Cutoff Theorems for the Equivalence of Parameterized Quantum Circuits

Neil J. Ross
Dalhousie University, Halifax, Canada

Scott Wesley
Dalhousie University, Halifax, Canada

Abstract

Many promising quantum algorithms in economics, medical science, and material science rely on circuits that are parameterized by a large number of angles. To ensure that these algorithms are efficient, these parameterized circuits must be heavily optimized. However, most quantum circuit optimizers are not verified, so this procedure is known to be error-prone. For this reason, there is growing interest in the design of equivalence checking algorithms for parameterized quantum circuits. In this paper, we define a generalized class of parameterized circuits with arbitrary rotations and show that this problem is decidable for cyclotomic gate sets. We propose a cutoff-based procedure which reduces the problem of verifying the equivalence of parameterized quantum circuits to the problem of verifying the equivalence of finitely many parameter-free quantum circuits. Because the number of parameter-free circuits grows exponentially with the number of parameters, we also propose a probabilistic variant of the algorithm for cases when the number of parameters is intractably large. We show that our techniques extend to equivalence modulo global phase, and describe an efficient angle sampling procedure for cyclotomic gate sets.

2012 ACM Subject Classification Hardware → Quantum computation; Hardware → Equivalence checking

Keywords and phrases Quantum Circuits, Parameterized Equivalence Checking

Digital Object Identifier 10.4230/LIPIcs.MFCS.2025.85

Related Version *Full Version*: `https://arxiv.org/abs/2506.20985` [39]

Acknowledgements We thank Mingkuan Xu for his feedback on the paper and our initial experiments, Linh Dinh for sharing their insights on the number-theoretic aspects of this paper, and Mingsheng Ying for his feedback on an early draft of the paper.

1 Introduction

In quantum mechanics, unitary operators describe how the probability distributions of quantum systems evolve over time. In quantum computing, primitive operators (known as *quantum gates*) are composed in sequence and parallel, to create *quantum circuits* which prepare quantum systems with desirable probability distributions. By sampling from these distributions, answers can be obtained to many high-value problems, such as those from economics [19], medical science [14, 40], and material science [29]. In these algorithms, an initial guess is made for the correct probability distribution, and then each sample is used to further refine the distribution. To make this search tractable, the probability distributions are sampled from a family of parameterized quantum circuits, known as *ansatz circuits*.

In practice, the structure of the ansatz circuit is static, so that the parameters only vary the operators which appear within the circuits. The parameterized operators within ansatz circuits can be understood geometrically as rotations by arbitrary angles. As a result, the gate sets used to construct ansatz circuits are necessarily infinite. In contrast, the gate sets implemented by real quantum computers are finite, due to limitations related to error-correction [15]. This means that for each parameter refinement, the ansatz circuit must be recompiled and optimized again. However, the compilation and optimization of quantum circuits are known to be highly error-prone [20, 50], so it is desirable to verify both the equivalence of the optimized circuit to the original circuit, and more generally, the correctness of each optimization. In both cases, it is necessary to reason equationally about parameterized relations between quantum circuits.

The problem of parameterized equivalence-checking has been well-studied in the context of distributed system. Given a set of parameters P and two programs parameterized by P, say C_1 and C_2, the parameterized-equivalence checking problem asks whether $C_1(\theta) = C_2(\theta), \forall \theta \in P$. When P is finite, this problem can be solved by simply testing the elements of P. When P is infinite, one approach to this problem is to find a cutoff n for which checking the equivalence of C_1 and C_2 for n distinct elements of P implies the equivalence of C_1 and C_2 for all elements of P [16]. Formally, one tries to find an $n \in \mathbb{N}$ such that for all $D \subseteq P$, if $|D| \geq n$, then $\forall \theta \in D \cdot C_1(\theta) = C_2(\theta)$ implies $\forall \theta \in P \cdot C_1(\theta) = C_2(\theta)$. Typically, the choice of n (and sometimes even D) will depend on both C_1 and C_2, and therefore this technique requires domain-specific insights (see, e.g., [2, 22, 25, 27, 34, 45]). When n becomes intractably large, probabilistic techniques have also been employed [13].

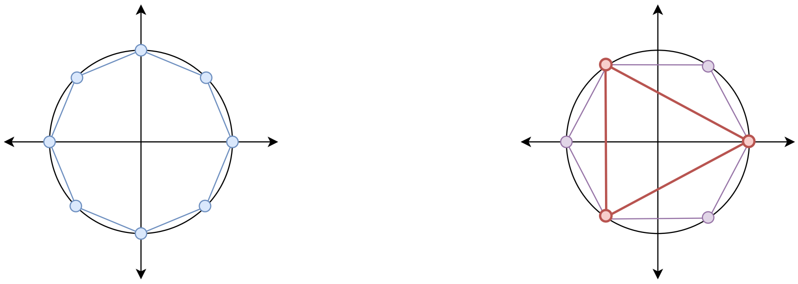
Cutoff-based techniques have yet to see wide application in the domain of parameterized quantum circuit equivalence-checking. In 2020, Miller-Bakewell developed a framework which adapts cutoff-based techniques to quantum circuits [32], though these techniques have yet to be applied in practice. The key insight of this work was to note that parameterized quantum circuits are analytic for realistic gate sets, and (up to a change of variable) can often be expressed as matrices over complex Laurent polynomials. The positive and negative degrees of these Laurent polynomials can be over-approximated in an inductive manner, and correspond to a cutoff for parameterized verification. The main challenge in applying the Miller-Bakewell framework is to identify an appropriate change-of-variables such that all parameterized matrices become matrices over complex Laurent polynomials. Once this change-of-variable has been identified, further steps may be taken, such as deriving a closed-form equation for the cutoff. In Miller-Bakewell's paper, the framework was applied to ZX-, ZW-, and ZH-diagrams, though closed-form bounds were not derived.

In this paper, we propose a cutoff-based technique for quantum circuits with arbitrary rotations with linear arguments. This technique can be understood as an instantiation of the Miller-Bakewell framework, insofar as each parameterized circuit is realized as a matrix over complex Laurent polynomials. However, the circuits considered in this paper correspond to ZXW-diagrams (i.e., with matrix exponentiation) [42], which are not addressed in Miller-Bakewell's original work. We derive closed-form equations for these cutoffs, which depend only on the coefficients of the parameters in the circuits. Furthermore, we provide an alternative proof for the correctness of the Miller-Bakewell framework, which depends on the distribution of zeros of Laurent polynomials as opposed to polynomial interpolation. This change in perspective motivates a probabilistic variant of the Miller-Bakewell framework, which is applicable for circuits with intractably large cutoffs.

In Sec. 3, we provide the syntax and semantics for our circuit language. In Sec. 4, we illustrate our technique on a simple real-world example. In Sec. 5, we prove a cutoff theorem, and propose a probabilistic variant. In Sec. 6, we identify and solve several challenges faced when implementing this technique. All appendices can be found in the full paper [39].

2 Background

We write $\mathbb{N}$ for the set of natural numbers (including zero), $\mathbb{Z}$ for the set of integers, $\mathbb{Q}$ for the set of rational numbers, $\mathbb{R}$ for the set of real numbers, and $\mathbb{C}$ for the set of complex numbers. If $z \in \mathbb{C}$, then $\overline{z}$ denotes the complex conjugate of z. If $n \in \mathbb{N}$, then $[n]$ denotes the set $\{j \in \mathbb{N} : 1 \leq j \leq n\}$ so that $[0] = \varnothing$. If $a \in \mathbb{R}$, then $a^+ = \max(0, a)$ and $a^- = \min(0, a)$.



(a) The roots of unity in $\mathbb{Q}(\zeta_8)$.

(b) $\mathbb{Q}(\zeta_6) = \mathbb{Q}(\zeta_3)$ since $\zeta_6 = -(\zeta_3)^2$.

Figure 1 Geometry of the cyclotomic numbers. The basis vectors of $\mathbb{Q}[\zeta_n]$ form the vertices of a regular n-gon on the complex unit circle, with one vertex at $(1, 0)$.

2.1 Linear Algebra

We assume familiarity with the basics of linear algebra. Otherwise, we refer the reader to an introductory text, such as [7]. Let M be a complex matrix. We let $M_{j,k}$ denote the entry of M in the j-th row and the k-th column. We recall the following definitions. The *conjugate of* M is the matrix $\overline{M}$ such that $\overline{M}_{j,k} = \overline{M_{j,k}}$. The *transpose of* M is the matrix M^T such that $(M^T)_{j,k} = M_{k,j}$. The *adjoint of* M is the matrix $\overline{M}^T$, and is denoted $M^\dagger$. A matrix H is called *Hermitian* if $H = H^\dagger$. A matrix U is called *unitary* if U is invertible and $U^{-1} = U^\dagger$.

2.2 Algebraic Numbers and Computation

We assume the reader is familiar with field theory, as found in standard abstract algebra textbooks, such as [17]. Let $\mathbb{F}$ be a subfield of $\mathbb{K}$. An element $\alpha \in \mathbb{K}$ is *algebraic over* $\mathbb{F}$ if there exists a polynomial $p \in \mathbb{F}[x]$ such that $p(\alpha) = 0$. We write $\mathbb{F}(\alpha)$ to denote the smallest subfield of $\mathbb{K}$ containing both $\mathbb{F}$ and α. If $\deg(p) = n$, then it can be shown that the elements of $\mathbb{F}(\alpha)$ form a finite-dimensional $\mathbb{F}$-vector space with basis vectors $\{1, \alpha, \alpha^2, \ldots, \alpha^{n-1}\}$. Furthermore, this vector space forms an $\mathbb{F}$-algebra under the multiplication of $\mathbb{F}(\alpha)$. In the case where $\mathbb{F} = \mathbb{Q}$ and $\mathbb{K} = \mathbb{C}$, we say that α is an *algebraic number.* The field of all algebraic numbers is denoted $\mathbb{Q}^{\mathrm{Alg}}$. Algebraic numbers are ideal from a computational perspective, since elements from n-dimensional $\mathbb{Q}$-vector spaces can be represented exactly using only $2n$ integers (i.e., the numerators and denominators). This is in contrast to floating-point arithemtic, which is inherently inexact.

A special class of algebraic numbers are the *cyclotomic numbers.* These are solutions to polynomial equations of the form $x^n - 1 = 0$. In other words, each cyclotomic number is a root of unity. We let ζ_n denote the *primitive n-th root of unity*, which can be defined analytically as $\zeta_n = e^{i2\pi/n}$. For example, $\zeta_2 = -1$ and $\zeta_4 = i$. The smallest subfield of $\mathbb{C}$ containing $\mathbb{Q}$ and all cyclotomic numbers is referred to as the *universal cyclotomic field.* Many algorithms exist to work efficiently with elements of the universal cyclotomic field, such as [10] and [11]. It is well-known that many quantum gate sets can be defined exactly using only finite-dimensional sub-fields of the universal cyclotomic field, such as the Clifford+T gate set [18] and its generalizations [4]. For this reason, recent work in the verification of quantum programs has advocated for the use of cyclotomic numbers as an exact representation [6].

