systems the HH rule had no restriction on the structure of Q, here we restrict $|\mathrm{Var}(Q)| \leq 1$ so that the substitution $[y \leftarrow Q]$ is simple.

Example 5. *Consider the path sum interpretation of the quantum circuit consisting of two Hadamard gates in sequence as in Example 4. The HH rule along with the Elim rule can be seen to enable the rewrite to the identity path sum I_1. The pivot and eliminated variables are marked in red at each rewrite.*

$$
\begin{aligned}
[\![\text{—}\boxed{H}\text{—}\boxed{H}\text{—}]\!] &= \frac{1}{4} \sum_{\{u,w,x,y,z\}} (-1)^{zw+xy+u(w+y)} |z\rangle \langle x| \\
&\to_{hh} \frac{1}{4} \sum_{\{w,x,y,z\}} (-1)^{y(z+x)} |z\rangle \langle x| \\
&\to_{hh} \frac{1}{4} \sum_{\{w,x,z\}} (-1)^{0} |x\rangle \langle x| \\
&\to_{elim} \frac{1}{2} \sum_{\{x,z\}} (-1)^{0} |x\rangle \langle x| \\
&\to_{elim} \sum_{\{x\}} (-1)^{0} |x\rangle \langle x| \;=\; I_1
\end{aligned}
$$

We can think the restriction of $|\mathrm{Var}(Q)| \leq 1$ in the HH rule as going hand-in-hand with the chosen notion of simple equivalence. That is, if we were to trade the restriction $|\mathrm{Var}(Q)| \leq 1$ with $\deg(Q) \leq 1$, then we would naturally arrive at *affine* equivalence rather than simple equivalence, whereby two path sums are equivalent if they are related by an affine transformation. Simple homomorphisms are desirable as for any polynomial P, not only is the substitution $P[y \leftarrow Q]$ degree-preserving, but also preserves the number of variables $\mathrm{Var}(P)$. As a method of *simplifying* path sums, the restriction to simple HH avoids complications where an affine HH may obstruct subsequent rule applications by introducing variables into $\mathrm{Var}(R, O, I)$ which are inessential. The following example illustrates this complication of allowing $\deg(Q) \leq 1$, as it relates to presentations of polynomials. The presence of both z, w in the second path sum can be seen to be inessential, in that there exists an affine transformation such that the polynomials are presented with fewer variables, for instance the transformation which sends $z \mapsto z + w$.

Example 6. *Let $\to_{hh}$ have the restriction on Q that $\deg(Q) \leq 1$. Then $A = \frac{1}{4\sqrt{2}} \sum_{\{w,x,y,z\}} (-1)^{x(y+z+w)+y} |y\rangle$ rewrites to*

$$
\begin{aligned}
A &\to_{hh} \frac{1}{4\sqrt{2}} \sum_{\{w,y,z\}} (-1)^{y} |y\rangle \\
A &\to_{hh} \frac{1}{4\sqrt{2}} \sum_{\{w,y,z\}} (-1)^{z+w} |z+w\rangle
\end{aligned}
$$

where the results of the rewrites are related by an affine transformation but not a simple transformation. Furthermore, the first path sum admits two eliminations while the second admits only one.

Since the RHS of the Z rule is identical for all path sums of the same signature, we will denote it $A \to_z 0$. We will let $\xrightarrow{*}$ be the reflexive-transitive closure of $\to$ throughout. A property of interest for rewrite systems is whether each rewrite sequence is of finite length which guarantees the existence of *normal forms* (not necessarily unique) which are expressions that do not admit any rewrite.

Definition 11 (Noetherian)**.** *A rewrite system $\to$ is noetherian iff there is no infinite sequence $x_1 \to \cdots \to x_n \to \cdots$.*

Lemma 7. *The rewrite system of Definition 10 is noetherian*

Proof. Trivial since each rewrite rule removes at least one variable. □

Corollary 8. *A rewrite sequence $A_0 \to A_1 \to \ldots \to A_l$ has length at most polynomial in the number of variables $|V|$ of A_0.*

The above actually shows that the rewrite system is *bounded* according the definitions given in [18], which is to say for every given path sum A, all rewrite sequences starting from A are at most some finite length.

3.3 Simulation

At this point, we can already state a general simulation algorithm for circuits C over gate set $\mathcal{G}$. For a circuit C implementing a unitary U, we consider the task of strong simulation — given computational basis states $|x\rangle, |y\rangle$, compute the output amplitude $\langle y| U |x\rangle$ whose magnitude squared is the probability of observing $|y\rangle$ on input $|x\rangle$.

Algorithm 1 PS strong simulation for quantum algorithms

Require: n-qubit Circuit C over the gate set $\mathcal{G}$ and strings $x, y \in \mathbb{F}_2^n$
  $f \leftarrow \langle y_1, y_2, \ldots, y_n| \circ [\![C]\!] \circ |x_1, x_2, \ldots, x_n\rangle$
  while $f \to f'$ **do**
    $f \leftarrow f'$
  end while
  return $eval(f)$

Hidden shift circuits are deterministic circuits and strong simulation would in the worst case, still require 2^n-many invocations of algorithm 1 for each value of y to find the unique shift s that returns amplitude 1. Indeed, each invocation would comprise of the hidden shift circuit C with input $x = 0 \ldots 0$, and a choice of y which returns 1 if and only if y is exactly the shift s. More generally for quantum algorithms with measurement on a subset of qubits, strong simulation is also not sufficient to simulate their outputs as in general, one must compute all of the relevant amplitudes by algorithm 1. We can modify algorithm 1 to accommodate these scenarios where measurement occurs on one qubit by the following, and can easily be extended for subsets of qubits. We note that unlike algorithm 1, algorithm 2 returns a probability.

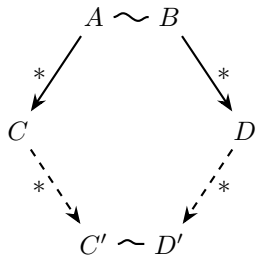
Algorithm 2 PS 1-qubit measurement simulation for quantum algorithms

Require: n-qubit Circuit C over the gate set $\mathcal{G}$, string $x \in \mathbb{F}_2^n$ and measurement index $i \in [n]$
  $g \leftarrow [\![C]\!] \circ |x_1, x_2, \ldots, x_n\rangle$
  $f \leftarrow g^\dagger \circ (I_{i-1} \otimes |1\rangle\langle 1| \otimes I_{n-i}) \circ g$
  while $f \to f'$ **do**
    $f \leftarrow f'$
  end while
  return $eval(f)$

We note that when post-selection in circuits is allowed by use of projectors, then algorithm 2 is just a strong simulation on a circuit of the form $C = U^\dagger P U$ with projector P and input strings $x = y$. We also note that for a uniform family of circuits $\{C_n\}$ deciding a language in BQP, a classical algorithm deciding the same language amounts to accepting input $x \in \mathbb{F}_2^n$ iff algorithm 2 outputs a value $\geq 2/3$ on input C_n and x. We can understand algorithms 1 and 2 to consist of essentially two phases. The first phase is to symbolically rewrite the path sum until a normal form is reached, and the second consists of an evaluation. One can notice by the rewrite system of Definition 10, that path sums remain polynomially-sized over the entire first phase. This is to say that for a circuit C over $\mathcal{G}$, whenever

$$[\![C]\!] \xrightarrow{*} A$$

Figure 1: Confluence Modulo $\sim$ of Definition 12



where $\xrightarrow{*}$ is the reflexive-transitive closure of $\to$, A has size bounded by $O(|C|^{m+2})$. Indeed, the bound on the size of $[\![C]\!]$ of Proposition 3 was due to the degree bound of all of its constituent polynomials. Since none of the rewrite rules of Definition 10 increase polynomial degrees, the same bound holds for A. Furthermore, by Corollary 8, we have that the while loop executes at most polynomially-many times, making the first phase terminate necessarily in polynomial time, since each rewrite takes polynomial time to identify and apply. The costly portion is the second evaluation phase which is exponential in the number of variables, which by proposition 3 is $\Omega(|C|)$ in the worst case.

Algorithms 1 and 2 always succeed, but in the worst case take exponential time in the volume of the circuit, due to the evaluation which takes exponential time in the number of variables. In the subsequent sections, in light of the preceding discussion on the efficiency of the first simplification phase, we will develop sufficient conditions for when these algorithms are efficient, which will boil down to demanding that the first phase sufficiently simplifies the path sum to reduce the complexity of the costly evaluation phase. Further, by showing our rewrite system to be *confluent*, it will follow that any choice of $f \to f'$ at each iteration of the loop ultimately yields an equivalent path sum in the evaluation phase, removing the need for any specific simplification strategy.

