

# Cyclotomic Identity Testing and Applications

Nikhil Balaji\*

nbalaji@cse.iitd.ac.in  
Department of CSE, IIT Delhi  
India

Sylvain Perifel\*

Mahsa Shirmohammadi\*  
sylvain.perifel@irif.fr  
mahsa@irif.fr  
Université de Paris, CNRS, IRIF  
Paris, France

James Worrell\*

james.worrell@cs.ox.ac.uk  
Department of Computer Science,  
University of Oxford  
Oxford, UK

## ABSTRACT

We consider the cyclotomic identity testing (CIT) problem: given a polynomial  $f(x_1, \dots, x_k)$ , decide whether  $f(\zeta_n^{e_1}, \dots, \zeta_n^{e_k})$  is zero, where  $\zeta_n = e^{2\pi i/n}$  is a primitive complex  $n$ -th root of unity and  $e_1, \dots, e_k$  are integers, represented in binary. When  $f$  is given by an algebraic circuit, we give a randomized polynomial-time algorithm for CIT assuming the generalised Riemann hypothesis (GRH), and show that the problem is in coNP unconditionally. When  $f$  is given by a circuit of polynomially bounded degree, we give a randomized NC algorithm. In case  $f$  is a linear form we show that the problem lies in NC. Towards understanding when CIT can be solved in deterministic polynomial-time, we consider so-called diagonal depth-3 circuits, i.e., polynomials  $f = \sum_{i=1}^m g_i^{d_i}$ , where  $g_i$  is a linear form and  $d_i$  a positive integer given in unary. We observe that a polynomial-time algorithm for CIT on this class would yield a sub-exponential-time algorithm for polynomial identity testing. However, assuming GRH, we show that if the linear forms  $g_i$  are all identical then CIT can be solved in polynomial time. Finally, we use our results to give a new proof that equality of compressed strings, i.e., strings presented using context-free grammars, can be decided in randomized NC.

## CCS CONCEPTS

• **Theory of computation** → **Algebraic complexity theory**; **Algebraic complexity theory**; **Circuit complexity**; **Problems, reductions and completeness**.

## KEYWORDS

Cyclotomic Integers, Cyclotomic Fields, Randomised Algorithms, Straight Line Programs, Polynomial Identity testing

### ACM Reference Format:

Nikhil Balaji, Sylvain Perifel, Mahsa Shirmohammadi, and James Worrell. 2021. Cyclotomic Identity Testing and Applications. In *Proceedings of the 2021 International Symposium on Symbolic and Algebraic Computation (ISSAC '21)*, July 18–23, 2021, Virtual Event, Russia Federation.

\*All authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ISSAC '21, July 18–23, 2021, Virtual Event, Russia Federation

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8382-0/21/07...\$15.00

<https://doi.org/10.1145/XXXXXX.XXXXXX>

ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/XXXXXX.XXXXXX>

## 1 INTRODUCTION

Identity testing in number fields is a fundamental problem in algorithmic algebra that has been studied in relation to solving systems of polynomial equations [17, 22] and polynomial identity testing [10]. Among number fields, cyclotomic fields, i.e., those generated by roots of unity, play a central role. The aim of this paper is a comprehensive study of the computational complexity of identity testing in cyclotomic fields.

We consider *cyclotomic identity testing problems*, where the input consists of a polynomial  $f(x_1, \dots, x_k)$  with integer coefficients, together with integers  $n, e_1, \dots, e_k$ , and the task is to decide whether  $f(\zeta_n^{e_1}, \dots, \zeta_n^{e_k})$  is zero for  $\zeta_n = e^{2\pi i/n}$ . We consider four variants of this problem according to the representation of  $f$ : (i)  $f$  is given as an algebraic circuit; (ii)  $f$  is given by a circuit of polynomially bounded syntactic degree; (iii)  $f$  is a linear form; (iv)  $f = \sum_{i=1}^m g_i^{d_i}$  where each  $g_i$  is a linear form and  $d_i$  is an integer in unary. Although  $f$  is a multivariate polynomial, since it is evaluated on powers of a common primitive  $n$ -th root of unity, we formalise the above problems in terms of circuits whose input gates are labelled by powers of a single variable  $x$ .

Formally, for our purposes an algebraic circuit  $C$  is a directed, acyclic graph with labelled vertices and edges. Vertices of in-degree zero are labelled in the set of monomials  $\{x^e : e \in \mathbb{N}\}$  and the remaining vertices have labels in  $\{+, \times\}$ . Moreover the incoming edges to  $+$ -vertices have labels in  $\mathbb{Z}$ , that is, the  $+$ -gates compute integer-weighted sums. There is a unique vertex of out-degree zero which determines the output of the circuit, a univariate polynomial, in an obvious manner. The size of  $C$  is the sum of the number of edges in the underlying graph and the bit-length of all integer constants appearing in  $C$ . The syntactic degree of  $C$  is defined inductively as follows: input gates have degree 1, the degree of an addition gate is the maximum of the degrees of its inputs, the degree of a multiplication gate is the sum of the degrees of its inputs, and the degree of  $C$  is the degree of the output gate. Note that the syntactic degree of  $C$  is *not* an upper bound on the degree of the computed polynomial since we allow monomials as inputs. Unless otherwise stated, we assume that all integers are represented in binary.

The four main variants of the cyclotomic identity testing problem are as follows:

In the *Cyclotomic Identity Testing (CIT)* problem the input is an algebraic circuit  $C$  representing a polynomial  $f(x)$ , together

with an integer  $n$ , and the task is to determine whether  $f(\zeta_n) = 0$ , where  $\zeta_n = e^{2\pi i/n}$  is a primitive complex  $n$ -th root of unity.

The *Bounded-CIT* problem is defined exactly as the CIT problem, except that the input also includes an upper bound on the syntactic degree of the circuit  $C$  that is given in unary. Thus in Bounded-CIT the degree of the circuit is at most the length of the input.

In the *Sparse-CIT* problem the circuit  $C$  has syntactic degree 1. This is equivalent to giving the input polynomial  $f$  in sparse representation, i.e., where  $f = \sum_{i=1}^s a_i x^{k_i}$  is encoded as a list of pairs of integers  $(a_1, k_1), \dots, (a_s, k_s)$ .