In this paper, we also utilize analytic properties of cyclotomic numbers. It follows from Euler's formula that $e^{i\theta} = \cos(\theta) + i\sin(\theta)$. We can then think of each cyclotomic number as a point of the complex unit circle (see Figure 1a). It follows geometrically that

$\mathbb{Q}(\zeta_n) = \mathbb{Q}(\zeta_{2n})$ whenever n is odd (see Figure 1b). Moreover, it can be shown by simple algebraic manipulations that the following equations hold.

$$\cos(\theta) = \frac{e^{i\theta} + e^{-i\theta}}{2} \qquad \sin(\theta) = \frac{e^{i\theta} - e^{-i\theta}}{2i}$$

If θ is a rational multiple of π, say $(q/r)2\pi$, this means that both $\cos(\theta)$ and $\sin(\theta)$ are elements of $\mathbb{Q}(i, \zeta_r)$. However, identifying roots of unity can be challenging, since not all elements of norm 1 in the universal cyclotomic field are roots of unity. A well-known example is $(3 + 4i)/5$, which has norm 1 but is not a root of unity.

2.3 Multivariate Laurent Polynomials

Let R be a ring. Then $R[x_1, \ldots, x_k]$ denotes the *ring of multivariate polynomials* with coefficients in R and indeterminates x_1 through x_k. An arbitrary element $f \in R[x_1, \ldots, x_k]$ is of the form $f(x_1, \ldots, x_k) = \sum_{t \in T}(a_t \prod_{j=1}^{k} {x_j}^{t_j})$ for some finite $T \subseteq \mathbb{N}^k \setminus \{0\}^k$ with a non-zero sequence $\{a_t\}_{t \in T}$ over R. We write $\deg_{x_j}(f)$ for the degree of f in variable x_j and $\deg(f)$ for the total degree of f, where $\deg_{x_j}(f) = \max\{t_j : t \in T\}$ and $\deg(f) = \max\{\sum_{j=1}^{k} t_j : t \in T\}$. When R is an integral domain, the following hold for all $f, g \in R[x_1, \ldots, x_k]$ and $j \in [k]$.

$$\deg_{x_j}(fg) = \deg_{x_j}(f) + \deg_{x_j}(g) \qquad \deg(fg) = \deg(f) + \deg(g)$$
$$\deg_{x_j}(f + g) \leq \max\{\deg_{x_j}(f), \deg_{x_j}(g)\} \qquad \deg(f + g) \leq \max\{\deg(f), \deg(g)\}$$

It is well known that when $k = 1$ and R is an integral domain, either $f = 0$ or f has at most $\deg(f)$ zeros. A consequence is that for any $S \subseteq R$, if $f \neq 0$ and $|S| > \deg_{x_1}(f)$, then there exists an $s \in S$ such $f(s) \neq 0$. Moreover, if s is sampled uniformly from S, then $\Pr(f(x) = 0) \leq \deg(f)/|S|$. The latter two remarks generalize to multivariate polynomials. Further generalization to Laurent polynomials are possible, by clearing the denominators.

▶ **Theorem 2.1** (Combinatorial Nullstellensatz [3]). *Let $\mathbb{F}$ be a field and f a polynomial in $\mathbb{F}[x_1, x_2, \ldots, x_k]$ with total degree $d_1 + d_2 + \cdots + d_k$ such that the coefficient of $\prod_{j=1}^{k} x_j^{d_j}$ is nonzero in f. If $S_1, S_2, \ldots, S_k$ are subsets of $\mathbb{F}$ with $|S_j| > d_j$ for each j, then there exists $x \in S_1 \times S_2 \times \cdots \times S_k$ such that $f(x) \neq 0$.*

▶ **Theorem 2.2** (DeMillo–Lipton–Schwartz–Zippel Lemma [13, 41, 51]). *Let R be an integral domain and $f \in R[x_1, x_2, \ldots, x_k]$ a polynomial with total degree d. For each finite subset S of R, if $s_1, s_2, \ldots, s_k$ are sampled at random, both independently and uniformly from S, then* $\Pr(f(s_1, s_2, \ldots, s_k) = 0) \leq d/|S|$.

We can further generalize multivariate polynomials to multivariate Laurent polynomials, denoted $R[x_1, x_1^{-1}, \ldots, x_k, x_k^{-1}]$. In this setting, $T \subseteq \mathbb{Z}^k$, so that powers may be positive or negative. For example, $f(x_1, x_2) = x_1 x_2 - x_1^{-3} + 5$ is a Laurent polynomial from $\mathbb{Z}[x_1, x_1^{-1}, x_2, x_2^{-1}]$. Since the exponents in a Laurent polynomial may be both positive and negative, each Laurent polynomial has both positive and negative degrees. We write $\deg^+_{x_j}(f)$ for the positive degree of f in variable x_j and $\deg^-_{x_j}$ for the negative degree of f in variable x_j, where $\deg^+_{x_j}(f) = \max\{t_j^+ : t \in T\}$ and $\deg^-_{x_j}(f) = \max\{-t_j^- : t \in T\}$. Similarly, the total positive degree of f is $\deg^+(f) = \max\{\sum_{j=1}^{k} t_j^+ : t \in T\}$.

3 A Syntax and Semantics for Parameterized Circuits

This section begins by reviewing quantum states, quantum operators, and their composition, as in [35, Ch. 4]. This background material is then used to give syntax and parameterized semantics for quantum circuits with arbitrary gates, and rotations around arbitrary axes.

3.1 Quantum States

The primitive unit of information in quantum computing is the qubit. As in classical computing, a qubit can be in the states zero and one, denoted $|0\rangle$ and $|1\rangle$. However, a qubit may also be in a *superposition* of the states $|0\rangle$ and $|1\rangle$. Formally, this means that the state of a qubit $|\psi\rangle$ can be described as $\alpha|0\rangle + \beta|1\rangle$ for any $\alpha \in \mathbb{C}$ and $\beta \in \mathbb{C}$ satisfying $|\alpha|^2 + |\beta|^2 = 1$. To simplify calculations, we think of $|0\rangle$ and $|1\rangle$ as the standard basis vectors for $\mathbb{C}^2$ to obtain the following vector equation: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right] + \beta\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right] = \left[\begin{smallmatrix}\alpha\\\beta\end{smallmatrix}\right]$.

Of course, the quantum algorithms described in the introduction of this paper require more than a single qubit of information. Given an n-qubit quantum system, there are clearly 2^n possible basis states. For example, when $n = 2$, these are $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. As before, an n-qubit quantum system may also be in an arbitrary superposition of these basis states with the modulus-squared of the coefficients summing to 1. For example, an arbitrary 2-qubit quantum system has state $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \rho|11\rangle$ for any $\alpha, \beta, \gamma, \rho \in \mathbb{C}$ satisfying $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\rho|^2 = 1$. This means that the states of an n-qubit quantum system correspond to the unit vectors in $\mathbb{C}^{2^n}$.

3.2 Quantum Operations

A quantum program evolves the state of a quantum system, after which all qubits are measured. Given a quantum state $|\psi\rangle = \sum_{j=1}^{2^n} \alpha_j |j\rangle$, the probability of observing state $|j\rangle$ is $|\alpha_j|^2$. Then the paradigm of quantum computing is to construct an n-qubit quantum system whose probability distribution assigns high probability to the correct output.

The evolution of a quantum system is described by a linear transformation of its state space. Since the laws of physics are reversible, then this transformation must be invertible. Moreover, the inverse of this transformation should be its conjugate transpose. This means that operations on n-qubit systems correspond to unitary matrices. Given an n-qubit state $|\psi\rangle$ and an $(2^n) \times (2^n)$ dimensional matrix M, the state obtained by applying M to $|\psi\rangle$ is $M|\psi\rangle$. For example, the following four matrices are unitary operations on a qubit.

$$I = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \qquad X = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix} \qquad Z = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix} \qquad Y = \begin{bmatrix}0 & -i\\i & 0\end{bmatrix}$$

The matrix I corresponds to a no-op and the matrix X corresponds to a not gate. The matrix Z can be understood as adjusting the coefficient of $|1\rangle$ by a factor of (-1). This has no classical analogue. The gate Y is equal to $(-iZ)X$, and therefore, corresponds to a not gate followed by some non-classical operation.

An important construct in classical computing is the if-then statement. This can be generalized to quantum computing as follows. Let M be a unitary transformation on an n-qubit quantum system. Then there exists a unitary transformation $I_{2^n} \oplus M$ on an $(n+1)$-qubit quantum system, such that $I_{2^n} \oplus M$ applies M to the last n qubits of a basis state if and only if the first qubit of the basis is in state $|1\rangle$. Formally, I_{2^n} is the $(2^n) \times (2^n)$ identity matrix, and $I_{2^n} \oplus M$ is the direct sum of I_{2^n} with M. In terms of matrices, $I_{2^n} \oplus M$ is simply the block diagonal matrix with blocks I_{2^n} and M, as shown below.

$$I_{2^n} \oplus M = \begin{bmatrix}I_{2^n} & 0\\0 & M\end{bmatrix} \qquad I_2 \oplus X = \begin{bmatrix}I_2 & 0\\0 & X\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}$$

The matrix for $I_2 \oplus X$, known as a *cnot gate*, is given above. This generalizes the classical conditional statement: if the first bit is in state $|1\rangle$, then apply a not gate to the second bit.

So far, all of the operations discussed are parameter-free. However, quantum algorithms also make use of rotation gates, which are parameterized by an angle of rotation. As the name suggests, a rotation gate is defined by its axis-of-rotation. Formally, each axis M is a Hermitian unitary matrix. Then one can define, as a generalization of Euler's formula, the rotation $R_M(\theta)$ as follows.

$$R_M(\theta) = e^{-iM\theta/2} = \sum_{n=0}^{\infty} \frac{(-iM\theta/2)^n}{n!} = \cos(-\theta/2)I + i\sin(-\theta/2)M$$

This definition can be extended to k parameters by taking any transformation $f : \mathbb{R}^k \to \mathbb{R}$. For example, given $f(\theta_1, \theta_2) = \theta_1 + \theta_2$, we can define a two parameter rotation $R_M(f)$ where $R_M(f)(\theta_1, \theta_2) = R_M(f(\theta_1, \theta_2)) = R_M(\theta_1 + \theta_2)$. In this work, we consider the family $\mathcal{F}$ of k-variable rational-linear functions with affine translations by rational multiples of π. That is, the set $\mathcal{F}$ is defined to be $\{f(\theta) = a_1\theta_1 + a_2\theta_2 + \cdots + a_k\theta_k + q\pi \mid a_1, a_2, \ldots, a_k, q \in \mathbb{Q}\}$.

The most common rotations in quantum circuits are the I-, X-, Y-, and Z-rotations. However, there are many single qubit rotations not of this form. For example, given any coefficients $\alpha, \beta, \gamma \in \mathbb{R}$, if $\alpha^2 + \beta^2 + \gamma^2 = 1$, the matrix $\alpha X + \beta Y + \gamma Z$ is also a Hermitian unitary matrix. Note that the matrix $R_I(-2\theta)$ is typically referred to as a *global phase gate*, rather than an I-rotation.