4 Confluent Rewriting for Path Sums

Having armed ourselves with a ring theoretic view of path sums, we are in a position to prove *confluence* of our rewrite system. The property of confluence as seen in Figure 1 can be intuitively understood as the property that "all roads lead to Rome". It is desirable as we don't ever need to backtrack while simplifying our expression and coupled with the noetherian property of a rewrite system, we are guaranteed unique normal forms so that our motto then becomes "all roads lead to Rome in finite time". It is readily seen that all of our rewrites can be identified and applied in polynomial time and since we've seen already that any rewrite sequence terminates in polynomially-many steps, this further improves our motto to

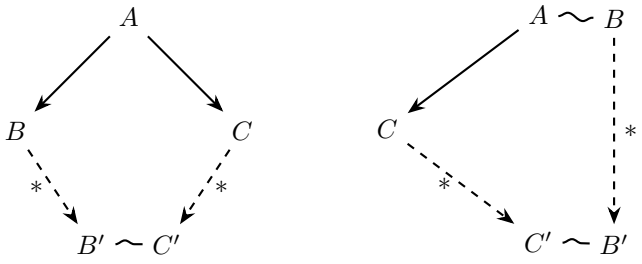
"all roads lead to Rome in polynomial time"

or more accurately, "all rewrites lead to a unique normal form in polynomial time". The rewrite system of [38] was not confluent and only complete when the rewrite system was made into an equational theory after taking the symmetric closure. We will show that the rewrite system 10 is confluent modulo simple equivalence. Confluence modulo an equivalence relation has the formal following definition, where $\xrightarrow{*}$ denotes the reflexive-transitive closure of a binary relation $\to$.

Definition 12 (Confluence Modulo $\sim$)**.** *If $\sim$ is an equivalence relation, then a rewrite system $\to$ is confluent modulo $\sim$ iff for all A, B such that $A \sim B$, and for any C, D such that $A \xrightarrow{*} C, B \xrightarrow{*} D$, there exists C', D' with $C' \sim D'$ and $C \xrightarrow{*} C', D \xrightarrow{*} D'$.*

Definition 12 is summarized by Figure 1. In the case of noetherian relations which we have by Lemma 7, we can consider a simpler characterization called local confluence modulo $\sim$.

Figure 2: α and β properties of Definition 13



Definition 13 (Local Confluence Modulo $\sim$)**.** *A rewrite system $\to$ is locally confluent modulo $\sim$ iff conditions α and β are satisfied:*

$$\alpha : \forall ABC \quad A \to B \wedge A \to C \implies B \overset{\sim}{\downarrow} C$$
$$\beta : \forall ABC \quad A \sim B \wedge A \to C \implies B \overset{\sim}{\downarrow} C$$

where $B \overset{\sim}{\downarrow} C$ denotes the existence of B', C' such that $B \overset{*}{\to} B'$, $C \overset{*}{\to} C'$ and $B' \sim C'$.

Lemma 9 ([18], Lemma 2.7)**.** *Let $\to$ be a noetherian relation. For any equivalence $\sim$, $\to$ is confluent modulo $\sim$ iff $\to$ is locally confluent modulo $\sim$.*

We will show that the rewrite system satisfies both the α and β properties. The key to proving confluence will be the universal property of the quotient for polynomial rings — seemingly different choices of the application of rewrite rules will correspond to different representations of polynomials modulo a common kernel. The universal property of the quotient will guarantee that the outcomes are related by an affine transformation in general, and in the case of the rewrites of Definition 10, related by a simple transformation. We will start by proving the β property for cases $\{\to_{elim}, \to_z\}$, then we'll show how the universal property of the quotient connects to path sums to prove the β property for $\to_{hh}$. Finally we'll borrow further ideas from ring theory to consider what can be said of sequences of $\to_{hh}$ and finish by giving a relatively succinct proof of α property.

Lemma 10. *The Elim rule satisfies property β of Definition 13.*

Proof. Let x be the variable eliminated from A to produce A', and $\phi(A) = B$. By the definition of simple equivalence it follows that the restriction of the domain of ϕ to $\mathbb{F}_2[V_A \setminus \{x\}]$ has image in $\mathbb{F}_2[V_B \setminus \{z\}]$ for some $z \in V_B$, so that ϕ with this restriction is

$$\phi' : \mathbb{F}_2[V_A \setminus \{x\}] \to \mathbb{F}_2[V_B \setminus \{z\}]$$

Thus $z \notin \mathrm{Var}(P_B, O_B, I_B)$ and z can be eliminated from B yielding B'. It follows that ϕ' with exactly this restriction is the simple transformation such that $\phi'(A') = B'$. □

Lemma 11. *The Z rule satisfies property β of Definition 13.*

Proof. Let ϕ be a simple transformation of path sums and $A, B \in \mathbf{PS}$ with $\phi(A) = B$, and let A admit an application of the Z rule $A \to A' = 0$ with

$$A = s_A \sum_{V_A} (-1)^{x+R} |O_A\rangle \langle I_A|$$

Then $\phi(x) = z + c$ for some $z \in V_B, c \in \mathbb{F}_2$, and it follows that

$$B = s_A \sum_{V_A} (-1)^{z+c+\phi(R)} |\phi(O_A)\rangle \langle \phi(I_A)|$$

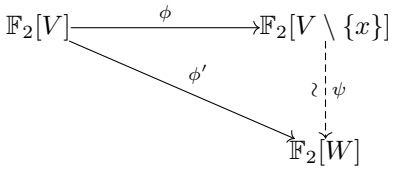
and $z \notin Var(\phi(R), \phi(O_A), \phi(I_A))$. Thus B satisfies the precondition for the Z rule, and can be thus be rewritten to exactly $A' = 0$. □

The β property for the HH rule will follow from the fact that the substitution induced satisfies the universal property for quotients for the relevant rings. Recall the definition of the universal property of the quotient in the category of rings. Let R be a ring and $I \subseteq R$ be an ideal, then the pair $\langle \pi, S \rangle$ where S is a ring, and $\pi : R \to S$ a ring homomorphism satisfies the universal property of the quotient for I if for all $\psi : R \to T$ such that $\ker \psi \supseteq I$, there exists a unique $\phi : S \to T$ such that $\psi = \phi \circ \pi$. For a more in-depth account of ring theory, the interested reader is encouraged to consult chapter 3 of [3].

Lemma 12. *Let V be a set of indeterminates containing x, and let $Q \in \mathbb{F}_2[V \setminus \{x\}]$ with $\deg(Q) \leq 1$. Then the substitution homomorphism $\phi : \mathbb{F}_2[V] \to \mathbb{F}_2[V \setminus \{x\}]$ defined by*

$$\phi : x \mapsto Q$$

and the identity on $v \in V \setminus \{x\}$, has the universal property of the quotient $\pi : \mathbb{F}_2[V] \to \mathbb{F}_2[V]/(x+Q)$. Furthermore, whenever $\phi' : \mathbb{F}_2[V] \to \mathbb{F}_2[W]$ is a degree-nonincreasing surjective homomorphism with $\ker(\phi') = (x+Q)$, the isomorphism ψ induced in the following diagram is degree-preserving. If in addition, ϕ' is simple then ψ is simple.



(6)

Proof. Since $\ker(\phi) = (x+Q)$ and $\mathrm{Im}(\phi) = \mathbb{F}_2[V \setminus \{x\}]$, by the first isomorphism theorem for rings there is a unique isomorphism $\mathbb{F}_2[V]/(x+Q) \cong \mathbb{F}_2[V \setminus \{x\}]$, which gives the desired universal property.

Now, suppose that $\phi' : \mathbb{F}_2[V] \to \mathbb{F}_2[W]$ is degree-nonincreasing and $\ker(\phi') = (x+Q)$. We notice that for $f \in \mathbb{F}_2[V]$, with $\deg(f) = 1$ for $c \in \mathbb{F}_2$,

$$f + (x+Q) = c + (x+Q) \iff \phi(f) = c \iff \phi'(f) = c \tag{7}$$

so that $\deg(\phi(f)) = 0$ iff $\deg(\phi'(f)) = 0$. Since ψ sends $\phi(f)$ to $\phi'(f)$, and all degree 1 polynomials of $\mathbb{F}_2[V \setminus \{x\}]$ are in the image of degree 1 polynomials of $\mathbb{F}_2[V]$ under ϕ, it follows that ψ is degree preserving.

In addition, let ϕ' be simple. Then for all $v \in V \setminus \{x\}$, ψ sends $\phi(v) = v$ to $\phi'(v)$ which by assumption must have $|\mathrm{Var}(\phi'(v))| = 1$, and so ψ is simple.

□

By Diagram 6 which gives the universal property and conclusions on the induced isomorphisms in Lemma 12, we can make statements about the equivalence of path sums. For example, consider $Q = y + Q_1 = w + Q_2$ where $\deg(Q) = 1$, $y \notin \mathrm{Var}(Q_1)$, and $w \notin \mathrm{Var}(Q_2)$, and let

$$A = s_A \sum_{V_A} (-1)^{xQ+R} |O\rangle \langle I|$$

satisfy the precondition of the HH rule. Then by choice of either $\phi = [y \leftarrow Q_1]$ or $\phi' = [w \leftarrow Q_2]$ we have two different HH rewrites $A \to B_1$ and $A \to B_2$ respectively. However, by Lemma 12 we see that there exists an affine transformation ψ such that $\psi(B_1) = B_2$. And furthermore, if $|\mathrm{Var}(Q)| \leq 2$ then ϕ, ϕ' are simple and thus so is ψ, and we have that $B_1 \sim B_2$ under our definition of equivalence. In particular this implies that the choice of target variable in the HH rule yields unique path sums up to simple equivalence. As a consequence, we will abuse notation to write $[Q \leftarrow 0]$ for a degree 1 polynomial Q to mean $[y \leftarrow Q']$ for any variable $y \in \mathrm{Var}(Q)$, $Q = y + Q'$. The following lemma stated in general for affine equivalence intuitively says that if the affine transformation between A and B is compatible with a HH rewrite applied on A in the sense that there is a corresponding rewrite available on B, then the results are affine equivalent.