Finally we consider CIT in case  $C$  is a diagonal circuit (see [36]), that is, the input polynomial has the form  $f = \sum_{i=1}^m g_i^{d_i}$  where each  $g_i$  is in sparse representation and  $d_1, \dots, d_m$  are integers represented in unary.

The representation of polynomials in the CIT problem can be exponentially more succinct than in the Bounded-CIT problem, since the syntactic degree of a circuit can be exponential in its size. Likewise the representation in the Bounded-CIT problem can be exponentially more succinct than in the Sparse-CIT problem, since the former allows the number of monomials to be exponential in the circuit size.

The problem Sparse-CIT was first studied by Plaisted [32], who gave a randomised polynomial-time algorithm. Subsequently, two different deterministic polynomial-time algorithms were given by Cheng et al. [11, 12]. A natural approach to decide zeroness of  $f(\zeta_n)$  is to compute an approximation of sufficient precision. However, given existing separation bounds for algebraic numbers, the precision required to distinguish between zero and a non-zero value precludes a polynomial-time bound via this method.

The conclusion of [12] raises the question of the complexity of CIT. The authors note that this problem lies in the counting hierarchy (which lies between NP and PSPACE), based on results of [2]. Our first main result is that CIT can be placed in BPP (i.e., randomized polynomial time) assuming GRH, and is in coNP unconditionally. The algorithm works by computing modulo a suitable prime ideal in the ring of integers of the number field  $\mathbb{Q}(\zeta_n)$ .

**THEOREM 1.** *The CIT problem is in BPP assuming GRH, and is in coNP unconditionally.*

Observe that the CIT problem is at least as hard as the Polynomial Identity Testing problem for circuits of unbounded degree, which is well-known to be P-hard [29, Theorem 2.4.6, Theorem 2.6.3]. As a result, algorithms for CIT are inherently sequential. We inspect two natural restrictions of CIT and show that they admit efficient parallel algorithms.

The complexity class NC formalises those polynomial-time-computable problems that are considered to be efficiently parallelizable. Formally, a problem is in NC if on instances of size  $n$ , it can be solved in  $(\log n)^{O(1)}$  time using  $n^{O(1)}$  processors in the PRAM model. Almost all natural problems in arithmetic [7] and linear algebra are known to be in NC [14]. However, membership in NC is open for GCD computation, modular powering, and primality testing. (Several of these problems are, however, known to admit efficient *randomized* NC algorithms.)

We consider the Bounded-CIT problem, in which the syntactic degree of the circuit is polynomially bounded, and give a randomized NC procedure with two-sided errors. Here we forsake the approach via finite arithmetic because computing powers in a finite field is not known to be in NC. Instead, we follow the identity testing method of Chen and Kao [10]: we pick a Galois conjugate of  $f(\zeta_n)$  uniformly at random and determine the zeroness of the conjugate by numerical computation. Thus we have:

**THEOREM 2.** *The Bounded-CIT problem is in randomized NC.*

Moving to the problem Sparse-CIT, we revisit the approach of [12], who gave a polynomial-time decision procedure. Here we give a simpler reformulation of their method and, as a by-product, we observe that the problem can be solved in NC.

**THEOREM 3.** *The Sparse-CIT problem is in NC.*

Theorems 1, 2, and 3 all take different approaches to the CIT problem: respectively using finite arithmetic, numerical approximation, and multilinear algebra. However it is interesting to note that all three approaches involve computing a partial prime factorisation of the order of the root of unity (or some multiple thereof).

Intermediate between Bounded-CIT and Sparse-CIT, we consider CIT for diagonal circuits. Here we observe that if CIT for diagonal circuits were solvable in polynomial time, then polynomial identity testing (PIT) for algebraic circuits of size  $s$  and degree  $d$  would be solvable in time polynomial in  $s^{O(\sqrt{d})}$ . However we have:

**THEOREM 4.** *Assuming GRH, CIT can be solved in polynomial time on the class of polynomials of the form  $f = \sum_{i=1}^m g_i^{d_i}$  with  $g$  in sparse representation and  $d_i$  an integer in unary.*

In terms of applications, we observe that cyclotomic identity testing can be used to obtain a new randomized NC algorithm to decide equality of compressed strings, that is, strings presented by acyclic context-free grammars (see Section 1.1 for previous work on this problem.)

## 1.1 Related work

As discussed in [12], cyclotomic identity testing is related to the so-called *torsion-point problem*, which asks whether a given multivariate polynomial has a zero in which all components are roots of unity [35]. The univariate version of this problem is known to be NP-hard [32].

Theorem 1 is a generalisation of the problem of testing equality of straight-line programs over the integers as studied by Schönhage [37] (see also Allender et al. [2]) to cyclotomic number fields. Testing zeroness of expressions involving real roots of rational numbers is considered in [8].

Lenstra [25] and Kaltofen and Koiran [21] gave polynomial-time algorithms for testing zeroness of sparse univariate and multivariate polynomials respectively on algebraic numbers of degree polynomially bounded in the problem instance (whereas Sparse-CIT features roots of unity whose degree can be exponential in the size of the problem instance).

There has been extensive work on the problem of testing equality of compressed strings, including Hirschfeld et al. [18]

( $O(n^4)$  time), Plandowski [33] and Melhorn et al. [26] ( $O(n^3)$  time), and Jeř [20] ( $O(n^2)$  time). Note that the quadratic running time in the latter is in a RAM model where arithmetic on integers (the binary encoding of a position in the uncompressed string fits into a single machine word) can be performed in a single time step.

König and Lohrey [23] show that the problem admits a randomised NC algorithm by reduction to the identity testing problem for univariate polynomials given as so-called powerful skew circuits. The main contribution of [23] is a randomised NC algorithm for the latter problem. Following the identity testing technique of Agrawal and Biswas [1], their algorithm works, by computing the value of the circuit modulo a randomly chosen polynomial  $p(x)$ . In order to perform this computation in NC they rely on the result of Fich and Tompa [16] that computing  $x^m \bmod p(x)$  for large powers  $m$  can be done in NC (assuming  $p$  is given in dense representation). By contrast, we observe that the same identity testing problem can be solved by numerically evaluating a polynomial at a randomly chosen conjugate of a root of unity  $\zeta_n$  of sufficiently high order. To obtain an NC bound we rely on the fact that it is straightforward to compute powers of  $\zeta_n$ . We also observe that our technique yields a randomised sequential algorithm that runs in  $\tilde{O}(n^2)$  time in the standard Turing machine model.