▶ **Example 3.1** (Real Amplitude Ansatz Circuit). In quantum machine learning, convolutional layers are often implemented using the real amplitude ansatz circuit [1, 5, 26, 31, 48]. This circuit is composed from one or more layers of Z-rotations, each followed by a layer of controlled-not gates. Since Z-rotations do not commute with the targets of controlled-not gates, then these layers can interact in non-trivial ways. The choice of parameter to each Z-rotation is treated as a weight in the quantum machine learning model.

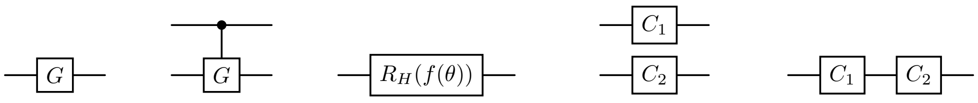
3.3 Composing Quantum Operations

Just like classical operations, quantum operations can also be composed in sequence and in parallel. Of the two, sequential composition is the simplest to describe. Assume that both M and N are operations on an n-qubit quantum system. If N is applied to an n-qubit system $|\psi\rangle$, then the state $N|\psi\rangle$ is obtained. If M is then applied to this intermediate state, then the state $M(N|\psi\rangle)$ is obtained. This is equivalent to applying MN to $|\psi\rangle$. In other words, the sequential composition of quantum operations corresponds to matrix multiplication.

Now let M denote a quantum operation on an m-qubit quantum system and N denote a quantum operation on an n-qubit quantum system. Intuitively, the parallel composition of M and N should act on the first m-qubits by M, and the last n-qubits by N. However, this composition must also respect superposition, through a property known an *bilinearity.* To compute this new operation, the *Kronecker tensor product* is required, which is denoted $\otimes$ and defined as follows for matrices of any dimension.

$$\begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,n} \end{bmatrix} \otimes M = \begin{bmatrix} c_{1,1}M & c_{1,2}M & \cdots & c_{1,n}M \\ c_{2,1}M & c_{2,2}M & \cdots & c_{2,n}M \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1}M & c_{m,2}M & \cdots & c_{m,n}M \end{bmatrix}$$

It follows that $(M \otimes N)(|\psi\rangle \otimes |\varphi\rangle) = (M|\phi\rangle) \otimes (N|\varphi\rangle)$ as desired.



(a) $G \in \mathcal{G}$. **(b)** $C(G)$. **(c)** $R_H(f)$. **(d)** $C_1//C_2$. **(e)** $C_2 \circ C_1$.

Figure 2 The graphical language for circuits in $\mathsf{Circ}(\mathcal{G}, \mathcal{H})$.

3.4 Quantum Circuits

Quantum circuits are constructed from primitive gates, under sequential and parallel composition. In this section, we first define what we take to be primitive gates, and then define what it means to be a circuit over this gate set. The distinction between syntax and semantics is emphasized. In both cases, we introduce inductive principles which will be used later in this paper. Formally, these circuits correspond to diagrams in a certain PROP category [8], with semantics given functorially [28], though this is only used to prove the inductive principles used throughout the paper, and to establish that our semantics and circuit transformations are well-defined (see the full paper for more details).

In what follows, $C(-)$ is a function symbol used to denote conditional control. A gate set is a collection of basic gates, closed under conditional control. A basic gate is a complex matrix (e.g. unitary operations, state preparation, post-selection) or parameterized rotation. Formally, we take some set $\mathcal{G}$ of complex matrices and some set $\mathcal{H}$ of Hermitian unitary matrices. The associated gate set, denoted $\Sigma(\mathcal{G}, \mathcal{H})$ is defined inductively as follows.

- If $G \in \mathcal{G}$, then $G \in \Sigma(\mathcal{G}, \mathcal{H})$.
- If $M \in \mathcal{H}$, then $R_M(f) \in \Sigma(\mathcal{G}, \mathcal{H})$ for each parameterization $f \in \mathcal{F}$.
- If $G \in \Sigma(\mathcal{G}, \mathcal{H})$ and G is unitary, then $C(G) \in \Sigma(\mathcal{G}, \mathcal{H})$.

We let $\mathsf{in}(-)$ and $\mathsf{out}(-)$ denote the input and output arities of these gates, which are defined as follows.

- If $G \in \mathcal{G}$ is $(2^n) \times (2^m)$, then $\mathsf{in}(G) = n$ and $\mathsf{out}(G) = m$.
- If $M \in \mathcal{H}$ is $(2^n) \times (2^n)$ and $f \in \mathcal{F}$, then $\mathsf{in}(R_M(f)) = \mathsf{out}(R_M(f)) = n$.
- If $G \in \Sigma(\mathcal{G}, \mathcal{H})$, then $\mathsf{in}(C(G)) = \mathsf{in}(G) + 1$ and $\mathsf{out}(C(G)) = \mathsf{out}(G) + 1$.

We let $[\![-]\!]$ denote the parameterized semantics of each gate, which are defined as expected.

- If $G \in \mathcal{G}$, then $[\![G]\!](\theta) = G$.
- If $M \in \mathcal{H}$ and $f \in \mathcal{F}$, then $[\![R_M(f)]\!](\theta) = \cos(-f(\theta)/2)I + i\sin(-f(\theta)/2)M$.
- If $G \in \Sigma(\mathcal{G}, \mathcal{H})$ with G an $(2^n) \times (2^n)$ unitary, then $[\![C(G)]\!](\theta) = I_{2^n} \oplus [\![G]\!](\theta)$.

Since this gate set is defined inductively, then to prove that every gate satisfies a predicate P, it suffices to use well-founded induction (see the full paper).

▶ **Proposition 3.2.** *Assume that a predicate P on $\Sigma(\mathcal{G}, \mathcal{H})$ satisfies the following.*

- ***Base Case (1).*** *$\forall G \in \mathcal{G}, P(G)$.*
- ***Base Case (2).*** *$\forall M \in \mathcal{H}, \forall f \in \mathcal{F}, P(R_M(f))$.*
- ***Control Induction.*** *$\forall G \in \Sigma(\mathcal{G}, \mathcal{H})$, G unitary and $P(G)$ implies $P(C(G))$.*

Then $P(G)$ holds for each $G \in \Sigma(\mathcal{G}, \mathcal{H})$.

Circuits are then constructed from the elements of $\Sigma(\mathcal{G}, \mathcal{H})$ through sequential and parallel composition. We let $(\circ)$ denote sequential composition and $(//)$ denote parallel composition, to distinguish between syntactic compositions and their semantic counterparts. Of course, sequential composition requires that the outputs of the first sub-circuit matches the inputs of the second sub-circuit. To handle this, we extend $\mathsf{in}(-)$ and $\mathsf{out}(-)$ as follows.

- $\mathsf{in}(C_1//C_2) = \mathsf{in}(C_1) + \mathsf{in}(C_2)$ and $\mathsf{out}(C_1//C_2) = \mathsf{out}(C_1) + \mathsf{out}(C_2)$.
- $\mathsf{in}(C_2 \circ C_1) = \mathsf{in}(C_1)$ and $\mathsf{out}(C_2 \circ C_1) = \mathsf{out}(C_2)$.

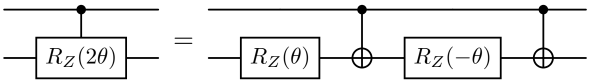


Figure 3 A parameterized equality used to compile controlled rotations.

Then $\mathsf{Circ}(\mathcal{G},\mathcal{H})$, the family of circuits over the gate set $\Sigma(\mathcal{G},\mathcal{H})$, is defined inductively as follows where ϵ denotes the *empty* wire with $\mathsf{in}(\epsilon) = \mathsf{out}(\epsilon) = 1$.

- If $C \in \Sigma(\mathcal{G},\mathcal{H})$, then $C \in \mathsf{Circ}(\mathcal{G},\mathcal{H})$.
- If $C_1, C_2 \in \mathsf{Circ}(\mathcal{G},\mathcal{H})$, then $C_1 // C_2 \in \mathsf{Circ}(\mathcal{G},\mathcal{H})$.
- If $C_1, C_2 \in \mathsf{Circ}(\mathcal{G},\mathcal{H})$ and $\mathsf{in}(C_2) = \mathsf{out}(C_1)$, then $C_2 \circ C_1 \in \mathsf{Circ}(\mathcal{G},\mathcal{H})$.

A graphical language for $\mathsf{Circ}(\mathcal{G},\mathcal{H})$ is given in Figure 2. The semantic map $[\![-]\!]$ extends to these circuits as expected: $[\![C_2//C_1]\!](\theta) = [\![C_2]\!](\theta) \otimes [\![C_1]\!](\theta)$, $[\![C_2 \circ C_1]\!](\theta) = ([\![C_2]\!](\theta))([\![C_1]\!](\theta))$, and $[\![\epsilon]\!] = I_2$. As with quantum gates, an inductive principle also holds for quantum circuits.

▶ **Proposition 3.3.** *Assume that a predicate P on $\mathsf{Circ}(\mathcal{G},\mathcal{H})$ satisfies the following.*

- ***Base Case (1).*** *$P(\epsilon)$.*
- ***Base Case (2).*** *$\forall G \in \Sigma(\mathcal{G},\mathcal{H}), P(G)$.*
- ***Parallel Induction.*** *If $C_1, C_2 \in \mathsf{Circ}(\mathcal{G},\mathcal{H})$ such that $P(C_1)$ and $P(C_2)$, then $P(C_1//C_2)$.*
- ***Sequential Induction.*** *If $C_1, C_2 \in \mathsf{Circ}(\mathcal{G},\mathcal{H})$ such that $\mathsf{in}(C_2) = \mathsf{out}(C_1)$ with $P(C_1)$ and $P(C_2)$, then $P(C_2 \circ C_1)$.*

Then $P(C)$ holds for each $C \in \mathsf{Circ}(\mathcal{G},\mathcal{H})$.

4 A Motivating Example: Circuit Compilation

We now discuss the verification of a concrete circuit equation. The example is simple but illustrative of the techniques we will develop in the next section. Consider the equation in Figure 3. A naive approach to establishing this equation is to evaluate the right-hand side to obtain the following operator.

$$(I \oplus X)(I \otimes R_Z(-\theta))(I \oplus X)(I \otimes R_Z(\theta)) = \begin{bmatrix} R_Z(-\theta)R_Z(\theta) & 0 \\ 0 & XR_Z(-\theta)XR_Z(\theta) \end{bmatrix}$$

Then, by further simplification, we obtain the following equations.

$$R_Z(-\theta)R_Z(\theta) = \begin{bmatrix} e^{-i\theta/2}e^{i\theta/2} & 0 \\ 0 & e^{i\theta/2}e^{-i\theta/2} \end{bmatrix} \quad XR_Z(-\theta)XR_Z(\theta) = \begin{bmatrix} e^{-i\theta/2}e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2}e^{i\theta/2} \end{bmatrix}$$

Using the identities $e^a e^b = e^{a+b}$ and $e^0 = 1$, it then follows that $XR_Z(-\theta)XR_Z(\theta) = R_Z(2\theta)$ and $R_Z(-\theta)R_Z(\theta) = I$. Consequently,

$$(I \oplus X)(I \otimes R_Z(-\theta))(I \oplus X)(I \otimes R_Z(\theta)) = \begin{bmatrix} I & 0 \\ 0 & R_Z(2\theta) \end{bmatrix} = (I \oplus R_Z(2\theta)).$$

This establishes the equation in Figure 3 for all choices of θ. However, this proof depends on the parameterized equations $e^{a+b} = e^a e^b$ and $e^0 = 1$. In general, it is challenging to find a complete set of parameterized relations for a parameterized gate set [33]. Moreover, given an arbitrary set of complete relations, the problem of deciding if two expressions are equivalent is known to be undecidable [36]. For these reasons, we adopt a different approach.