Lemma 13. *Let $A \in \boldsymbol{PS}$ have the form consistent with the precondition of the HH rule, and let $A \to_{hh} A'$ where $\deg(Q) = 1$.*

$$A = s_A \sum_{V_A} (-1)^{xQ+R} |O\rangle \langle I|$$

$$A' = s_A \sum_{V_A \setminus \{x\}} (-1)^{R[Q \leftarrow 0]} |O[Q \leftarrow 0]\rangle \langle I[Q \leftarrow 0]|$$

Suppose ϕ is an affine transformation of A such that $B = \phi(A)$ admits an application of the HH rule yielding

$$B' = s_A \sum_{V_B \setminus \{z\}} (-1)^{\phi(R)[\phi(Q) \leftarrow 0]} |\phi(O)[\phi(Q) \leftarrow 0]\rangle \langle \phi(I)[\phi(Q) \leftarrow 0]|$$

If $\phi(\mathbb{F}_2[V_A \setminus \{x\}]) \subseteq \mathbb{F}_2[V_B \setminus \{z\}]$, then there exists an affine transformation ψ such that $\psi(A') = B'$. In addition if $[Q \leftarrow 0]$ and ϕ are simple, then ψ is simple.

Proof. Letting $\phi' : \mathbb{F}_2[V_A \setminus \{x\}] \to \mathbb{F}_2[V_B \setminus \{z\}]$ be the restriction of ϕ, since $x \notin \mathrm{Var}(Q, R, O, I)$, it follows that

$$B' = s_A \sum_{V_A \setminus \{z\}} (-1)^{\phi'(R)[\phi'(Q) \leftarrow 0]} |\phi'(O)[\phi'(Q) \leftarrow 0]\rangle \langle \phi'(I)[\phi'(Q) \leftarrow 0]|$$

By the universal property 12 of $[Q \leftarrow 0]$ we have that the affine transformation ϕ' induces an affine transformation ψ such that the following diagram commutes.

$$\begin{array}{ccc} \mathbb{F}_2[V_{A'}] & \xrightarrow{[Q \leftarrow 0]} & \mathbb{F}_2[V_{A'}]/(Q) \\ \downarrow{\scriptstyle \phi'} & & \downarrow{\scriptstyle \psi} \\ \mathbb{F}_2[V_{B'}] & \xrightarrow[[\phi'(Q) \leftarrow 0]]{} & \mathbb{F}_2[V_{B'}]/(\phi'(Q)) \end{array}$$

Where $\mathbb{F}_2[V_{A'}]/(Q)$ and $\mathbb{F}_2[V_{B'}]/(\phi'(Q))$ are understood as the codomain of the substitution homomorphism for arbitrary choice of variable as in Lemma 12. Furthermore if ϕ, and $[Q \leftarrow 0]$ are simple, then $[\phi'(Q) \leftarrow 0]$ is simple and by Lemma 12, it follows that ψ is simple. □

The β property follows from the fact that a simple equivalence $A \sim B$, is always compatible with HH rewrites. This is to say that there is always a corresponding rewrite on B to any given HH rewrite on A, and Lemma 13 guarantees that the results are simple equivalent.

Corollary 14. *The HH rule satisfies property β of Definition 13.*

Proof. Let $B = \phi(A)$ where ϕ is a simple transformation. Then $\phi(x) = z + c$, for some $z \in V_B, c \in \mathbb{F}_2$, and

$$B = s_A \sum_{V_B} (-1)^{z\phi(Q)+c\phi(Q)+\phi(R)} |\phi(O)\rangle \langle \phi(I)|$$

admits an HH rewrite of the form required by Lemma 13. Specifically, the rewrite with pivot variable z yielding the homomorphism $[\phi(Q) \leftarrow 0]$. □

Example 15. *Consider the equivalent path sums from Example 4 where the LHS admits the first rewrite from Example 5. Corollary 14 ensures that there is a corresponding rewrite to the RHS and furthermore that their results are equivalent.*

$$\begin{array}{ccc} \frac{1}{4}\sum_{\{u,w,x,y,z\}} (-1)^{zw+xy+u(w+y)} |z\rangle \langle x| & \overset{\phi}{\sim} & \frac{1}{4}\sum_{\{u,w,x,y,z\}} (-1)^{xw+zy+z+u(w+y+1)} |x+1\rangle \langle z+1| \\ \downarrow{\scriptstyle [w \leftarrow y]} & & \downarrow{\scriptstyle [y \leftarrow w+1]} \\ \frac{1}{4}\sum_{\{w,x,y,z\}} (-1)^{zy+xy} |z\rangle \langle x| & \overset{\psi}{\sim} & \frac{1}{4}\sum_{\{w,x,y,z\}} (-1)^{xw+zw} |x+1\rangle \langle z+1| \end{array}$$

The resulting path sums can be seen to be equivalent by the simple transformation ψ defined by

$$\psi : w \mapsto y, \quad x \mapsto z + 1, \quad y \mapsto w, \quad z \mapsto x + 1$$

as a direct consequence of the unique isomorphism $\bar{\psi} : \mathbb{F}_2[x, y, z] \to \mathbb{F}_2[w, x, z]$ such that

$$\bar{\psi} \circ [w \leftarrow y] = [y \leftarrow w + 1] \circ \phi$$
$$\textit{defined by} \quad \bar{\psi} : x \mapsto z + 1, \quad y \mapsto w, \quad z \mapsto x + 1$$

Finally to prove the α property, we will further borrow from ring theory to consider the case of a sequence of $\to_{hh}$. To this end, recall the following which is a consequence of the third isomorphism theorem for rings. For a ring R with ideals $I, J \subseteq R$, let $\pi : R \to R/I$ be the quotient homomorphism. Then $\pi(J) = \pi(I + J) = (I + J)/I$ is an ideal of R/I and

$$\frac{R/I}{\pi(J)} = \frac{R/I}{(I + J)/I} \cong \frac{R}{I + J} \tag{8}$$

which says that if $\rho : R/I \to \frac{R/I}{\pi(J)}$ is again the quotient homomorphism, then $\rho \circ \pi$ has the universal property of the quotient with respect to ideal $I + J = \{i + j \mid i \in I, j \in J\}$.

Since we've seen that the substitution homomorphism $[Q \leftarrow 0]$ can be identified with π for $I = (Q)$ up to an affine equivalence, we see that this allows for the universal property to extend to a sequence of multiple rewrites. To see how this applies to substitution homomorphisms, consider the following which in particular implies that for degree 1 polynomials Q_1, Q_2, we have $Q_1[Q_2 \leftarrow 0] = c$ iff $Q_2[Q_1 \leftarrow 0] = c$.

Lemma 16. *Let $Q_1, Q_2 \in \mathbb{F}_2[V]$ be polynomials such that $\deg(Q_1) = \deg(Q_2) = 1$. Then for $c \in \mathbb{F}_2$, $Q_1[Q_2 \leftarrow 0] = c$ iff $Q_1 + Q_2 = c$.*

Proof. In Appendix A □

Now let $Q_1, Q_2 \in \mathbb{F}_2[V]$ such that $\deg(Q_1 + Q_2) = 1$, so that $Q_2' = Q_2[Q_1 \leftarrow 0]$ has degree 1 by the above. Then letting $V'' \subsetneq V' \subsetneq V$ be the variable sets with the target variables of $[Q_2' \leftarrow 0], [Q_1 \leftarrow 0]$ removed respectively, we have a composition of homomorphisms

$$\mathbb{F}_2[V] \xrightarrow{[Q_1 \leftarrow 0]} \mathbb{F}_2[V'] \xrightarrow{[Q_2' \leftarrow 0]} \mathbb{F}_2[V'']$$

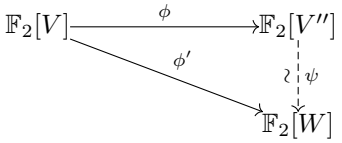
By Lemma 12, and Equation 8 we have that

$$\mathbb{F}_2[V''] \cong \frac{\mathbb{F}_2[V']}{(Q_2')} \cong \frac{\mathbb{F}_2[V]/(Q_1)}{(\pi(Q_2))} \cong \frac{\mathbb{F}_2[V]}{(Q_1, Q_2)}$$

where $\pi : \mathbb{F}_2[V] \to \mathbb{F}_2[V]/(Q_1)$ is the quotient map. Thus the homomorphism

$$\phi = [Q_1 \leftarrow 0][Q_2' \leftarrow 0] \tag{9}$$

satisfies the universal property of the quotient $\mathbb{F}_2[V] \to \frac{\mathbb{F}_2[V]}{(Q_1, Q_2)}$. Furthermore, the same consequences about the induced isomorphism ψ of Diagram 10 follow, by the same proof as Lemma 12. Indeed, consider an alternative $\phi' : \mathbb{F}_2[V] \to \mathbb{F}_2[W]$ with $\ker(\phi') = (Q_1, Q_2)$, then there is a unique isomorphism $\psi : \mathbb{F}_2[V''] \to \mathbb{F}_2[W]$ such that the following commutes



(10)

and furthermore, we have by the exact same reasoning as in Equation 7 of Lemma 12, that if ϕ' is degree-preserving (respectively, simple) then ψ is degree-preserving (respectively, simple).

This observation about chaining homomorphisms can be extended further, and in particular guarantees affine or simple equivalence of path sums which are results of different sequences of $\to_{hh}$ so long as

they have the same kernel. We use this reasoning to prove the α property by a case analysis of each pair of possible rewrites. The unification of path sums will involve sequences of rewrites, which in the view developed, will correspond to a composition of homomorphisms which is yet another homomorphism. Then by observing when two seemingly distinct rewrite sequences have the same kernel, we can reason by the universal property of the quotient that the results are in fact simple equivalent.