All missing proofs can be found in [5].

## 2 PRELIMINARIES

We give some background results on arithmetic and cyclotomic fields. Throughout the paper we use  $\log x$  to denote  $\log_2 x$ .

**Proposition 5.** Fix  $m \in \mathbb{N}$  and consider drawing an element  $k$  uniformly at random from the set  $\{1, \dots, m-1\}$ . Let  $A$  be the event that  $k$  and  $m$  are coprime and let  $B$  be the event that  $k$  and  $m$  share no common prime divisor  $p < 10 \log m$ . Then  $\Pr(A \mid B) > \frac{9}{10}$  for  $m$  sufficiently large.

The following estimates on the density of primes in an arithmetic progression can be found in [15, Chapter 20, page 125] and [19, Corollary 18.8] respectively.

**THEOREM 6.** Given  $a \in \mathbb{Z}_n^*$ , write  $\pi_{n,a}(x)$  for the number of primes less than  $x$  that are congruent to  $a$  modulo  $n$ . Then under GRH, there is an absolute constant  $C > 0$  such that

$$\pi_{n,a}(x) \geq \frac{x}{\varphi(n) \log x} - C\sqrt{x} \log(x).$$

Unconditionally, there exist effective absolute positive constants  $C_1$  and  $C_2$  such that for all  $n < C_1 x^{C_1}$ ,

$$\pi_{n,a}(x) \geq \frac{C_2 x}{\varphi(n) \sqrt{n} \log(x)}.$$

Recall that  $\alpha \in \mathbb{C}$  is an *algebraic number* if it is the root of a non-zero polynomial in  $\mathbb{Q}[x]$ . The *minimal polynomial* of  $\alpha$  is the unique monic polynomial in  $\mathbb{Q}[x]$  having  $\alpha$  as a root. If the minimal polynomial has integer coefficients then we say that  $\alpha$  is an *algebraic integer*. The degree and height of an algebraic integer are respectively the degree and the maximum absolute value of a coefficient in its minimal polynomial.

Fix  $n \in \mathbb{N}$  and write  $\mathbb{Q}(\zeta_n)$  for the field obtained by adjoining a primitive complex  $n$ -th root of unity  $\zeta_n = e^{\frac{2\pi i}{n}}$  to  $\mathbb{Q}$ . The  $n$ -th *cyclotomic polynomial* is the minimal polynomial of  $\zeta_n$ . It is well known that  $\Phi_n$  has degree  $\varphi(n)$ , where  $\varphi$  is the Euler totient function.

It is well known that the sub-ring of  $\mathbb{Q}(\zeta_n)$  comprised of algebraic integers, called the sub-ring of cyclotomic integers, is the ring  $\mathbb{Z}[\zeta_n]$  that is generated over  $\mathbb{Z}$  by  $\zeta_n$ .

Recall that the group  $\text{Gal}(\mathbb{Q}(\zeta_n)/\mathbb{Q})$  of automorphisms of  $\mathbb{Q}(\zeta_n)$  is isomorphic to the multiplicative group  $\mathbb{Z}_n^*$  of integers mod  $n$ . For each  $k \in \mathbb{Z}_n^*$ , the corresponding automorphism  $\sigma$  in  $\text{Gal}(\mathbb{Q}(\zeta_n)/\mathbb{Q})$  is defined by  $\sigma(\zeta_n) = \zeta_n^k$ .

The image of  $\alpha \in \mathbb{Q}(\zeta_n)$  under an automorphism of  $\mathbb{Q}(\zeta_n)$  is called a *Galois conjugate* of  $\alpha$ . The *norm* of  $\alpha$  is defined by

$$N_{\mathbb{Q}(\zeta_n)/\mathbb{Q}}(\alpha) := \prod_{\sigma \in \text{Gal}(\mathbb{Q}(\zeta_n)/\mathbb{Q})} \sigma(\alpha)$$

For short, we will write  $N(\alpha)$  for  $N_{\mathbb{Q}(\zeta_n)/\mathbb{Q}}(\alpha)$ , i.e., the underlying field will be understood from the context. Recall that the norm of a cyclotomic integer lies in  $\mathbb{Z}$ .

## 3 A RANDOMISED POLYNOMIAL-TIME ALGORITHM FOR CIT

In this section we give a randomised polynomial-time algorithm for the CIT problem, assuming GRH. Furthermore, we show that the problem is in coNP unconditionally. The idea is to work in a finite field, obtained by quotienting the ring of cyclotomic integers by a suitable prime ideal.

Throughout this section, we say that the cyclotomic integer  $f(\zeta_n)$  has a *description of size*  $s \in \mathbb{N}$  if it is computed by a circuit  $C$  where  $s$  is the sum of the size of  $C$  and the bit-length of  $n$ .

**Proposition 7.** Let  $\alpha \in \mathbb{Z}(\zeta_n)$  be a cyclotomic integer with a description of size  $s$ . Then  $|N(\alpha)| \leq 2^{2s}$ .

**THEOREM 8.** Let  $p \in \mathbb{Z}$  be a prime such that  $\mathbb{F}_p$  contains a primitive  $n$ -th root of unity  $\omega_n$ . Given  $g(x) \in \mathbb{Z}[x]$ , denoting by  $\bar{g} \in \mathbb{F}_p[x]$  the reduction of  $g$  modulo  $p$ , we have

- (1) if  $g(\zeta_n) = 0$  then  $\bar{g}(\omega_n) = 0$ , and
- (2) if  $\bar{g}(\omega_n) = 0$  then  $p \mid N(g(\zeta_n))$ .