A perhaps surprising result is that all parameterized circuit equalities can be established by checking only a finite number of rotation angles. In other words, if the equality in Figure 3 did not hold, then a counterexample could be found by checking only a fixed number of instances. To do this, we first convert the equality into a family of polynomials, such that the equality holds if and only if all of the polynomials are identically zero. We then find an integer n such that each of the polynomials has degree at most n. Since non-zero polynomials of degree n have at most n roots, then either the polynomial is zero and will evaluate to zero on $n+1$ angles, or the polynomial is non-zero and at least one of the $n+1$ angles yields a non-zero result.

To obtain the desired polynomials, we apply the change-of-variable $e^{-i\theta/2} \mapsto z$. Under this change of variable, the following equalities hold.

$$R_Z(-\theta)R_Z(\theta) = \begin{bmatrix} z^{-1}z & 0 \\ 0 & zz^{-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = z^{-2}\begin{bmatrix} z^2 & 0 \\ 0 & z^2 \end{bmatrix}$$

$$XR_Z(-\theta)XR_Z(\theta) = \begin{bmatrix} z^{-1}z^{-1} & 0 \\ 0 & zz \end{bmatrix} = \begin{bmatrix} z^{-2} & 0 \\ 0 & z^2 \end{bmatrix} = z^{-2}\begin{bmatrix} 1 & 0 \\ 0 & z^4 \end{bmatrix}$$

Continuing in this fashion, we can find that each matrix entry on the left-hand side or the right-hand side of Figure 3 has degree at most four. Then the difference between the left-hand side and the right-hand side also has degree at most four. Note that the z^{-2} terms correspond to a removable singularity at $z = 0$, which does not fall on the complex unit circle, and can be safely ignored. Since degree four polynomials have at most four roots, then it suffices to check the equality in Figure 3 using only 5 angles from $[0, 4\pi)$. For example, consider the five angles $\theta_j = j\pi/2$ for $0 \leq j \leq 4$. It is easily verified that $(I \oplus R_Z(2\theta_j)) = (I \oplus X)(I \otimes R_Z(-\theta_j))(I \oplus X)(I \otimes R_Z(\theta_j))$ for all $0 \leq j \leq 4$. Then $f(\theta) = (I \oplus R_Z(2\theta)) - (I \oplus X)(I \otimes R_Z(-\theta))(I \oplus X)(I \otimes R_Z(\theta))$ has at least five roots. Since each entry of $f(\theta)$ has degree at most four, then f is identically zero and Figure 3 must hold. Note that the angles were sampled from $[0, 4\pi)$ since $e^{-i\theta_j/2}$ has period 4π.

While this example was admittedly simplistic, we will see in the next section, that the technique generalizes to all parameterized circuits. In particular, just as in this example, we will see that computing the polynomials is inconsequential. Instead, it will suffice to find an efficient procedure which provides a reason bound on each degree.

5 Equivalence Checking Techniques

In this section, we consider parameterized quantum circuits where all coefficients are from $\mathbb{Z}$, rather than $\mathbb{Q}$. We denote these circuits $\mathsf{Circ}_{\mathbb{Z}}(\mathcal{G}, \mathcal{H})$. It is first shown that up to a change of variable, these circuits admit semantics as matrices over the ring of Laurent polynomials $\mathbb{C}[z_1, z_1^{-1}, \ldots, z_k, z_k^{-1}]$. This is then combined with Thm. 2.1 to establish a cutoff-based equivalence checking theorem for these circuits. Using Thm. 2.2, a probabilistic variant is also obtained. In Sec. 6, we show how these results generalize back to parameterized circuits with rational coefficients.

5.1 Polynomial Semantics

This section shows that, up to a change of variable, each circuit $\mathsf{Circ}(\mathcal{G}, \mathcal{H})$ has semantics given by a matrix with entries corresponding to complex Laurent polynomials. Moreover, these polynomials are shown to have degrees bounded by certain sums of the coefficients which appear in the circuit. It follows that the techniques used in Sec. 4 can be generalized to all integral circuits in $\mathsf{Circ}_{\mathbb{Z}}(\mathcal{G}, \mathcal{H})$.

As a first step, a new semantic interpretation $[\![-]\!]_{\mathsf{Poly}}$ is provided for $\mathsf{Circ}_{\mathbb{Z}}(\mathcal{G}, \mathcal{H})$, which interprets each circuit in $\mathsf{Circ}_{\mathbb{Z}}(\mathcal{G}, \mathcal{H})$ as a polynomial over $\mathbb{C}[z_1, z_1^{-1}, \ldots, z_k, z_k^{-1}]$. Since parameters only appear in trigonometric terms, then a first step is to give Laurent polynomials which abstract the trigonometric terms. Let $\alpha \in \mathbb{Z}^k$, $q \in \mathbb{Q}$, and $f(\theta) = \alpha_1\theta_1 + \cdots \alpha_k\theta_k + q$.

$$\cos\left(-\frac{f(\theta)}{2}\right) = \frac{e^{i(-f(\theta)/2)} + e^{-i(-f(\theta)/2)}}{2} = \frac{e^{-iq/2}}{2}\prod_{j=1}^{k}\left(e^{-i\theta_j/2}\right)^{a_j} + \frac{e^{iq/2}}{2}\prod_{j=1}^{k}\left(e^{i\theta_j/2}\right)^{a_j}$$

$$\sin\left(-\frac{f(\theta)}{2}\right) = \frac{e^{i(-f(\theta)/2)} - e^{-i(-f(\theta)/2)}}{2i} = \frac{e^{-iq/2}}{2i}\prod_{j=1}^{k}\left(e^{-i\theta_j/2}\right)^{a_j} - \frac{e^{iq/2}}{2i}\prod_{j=1}^{k}\left(e^{i\theta_j/2}\right)^{a_j}$$

By substituting $z_j = e^{-i\theta_j/2}$ for each $j \in [k]$ and letting $c = e^{-iq/2}$, the following Laurent polynomials are obtained.

$$\mathsf{CPoly}(f) = \frac{c}{2}\prod_{j=1}^{k} z_j^{\alpha_j} + \frac{1}{2c}\prod_{j=1}^{k} z_j^{-\alpha_j} \qquad \mathsf{SPoly}(f) = \frac{-ic}{2}\prod_{j=1}^{k} z_j^{\alpha_j} + \frac{i}{2c}\prod_{j=1}^{k} z_j^{-\alpha_j}$$

Then the following equations hold by construction.

$$\mathsf{CPoly}(f)\left(e^{-i\theta_1/2},\dots,e^{-i\theta_k/2}\right) = \frac{e^{-iq/2}}{2}\prod_{j=1}^{k}\left(e^{-i\theta_j/2}\right)^{a_j} + \frac{e^{iq/2}}{2}\prod_{j=1}^{k}\left(e^{i\theta_j/2}\right)^{a_j} = \cos\left(-\frac{f(\theta)}{2}\right)$$

$$\mathsf{SPoly}(f)\left(e^{-i\theta_1/2},\dots,e^{-i\theta_k/2}\right) = \frac{e^{-iq/2}}{2i}\prod_{j=1}^{k}\left(e^{-i\theta_j/2}\right)^{a_j} - \frac{e^{iq/2}}{2i}\prod_{j=1}^{k}\left(e^{i\theta_j/2}\right)^{a_j} = \sin\left(-\frac{f(\theta)}{2}\right)$$

Given these polynomials, $[\![-]\!]_{\mathsf{Poly}}$ is defined inductively on the gates as follows.

- If $G \in \mathcal{G}$, then $[\![G]\!]_{\mathsf{Poly}} = G$.
- If $M \in \mathcal{H}$ and $f \in \mathcal{F}$, then $[\![R_M(f)]\!]_{\mathsf{Poly}} = \mathsf{CPoly}(f)I + i\,\mathsf{SPoly}(f)M$.
- If $G \in \Sigma(\mathcal{G},\mathcal{H})$ with G an $(2^n) \times (2^n)$ unitary, then $[\![C(G)]\!]_{\mathsf{Poly}} = I_{2^n} \oplus [\![G]\!]_{\mathsf{Poly}}$.

The semantics extend as expected to sequential and parallel composition. This makes precise the change of variable used in Sec. 4.

▶ **Definition 5.1** (Polynomial Abstraction). *A* polynomial abstraction *is a function $[\![-]\!]_*$ from $\mathsf{Circ}_{\mathbb{Z}}(\mathcal{G},\mathcal{H})$ to collection of matrices over $\mathbb{C}[z_1, z_1^{-1}, \dots, z_k, z_k^{-1}]$ such that $[\![C]\!](\theta_1,\dots,\theta_k) = [\![C]\!]_*\left(e^{-i\theta_1/2},\dots,e^{-i\theta_k/2}\right)$ for all $C \in \mathsf{Circ}_{\mathbb{Z}}(\mathcal{G},\mathcal{H})$.*

▶ **Theorem 5.2.** *$[\![-]\!]_{\mathsf{Poly}}$ is a polynomial abstraction.*

▶ **Example 5.3** (Polynomial Semantics). The calculations from Sec. 4 can be revisited from the perspective of polynomial semantics. Of course, the circuit in Figure 3 is somewhat uninteresting, since the circuit has only one parameter. Instead, we will consider a new circuit with two parameters ρ_1 and ρ_2 obtained through the substitution $\theta = f(\rho_1,\rho_2)$ where $f(\rho_1,\rho_2) = \rho_1 - 2\rho_2$. The sine and cosine polynomials for f are as follows.

$$\mathsf{CPoly}(f) = \frac{1}{2}z_1z_2^{-2} + \frac{1}{2}z_1^{-1}z_2^{2} \qquad \mathsf{SPoly}(f) = \frac{-i}{2}z_1z_2^{-2} + \frac{i}{2}z_1^{-1}z_2^{2}$$

Then $\mathsf{CPoly}(f) + i\,\mathsf{SPoly}(f) = z_1z_2^{-2}$ and $\mathsf{CPoly}(f) - i\,\mathsf{SPoly}(f) = z_1^{-1}z_2^{2}$. Let C_1 denote the right-hand side of the equation in Figure 3. To compute $[\![C_1]\!]_{\mathsf{Poly}}$, we start by evaluating each gate. Clearly $[\![C(X)]\!]_{\mathsf{Poly}} = I_2 \oplus X$. Moreover,

$$[\![\epsilon//R_Z(f)]\!]_{\mathsf{Poly}} = I_2 \otimes [\![R_Z(f)]\!]_{\mathsf{Poly}} = I_2 \otimes \begin{bmatrix} z_1z_2^{-2} & 0 \\ 0 & z_1^{-1}z_2^{2} \end{bmatrix},$$

$$[\![\epsilon//R_Z(-f)]\!]_{\mathsf{Poly}} = I_2 \otimes [\![R_Z(-f)]\!]_{\mathsf{Poly}} = I_2 \otimes \begin{bmatrix} z_1^{-1}z_2^{2} & 0 \\ 0 & z_1z_2^{-2} \end{bmatrix}.$$