Lemma 17. *The rewrite system $\to$ of Definition 10 on **PS** satisfies the α property.*

Proof. Let $A \to B$ and $A \to C$ as in top two arrows in the alpha property in Figure 2. We consider cases of each rewrite in $\{\to_{elim}, \to_{hh}, \to_z\}$ and by symmetry need only consider each unordered pair. Whenever one of the rewrites is $\to_{elim}$, we can trivially obtain a common $D \in \mathbf{PS}$, since variable elimination and all other rewrites commute. So we consider only the pairs of rewrites taken from $\{\to_{hh}, \to_z\}$. The case when both $A \to_z B$, $A \to_z C$ is trivial as $B = C = 0$. So we show the remaining two cases.

case 1. $A \to_z B$ and $A \to_{hh} C$.

Let x be the pivot variable for $A \to_{hh} C$, and z be the variable of A as in z in the LHS for $A \to_z B = 0$. Then x and z must be distinct variables and $z \notin \mathrm{Var}(Q)$ where A, C have general forms

$$A = \sum_{V_A} (-1)^{(z+xQ+R)} |O_A\rangle \langle I_A|$$
$$C = \sum_{V_A \setminus \{x\}} (-1)^{(z+R[Q\leftarrow 0])} |O_A[Q \leftarrow 0]\rangle \langle I_A[Q \leftarrow 0]|$$

Where $x, z \notin \mathrm{Var}(Q, R, O_A, I_A)$. Then C rewrites to $B = 0$ by an application of $\to_z$.

case 2. $A \to_{hh} B$ and $A \to_{hh} C$.

We distinguish cases on the "pivot" variable x as in the HH rule of Definition 10. We let x, w be the "pivot" variables for $A \to B$ and $A \to C$ respectively. This is to say that A can be expressed in two forms,

$$A = s_A \sum_{V_A} (-1)^{xQ_1+R_1} |O_A\rangle \langle I_A| = s_A \sum_{V_A} (-1)^{wQ_2+R_2} |O_A\rangle \langle I_A|$$

both satisfying the precondition of the HH rule.

If $x = w$, then we have that $Q_1 = Q_2$ so that both $A \to B$, and $A \to C$ induce substitution homomorphism $[Q_1 \leftarrow 0]$, which by Lemma 12, implies that $B \sim C$.

Now suppose that $x \neq w$. We notice that $x \in Var(Q_2) \iff w \in Var(Q_1)$ since both sides imply that xw is a term of P_A. Consider the case where $x \notin Var(Q_2)$. It follows that A must have the general form

$$A = s_A \sum_{V_A} (-1)^{xQ_1+wQ_2+R} |O_A\rangle \langle I_A|$$

where $x, w \notin Var(Q_1, Q_2, R, O_A, I_A)$. It follows that B, C have forms

$$B = s_A \sum_{V_A \setminus \{x\}} (-1)^{wQ_2'+R[Q_1\leftarrow 0]} |O_A[Q_1 \leftarrow 0]\rangle \langle I_A[Q_1 \leftarrow 0]|$$
$$C = s_A \sum_{V_A \setminus \{w\}} (-1)^{xQ_1'+R[Q_2\leftarrow 0]} |O_A[Q_2 \leftarrow 0]\rangle \langle I_A[Q_2 \leftarrow 0]|$$

where $Q_2' = Q_2[Q_1 \leftarrow 0]$ and $Q_1' = Q_1[Q_2 \leftarrow 0]$. By Lemma 16, it follows that for $c \in \mathbb{F}_2$,

$$Q_2' = c \iff Q_1' = c \iff Q_2 + Q_1 = c$$

Thus if $Q_2' = 0$, then $Q_1' = 0$ and $Q_1 = Q_2$, so $B \sim C$ by simply mapping $w \mapsto x$. If $Q_2' = 1$, then $Q_1' = 1$ and both B, C rewrite to 0 by $\to_z$. So we are left with the case that $\deg(Q_2') = \deg(Q_1') = 1$. Since $|\mathrm{Var}(Q_1)| \leq 2$, $|\mathrm{Var}(Q_2)| \leq 2$, and $[Q_2 \leftarrow 0]$, $[Q_1 \leftarrow 0]$ are simple, it follows that $|\mathrm{Var}(Q_1')| \leq$

2, $|\mathrm{Var}(Q_2')| \leq 2$. Thus B and C rewrite to B' and C' respectively

$$B' = \left(s_A \sum_{V_A \setminus \{x,w\}} (-1)^R \left|O_A\right\rangle \left\langle I_A\right| \right) [Q_1 \leftarrow 0][Q_2' \leftarrow 0] \tag{11}$$

$$C' = \left(s_A \sum_{V_A \setminus \{x,w\}} (-1)^R \left|O_A\right\rangle \left\langle I_A\right| \right) [Q_2 \leftarrow 0][Q_1' \leftarrow 0] \tag{12}$$

Since both $\phi_1 = [Q_1 \leftarrow 0][Q_2' \leftarrow 0], \phi_2 = [Q_2 \leftarrow 0][Q_1' \leftarrow 0]$ have the same kernel (Q_1, Q_2), it follows by the discussion of 9, that B', C' are affinely related by some $\phi(B') = C'$, and furthermore ϕ is simple since both ϕ_1, ϕ_2 are simple.

To finish the proof, we consider lastly when $x \in \mathrm{Var}(Q_2)$. It follows that A has the general form

$$A = s_A \sum_{V_A} (-1)^{xQ_1' + wQ_2' + xw + R} \left|O_A\right\rangle \left\langle I_A\right|$$

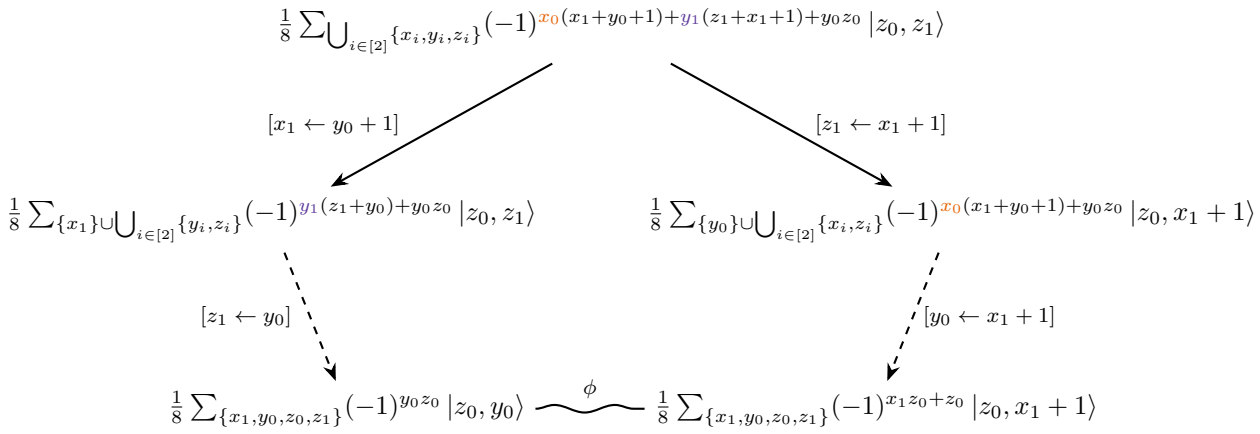
where $x, w \notin Var(Q_1', Q_2', R, O_A, I_A)$. Then by Lemma 12, up to a simple equivalence, we can let $A \to B$, $A \to C$ induce $[w \leftarrow Q_1'], [x \leftarrow Q_2']$ respectively. Thus we have

$$B \sim s_A \sum_{V_A \setminus \{x\}} (-1)^{Q_1' Q_2' + R} \left|O_A\right\rangle \left\langle I_A\right| \qquad C \sim s_A \sum_{V_A \setminus \{w\}} (-1)^{Q_1' Q_2' + R} \left|O_A\right\rangle \left\langle I_A\right|$$

where the RHS of each equivalence are equivalent by the isomorphism which sends $w \mapsto x$. □

The following example illustrates the case of the α property expressed by Equations (11) and (12), where the universal property of the homomorphism of the form 9, ensures that the resulting path sums are simple equivalent.

Example 18. *Consider the two potential rewrites due to the HH rule with respect to pivot variables x_0 and y_1 respectively. They can be seen to be unified by HH rule applications with pivots y_1 and x_0 respectively.*



By the fact that the composite homomorphisms down each path have the same kernel $(x_1 + y_0 + 1, z_1 + x_1 + 1)$, they are related by the simple transformation ϕ that has the following action,

$$\phi : x_1 \mapsto y_0, \quad y_0 \mapsto x_1 + 1, \quad z_0 \mapsto z_0, \quad z_1 \mapsto z_1$$

which is a direct consequence of the unique isomorphism $\bar{\phi} : \mathbb{F}_2[y_0, z_0] \to \mathbb{F}_2[x_1, z_0]$ such that

$$\bar{\phi} \circ ([z_1 \leftarrow y_0][x_1 \leftarrow y_0 + 1]) = [y_0 \leftarrow x_1 + 1][z_1 \leftarrow x_1 + 1]$$
$$\textit{defined by} \quad \bar{\phi} : y_0 \mapsto x_1 + 1, \quad z_0 \mapsto z_0$$

Now, combining Lemma 17 which proves the α property, and Lemmas 11, 10, 14 which prove the β property, we have the following theorem.