**PROOF.** Define a ring homomorphism  $\text{ev} : \mathbb{Z}[x] \rightarrow \mathbb{F}_p$  by  $\text{ev}(g) = \bar{g}(\omega_n)$ . For  $d < n$ , since  $\Phi_d \mid x^d - 1$  and  $\text{ev}(x^d - 1) \neq 0$ , we have  $\text{ev}(\Phi_d) \neq 0$ . Since also  $x^n - 1 = \prod_{d \mid n} \Phi_d$ , we have  $\text{ev}(\Phi_n) = 0$ . It follows that the map  $\text{ev}$  factors through the homomorphism  $\text{ev}' : \mathbb{Z}(\zeta_n) \rightarrow \mathbb{F}_p$  given by  $\text{ev}'(g(\zeta_n)) = \bar{g}(\omega_n)$  for  $g \in \mathbb{Z}[x]$ .

For Item 1, if  $g(\zeta_n) = 0$  then  $\bar{g}(\omega_n) = \text{ev}'(g(\zeta_n)) = 0$ .

For Item 2, observe that the kernel of  $\text{ev}'$  is a prime ideal  $\mathfrak{p}$  in  $\mathbb{Z}(\zeta_n)$  satisfying  $\mathfrak{p} \cap \mathbb{Z} = p\mathbb{Z}$ . Hence if  $\bar{g}(\omega_n) = 0$  then  $g(\zeta_n) \in \mathfrak{p}$  and so  $p \mid N(g(\zeta_n))$ .  $\square$

Theorem 8 suggests a natural test for CIT: evaluate the circuit in a finite field  $\mathbb{F}_p$  that contains a primitive  $n$ -th root of unity. Since the multiplicative group  $\mathbb{F}_p^*$  is cyclic, it is clear that  $\mathbb{F}_p^*$  contains a primitive  $n$ -th root of unity just in case  $n \mid (p-1)$ , i.e.,  $p \equiv 1 \pmod n$ .

<b>Cyclotomic Identity Testing</b>	
<b>Input:</b>	Algebraic circuit $C$ and integer $n$ , written in binary, of combined size $s$ .
<b>Output:</b>	Whether $f(\zeta_n) = 0$ for the polynomial $f(x)$ computed by $C$ .
1:	Pick $p$ uniformly at random from $\{q \in \mathbb{N} : q \leq 2^{5s}, q \text{ prime, and } q \equiv 1 \pmod n\}$ .
2:	Pick $h$ uniformly at random from $\{a : a \in \mathbb{F}_p^*, \bigwedge_{\substack{2 \leq q < 10 \log(p-1) \\ q p-1}} a^{\frac{p-1}{q}} \neq 1\}$ .
3:	Set $\omega_n := h^{\frac{p-1}{n}} \in \mathbb{F}_p^*$ .
4:	Output ‘Zero’ if $\bar{f}(\omega_n) = 0$ where $\bar{f}$ is the reduction of $f$ modulo $p$ ; otherwise output ‘Non-Zero’.

**Figure 1: Algorithm for the CIT problem**

**Proposition 9.** *Let  $\alpha \in \mathbb{Z}(\zeta_n)$  be a non-zero cyclotomic integer whose description has size at most  $s$ . Suppose that  $p$  is chosen uniformly at random from  $\{q \in \mathbb{N} : q \leq 2^{5s} \text{ and } q \equiv 1 \pmod n\}$ . Assuming GRH, (i)  $p$  is prime with probability at least  $\frac{1}{6s}$ , and (ii) given that  $p$  is prime, the probability that it divides  $N(\alpha)$  is at most  $2^{-s}$ .*

**PROOF.** For (i), we note that by Theorem 6, the probability that  $p$  is prime is at most

$$\frac{\pi_{n,1}(2^{5s})}{2^{5s}/n} \geq \frac{n}{\varphi(n) \log(2^{5s})} - \frac{Cn \log(2^{5s})}{\sqrt{2^{5s}}} \geq \frac{1}{5s} - \frac{C2^s 5s}{2^{2s}},$$

where  $C$  is the absolute constant mentioned in the theorem. But the above is at least  $\frac{1}{6s}$  for  $s$  sufficiently large.

For (ii), by Proposition 7 the norm of  $\alpha$  has absolute value at most  $2^{2s}$ , and hence  $N(\alpha)$  has at most  $2^{2s}$  distinct prime factors. Then, for  $s$  sufficiently large, the probability that  $p$  divides  $N(\alpha)$  given that  $p$  is prime is at most  $\frac{6sn2^{2s}}{2^{5s}} \leq 2^{-s}$ .  $\square$

A straightforward application of Proposition 5 gives the following proposition, enabling us to find primitive  $n$ -th roots of unity in  $\mathbb{F}_p$  in case  $p \equiv 1 \pmod n$ .

**Proposition 10.** *For a prime  $p$ , let  $h$  be chosen uniformly at random from the set*

$$\left\{ a \in \mathbb{F}_p^* : \bigwedge_{\substack{2 \leq q < 10 \log(p-1) \\ q|p-1}} a^{\frac{p-1}{q}} \neq 1 \right\}.$$

*Then  $h$  is a primitive root of  $\mathbb{F}_p^*$  with probability at least 0.9.*

**PROOF.** Fix a primitive root  $g \in \mathbb{F}_p^*$ . For  $a$  distributed uniformly at random over  $\mathbb{F}_p^*$ , we have that  $\log_g a$  (the discrete logarithm of  $a$  in  $\mathbb{F}_p^*$ ) is distributed uniformly at random over  $\{0, \dots, p-2\}$ . Moreover, for every divisor  $q$  of  $p-1$ ,  $q$  divides  $\log_g a$  if and only if  $a^{\frac{p-1}{q}} = 1 \pmod p$ . It follows that for  $h$  as in the statement of the proposition,  $\log_g h$  is distributed uniformly at random among those elements in  $\{2, \dots, p-2\}$  that do not share a divisor less than  $10 \log(p-1)$  with  $p-1$ . Applying Proposition 5 we have that  $\log_g h$  is coprime with  $p-1$

with probability at least 0.9. But  $\log_g h$  is coprime with  $p-1$  if and only if  $h$  is itself a primitive root of  $\mathbb{F}_p^*$ .  $\square$

We are now in a position to prove the main result of this section:

**THEOREM 1.** *The CIT problem is in BPP assuming GRH, and is in coNP unconditionally.*

**PROOF.** Figure 1 presents a Monte Carlo randomized algorithm for the CIT problem. The argument for the correctness of the algorithm is as follows. Let  $p$  be a prime such that  $p \equiv 1 \pmod n$ , as chosen in Line 1.