It follows by calculations similar to those in Sec. 4 that,

$$[\![C_1]\!]_{\mathsf{Poly}} = [\![C(X)]\!]_{\mathsf{Poly}}[\![\epsilon//R_Z(-f)]\!]_{\mathsf{Poly}}[\![C(X)]\!]_{\mathsf{Poly}}[\![\epsilon//R_Z(f)]\!]_{\mathsf{Poly}} = I_2 \oplus \begin{bmatrix} z_1^{-2}z_2^4 & 0 \\ 0 & z_1^2 z_2^{-4} \end{bmatrix}.$$

Then $[\![C_1]\!]_{\mathsf{Poly}}(e^{-i\rho_1/2}, e^{-i\rho_2/2}) = I \oplus R_Z(2f(\rho_1, \rho_2)) = [\![C_1]\!](\rho_1, \rho_2)$ as expected. ⌟

To check that $[\![C_1]\!] = [\![C_2]\!]$, it suffices to check symbolically that $[\![C_1]\!]_{\mathsf{Poly}} = [\![C_2]\!]_{\mathsf{Poly}}$. However, it is often too computationally expensive to compute the polynomials explicitly. Instead, one could first upper-bound the degree of each polynomial, and then combine these degree bounds with the theorems of Sec. 2.3. It is not hard to see that for each component of $[\![R_H(f)]\!]_{\mathsf{Poly}}$, its degrees are all bounded by the coefficients of f. This property extends to all circuits in $\mathsf{Circ}_{\mathbb{Z}}(\mathcal{G}, \mathcal{H})$ by studying their coefficient sequences. Intuitively, the coefficient sequence of a circuit C is a sequence $A(C)$ over $\mathbb{Q}^k$ such that $A(C)_j$ is the list of coefficients for the j-th rotation in C. More formally, let $(\mathbb{Q}^k)^*$ denote the set of all finite sequences over $\mathbb{Q}^k$ and $(\cdot)$ denote sequence concatenation. Then $A(-)$ is defined inductively as follows.

- If $G \in \mathcal{G}$, then $A(G) = \epsilon$.
- If $M \in \mathcal{H}$ and $f(\theta) = a_1\theta_1 + \cdots + a_k\theta_k + q$, then $A(R_M(f)) = ((a_1, \ldots, a_k))$.
- If $G \in \Sigma(\mathcal{G}, \mathcal{H})$, then $A(C(G)) = A(G)$.
- If $C_1, C_2 \in \mathsf{Circ}(\mathcal{G}, \mathcal{H})$, then $A(C_1 // C_2) = A(C_1) \cdot A(C_2)$.
- If $C_1, C_2 \in \mathsf{Circ}(\mathcal{G}, \mathcal{H})$ and $\mathsf{in}(C_2) = \mathsf{out}(C_1)$, then $A(C_2 \circ C_1) = A(C_2) \cdot A(C_1)$.

Then $\mathsf{Circ}_{\mathbb{Z}}(\mathcal{G}, \mathcal{H})$ is precisely the set of circuits in $\mathsf{Circ}_{\mathbb{Z}}(\mathcal{G}, \mathcal{H})$ such that $A(C) \in (\mathbb{Z}^k)^*$. We define $\Sigma_{\mathbb{Z}}(\mathcal{G}, \mathcal{H})$ analogously. The following definition generalizes the coefficient bound of the degree of a gate to a coefficient bound on the degree of all circuits.

▶ **Definition 5.4** (Coefficient Bounded Semantics). *Let $[\![-]\!]_*$ be a polynomial abstraction. A circuit $C \in \mathsf{Circ}(\mathcal{G}, \mathcal{H})$ with $\mathsf{in}(C) = n$ and $\mathsf{out}(C) = m$ is* coefficient bounded with respect to *$[\![-]\!]_*$, denoted $\mathsf{Bnd}_*(C)$, if for each $s \in [2^n]$ and $t \in [2^m]$ with $f = ([\![C]\!]_*)_{s,t}$,*

- *(B1). $\deg^+_{z_j}(f) \leq \sum_{a \in A(C)} |a_j|$ for each $j \in [k]$,*
- *(B2). $\deg^-_{z_j}(f) \leq \sum_{a \in A(C)} |a_j|$ for each $j \in [k]$,*
- *(B3). $\deg^+(f) \leq \sum_{a \in A(C)} \kappa(a)$ where $\kappa(a) = \max\{\sum_{j=1}^k a_j^+, \sum_{j=1}^k -a_j^-\}$.*

▶ **Example 5.5** (Coefficient Bounded Semantics). Recall C_1 from Ex. 5.3. It will be shown that $\mathsf{Bnd}_{\mathsf{Poly}}(C_1)$ holds. First, the coefficient sequence of C_1 must be computed. As illustrated in the previous example, C_1 contains only the rotations: $R_1 = C(R_Z(-\rho_1 + 2\rho_2))$ and $R_2 = C(R_Z(\rho_1 - 2\rho_2))$. The coefficient sequences of these rotations are $\beta = (-1, 2)$ and $\gamma = (1, -2)$ respectively. Then $A(C_2) = A(R_1) \cdot A(R_2) = (\beta) \cdot (\gamma) = (\beta, \gamma)$. Moreover, $\kappa(\beta) = \max\{0 + 2, 1 + 0\} = 2$ and $\kappa(\gamma) = \max\{1 + 0, 0 + 2\} = 2$. By inspecting the matrices in Ex. 5.3, it is clear that the following bounds hold for all $j \in [2]$ and $s, t \in [4]$.

$$\deg^+_{z_j}(([\![R_1]\!]_{\mathsf{Poly}})_{s,t}) \leq |\beta_j| \qquad \deg^-_{z_j}(([\![R_1]\!]_{\mathsf{Poly}})_{s,t}) \leq |\beta_j| \qquad \deg^+(([\![R_1]\!]_{\mathsf{Poly}})_{s,t}) \leq \kappa(\beta)$$
$$\deg^+_{z_j}(([\![R_2]\!]_{\mathsf{Poly}})_{s,t}) \leq |\gamma_j| \qquad \deg^-_{z_j}(([\![R_2]\!]_{\mathsf{Poly}})_{s,t}) \leq |\gamma_j| \qquad \deg^+(([\![R_2]\!]_{\mathsf{Poly}})_{s,t}) \leq \kappa(\gamma)$$

The κ terms can be thought of as adding together the maximum positive degrees of the two terms in each sine or cosine polynomial It turns out that these bounds compose additively under the composition of matrices, motivating properties (B1) through to (B3). In this example $\sum_{\alpha \in A(C_1)} |\alpha_1| = |-1| + |1| = 2$, $\sum_{\alpha \in A(C_1)} |\alpha_2| = |2| + |-2| = 4$, and $\sum_{\alpha \in A(C_1)} \kappa(\alpha) = 2 + 2 = 4$ By inspecting the final matrix in Ex. 5.3, it is clear that the following bounds hold for all $s, t \in [4]$ where $f = ([\![C_1]\!]_{\mathsf{Poly}})_{s,t}$.

$$\deg^+_{z_1}(f) \leq 2 \qquad \deg^-_{z_1}(f) \leq 2 \qquad \deg^+_{z_2}(f) \leq 4 \qquad \deg^-_{z_2}(f) \leq 4 \qquad \deg^+(f) \leq 4$$

Then C_1 satisfies (B1) through to (B3). Therefore, $\mathsf{Bnd}_{\mathsf{Poly}}(C_1)$ holds ⌟

$R_X(\theta_1 + 2\theta_2)$
$R_X(-\theta_2)$
$R_X(n\theta)$
Z
$R_X(n\theta)$

(a) The circuit for $\alpha = ((1,2),(0,-1))$.

(b) $(R_X(n\theta)) \circ Z \circ (R_X(n\theta))$.

Figure 4 Circuits used in Ex. 5.8 and Ex. 5.9 to illustrate the precision of $\mathsf{Bnd}(-)$.

This rationale given in Ex. 5.5 extends to all circuits in $\mathsf{Circ}_{\mathbb{Z}}(\mathcal{G},\mathcal{H})$. Since primitive gates map to constant matrices, then they trivially satisfy $\mathsf{Bnd}_{\mathsf{Poly}}(-)$. By construction of $\mathsf{CPoly}(f)$ and $\mathsf{SPoly}(f)$, then rotation matrices also satisfy $\mathsf{Bnd}_{\mathsf{Poly}}(-)$. It is then easy to show, using Prop. 3.2, that every gate in $\Sigma_{\mathbb{Z}}(\mathcal{G},\mathcal{H})$ satisfies $\mathsf{Bnd}_{\mathsf{Poly}}(-)$. With a slightly more careful analysis, it can then be shown that this invariant is closed under sequential and parallel composition. Intuitively, both matrix multiplication and the Kronecker tensor product yields sums of products of polynomials, in which each term can be shown to satisfy the degree bounds. Then by Prop. 3.3, every circuit in $\mathsf{Circ}_{\mathbb{Z}}(\mathcal{G},\mathcal{H})$ also satisfies $\mathsf{Bnd}_{\mathsf{Poly}}(-)$. Given these coefficient bounded semantics, the singularity factoring techniques of Sec. 4 can then be applied to obtain Cor. 5.7. All proof details can be found in the full paper.

▶ **Theorem 5.6.** *If $C \in \mathsf{Circ}_{\mathbb{Z}}(\mathcal{G},\mathcal{H})$, then $\mathsf{Bnd}_{\mathsf{Poly}}(C)$.*

▶ **Corollary 5.7.** *If $C_1 \in \mathsf{Circ}_{\mathbb{Z}}(\mathcal{G},\mathcal{H})$ and $C_2 \in \mathsf{Circ}_{\mathbb{Z}}(\mathcal{G},\mathcal{H})$ with $\mathsf{in}(C_1) = \mathsf{in}(C_2) = n$ and $\mathsf{out}(C_1) = \mathsf{out}(C_2) = m$, then for each pair of indices $s \in [2^n]$ and $t \in [2^m]$, there exists a polynomial $f \in \mathbb{C}[x_1,\ldots,x_k]$ such that,*

- *(D1).* $\deg_{x_j}(f) \leq 2\lambda_j$ *for each* $j \in [k]$*,*
- *(D2).* $\deg(f) \leq \max\{\sum_{a\in A(C)} \kappa(a) : C \in \{C_1, C_2\}\} + \sum_{j=1}^{k} \lambda_j$*,*
- *(D3).* $([\![C_1]\!] - [\![C_2]\!])_{s,t}(\theta) = 0$ *if and only if* $f(e^{-i\theta_1/2},\ldots,e^{-i\theta_k/2}) = 0$*,*

where $\lambda_j = \max\{\sum_{a\in A(C)} |a_j| : C \in \{C_1, C_2\}\}$ *for each* $j \in [k]$*.*

An interesting observation is that the bounds obtained through Thm. 5.6 were tight in Ex. 5.5. A natural question is whether these bounds are always tight, with respect to the granularity of the abstraction. We answer this question in the positive, by showing that for each coefficient sequence α, there exists a circuit C with $A(C) = \alpha$ such that the corresponding bound is tight. Of course, it is not possible to reconstruct a circuit from its coefficient sequence, so some information must be lost. To this end, we exhibit a family of circuits in Ex. 5.9, each of degree zero, for which arbitrarily large bounds can be obtained. In this example, relations exist between the rotations that depend on the axes-of-rotation and the parameter-free gates in the circuit, both of which are not captured by the coefficient sequence. In particular, both examples rely on the relations $(R_X(\beta))(R_X(\gamma)) = R_X(\beta+\gamma)$ and $Z(R_X(\beta)) = (R_X(-\beta))Z$.