Theorem 19. *The rewrite system $\to$ of Definition 10 on **PS** is confluent modulo simple equivalence $\sim$.*

5 Polynomial-Time Simulation of Hidden Shift Circuits

Having shown that a circuit C over gate set $\mathcal{G} = \{H, X, \text{SWAP}, Z, CZ, \cdots, C^{(m)}Z\}$ can be interpreted as a path sum $[\![C]\!]$ in polynomial time in the volume of C, and that the rewrite system $\to$ to simplify $[\![C]\!]$ is confluent, it remains to determine how confluence of the rewrite system 10 affects the time complexity of the simplification and evaluation phases of algorithms 1 and 2. From Corollary 8, we have on input circuit C, that the number of rewrites is bounded by $\text{poly}(|C|)$ where $|C|$ is the volume of circuit C. Furthermore, each rewrite takes at most polynomial time in the size of path sum at the beginning of the iteration to identify and apply by a linear scan of the path sum expression. By the discussion at the end of section 3.3, we have that the size of the path sum remains $\text{poly}(|C|)$ throughout repeated rewrites, so that the path sum at the beginning of each iteration is in fact of size $\text{poly}(|C|)$. By combining these facts together, namely that we have $\text{poly}(|C|)$ many iterations, each taking $\text{poly}(|C|)$ time, we have that the while loop in algorithms 1 and 2 terminate in $\text{poly}(|C|)$ time. Thus, for the algorithms to be efficient, it is sufficient for the final evaluation to take polynomial time. This is achieved when the path sum $[\![C]\!]$ can be simplified to a path sum of sufficiently few variables. As the preceding discussion suggests, the notion of efficiency for a family of circuits will be polynomial-time in the volume of each circuit.

Lemma 20. *Let $\mathcal{C}$ be a family of circuits over $\mathcal{G}$. For circuit $C \in \mathcal{C}$ with signature $n \to l$, $x \in \mathbb{F}_2^n$, and $y \in \mathbb{F}_2^l$, if $\langle y| \circ [\![C]\!] \circ |x\rangle \xrightarrow{*} A$ such that $|V_A| = O(\log|C|)$, then algorithm 1 outputs $\langle y| C |x\rangle$ in polynomial time.*

Proof. Construction of $[\![C]\!]$ takes time and has size $O(|C|^{m+2})$. For at most $O(|C|^{m+2})$ times, rewrites are applied which each take $O(|C|^{m+2})$ time. After rewriting to a normal form A' which is $A' \sim A$, we have evaluation on $2^{O(\log|C|)}$ points. Let $|V_{A'}| = |V_A| \leq d \log|C|$ asymptotically. Then we have at most $|C|^d$ points, each polynomial evaluation takes $O(|V_A|^{m+1})$ time. So the total time for evaluating A' is

$$|C|^d O(|V_A|^{m+1}) = O(|C|^{m+d+1})$$

The time complexity for algorithm is then

$$O(|C|^{2m+4}) + O(|C|^{m+d+1}) = O(|C|^{2m+d+4})$$

□

To state a similar result for algorithm 2, it will be useful to state a lemma relating rewrites and path sum compositions and tensor products. In the following, let $\xrightarrow{\epsilon}$ be the reflexive closure of the binary relation $\to$.

Lemma 21. *Let $A, B \in \boldsymbol{PS}$, be path sums such that $A \xrightarrow{\epsilon} A'$ and $B \xrightarrow{\epsilon} B'$. Then whenever the following are well-defined,*

$$A \circ B \xrightarrow{*} A' \circ B' \tag{13}$$

$$A \otimes B \xrightarrow{*} A' \otimes B' \tag{14}$$

$$A^\dagger \xrightarrow{\epsilon} (A')^\dagger \tag{15}$$

where the RHS of Equations (13) and (14) is 0 in the exceptional case where either $A \to_z 0$ or $B \to_z 0$.

Proof. In Appendix C □

Lemma 22. *Let $\mathcal{C}$ be a family of circuits over $\mathcal{G}$. For circuit $C \in \mathcal{C}$ on n qubits, and $x \in \mathbb{F}_2^n$, if $[\![C]\!] \circ |x\rangle \xrightarrow{*} A$ such that $|V_A| = O(\log|C|)$, then for all $i \in [n]$, algorithm 2 terminates in polynomial time on input (C, x, i).*

Proof. Without loss of generality let $i = 1$. The path sum g in algorithm 2 is constructed in time and has size $O(|C|^{m+2})$. Then path sum f of the algorithm has size at most

$$2O(|C|^{m+2}) + O(n) = O(|C|^{m+2})$$

thus simplification of f consists of at most $O(|C|^{m+2})$ rewrites each taking $O(|C|^{m+2})$ time, which takes time $O(|C|^{2m+4})$. By assumption $g \xrightarrow{*} A$ such that $|V_A| = O(\log|C|)$ so that $|V_A| \leq d \log|C|$

asymptotically for some constant d. It follows by Lemma 21 that $g^\dagger \xrightarrow{*} A^\dagger$ which has the same number of variables as A. Let $B = A^\dagger$ to simplify notation. By Lemma 21, it follows that for path sum f of the algorithm

$$\begin{aligned}
f &\xrightarrow{*} B \circ (|1\rangle\langle 1| \otimes I_{n-1}) \circ A \\
&= B \circ \frac{s_A}{2^n} \sum_{\substack{V_A \cup \{x_2,\ldots,x_n\} \\ \cup \{z,y_2,\ldots,y_n\}}} (-1)^{P_A + z(1+O_{A,1}) + \sum y_i(x_i + O_{A,i})} |1, x_2, \ldots, x_n\rangle \\
&\xrightarrow{*} B \circ \frac{s_A}{2} \sum_{V_A \cup \{z\}} (-1)^{P_A + z(1+O_{A,1})} |1, O_{A,2}, \ldots, O_{A,n}\rangle \\
&= \frac{s_B s_A}{2^{n+1}} \sum_{\substack{V_A \cup V_B \cup \{z\} \\ \cup \{w,y_2,\ldots,y_n\}}} (-1)^{P_A + z(1+O_{A,1}) + P_B + w(1+I_{B,1})s + \sum y_i(I_{B,i} + O_{A,i})} |\rangle\langle|
\end{aligned}$$

where $|\mathrm{Var}(I_{B,i}+O_{A,i})| \leq 2$ for $i = 2, \ldots, n$ so that either an HH, Elim, or Z rule can be applied to remove each y_i. Thus we have that $f \xrightarrow{*} C$ such that $V_C = V_A \cup V_B \cup \{w, z\}$, so that $|V_C| \leq 2d\log|C| + 2 = O(\log|C|)$. Let $|V_C| \leq d' \log|C|$ asymptotically. Then after the simplification loop of the algorithm, $f \sim C$ and so it has the same number of variables. Thus evaluation occurs on at most $2^{d' \log|C|} = |C|^{d'}$ points for a polynomial with at most $O(|V_C|^{m+1})$ terms. Each term takes m multiplications which is a constant, so that evaluation takes

$$|C|^{d'} O(|V_C|^{m+1}) = O(|C|^{d'} |C|^{m+1}) = O(|C|^{m+d'+1})$$

Thus, algorithm 2 terminates in time

$$O(|C|^{2m+4}) + O(|C|^{m+d'+1}) = O(|C|^{2m+d'+4})$$

□

Finally we prove our main result, that the family of hidden shift circuits $C_{(\pi,g,s)}$ for Roetteler's shifted bent function algorithm is polynomial-time simulable via rewrites of its path sum. For a hidden shift circuit $C_{(\pi,g,s)}$, its classical simulation amounts to applying algorithm 2 on

$$\big(C_{(\pi,g,s)}, 0, i\big)$$

for each $i \in [n]$ to build the shift s. To prove that Algorithm 2 simulates hidden shift circuits efficiently, by Lemma 22, it suffices to prove that $[\![C_{(\pi,g,s)}]\!] \circ |00\ldots0\rangle$ simplifies to sufficiently few variables. In the following proposition, we will see that in fact, all variables can be removed by virtue of the proof of correctness seen in section 2. This is to say that the proof of correctness can essentially be reproduced formally with the rewrite system $\to$. Then by confluence of the rewrite system $\to$, algorithm 2, is guaranteed to sufficiently simplify the corresponding path sum by applying rewrites in a structure-oblivious manner, to yield the i-th bit of the shift in polynomial time.