It follows from Proposition 10 that with probability at least 0.9, the element  $h \in \mathbb{F}_p^*$  that is selected in Line 2 is a primitive root of  $\mathbb{F}_p^*$ . Now let us bound the error of the algorithm under the assumption that  $h$  is indeed a primitive root of  $\mathbb{F}_p^*$ . Note that in this case we have that  $\omega_n$ , as chosen in Line 3, is a primitive  $n$ -th root of unity in the field  $\mathbb{F}_p$ . We consider two cases. First, suppose that  $f(\zeta_n) = 0$ ; then by Theorem 8 we have  $\bar{f}(\omega_n) = 0$ , and hence the output is ‘Zero’. Second, suppose that  $f(\zeta_n) \neq 0$ . Then by Theorem 8 the output will be ‘Non-Zero’ provided that  $p$  does not divide  $N(f(\zeta_n))$ . But by Proposition 9(ii) the probability that  $p$  does not divide  $N(f(\zeta_n))$  is at least  $1 - 2^{-s}$ . Thus, in total, the probability that the algorithm gives the wrong output is at most  $0.1 + 2^{-s}$ .

It is clear that the algorithm runs in polynomial time. In particular, in Line 1, by Proposition 9(i) we can choose a prime uniformly at random from the set  $\{q \in \mathbb{N} : q \leq 2^{5s}, q \equiv 1 \pmod n\}$  by random sampling with  $O(s)$  repetitions with a small constant failure probability.

It remains to show that the CIT problem lies in coNP. The idea is to modify the algorithm in Figure 1, replacing randomisation with guessing. Suppose  $f(\zeta_n) \neq 0$ . The unconditional lower bound in Theorem 6 shows that  $\pi_{n,1}(2^{s'}) > 2^{2s}$  for  $s$  sufficiently large and  $s' > \max(4s, s/C_1)$ . It follows that there exists a prime  $p \leq 2^{s'}$  that does not divide  $N(f(\zeta_n))$  such that  $p \equiv 1 \pmod n$ . The polynomial certificate of non-zerosness of  $f(\zeta_n)$  then comprises, the above prime  $p$ , a list of the prime factors of  $p-1$ , and an element  $h \in \mathbb{F}_p^*$  such that  $h^{\frac{p-1}{q}} \neq 1$  for all prime factors  $q$  of  $p-1$ . Such an  $h$  is a generator of  $\mathbb{F}_p^*$  and so  $\omega_n := h^{\frac{p-1}{n}}$  is a primitive  $n$ -th root of unity. We then have that  $\bar{f}(\omega_n) \neq 0$ . On the other hand, as noted above, for any prime  $p$  and primitive  $n$ -th root of unity  $\omega_n \in \mathbb{F}_p^*$ , if  $f(\zeta_n) = 0$  then  $\bar{f}(\omega_n) = 0$ .  $\square$

**Corollary 11.** *Assuming GRH, there is a Monte Carlo randomised algorithm that solves CIT for cyclotomic integer of description size  $s$ , using  $O(s^2)$  random bits and  $O(s^3)$  arithmetic operations.*

In [5] we give an example showing that the above algorithm has two-sided errors.

## 4 A RANDOMISED NC ALGORITHM FOR BOUNDED-CIT

In this section we give a randomized NC algorithm for the Bounded-CIT problem. The algorithm is shown in Figure 2.

<b>Bounded Cyclotomic Identity Testing</b>	
<b>Input:</b>	Algebraic circuit $C$ with a unary upper bound on its syntactic degree, and integer $n$ written in binary, of combined size $s$ .
<b>Output:</b>	Whether $f(\zeta_n) = 0$ for the polynomial $f(x)$ computed by $C$ .
1:	Pick uniformly at random $a \in \{1, \dots, n-1\}$ such that $a$ and $n$ have no common divisor less than $10 \log n$ .
2:	Compute $\alpha \in \mathbb{Q}(i)$ such that $ f(\zeta_n^a) - \alpha  < 2^{-4s-1}$ .
3:	Output ‘Zero’ if $ \alpha  < 2^{-4s-1}$ , otherwise output ‘Non-Zero’.

**Figure 2: Algorithm for the Bounded-CIT problem**

The input is an algebraic circuit  $C$  with a unary upper bound on its syntactic degree (but with high-powered inputs), together with an integer  $n$  written in binary. The desired output is whether or not  $f(\zeta_n) = 0$  for  $f(x)$  the polynomial represented by  $C$ .

Intuitively, Line 1 of the algorithm attempts to select a Galois conjugate  $\zeta_n^a$  of  $\zeta_n$  uniformly at random. Here, to avoid having to check whether  $\gcd(a, n) = 1$ —which is not known to be in NC—we only have a Galois conjugate with high probability. Line 2 computes a numerical approximation  $\alpha$  of  $f(\zeta_n^a)$  to precision  $2^{-4s-1}$ . Finally, Line 3 outputs ‘Zero’ if and only if  $\alpha$  has absolute value at most  $2^{-4s-1}$ .

**Numerical approximation.** We now explain how the numerical approximation  $\alpha$  in Line 2 can be computed in NC with the desired precision.

Write  $\varepsilon := 2^{-s^2-5s-1}$ . Taking  $O(s^2)$  terms of the power series involved in Machin’s formula [39] for  $\pi$  we obtain  $\tilde{\pi} \in \mathbb{Q}$  such that  $|\pi - \tilde{\pi}| < \varepsilon/3$ . Likewise, for each  $n$ -th root of unity  $\zeta_n^\ell$ , by taking  $O(s^2)$  terms in the power series expansion for  $e^{2\pi i \ell/n}$ , we obtain  $\tilde{\zeta}_n^\ell \in \mathbb{Q}(i)$  such that  $|\tilde{\zeta}_n^\ell - e^{2\pi i \ell/n}| < \varepsilon/3$ . We then have that, for all  $0 \leq \ell < n$ ,

$$\begin{aligned}
|\tilde{\zeta}_n^\ell - \zeta_n^\ell| &\leq |\tilde{\zeta}_n^\ell - e^{2\pi i \ell/n}| + |e^{2\pi i \ell/n} - \zeta_n^\ell| \\
&\leq \varepsilon/3 + |1 - e^{2(\tilde{\pi}-\pi)i\ell/n}| \\
&\leq \varepsilon/3 + 2|\tilde{\pi} - \pi| \\
&\leq \varepsilon.
\end{aligned}$$