▶ **Example 5.8** (Necessary Bounds)**.** Let α be any sequence over $\mathbb{Z}^k$ with $|\alpha| = n$. For each $j \in [n]$, define a linear function $f_j(\theta) = (\alpha_j)_1\theta_1 + \cdots + (\alpha_j)_k\theta_k$ and a rotation gate $G_j = R_X(f_j)$. Now consider the circuit $C = G_1 // \cdots // G_n$ (see Figure 4a). It follows that $A(C) = \alpha$. Moreover, $([\![C]\!](\theta))_{0,0} = \prod_{j=1}^{n} \cos(f_j(\theta)/2)$. With regard to the polynomial semantics, $[\![C]\!]_{\mathsf{Poly}} = 2^{-n}\prod_{a\in\alpha}(\prod_{j=1}^{k} z_k^{a_j} - \prod_{j=1}^{k} z_k^{-a_j})$. Clearly $\deg_{x_j}^{+}(([\![C]\!]_{\mathsf{Poly}})_{0,0}) = \sum_{a\in\alpha}|a_j|$ and $\deg_{x_j}^{-}(g) = \sum_{a\in\alpha}|a_j|$ for each $j \in [k]$. Then $\mathsf{Bnd}_{\mathsf{Poly}}(C)$ is tight. Since α was arbitrary, then every coefficient sequence is realizable with tight bounds. ⌟

▶ **Example 5.9** (Impact of Circuit Relations)**.** Fix $k = 1$ as the number of parameters and let $n \in \mathbb{N}$. Consider the circuit $C = R_X(n\theta) \circ Z \circ R_X(n\theta)$, as illustrated in Figure 4b. It follows that $[\![C]\!](\theta) = (R_X(n\theta))Z(R_X(n\theta)) = (R_X(n\theta))(R_X(-n\theta))Z = R_X(0)Z = Z$. Since

$[\![C]\!](\theta)$ is constant, its associated polynomials have degree zero. However, $\mathsf{Bnd}_{\mathsf{Poly}}(C)$ yields an upper bound of $\sum_{a\in A(C)}|a_1| = |n| + |n| = 2n$, which exceeds the true degree by $2n$. Since n was arbitrary, this error can be made arbitrarily large. ⌟

5.2 A Cutoff Theorem for Parameterized Equivalence

This section shows that parameterized equivalence checking reduces to parameter-free equivalence checking for quantum circuits (Thm. 5.10). The proof proceeds as follows. First, Cor. 5.7 is used to characterize a family of polynomials which are identically zero if and only if the two circuits are equal. Using Thm. 2.1, a finite set of points $S \subseteq \mathbb{Q}^k$ can be constructed to determine if these polynomials are identically zero. The points in S are in bijection with a set of points on the complex unit circle under the transformation $x \mapsto e^{-ix/2}$. It follows that each polynomial is identically zero if and only if $[\![C_1]\!](s) = [\![C_2]\!](s)$ for all points $s \in S$. Note that the polynomials are never explicitly constructed. All proof details are in the full paper.

▶ **Theorem 5.10.** *Let $C_1 \in \mathsf{Circ}_{\mathbb{Z}}(\mathcal{G},\mathcal{H})$ and $C_2 \in \mathsf{Circ}_{\mathbb{Z}}(\mathcal{G},\mathcal{H})$ with $\mathsf{in}(C_1) = \mathsf{in}(C_2)$ and $\mathsf{out}(C_1) = \mathsf{out}(C_2)$. If $S_1, S_2, \ldots, S_k \subseteq [0, 4\pi)$ such that $|S_j| > 2\lambda_j$ for each $j \in [k]$, then $[\![C_1]\!](\theta) = [\![C_2]\!](\theta)$ for all $\theta \in \mathbb{R}^k$ if and only if $[\![C_1]\!](v) = [\![C_2]\!](v)$ for all $v \in S_1 \times S_2 \times \cdots \times S_k$.*

▶ **Corollary 5.11.** *If $\mathcal{G}$ and $\mathcal{H}$ consist of matrices over the universal cyclotomic field, then the parameterized equivalence checking problem is decidable for $\mathsf{Circ}_{\mathbb{Z}}(\mathcal{G},\mathcal{H})$.*

As k grows large, the utility of Thm. 5.10 decreases. For example, if each λ_j is b, then $|S_1 \times \cdots \times S_k| = (2b+1)^k$. That is, the number of instances grows exponentially with k. However, this exponential growth can be overcome by a probabilistic algorithm. Fix a finite subset S of $[0, 4\pi)^k$ and assume that s is chosen at random from S. If $[\![C_1]\!](s) = [\![C_2]\!](s)$, then conclude that $[\![C_1]\!] = [\![C_2]\!]$, otherwise conclude that $[\![C_2]\!] \neq [\![C_2]\!]$. Clearly, this algorithm has no false negatives, since $[\![C_1]\!](s) \neq [\![C_2]\!](s)$ implies $[\![C_2]\!] \neq [\![C_2]\!]$. A more interesting question is the false positive rate. Note that a false positive occurs when $[\![C_1]\!](s) = [\![C_2]\!](s)$ but $[\![C_1]\!] \neq [\![C_2]\!]$. It is shown in the following theorem that the probability of a false positive decreases with order $O(1/|S|)$, as an application of Thm. 2.2.

▶ **Theorem 5.12.** *Let $C_1 \in \mathsf{Circ}_{\mathbb{Z}}(\mathcal{G},\mathcal{H})$ and $C_2 \in \mathsf{Circ}_{\mathbb{Z}}(\mathcal{G},\mathcal{H})$ with $\mathsf{in}(C_1) = \mathsf{in}(C_2)$, $\mathsf{out}(C_1) = \mathsf{out}(C_2)$, and $[\![C_1]\!] \neq [\![C_2]\!]$. For each finite subset $S \subseteq [0, 4\pi)$, if $s_1, \ldots, s_k$ are sampled at random both independently and uniformly from S, then*

$$\Pr\left([\![C_1]\!](s_1, \ldots, s_k) = [\![C_2]\!](s_1, \ldots, s_k)\right) \leq d/|S|$$

where $d = \max\{\sum_{\alpha\in A(C)} \kappa(\alpha) : C \in \{C_1, C_2\}\} + \sum_{j=1}^{k} \lambda_j$.

6 Extending to Rational Coefficients and Global Phase

The methods presented in Sec. 5 face several limitations. In particular, both Thm. 5.10 and Thm. 5.12 assume that the circuits are integral, and do not allow for equivalence up to global phase. In this section, we show how to extend the techniques of Sec. 5 to handle rational circuits and global phase. We also expand Thm. 5.12 into an algorithm, and consider the problem of angle sampling given a gate set over the universal cyclotomic field.

6.1 Verifying Circuits with Rational Coefficients

Most parameterized quantum circuits have fractional coefficients. For example, the equality in Figure 3 is typically stated with a parameter θ on the left-hand side and the parameters $\pm\theta/2$ on the right-hand side. The circuits in Figure 3 are related to these fractional circuits by the substitution $f(\theta) = \theta/2$. Conceptually, $f : \mathbb{R}^k \to \mathbb{R}^k$ *reparameterizes* the circuit, by inducing a bijection between the parameter space of the rational circuits and the parameter space of the integral circuits. This generalizes to all examples (see the full paper for proofs).

▶ **Lemma 6.1.** *Let $C_1, C_2 \in \mathsf{Circ}(\mathcal{G}, \mathcal{H})$. If $f : \mathbb{R}^k \to \mathbb{R}^k$ is a bijective function, then $[\![C_1]\!] = [\![C_2]\!]$ if and only if $[\![C_1]\!] \circ f = [\![C_2]\!] \circ f$.*

The goal of this section is to construct a syntactic transformation which eliminates all rational coefficients, which preserving the semantic interpretation via a bijective reparameterization. A *syntactic reparameterization* is a map $F : \mathsf{Circ}(\mathcal{G}, \mathcal{H}) \to \mathsf{Circ}(\mathcal{G}, \mathcal{H})$ with a bijective function $f : \mathbb{R}^k \to \mathbb{R}^k$ such that $[\![F(C)]\!] = [\![C]\!] \circ f$. The simplest syntactic reparameterization is a linear rescaling of the parameters in the circuit by a non-zero rational vector. For each vector $v \in (\mathbb{Q} \setminus \{0\})^k$, define the map $F_v : \mathsf{Circ}(\mathcal{G}, \mathcal{H}) \to \mathsf{Circ}(\mathcal{G}, \mathcal{H})$ as follows.

- If $G \in \mathcal{G}$, then $F_v(G) = G$.
- If $M \in \mathcal{H}$ and $f(\theta) = a_1\theta_1 + a_2\theta_2 + \cdots + a_k\theta_k + q$, then $F_v(R_M(f)) = R_M(g)$ where $g(\theta) = (v_1a_1)\theta_1 + (v_2a_2)\theta_2 + \cdots + (v_ka_k)\theta_k + q$.
- If $G \in \Sigma(\mathcal{G}, \mathcal{H})$, then $F_v(C(G)) = C(F_v(G))$.
- If $C_1, C_2 \in \mathsf{Circ}(\mathcal{G}, \mathcal{H})$, then $F_v(C_1 // C_2) = F_v(C_1) // F_v(C_2)$.
- If $C_1, C_2 \in \mathsf{Circ}(\mathcal{G}, \mathcal{H})$, then $F_v(C_2 \circ C_1) = F_v(C_2) \circ F_v(C_1)$.

▶ **Theorem 6.2.** *For each $v \in (\mathbb{Q}\setminus\{0\})^k$, $f : \mathbb{R}^k \to \mathbb{R}^k$ defined by $f(\theta) = (v_1\theta_1, v_2\theta_2, \ldots, v_k\theta_k)$ is bijective and F_v is syntactic reparameterization with respect to f.*

Now assume that C_1 and C_2 are circuits in $\mathsf{Circ}(\mathcal{G}, \mathcal{H})$. For the correct choice of v, both $F_v(C_1)$ and $F_v(C_2)$ are elements of $\mathsf{Circ}_{\mathbb{Z}}(\mathcal{G}, \mathcal{H})$. Intuitively, each v_j must be chosen such that it clears the denominators of all coefficients tied to θ_k in both C_1 and C_2. Formally, let $\mathrm{denom}(q)$ denote the denominator of $q \in \mathbb{Q}$ and $\mathrm{lcm}\{x_1, x_2, \ldots, x_n\}$ denote the least common multiple of $x_1, x_2, \ldots, x_n \in \mathbb{Z}$. Then for each $j \in [k]$, $X_j = \{\mathrm{denom}(\alpha_j) : \alpha \in A(C_1) \cdot A(C_2)\}$ is the set of all denominators of coefficients tied to θ_k in both C_1 and C_2. Then $v_j = \mathrm{lcm}(X_j)$ for each $j \in [k]$. Let $\mathsf{circLcm}(C_1, C_2)$ denote this vector.