Proposition 23. *Given a hidden shift circuit $C_{(\pi,g,s)}$, the path sum $[\![C_{(\pi,g,s)}]\!] \circ |00\ldots0\rangle$ can be computed and has size polynomial in the volume $|C_{(\pi,g,s)}|$ of $C_{(\pi,g,s)}$. Furthermore with the rewrite system $\to$ of Definition 10,*

$$[\![C_{(\pi,g,s)}]\!] \circ |00\ldots0\rangle \xrightarrow{*} |s\rangle$$

Proof. We first note that through at most $|C_{(\pi,g,s)}|$ applications of $\to_{hh}$, the path sum $[\![C]\!] \circ |00\ldots0\rangle$ may be re-written to the form

$$\frac{1}{2^{3n}} \sum_{x,y,z} (-1)^{g(x) + \langle x,y\rangle + \overline{f}(y) + \langle y,z\rangle} |z\rangle$$

as in the proof of correctness in section 2, where the phase polynomial is explicitly expanded to algebraic normal form. Now it can be observed that the proof of correctness given in section 2 follows from $3n$ applications of $\to_{hh}$: the first sequence of n rewrites pivots on variables of x_1 giving the simple substitutions $[y_1 \leftarrow \phi(x_2) + \pi(s_2)]$, while the remaining $2n$ independent rewrites correspond to pivots y_2 and x_2 with substitutions $[z_2 \leftarrow s_2]$ and $[z_1 \leftarrow s_1]$ Since the proof ends in $|s_1, s_2\rangle = |s\rangle$, we have the desired result. □

Proposition 23 ensures the existence of a rewrite sequence resulting in a path sum of sufficiently few variables. Thus by applying Lemma 22, we can now derive the main result of our paper, namely that hidden shift circuits are simulated by Algorithm 2 in polynomial time.

Corollary 24. *Let $\mathcal{C} = \{C_{(\pi,g,s)}\}$ be the hidden shift circuit family of Definition 3. Then for each $C_{(\pi,g,s)} \in \mathcal{C}$, algorithm 2 simulates $C_{(\pi,g,s)}$ on input $|00\ldots0\rangle$ in time $\mathrm{poly}(|C_{(\pi,g,s)}|)$.*

Proof. By Proposition 23, for each $C_{(\pi,g,s)} \in \mathcal{C}$ on n qubits, we have that $[\![C_{(\pi,g,s)}]\!] \circ |00\ldots0\rangle \xrightarrow{*} |s\rangle$ which has no variables which is trivially $O(\log|C_{(\pi,g,s)}|)$. Thus by Lemma 22, algorithm 2 terminates in time $O(|C_{(\pi,g,s)}|^{2m+4})$. An n-fold application of algorithm 2 for each $i \in [n]$ yields the shift s, taking time

$$n \cdot O(|C_{(\pi,g,s)}|^{2m+4}) = O(|C_{(\pi,g,s)}|^{2m+5})$$

□

Remark 25. *It can be noted that in the case of hidden shift circuits, we can avoid the n-fold application of algorithm 2 by simply rewriting $[\![C]\!] \circ |00\ldots0\rangle$ directly. Indeed, since hidden shift circuits are deterministic, we have that*

$$\mathit{eval}\big([\![C]\!] \circ |00\ldots0\rangle\big) = |s\rangle$$

hence in algorithm 2, we can instead simply set $f \leftarrow g$ in the second line, so that the final evaluation yields $|s\rangle$ exactly, saving a factor of n by applying the algorithm exactly once.

6 Conclusion

In this paper we have shown that a family of quantum circuits for Roetteler's shifted bent function algorithm which requires non-Clifford resources and is not trivially efficient to simulate can in fact be simulated in polynomial time using path sums. We do so by giving a rewrite system on path sums which is both restrictive enough to prove *confluence*, while also powerful enough to deterministically reduce implementations of Roetteler's algorithm to the hidden shift in a small (linear) number of steps. This rewrite system is applied within a simulation algorithm which first simplifies a path sum to reduce the number of variables before explicitly evaluating the sum. As simplifications of the path sums are closely related to tensor network contractions, it stands to reason that our work can be seen as a type of strategy for tensor network contraction. In this sense, a novel aspect of the work is that through confluence we show not just the *existence* of (an analogue of) a contraction order which gives an efficient simulation, but that *every* order with the simple restriction yields an efficient simulation. As our results apply to oracles implemented over a gate set which necessarily produce bent functions with multilinear polynomial representations having polynomially-many terms, the theoretical implications of this work appear to be limited. Moreover, as the query complexity separation between quantum and classical solutions to the shifted bent function problem do not necessarily translate to realistic quantum speedup, the utility of a polynomial-time classical simulation of such circuits is dubious. Instead, this work shows that this particular class of circuits is not a suitable benchmark for classical simulation algorithms, a point alluded to by [11, 22]. More generally, our work suggests that the sum-over-paths approach may yield efficient simulations of certain classes of circuits.

The effectiveness of rewriting in the context of general circuit simulation remains an interesting question. In the worst case, our rewrite system cannot simplify the path sum at all, leading to a very costly evaluation of the sum. It happens that nothing is lost in this case, as the resulting sum may then be decomposed into a sum of stabilizer sums, or more generally other polynomial-time simulable sums. It remains a question for future work to explore these decompositions in practice and benchmark them against other methods. More generally, it remains an open question to find other rewrite systems, ideally which satisfy strong guarantees on their time complexity to reach normal forms and determinism as the one we have developed here, which are effective in the context of circuit simulation. One potential avenue of simulation which was alluded to in [5] is to generalize the sum from variables to *varieties* over $\mathbb{F}_2^k$, which allows the entire phase polynomial to be deterministically absorbed into an ideal of polynomial equations. In one sense this offloads the difficulty of simulation to the similarly difficult problem of computing Gröbner bases and counting points in varieties, and so it remains a question for future work to explore this technique.

7 Acknowledgements

MA acknowledges support from the Canada Research Chair program, NSERC Discovery grant RGPIN-2022-03319, NSERC Alliance Quantum Software Consortium, and the Google Research Scholar program. MA and LSS thank the reviewers for their constructive comments and suggestions.

References

[1] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5), Nov. 2004. Publisher: American Physical Society (APS). https://doi.org/10.1103/PhysRevA.70.052328.

[2] D. Aharonov. A Simple Proof that Toffoli and Hadamard are Quantum Universal, 2003. _eprint: quant-ph/0301040. https://doi.org/10.48550/arXiv.quant-ph/0301040.

[3] P. Aluffi. *Algebra: Chapter 0*. Graduate studies in mathematics ; 104. American Mathematical Society, Providence, R.I, 2009. Publication Title: Algebra: Chapter 0.

[4] M. Amy. Towards Large-scale Functional Verification of Universal Quantum Circuits. In P. Selinger and G. Chiribella, editors, *Proceedings 15th International Conference on Quantum Physics and Logic, QPL 2018, Halifax, Canada, 3-7th June 2018*, volume 287 of *EPTCS*, pages 1–21, 2018. https://doi.org/10.4204/EPTCS.287.1.

[5] M. Amy. Complete Equational Theories for the Sum-Over-Paths with Unbalanced Amplitudes. In S. Mansfield, B. Valiron, and V. Zamdzhiev, editors, *Proceedings of the Twentieth International Conference on Quantum Physics and Logic, QPL 2023, Paris, France, 17-21st July 2023*, volume 384 of *EPTCS*, pages 127–141, 2023. https://doi.org/10.4204/EPTCS.384.8.

[6] M. Amy, O. Bennett-Gibbs, and N. J. Ross. Symbolic Synthesis of Clifford Circuits and Beyond. In S. Gogioso and M. Hoban, editors, *Proceedings 19th International Conference on Quantum Physics and Logic, QPL 2022, Wolfson College, Oxford, UK, 27 June - 1 July 2022*, volume 394 of *EPTCS*, pages 343–362, 2022. https://doi.org/10.4204/EPTCS.394.17.

[7] S. Bravyi, D. Browne, P. Calpin, E. Campbell, D. Gosset, and M. Howard. Simulation of quantum circuits by low-rank stabilizer decompositions. *Quantum*, 3:181, Sept. 2019. Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften. https://doi.org/10.22331/q-2019-09-02-181.

[8] S. Bravyi and D. Gosset. Improved Classical Simulation of Quantum Circuits Dominated by Clifford Gates. *Physical Review Letters*, 116(25), June 2016. Publisher: American Physical Society (APS). https://doi.org/10.1103/PhysRevLett.116.250501.

[9] S. Bravyi, G. Smith, and J. A. Smolin. Trading Classical and Quantum Computational Resources. *Phys. Rev. X*, 6(2):021043, June 2016. Publisher: American Physical Society. https://doi.org/10.1103/PhysRevX.6.021043.

[10] C. Chareton, S. Bardin, F. Bobot, V. Perrelle, and B. Valiron. An Automated Deductive Verification Framework for Circuit-building Quantum Programs. In N. Yoshida, editor, *Programming Languages and Systems*, pages 148–177, Cham, 2021. Springer International Publishing. https://doi.org/10.1007/978-3-030-72019-3_6.

[11] J. Codsi. Cutting Edge Graphical Stabilizer Decompositions for Classical Simulation of Quantum Circuits. Master's thesis, University of Oxford, 2022.

[12] B. Coecke and R. Duncan. Interacting Quantum Observables. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *Automata, Languages and Programming*, pages 298–310, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-70583-3_25.

[13] A. W. Cross, E. Magesan, L. S. Bishop, J. A. Smolin, and J. M. Gambetta. Scalable randomised benchmarking of non-Clifford gates. *npj Quantum Information*, 2(1), Apr. 2016. Publisher: Springer Science and Business Media LLC. https://doi.org/10.1038/npjqi.2016.12.

[14] C. M. Dawson, H. L. Haselgrove, A. P. Hines, D. Mortimer, M. A. Nielsen, and T. J. Osborne. Quantum computing and polynomial equations over the finite field Z_2, 2004. _eprint: quant-ph/0408129. https://doi.org/10.26421/QIC5.2-2.

[15] J.-C. Faugère and L. Perret. Polynomial Equivalence Problems: Algorithmic and Theoretical Aspects. In S. Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, pages 30–47, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. https://doi.org/10.1007/11761679_3.