Now  $f(x) = g(x^{e_1}, \dots, x^{e_k})$ , where  $e_1, \dots, e_k$  are positive integers and  $g$  is a multivariate polynomial represented by a circuit of size  $s$  and syntactic degree  $d \leq s$ . We define

$$\alpha := g(\tilde{\zeta}_n^{ae_1}, \dots, \tilde{\zeta}_n^{ae_k}) \in \mathbb{Q}(i)$$

In other words,  $\alpha$  is obtained by evaluating the circuit  $C$  on inputs  $\tilde{\zeta}_n^{ae_1}, \dots, \tilde{\zeta}_n^{ae_k}$ . Now each monomial in  $g$  has total degree at most  $d$  and the sum of the absolute values of the coefficients of  $g$  is at most  $2^{sd}$ . Thus we have, by definition of  $\varepsilon$ ,

$$\begin{aligned}
|f(\zeta_n^a) - \alpha| &= |g(\zeta_n^{ae_1}, \dots, \zeta_n^{ae_k}) - g(\tilde{\zeta}_n^{ae_1}, \dots, \tilde{\zeta}_n^{ae_k})| \\
&\leq 2^{sd} |1 - (1 + \varepsilon)^d| \\
&\leq 2^{sd} \varepsilon 2^d \leq 2^{-4s-1}.
\end{aligned}$$

By construction, the approximants  $\tilde{\zeta}_n^{ae_1}, \dots, \tilde{\zeta}_n^{ae_k}$  are represented by arithmetic circuits of size and degree polynomial in  $s$  that can moreover be computed in space logarithmic in  $s$ . Composing these circuits with  $C$  gives an arithmetic circuit for  $\alpha$  that has size and degree polynomial in  $s$ . We now use the classical result of Valiant *et al.* [38] that given an arithmetic circuit of degree  $d$  and size  $\sigma$  one can construct (in logarithmic space [3]) an equivalent arithmetic circuit of depth  $O(\log(d) \log(\sigma))$  and size polynomial in  $\log d$  and  $\sigma$ . Every bit of the numbers produced at each gate of the resulting compressed-depth circuit can be computed by Boolean NC circuits of size at most  $O(\sigma \log \sigma)$  [34]. Applying this transformation to the algebraic circuit for  $\alpha$  results in a Boolean circuit for (the bits of)  $\alpha$  of size polynomial in  $s$  and depth polynomial in  $\log(s)$  that is moreover computable in space  $\log s$ . This shows that  $\alpha$  is computable in NC.

**Correctness.** The probabilistic correctness of the algorithm relies on the following well-known result:

**Proposition 12.** [Chen and Kao [10] and Blömer [8]] *Let  $\alpha$  be a non-zero algebraic integer where the absolute value of all its conjugates is at most  $2^B$ . For all  $b \in \mathbb{N}$ , a random conjugate  $\alpha'$  of  $\alpha$  satisfies  $|\alpha'| \leq 2^{-b}$ , with probability at most  $B/(b+B)$ .*

Observe that the constants appearing in  $f$  have magnitude at most  $2^s$  and that  $f$  has degree less than  $2^s$ . It follows that  $|f(\zeta_n^\ell)| \leq 2^{2s}$  for all  $\ell$ . Using Proposition 12 with  $B = 2s$  and  $b = 4s$ , whenever  $f(\zeta_n)$  is non-zero, a random conjugate  $f(\zeta_n^a)$  of  $f(\zeta_n)$  has absolute value larger than  $2^{-4s}$  with probability at least  $\frac{2}{3}$ .

In Line 1, the algorithm takes a polynomial number of samples from  $\{1, \dots, n-1\}$  uniformly at random (in parallel) and returns any  $a$  such that  $a$  and  $n$  have no common divisors less than  $10 \log n$ . By Proposition 5, we have that  $a$  is coprime with  $n$ , and hence  $\zeta_n^a$  is a conjugate of  $\zeta_n$ , with probability at least  $\frac{9}{10}$ .

To estimate the error probability we consider two cases. First, suppose that  $f(\zeta_n) \neq 0$ . Then with probability at least  $\frac{9}{10} \cdot \frac{2}{3} = \frac{3}{5}$  we have that  $|f(\zeta_n^a)| > 2^{-4s}$  and hence  $|\alpha| > 2^{-4s-1}$ . Second, suppose that  $f(\zeta_n) = 0$ . Then with probability at least  $\frac{2}{3}$  we have  $f(\zeta_n^a) = 0$  and hence  $|\alpha| < 2^{-4s-1}$ . Thus the error probability is at most  $\frac{2}{5}$ . Finally we obtain:

**THEOREM 2.** *The Bounded-CIT problem is in randomized NC.*

#### 4.1 Compressed Words and Powerful Skew Circuits

An algebraic circuit computing a univariate polynomial is said to be a *powerful skew circuit* if at least one input of every multiplication gate is a leaf. Here the word *powerful* reflects our convention that leaves can be labelled with monomials  $x^m$ , where  $m$  is given in binary. The main motivation for studying this identity testing problem is that there is an NC reduction of the equivalence testing problem for compressed strings to identity testing for powerful skew circuits [23]. Briefly, a compressed word is one that is given by an acyclic context-free grammar in which each non-terminal occurs on the left-hand side of exactly one production. Such a grammar produces a

single word, whose length can be exponential in the number of non-terminals and productions. See [23] for more details.

In this section we give an alternative randomised NC algorithm for PIT on powerful skew circuits, employing the same random conjugate technique used to solve the Bounded-CIT problem. Since the syntactic degree of a powerful skew circuit is at most the number of gates we can use our Algorithm in Figure 2 to decide PIT over the class of powerful skew circuits: we simply pick a root of unity  $\zeta_n$  with  $n$  higher than the degree of the given polynomial  $f \in \mathbb{Z}[x]$ , and approximate a random conjugate of  $f(\zeta_n)$ .