▶ **Theorem 6.3.** *If $C_1, C_2 \in \mathsf{Circ}(\mathcal{G}, \mathcal{H})$ and $v = \mathsf{circLcm}(C_1, C_2)$, then $F_v(C_1) \in \mathsf{Circ}_{\mathbb{Z}}(\mathcal{G}, \mathcal{H})$ and $F_v(C_2) \in \mathsf{Circ}_{\mathbb{Z}}(\mathcal{G}, \mathcal{H})$. Moreover, $[\![C_1]\!] = [\![C_2]\!]$ if and only if $[\![F_v(C_1)]\!] = [\![F_v(C_2)]\!]$.*

▶ **Corollary 6.4.** *If $\mathcal{G}$ and $\mathcal{H}$ consist of matrices over the universal cyclotomic field, then the parameterized equivalence checking problem is decidable for $\mathsf{Circ}(\mathcal{G}, \mathcal{H})$.*

6.2 Verifying Circuits Modulo Global Phase

In Sec. 5 the circuits C_1 and C_2 where defined to be equivalence when $[\![C_1]\!](\theta) = [\![C_2]\!](\theta)$ for all $\theta \in \mathbb{R}^k$. For many applications, this notion of equivalence is far too strict. This is because C_1 and C_2 will prepare the same probability distribution provided there exists some function $p : \mathbb{R}^k \to \mathbb{R}$ such that $[\![C_1]\!](\theta) = e^{ip(\theta)}[\![C_2]\!](\theta)$ for all $\theta \in \mathbb{R}^k$. When such a function exists, we say that C_1 and C_2 are equivalent modulo global phase. Of course, verifying the existence of an arbitrary p is infeasible. Prior work has assumed p to be affine linear [21,38,47]. That is, $p(\theta) = \alpha_1\theta_1 + \cdots \alpha_k\theta_k + \beta$. We further assume that α_1 through to α_k are rational. In this section we show how to verify the equivalence of C_1 and C_2 modulo affine rational linear global phase, under the following assumptions.

1. All matrices in $\mathcal{H}$ are defined over the universal cyclotomic field.
2. All matrices in $\mathcal{G}$ are injective and defined over the universal cyclotomic field.

In practice, the second assumption restricts $\mathcal{G}$ to unitary operations and state preparation.

Since the universal cyclotomic field is closed under addition and multiplication, then every global phase will be cyclotomic when evaluated at rational multiples of π. In general, α need not be rational, since there exists cyclotomic numbers of norm 1 which are not roots of unity. However, the periodicity of $[\![C_1]\!]$ an $[\![C_2]\!]$ ensure that $\alpha \in \mathbb{Q}^k$. Using properties of cyclotomic numbers, such as the fact that $\mathbb{Q}(\zeta_{2n}) = \mathbb{Q}(\zeta_n)$ for odd n, it is then possible to solve for α (if it exists). In the full paper, an algorithm $\mathsf{FindPhase}(C_1, C_2)$ is described to compute these coefficients. The injectivity of $\mathcal{G}$ ensures that all coefficients can be isolated (this condition is sufficient but not necessary). In the case where C_1 and C_2 are not equivalent up to global phase, then arbitrary coefficients are returned. Otherwise, the function $\mathsf{FindPhase}(C_1, C_2)$ returns a tuple (z, f) such that $z = e^{i\beta}$ and $f(\theta) = (-2\alpha_1)\theta_1 + \cdots + (-2\alpha_k)\theta_k$. Then the global phase can be offset by introducing a unitary gate zI and a global phase gate $R_I(f)$. Then equivalence modulo global phase reduces to exact equivalence as follows.

▶ **Theorem 6.5.** *Assume $\mathcal{G}$ and $\mathcal{H}$ consist of matrices over the universal cyclotomic field, with all gates in $\mathcal{G}$ injective. If $C_1, C_2 \in \mathsf{Circ}_{\mathbb{Z}}(\mathcal{G}, \mathcal{H})$ and $(z, f) = \mathsf{FindPhase}(C_1, C_2)$, then C_1 is equivalent to C_2 modulo affine rational linear global phase if and only if the equation $[\![C_1]\!] = [\![zI \circ R_I(f) \circ C_2]\!]$ holds.*

▶ **Corollary 6.6.** *If $\mathcal{G}$ and $\mathcal{H}$ satisfy assumptions (1–2), then the parameterized equivalence checking problem is decidable modulo affine rational linear global phase for $\mathsf{Circ}(\mathcal{G}, \mathcal{H})$.*

6.3 A Probabilistic Equivalence Checking Procedure

Imagine applying Thm. 5.12 to a pair of quantum circuits C_1 and C_2. In practice, an end-user would have some desired upper bound $p \in (0, 1]$ on the false positive rate. A simply way to bound the false positive rate is to require that $d/|S| \leq p$, meaning that $d/p \leq |S|$. Since d/p is positive and $|S|$ is a natural number, then the minimum value of $|S|$ which satisfies this inequality is $N = \lceil d/p \rceil$. Using this optimal solution, the following algorithm is obtained.

1. Compute $d = \max\{\sum_{\alpha \in A(C)} \kappa(\alpha) : C \in \{C_1, C_2\}\} + \sum_{j=1}^{k} \lambda_j$.
2. Select a set $S \subseteq [0, 4\pi)$ such that $|S| = \lceil d/p \rceil$.
3. Sample $s_1, \ldots, s_k$ at random both independently and uniformly from S.
4. Determine if $[\![C_1]\!](s_1, \ldots, s_k) = [\![C_2]\!](s_1, \ldots, s_k)$.

The most crucial step of this algorithm is the second step. First, the choice of S must ensure that the values of $\sin(-)$ and $\cos(-)$ are exact. As outlined in Sec. 2.2, the simplest way to do this is to sample S from $[0, 4\pi) \cap \mathbb{Q}\pi$ for with for which $\sin(-)$ and $\cos(-)$ must evaluate to cyclotomic numbers. This method is particularly effective when $\mathcal{G}$ and $\mathcal{H}$ consists purely of matrices over the universal cyclotomic field, in which case all computation can be carried out over the universal cyclotomic field.

Now, consider the elements of $\sin(S)$ and $\cos(S)$. For each $(j/n)\pi$ in S, the elements $\sin(j/n)$ and $\cos(j/n)$ will be elements of $\mathbb{Q}[\zeta_n]$. Likewise, if ℓ is the least common denominator of all fractions in S, then $S \subseteq \mathbb{Q}[\zeta_\ell]$. In the worst case, $\mathbb{Q}[\zeta_\ell]$ will be an ℓ-dimensional vector space. This means that the cost of addition will grow at least linearly with ℓ, and the cost of multiplication will grow at least quadratically with ℓ.

▶ **Theorem 6.7.** *If $k \in \mathbb{N}$, $S \subseteq [0, k) \cap \mathbb{Q}$ and $b = |S|$, then $\operatorname{lcm}\{\operatorname{denom}(s) : s \in S\} \geq \lceil b/k \rceil$.*

Let M be the smallest multiple of 4 which is greater than or equal to N. It follows from Thm. 6.7 that $S = \{0, (1/M)4\pi, (2/M)4\pi, \ldots, ((M-1)/M)4\pi\}$ minimizes ℓ. This set is also easy to compute, and is therefore taken to be the definition of S.

7 Related Work

In the introduction, we discussed the cutoff-based techniques [32], which subsumes prior work such as [23]. In this section, we compare to other approaches.

Circuit Rewriting. It was highlighted in Ex. 5.9 that circuit rewriting intersects with parameterized equivalence checking. In [38], an incomplete equational theory is given for a family of parameterized circuits, which is shown to be effective for equivalence checking. In [44], a complete set of relations are derived, under the assumption that each parameter appears exactly once in the circuit. Relations which hold for abstract gate sets, such as $\Sigma(\mathcal{G}, \mathcal{H})$, have yet to be explored.

Symbolic Techniques. In [47], symbolic techniques are used to determine parameterized equivalence. Particularly, trigonometric relations, together with the Pythagorean relation $\cos(\theta)^2 + \sin(\theta)^2 = 1$, are used to reduce equivalence checking to a family of equations over the theory of non-linear real arithmetic. This is then solved using the Z3 [12] solver as a black box. However, the decision problem for non-linear real arithmetic is known to be double-exponential in the number of variables [9, 24], whereas our approach is exponential in the number of variables.

Probabilistic Techniques. In [46], Thm. 2.2 was used to determine the equivalence of parameterized quantum circuits. However, our technique yields Laurent polynomials rather than ordinary polynomials, which we do not compute explicitly. In [38], Peham et al. show that if v is sampled uniformly at random from $[0, 4\pi)^k$, then $\Pr(\llbracket C_1 \rrbracket(v) = \llbracket C_2 \rrbracket(v)) = 0$ given $\llbracket C_1 \rrbracket \neq \llbracket C_2 \rrbracket$. However, sampling v from a uniform continuous distribution is impossible on a digital computer, which can only represent a countable and non-enumerable subset of real numbers [43]. In Peham et al., floating-point is used, and presumably, the error is assumed to be uniform as well. In our work, all computation is exact, and therefore, such assumptions do not apply. Since there does not exist a uniform distribution for countable sets, we instead sample uniformly from a finite subset of $[0, 4\pi)$, in which case Thm. 2.2 applies, rather than the analytic results of Peham et al.

8 Conclusion and Future Work

In this paper, we considered the problem of parameterized equivalence checking for quantum circuits. We show that the parameterized problem can be reduced to finitely many instances of the parameter-free problem, regardless of the gate set or axes of rotation. Consequently, the problem is decidable in the case of gate sets defined over the universal cyclotomic field. Moreover, we show that when the number of instances becomes intractable large, there exists a probabilistic variation of the algorithm where the probability of being incorrect can be made arbitrarily small. We have outlined how the techniques can be implemented in practice, taking into account rational coefficients, global phase, and angle sampling.

In future work, we would like to explore how these decision procedures can be implemented efficiently using circuit rewriting and sparse matrix representations. In particular, we would like to explore angle sampling and circuit evaluation using ZX-diagrams [37], tensor decision-diagrams [49], and model-counting [30], which have all proven effective in parameter-free equivalence checking. We would also like to explore how rewriting-based techniques and symmetry reductions might help to tighten the cutoffs obtained from $\mathsf{Bnd}(-)$. For example, the bound obtained in Ex. 5.9 could be reduced to zero by viewing each relation as a rewriting rule, and then searching for a derivation which reduces the bound.

References

1 Amira Abbas, David Sutter, Christa Zoufal, Aurelien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409, 2021. doi:10.1038/s43588-021-00084-1.