[16] K. Geddes, S. Czapor, and G. Labahn. *Algorithms for computer algebra*. Springer Science+Business Media New York, 1992. https://doi.org/10.1007/b102438.

[17] S. Hallgren and A. W. Harrow. Superpolynomial Speedups Based on Almost Any Quantum Circuit. In *Lecture Notes in Computer Science*, pages 782–795. Springer Berlin Heidelberg, 2008. ISSN: 1611-3349. https://doi.org/10.1007/978-3-540-70575-8_64.

[18] G. Huet. Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems: Abstract Properties and Applications to Term Rewriting Systems. *Journal of the ACM*, 27(4):797–821, Oct. 1980. Place: New York, NY, USA Publisher: Association for Computing Machinery. https://doi.org/10.1145/322217.322230.

[19] C. Jones. Low-overhead constructions for the fault-tolerant Toffoli gate. *Physical Review A*, 87(2):022328, Feb. 2013. Publisher: American Physical Society. https://doi.org/10.1103/PhysRevA.87.022328.

[20] A. Kissinger and J. van de Wetering. Simulating quantum circuits with ZX-calculus reduced stabiliser decompositions. *Quantum Science and Technology*, 7(4):044001, July 2022. Publisher: IOP Publishing. https://doi.org/10.1088/2058-9565/ac5d20.

[21] A. Kissinger, J. van de Wetering, and R. Vilmart. Classical Simulation of Quantum Circuits with Partial and Graphical Stabiliser Decompositions. In F. Le Gall and T. Morimae, editors, *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, volume 232 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:13, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. https://doi.org/10.4230/LIPIcs.TQC.2022.5.

[22] M. Koch, R. Yeung, and Q. Wang. Contraction of zx diagrams with triangles via stabiliser decompositions. *Physica Scripta*, 99(10):105122, Sept. 2024. https://doi.org/10.1088/1402-4896/ad6fd8.

[23] L. Kocia and M. Sarovar. Classical simulation of quantum circuits using fewer Gaussian eliminations. *Phys. Rev. A*, 103(2):022603, Feb. 2021. Publisher: American Physical Society. https://doi.org/10.1103/PhysRevA.103.022603.

[24] B. Lovitz and V. Steffan. New techniques for bounding stabilizer rank. *Quantum*, 6:692, Apr. 2022. Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften. https://doi.org/10.22331/q-2022-04-20-692.

[25] T. Lubinski, S. Johri, P. Varosy, J. Coleman, L. Zhao, J. Necaise, C. H. Baldwin, K. Mayer, and T. Proctor. Application-oriented performance benchmarks for quantum computing. *IEEE Transactions on Quantum Engineering*, 4:1–32, 2023. https://doi.org/10.1109/TQE.2023.3253761.

[26] A. Montanaro. Quantum circuits and low-degree polynomials over \mathbb{F}_2. *Journal of Physics A: Mathematical and Theoretical*, 50(8):084002, Jan. 2017. Publisher: IOP Publishing. https://doi.org/10.1088/1751-8121/aa565f.

[27] X. Ni and M. van den Nest. Commuting quantum circuits: efficiently classical simulations versus hardness results. *Quant. Inf. Comput.*, 13(1-2):0054–0072, 2013. https://doi.org/10.26421/QIC13.1-2-5.

[28] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2011. https://doi.org/10.1017/CBO9780511976667.

[29] H. Pashayan, O. Reardon-Smith, K. Korzekwa, and S. D. Bartlett. Fast Estimation of Outcome Probabilities for Quantum Circuits. *PRX Quantum*, 3(2), June 2022. Publisher: American Physical Society (APS). https://doi.org/10.1103/PRXQuantum.3.020361.

[30] J. Patarin. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In U. Maurer, editor, *Advances in Cryptology — EUROCRYPT '96*, pages 33–48, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-68339-9_4.

[31] J. Patarin, L. Goubin, and N. Courtois. Improved algorithms for isomorphisms of polynomials. In K. Nyberg, editor, *Advances in Cryptology — EUROCRYPT'98*, pages 184–200, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0054126.

[32] S. Peleg, A. Shpilka, and B. L. Volk. Lower Bounds on Stabilizer Rank. *Quantum*, 6:652, Feb. 2022. Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften. https://doi.org/10.22331/q-2022-02-15-652.

[33] F. C. R. Peres and E. F. Galvão. Quantum circuit compilation and hybrid computation using Pauli-based computation. *Quantum*, 7:1126, Oct. 2023. Publisher: Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften. https://doi.org/10.22331/q-2023-10-03-1126.

[34] L. Perret. A Fast Cryptanalysis of the Isomorphism of Polynomials with One Secret Problem. In R. Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, pages 354–370, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. https://doi.org/10.1007/11426639_21.

[35] H. Qassim, H. Pashayan, and D. Gosset. Improved upper bounds on the stabilizer rank of magic states. *Quantum*, 5:606, Dec. 2021. Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften. https://doi.org/10.22331/q-2021-12-20-606.

[36] M. Rötteler. Quantum algorithms for highly non-linear Boolean functions. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 448–457, USA, 2010. Society for Industrial and Applied Mathematics. event-place: Austin, Texas. https://doi.org/10.1137/1.9781611973075.37.

[37] R. Vilmart. The Structure of Sum-Over-Paths, its Consequences, and Completeness for Clifford. In S. Kiefer and C. Tasson, editors, *Foundations of Software Science and Computation Structures*, pages 531–550, Cham, 2021. Springer International Publishing. https://doi.org/10.1007/978-3-030-71995-1_27.

[38] R. Vilmart. Completeness of Sum-Over-Paths for Toffoli-Hadamard and the Dyadic Fragments of Quantum Computation. In *CSL 2023 - 31st EACSL Annual Conference on Computer Science Logic*, volume 252, pages 36:1–36:17, Warsaw, Poland, Feb. 2023. https://doi.org/10.4230/LIPIcs.CSL.2023.36.

[39] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins, K. M. Hudek, J. Mizrahi, J. D. Wong-Campos, S. Allen, J. Apisdorf, P. Solomon, M. Williams, A. M. Ducore, A. Blinov, S. M. Kreikemeier, V. Chaplin, M. Keesan, C. Monroe, and J. Kim. Benchmarking an 11-qubit quantum computer. *Nature Communications*, 10(1):5464, Nov. 2019. Publisher: Nature Publishing Group. https://doi.org/10.1038/s41467-019-13534-2.

A Multilinear Forms and Polynomials Over $\mathbb{F}_2$

Here we review some facts which are relevant to multilinear polynomials. We prove a fact that a degree-nonincreasing homomorphism between polynomial rings over $\mathbb{F}_2$ induce a unique ring homomorphism between their respective Boolean rings.

Lemma 26. *Let $\phi : \mathbb{F}_2[V] \to \mathbb{F}_2[V]$ be a homomorphism. Then ϕ is degree-nonincreasing on $\mathbb{F}_2[V]$ iff it is degree-nonincreasing on linear forms in $\mathbb{F}_2[V]$.*

Proof. Forward direction is immediate. Conversely, we have that ϕ is degree-nonincreasing on all variables $v \in V$. Then if for $f = \sum_I c_I V^I \in \mathbb{F}_2[V]$, let $\deg(f) = \max\{deg(V_I) \mid c_I \neq 0\}$ so that

$$\begin{aligned}\deg(\phi(f)) &\leq \max\{deg(\phi(V_I)) \mid c_I \neq 0\} \\ &\leq \max\{deg(V_I) \mid c_I \neq 0\} \\ &= \deg(f)\end{aligned}$$

□

Lemma 27. *Let $\phi : \mathbb{F}_2[V] \to \mathbb{F}_2[V]$ be an isomorphism. Then ϕ preserves total degree on $\mathbb{F}_2[V]$ iff it preserves total degree on linear forms in $\mathbb{F}_2[V]$.*

Proof. The forward direction is clear. Suppose that ϕ preserves degree on linear forms and assume for the moment that ϕ is linear in that for linear forms y, $\phi(y)$ is a linear form. Then ϕ can be viewed as an invertible linear transformation on $\mathbb{F}_2^{\oplus V}$. Then by Lemma 26, ϕ, and ϕ^{-1} are degree-nonincreasing on $\mathbb{F}_2[V]$, so that ϕ is degree preserving. The implication for general ϕ follows by writing $\phi = \phi' \circ C$ for linear ϕ' and C which adds constants to each variable as appropriate. □

Lemma 28. *Let $\phi : \mathbb{F}_2[V] \to \mathbb{F}_2[W]$ be a degree-preserving homomorphism, then ϕ induces a degree preserving homomorphism $\phi' : \mathbb{F}_2[V]/I_V \to \mathbb{F}_2[W]/I_W$ and this association preserves composition.*

Proof. Let $\psi := \pi \circ \phi$ where $\pi : \mathbb{F}_2[W] \to \mathbb{F}_2[W]/I_W$ is the projection map. Let α be an arbitrary element of I_V. Then $\alpha = \sum_i \alpha_i(v_i^2 - v_i)$, where $\alpha_i \in \mathbb{F}_2[V]$ and $v_i \in V$ are indeterminates. Since ϕ is degree preserving, $\phi(v_i) = \sum_j \beta_j^i w_j + c_i$, for some $\beta_j^i, c_i \in \mathbb{F}_2$. Noticing in characteristic 2,

$$\begin{aligned}\phi(v_i)^2 &= \left(\sum_j \beta_j^i w_j + c_i\right)^2 \\ &= \sum_j \beta_j^i w_j^2 + c_i\end{aligned}$$

We have

$$\begin{aligned}\phi(\alpha) &= \sum_i \phi(\alpha_i)(\phi(v_i)^2 - \phi(v_i)) \\ &= \sum_i \phi(\alpha_i)(\sum_j \beta_j^i w_j^2 + c_i - \sum_j \beta_j^i w_j + c_i) \\ &= \sum_i \phi(\alpha_i)(\sum_j \beta_j^i (w_j^2 - w_j)) \\ &= \sum_{i,j} \beta_j^i \phi(\alpha_i)(w_j^2 - w_j) \in I_W\end{aligned}$$

Thus ψ factors through $\mathbb{F}_2/I_V$, inducing ϕ' as desired.

$$\begin{array}{ccc} \mathbb{F}_2[V] & \xrightarrow{\phi} & \mathbb{F}_2[W] \\ \downarrow{\scriptstyle \pi} & & \downarrow{\scriptstyle \pi} \\ \mathbb{F}_2[V]/I_V & \xrightarrow[\phi']{} & \mathbb{F}_2[W]/I_W \end{array}$$

By uniqueness of ϕ', it follows that compositions and inverses are preserved. □

It is a well-known fact that any polynomial ring over a unique factorization domain (UFD) is a UFD.