Since the algorithm is insensitive to the choice of  $n$ , as long as it is larger than the degree of  $f$  (that is at most  $2^s$  where  $s$  is the size of circuit), we choose  $n = 2^{4s}$  ensuring that  $\zeta_n^a$  is a conjugate of  $\zeta_n$  for all odd numbers  $a$ ,  $1 \leq a < n$ . This prevents one type of error in our randomised algorithm for the Bounded-CIT problem (the error caused by picking a non-conjugate in Line 1 of Figure 2); indeed, whenever  $f(\zeta_n) = 0$  our algorithm returns ‘Zero’ with probability 1. Then we conclude the following theorem noting that the approximation is efficiently computable in randomized sequential time using Brent’s algorithm [9].

**THEOREM 13.** *Testing equality of two compressed words, of combined size  $s$ , is solvable in  $\tilde{O}(s^2)$ -time by a randomized sequential algorithm. Furthermore, it can be implemented by  $\tilde{O}(s^3)$ -sized NC circuits using  $O(s)$  random bits.*

## 5 AN NC ALGORITHM FOR SPARSE-CIT

Cheng *et al.* [12] showed how to solve the Sparse-CIT in polynomial time. Their method involves a tensor decomposition of the space of all polynomials that vanish on a given root of unity  $\zeta_n$ , based on a partial factorisation of  $n$ . They then exploit sparsity to efficiently determine membership of this space (which has dimension  $n$ , exponential in the length of the problem instance). Below we reformulate this idea to avoid working with vector spaces of exponential dimension. We work instead with a space of vanishing sums (see (1)) whose dimension equals the number of monomials of the input polynomial. Our reformulation relies on a simple proposition in linear algebra (Proposition 15), which not only simplifies the approach of Cheng *et al.*, but allows to place the problem Sparse-CIT in NC.

Let  $\zeta_n$  denote a primitive  $n$ -th root of unity for a positive integer  $n$ . Given a vector  $\mathbf{k} = (k_1, \dots, k_s)$  of nonnegative integers where  $k_1 < \dots < k_s$ , we aim to compute the space of *vanishing sums*

$$V_n^{\mathbf{k}} := \left\{ a \in \mathbb{Q}^s : \sum_{i=1}^s a_i \zeta_n^{k_i} = 0 \right\} \quad (1)$$

in polylogarithmic parallel time in the total bit length of  $n$  and  $k_1, \dots, k_s$ .

### 5.1 Composing spaces of vanishing sums

In the approach of [12][Section 2.1] the following (which is an easy consequence of the Chinese Remainder Theorem) plays a central role:

**Proposition 14.** *Suppose that  $n = n_1 n_2$  for  $n, n_1, n_2 \in \mathbb{N}$ , with  $n_1$  and  $n_2$  coprime. Then the map  $\zeta_n \mapsto \zeta_{n_1} \otimes \zeta_{n_2}$  defines a  $\mathbb{Q}$ -algebra isomorphism between  $\mathbb{Q}(\zeta_n)$  and  $\mathbb{Q}(\zeta_{n_1}) \otimes \mathbb{Q}(\zeta_{n_2})$ .*

Given vectors  $a, b \in \mathbb{Q}^s$ , define the *Hadamard product*  $a \odot b \in \mathbb{Q}^s$  by  $a \odot b := (a_1 b_1, \dots, a_s b_s)$ . Furthermore, given  $U, V$  vector sub-spaces of  $\mathbb{Q}^s$ , define

$$U \odot V = \text{span}\{u \odot v : u \in U, v \in V\}.$$

Given bases of  $U$  and  $V$ , we can compute a basis of  $U \odot V$  in NC by constructing the set of products  $u \odot v$  as  $u$  ranges over the basis of  $U$  and  $v$  ranges over the basis of  $V$ , and then selecting a maximally linearly independent subset of the resulting collection of vectors (e.g., by the algorithm of [13]). The operator  $\odot$  is moreover associative, so given a list of sub-spaces  $U_1, \dots, U_\ell \subseteq \mathbb{Q}^s$ , we can compute the iterated product  $U_1 \odot \dots \odot U_\ell$  in NC using the parallel prefix technique [24].

Denote by  $U^\perp$  the orthogonal complement of  $U \subseteq \mathbb{Q}^s$ .

**Proposition 15.** *Let  $U, V$  be finite dimensional vector spaces over  $\mathbb{Q}$  with  $u_1, \dots, u_s \in U$  and  $v_1, \dots, v_s \in V$  for some  $s \in \mathbb{N}$ . Define the following three vector subspaces of  $\mathbb{Q}^s$ :*

$$\begin{aligned} A &:= \{a \in \mathbb{Q}^s : \sum_{i=1}^s a_i u_i = 0\} \\ B &:= \{b \in \mathbb{Q}^s : \sum_{i=1}^s b_i v_i = 0\} \\ C &:= \{c \in \mathbb{Q}^s : \sum_{i=1}^s c_i (u_i \otimes v_i) = 0\}. \end{aligned}$$

Then  $C^\perp = A^\perp \odot B^\perp$ .

**PROOF.** For a non-negative integer  $k$  and list of vectors  $w_1, \dots, w_s \in \mathbb{Q}^k$ , write  $R(w_1, \dots, w_s)$  for the row space of the matrix with columns  $w_1, \dots, w_s$ . Recall that  $R(w_1, \dots, w_s)$  is the orthogonal complement of  $\{a \in \mathbb{Q}^s : \sum_{i=1}^s a_i w_i = 0\}$ .

Without loss of generality, suppose that  $U = \mathbb{Q}^m$  and  $V = \mathbb{Q}^n$ . Then we can identify  $U \otimes V$  with  $\mathbb{Q}^{mn}$  by taking  $u \otimes v$  to be the Kronecker product of  $u \in U$  and  $v \in V$ . Now we have

$$\begin{aligned} A^\perp &= R(u_1, \dots, u_s) \\ B^\perp &= R(v_1, \dots, v_s) \\ C^\perp &= R(u_1 \otimes v_1, \dots, u_s \otimes v_s). \end{aligned} \quad (2)$$

But it clearly also holds that

$$R(u_1 \otimes v_1, \dots, u_s \otimes v_s) = R(u_1, \dots, u_s) \odot R(v_1, \dots, v_s). \quad (3)$$

The result follows from Equations (2) and (3).  $\square$

The following result follows from Propositions 14 and 15:

**Corollary 16.** *Let  $n_1$  and  $n_2$  be coprime positive integers, then*

$$(V_{n_1 n_2}^{\mathbf{k}})^\perp = (V_{n_1}^{\mathbf{k}})^\perp \odot (V_{n_2}^{\mathbf{k}})^\perp.$$

### 5.2 Base cases

We will use Corollary 16 in tandem with the following known characterisations of the vanishing spaces for prime powers and composite numbers.