2 Parosh Aziz Abdulla, Frédéric Haziza, and Lukás Holík. All for the price of few. In *VMCAI*, volume 7737 of *LNCS*, pages 476–495. Springer, 2013. doi:10.1007/978-3-642-35873-9_28.

3 Noga Alon. Combinatorial Nullstellensatz. *Combinatorics, Probability and Computing*, 8(1–2):7–29, 1999. doi:10.1017/S0963548398003411.

4 Matthew Amy, Andrew N. Glaudell, Shaun Kelso, William Maxwell, Samuel S. Mendelson, and Neil J. Ross. Exact synthesis of multiqubit Clifford-cyclotomic circuits. In *RC*, volume 14680 of *LNCS*, pages 238–245. Springer, 2024. doi:10.1007/978-3-031-62076-8_15.

5 Davis Arthur and Prasanna Date. A hybrid quantum-classical neural network architecture for binary classification, 2022. arXiv:2201.01820.

6 Martin Avanzini, Georg Moser, Romain Péchoux, and Simon Perdrix. On the hardness of analyzing quantum programs quantitatively. In *Programming Languages and Systems*, volume 14577 of *LNCS*, pages 31–58. Springer, 2024. doi:10.1007/978-3-031-57267-8_2.

7 Sheldon Axler. *Linear Algebra Done Right*. Springer, 3rd edition, 2014. doi:10.1007/978-3-031-41026-0.

8 John C. Baez, Brandon Coya, and Franciscus Rebro. Props in network theory. *Theory and Applications of Categories*, 33(25):727–783, 2010.

9 Nikolaj Bjørne, Leonardo de Moura, Lev Nachmanson, and Christoph M. Wintersteiger. *Programming Z3*, volume 11430 of *LNPSE*, pages 148–201. Springer, 2019. doi:10.1007/978-3-030-17601-3_4.

10 Wieb Bosma. Canonical bases for cyclotomic fields. *Applicable Algebra in Engineering, Communication and Computing*, 1:125–134, 1990. doi:10.1007/BF01810296.

11 Thomas Breuer. Integral bases for subfields of cyclotomic fields. *Applicable Algebra in Engineering, Communication and Computing*, 8:279–289, 1997. doi:10.1007/s002000050065.

12 Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *TACAS*, volume 4963 of *LNCS*, pages 337–340. Springer, 2008. doi:10.1007/978-3-540-78800-3_24.

13 Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Info. Proc. Letters*, 7(4):193–195, 1978. doi:10.1016/0020-0190(78)90067-4.

14 Qi-Ming Ding, Yi-Ming Huang, and Xiao Yuan. Molecular docking via quantum approximate optimization algorithm. *Phys. Rev. Appl.*, 21:034036, 2024. doi:10.1103/PhysRevApplied.21.034036.

15 Bryan Eastin and Emanuel Knill. Restrictions on transversal encoded quantum gate sets. *Phys. Rev. Let.*, 102(11):110502, 2009. doi:10.1103/physrevlett.102.110502.

16 E. Allen Emerson and Kedar S. Namjoshi. On reasoning about rings. *Int. J. Found. Comput. Sci.*, 14(4):527–550, 2003. doi:10.1142/S0129054103001881.

17 Richard M. Foote and David S. Dummit. *Abstract Algebra*. Wiley, 3rd edition, 2003.

18 Brett Giles and Peter Selinger. Exact synthesis of multiqubit Clifford+T circuits. *Phys. Rev. A*, 87:032332, 2013. doi:10.1103/PhysRevA.87.032332.

19 Dylan Herman, Cody Googin, Xiaoyuan Liu, Yue Sun, Alexey Galda, Ilya Safro, Marco Pistoia, and Yuri Alexeev. Quantum computing for finance. *Nature Rev. Phys.*, 5(8):450–465, 2023. doi:10.1038/s42254-023-00603-1.

20 Kesha Hietala, Robert Rand, Liyi Li, Shih-Han Hung, Xiaodi Wu, and Michael Hicks. A verified optimizer for quantum circuits. *ACM Trans. Program. Lang. Syst.*, 45(3), 2023. doi:10.1145/3604630.

21 Xin Hong, Wei-Jia Huang, Wei-Chen Chien, Yuan Feng, Min-Hsiu Hsieh, Sanjiang Li, and Mingsheng Ying. Equivalence checking of parameterised quantum circuits, 2024. arXiv:2404.18456.

22 C. Norris Ip and David L. Dill. Better verification through symmetry. In *CHDL*, volume A-32 of *IFIP Transactions*, pages 97–111. North-Holland, 1993. doi:10.5555/648251.752211.

23 Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Diagrammatic reasoning beyond clifford+t quantum mechanics. In *LICS*. ACM, 2018. doi:10.1145/3209108.3209139.

24 Dejan Jovanović and Leonardo de Moura. Solving non-linear arithmetic. In *AR*, volume 7364 of *LNAI*, pages 339–354. Springer, 2012. doi:10.1007/978-3-642-31365-3_27.

25 Alexander Kaiser, Daniel Kroening, and Thomas Wahl. Dynamic cutoff detection in parameterized concurrent programs. In *CAV*, volume 6174 of *LNCS*, pages 645–659. Springer, 2010. doi:10.1007/978-3-642-14295-6_55.

26 Shu Kanno, Hajime Nakamura, Takao Kobayashi, Shigeki Gocho, Miho Hatanaka, Naoki Yamamoto, and Qi Gao. Quantum computing quantum Monte Carlo with hybrid tensor network for electronic structure calculations. *npj Quantum Information*, 10(1), 2024. doi:10.1038/s41534-024-00851-8.

27 Ayrat Khalimov, Swen Jacobs, and Roderick Bloem. Towards efficient parameterized synthesis. In *VMCAI*, volume 7737 of *LNCS*, pages 108–127. Springer, 2013. doi:10.1007/978-3-642-35873-9_9.

28 F. William Lawvere. Functorial semantics of algebraic theories. *Proc. Natl. Acad. Sci. U.S.A.*, 50(5):869–872, 1963. doi:10.1073/pnas.50.5.869.

29 He Ma, Govoni Marco, and Giulia Galli. Quantum simulations of materials on near-term quantum computers. *npj Computational Materials*, 6:85, 2020. doi:10.1038/s41524-020-00353-z.

30 Jingyi Mei, Marcello Bonsangue, and Alfons Laarman. Simulating quantum circuits by model counting. In *CAV*, volume 14683 of *LNCS*, pages 555–578. Springer, 2024. doi:10.1007/978-3-031-65633-0_25.

31 Dekel Meirom and Steven H. Frankel. PANSATZ: pulse-based ansatz for variational quantum algorithms. *Frontiers in Quantum Science and Technology*, 2, 2023. doi:10.3389/frqst.2023.1273581.

32 Hector Miller-Bakewell. Finite verification of infinite families of diagram equations. *EPTCS*, 318:27–52, 2020. doi:10.4204/eptcs.318.3.

33 Hector Miller-Bakewell. *Graphical Calculi and their Conjecture Synthesis*. PhD thesis, University of Oxford, 2020.

34 Kedar S. Namjoshi and Richard J. Trefler. Parameterized compositional model checking. In *TACAS*, volume 9636 of *LNCS*, pages 589–606. Springer, 2016. doi:10.1007/978-3-662-49674-9_39.

35 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2011.

36 Pyotr Novikov. On the algorithmic unsolvability of the word problem in group theory. *Trudy Matematicheskogo Instituta imeni V.A. Steklova*, 44:3–143, 1955.

37 Tom Peham, Lukas Burgholzer, and Robert Wille. Equivalence checking of quantum circuits with the ZX-calculus. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 12(3):662–675, 2022. doi:10.1109/jetcas.2022.3202204.

38 Tom Peham, Lukas Burgholzer, and Robert Wille. Equivalence checking of parameterized quantum circuits: Verifying the compilation of variational quantum algorithms. In *ASPDAC*, pages 702–708. ACM, 2023. doi:10.1145/3566097.3567932.

39 Neil J. Ross and Scott Wesley. Cutoff theorems for the equivalence of parameterized quantum circuits (extended), 2025. arXiv:2506.20985.

40 Raffaele Santagati, Alan Aspuru-Guzik, Ryan Babbush, Matthias Degroote, Leticia González, Elica Kyoseva, Nikolaj Moll, Markus Oppel, Robert M. Parrish, Nicholas C. Rubin, Michael Streif, Christofer S. Tautermann, Horst Weiss, Nathan Wiebe, and Clemens Utschig-Utschig. Drug design on quantum computers. *Nature Phys.*, 20:549–557, 2024. doi:10.1038/s41567-024-02411-5.

41 J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980. doi:10.1145/322217.322225.

42 Razin A. Shaikh, Quanlong Wang, and Richie Yeung. How to sum and exponentiate Hamiltonians in ZXW calculus. *EPTCS*, 394:236–261, 2023. doi:10.4204/eptcs.394.14.

43 A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proc. of the London Math. Soc.*, s2-42(1):230–265, 1937. doi:10.1112/plms/s2-42.1.230.

44 John van de Wetering, Richie Yeung, Tuomas Laakkonen, and Aleks Kissinger. Optimal compilation of parametrised quantum circuits, 2024. doi:10.48550/arXiv.2401.12877.

45 Scott Wesley, Maria Christakis, Jorge A. Navas, Richard Trefler, Valentin Wüstholz, and Arie Gurfinkel. Verifying Solidity smart contracts via communication abstraction in SmartACE. In *VMCAI*, pages 425–449. Springer, 2022. doi:10.1007/978-3-030-94583-1_21.

46 Amanda Xu, Abtin Molavi, Lauren Pick, Swamit Tannu, and Aws Albarghouthi. Synthesizing quantum-circuit optimizers. *Proc. ACM Program. Lang.*, 7(PLDI), 2023. doi:10.1145/3591254.

47 Mingkuan Xu, Zikun Li, Oded Padon, Sina Lin, Jessica Pointing, Auguste Hirth, Henry Ma, Jens Palsberg, Alex Aiken, Umut A. Acar, and Zhihao Jia. Quartz: superoptimization of quantum circuits. In *PLDI*, pages 625–640. ACM, 2022. doi:10.1145/3519939.3523433.

48 Daniel Yoffe, Noga Entin, Amir Natan, and Adi Makmal. A qubit-efficient variational selected configuration-interaction method. *Quantum Science and Technology*, 10(1):015020, 2024. doi:10.1088/2058-9565/ad7d32.

49 Qirui Zhang, Mehdi Saligane, Hun-Seok Kim, David Blaauw, Georgios Tzimpragos, and Dennis Sylvester. Quantum circuit simulation with fast tensor decision diagram. In *ISQED*, pages 1–8. IEEE, 2024. doi:10.1109/isqed60706.2024.10528748.

50 Pengzhan Zhao, Zhongtao Miao, Shuhan Lan, and Jianjun Zhao. Bugs4Q: A benchmark of existing bugs to enable controlled testing and debugging studies for quantum programs. *J. of Systems and Software*, 205:111805, 2023. doi:10.1016/j.jss.2023.111805.

51 Richard Zippel. Probabilistic algorithms for sparse polynomials. In *EUROSAM*, volume 72 of *LNCS*, pages 216–226. Springer, 1979. doi:10.1007/3-540-09519-5_73.