Lemma 29 (Pseudo Division in UFDs - pg.54 of [16])**.** *Let $D[x]$ be a polynomial ring over a UFD D. For all $a(x), b(x) \in D[x]$ with $b(x) \neq 0$, there exist unique polynomials $q(x), r(x) \in D[x]$ such that*

$$\beta^l a(x) = b(x)q(x) + r(x)$$

where $\deg(r(x)) < \deg(b(x))$, β is the leading coefficient of $b(x)$ and $l = \deg(a(x)) - \deg(b(x)) + 1$.

Lemma 16. *Let $a, b \in \mathbb{F}_2[V]$ be polynomials such that $\deg(a) = \deg(b) = 1$. Then for $c \in \mathbb{F}_2$, $a[b \leftarrow 0] = c$ iff $a + b = c$.*

Proof. Let $a + b = c \in \mathbb{F}_2$, then $(a + b)[b \leftarrow 0] = a[b \leftarrow 0] = c$. Conversely let $a[b \leftarrow 0] = c \in \mathbb{F}_2$ and $x \in \mathrm{Var}(b)$ be the target variable of the substitution $[b \leftarrow 0]$. Then by pseudo-division, we have

$$a = qb + r$$

where $x \notin \mathrm{Var}(r)$. Thus $a[b \leftarrow 0] = r[b \leftarrow 0] = r = c$. Since $\deg(a) = 1$, it follows that $q = 1$, so

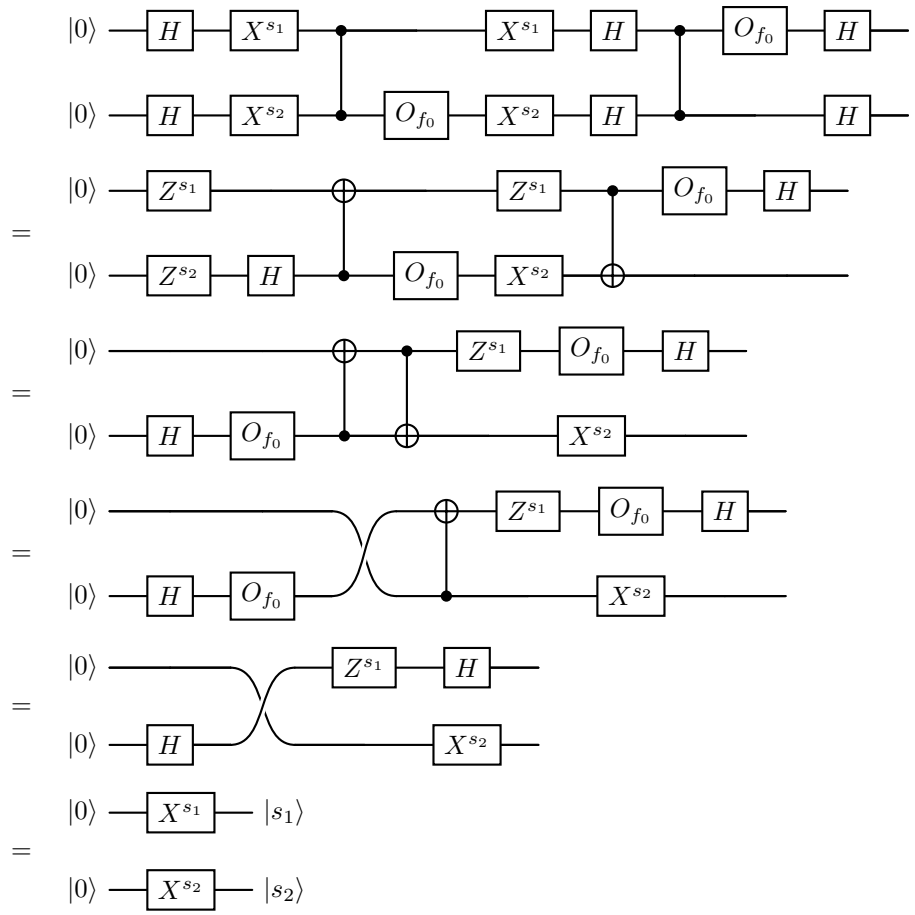
$$a = b + r = b + c$$

and $a + b = c$. □

In particular $a[b \leftarrow 0] = c$ iff $b[a \leftarrow 0] = c$.

B Correctness of Bravyi-Gosset Construction of Hidden Shift Circuits via Circuit Equalities

We show that the implementation of Roetteler's algorithm with $\pi = \mathrm{id}$ in [8] can be reduced to the unique hidden shift using circuit equalities. As shown in the circuit-level proof below, the application of X^s to apply a literal shift, as in [8] (note that Z^s was equivalently used in their implementation), allows the two applications of $O_{f_0} : |x\rangle \mapsto (-1)^{f_0(x)} |x\rangle$ to cancel, at which point the rest follows from basic simplifications.



C Composition of Path Sums

We state useful lemmas relating compositions and tensor products to the rewrites of their constituent parts. We also give two easy lemmas pertaining to the time complexity of simplification and evaluation of path sums.

Lemma 30. *Let $A, B \in \boldsymbol{PS}$ be path sums and suppose $A \to A'$. Then we have the following whenever compositions are well-defined.*

$$A \circ B \to A' \circ B$$
$$A \otimes B \to A' \otimes B$$
$$B \circ A \to B \circ A'$$
$$B \otimes A \to B \otimes A'$$

Proof. We give the proof for the first statement when $A \to_{hh} D$. The rest follow similarly. Let A have the general form of the LHS of the HH rule.

$$A = s_A \sum_{V_A} (-1)^{x(y+Q)+R} |O_A\rangle \langle I_I|$$

Let $A : m \to n$, and $B : l \to m$. Then we have

$$\begin{aligned}
A \circ B &= \frac{s_A s_B}{2^m} \sum_{V_A \cup V_B \cup \{z_1,\ldots,z_m\}} (-1)^{x(y+Q)+R+P_B+\sum z_i(I_{A,i}+O_{B,i})} |O_A\rangle \langle I_B| \\
&\to_{hh} \frac{s_A s_B}{2^m} \sum_{(V_A \setminus \{x\}) \cup V_B \cup \{z_1,\ldots,z_m\}} (-1)^{R[y\leftarrow Q]+P_B+\sum z_i(I_{A,i}[y\leftarrow Q]+O_{B,i})} |O_A[y \leftarrow Q]\rangle \langle I_B| \\
&= \frac{s_D s_B}{2^m} \sum_{V_D \cup V_B \cup \{z_1,\ldots,z_m\}} (-1)^{P_D+P_B+\sum z_i(I_{D,i}+O_{B,i})} |O_D\rangle \langle I_B| \\
&= D \circ B
\end{aligned}$$

which follows from the fact that

$$D = s_A \sum_{V_A \setminus \{x\}} (-1)^{R[y\leftarrow Q]} |O_A\rangle \langle I_I| [y \leftarrow Q]$$

□

Lemma 21. *Let $A, B \in \boldsymbol{PS}$, be path sums such that $A \xrightarrow{\epsilon} A'$ and $B \xrightarrow{\epsilon} B'$. Then*

$$A \circ B \to A' \circ B'$$
$$A \otimes B \to A' \otimes B'$$

where $\xrightarrow{\epsilon}$ is the reflexive closure of $\to$, and $A \circ B$ is well-defined.

Proof. Immediately follows from Lemma 30. □

Lemma 31. *Let $A \in \boldsymbol{PS}$ be a path sum with signature $n \to l$ such that all constituent polynomials are bounded in degree by a constant $m \in \mathbb{N}$, then simplifying A to a normal form by $\to$ takes time*

$$O((n + l + 1)|V_A|^{m+2})$$

Lemma 32. *Let $A \in \boldsymbol{PS}$ be a path sum with signature $n \to l$ such that all constituent polynomials are bounded in degree by a constant $m \in \mathbb{N}$, then $eval(A)$ is computed in time*

$$O((n + l + 1)|V_A|^{m+1} 2^{|V_A|})$$

and yields a linear operator with description size $O((n + l)2^{|V_A|})$.