**Proposition 17.** *Let  $p$  be a prime,  $e$  a positive integer, and let  $0 \leq k_1 < \dots < k_s < p^e$  be non-negative integers. Given  $a \in \mathbb{Q}^s$ , we have  $\sum_{i=1}^s a_i \zeta_{p^e}^{k_i} = 0$  if and only if (i)  $a_i = a_j$  for all  $i, j$  such that  $k_i \equiv k_j \pmod{p^{e-1}}$  and (ii)  $a_i = 0$  for all  $i$  such that  $\#\{k_j : k_j \equiv k_i \pmod{p^{e-1}}\} < p$ .*

**Proposition 18.** Let  $f(x) = \sum_{i=1}^s a_i x^{k_i} \in \mathbb{Q}[x]$  be a polynomial such that  $0 \leq k_1 < \dots < k_s < n$  and suppose that  $p > s$  for all prime divisors  $p$  of  $n$ . Then  $f(\zeta_n) = 0$  only if  $f$  is identically zero.

PROOF. Write  $n = p_1^{e_1} \dots p_m^{e_m}$  for the prime factorization of  $n$ . Write  $\ell_{ij} := k_i \bmod p_j^{e_j}$  for  $i = 1, \dots, s$  and  $j = 1, \dots, m$ . By the Chinese Remainder Theorem the  $m$ -tuples  $\ell_i = (\ell_{i1}, \dots, \ell_{im})$ ,  $i = 1, \dots, s$ , are all distinct. Now we have

$$\begin{aligned} f(\zeta_n) = 0 &\Leftrightarrow \sum_{i=1}^s a_i \zeta_n^{k_i} = 0 \\ &\Leftrightarrow \sum_{i=1}^s a_i (\zeta_{p_1^{e_1}}^{\ell_{i1}} \otimes \dots \otimes \zeta_{p_m^{e_m}}^{\ell_{im}}) = 0. \end{aligned}$$

But, by Proposition 17,  $\left\{ \zeta_{p_j^{e_j}}^{\ell_{1j}}, \dots, \zeta_{p_j^{e_j}}^{\ell_{sj}} \right\}$  is a linearly independent set in  $\mathbb{Q}(\zeta_{p_j^{e_j}})$  for all  $j = 1, \dots, m$  (possibly listed with repetitions). It follows that

$$\left\{ \zeta_{p_1^{e_1}}^{\ell_{i1}} \otimes \dots \otimes \zeta_{p_m^{e_m}}^{\ell_{im}} : i = 1, \dots, s \right\}$$

is a linearly independent set in  $\mathbb{Q}(\zeta_n)$ . Since the  $\ell_i$  are all distinct we conclude that  $a_1 = \dots = a_s = 0$ .  $\square$

### 5.3 Putting Things Together

**THEOREM 3.** *The Sparse-CIT problem is in NC.*

PROOF. Given  $f(x) = \sum_{i=0}^s a_i x^{k_i}$  and  $n \in \mathbb{N}$ , we wish to determine whether  $f(\zeta_n) = 0$ .

Since integer division is in NC, given  $n \in \mathbb{N}$  one can compute in NC a factorisation  $n = p_1^{e_1} \dots p_\ell^{e_\ell} m$  such that all  $p_1, \dots, p_\ell \leq s$  are prime and all prime factors of  $m$  are strictly greater than  $s$ .

Let  $\mathbf{k} = (k_1, \dots, k_s)$ . We use Propositions 17 and 18 to compute the vanishing spaces  $V_{p_i^{e_i}}^{\mathbf{k}}$  for  $i = 1, \dots, \ell$  and  $V_m^{\mathbf{k}}$ . More precisely, to compute  $V_m^{\mathbf{k}}$  we let  $0 \leq k'_1 < \dots < k'_t < m$  be a list of the distinct residues of  $k_1, \dots, k_s$  modulo  $m$ , and define a map  $T : \mathbb{Q}^s \rightarrow \mathbb{Q}^t$  by  $T(a_1, \dots, a_s) = (b_1, \dots, b_t)$ , where  $b_i := \sum \{a_j : k_j \equiv k'_i \bmod m\}$  for  $i = 1, \dots, t$ . Then  $V_m^{\mathbf{k}}$  is the pre-image under  $T$  of  $V_m^{(k'_1, \dots, k'_t)}$ . Since all prime factors of  $m$  are greater than  $s$ , by Proposition 18,  $V_m^{\mathbf{k}}$  is the preimage of  $T(0)$ . The computation of  $V_{p_i^{e_i}}^{\mathbf{k}}$  is analogous and uses Proposition 17. Moreover, since only integer division is required to specify the linear map  $T$ , the given characterisations can be computed in NC [7]. Finally, the orthogonal complements of the above-computed vanishing spaces can also be derived in NC [31].

By Corollary 16 we have that

$$(V_n^{\mathbf{k}})^\perp = (V_{p_1^{e_1}}^{\mathbf{k}})^\perp \odot \dots \odot (V_{p_\ell^{e_\ell}}^{\mathbf{k}})^\perp \odot (V_m^{\mathbf{k}})^\perp.$$

But, as observed above, such an iterated product can be computed in NC. Finally, with  $(V_n^{\mathbf{k}})^\perp$  in hand we can directly test whether  $f(\zeta_n) = 0$ .  $\square$

In [5], we use the above compositional technique to recover the result of Migotti [28] (see also Bang [6]) that all coefficients

Restricted Diagonal Circuits	
<b>Input:</b>	Polynomial $f = \sum_{i=1}^s g^{d_i}$ , with $g$ in sparse representation and $d_i$ in unary
<b>Output:</b>	Whether $f(\zeta_n) = 0$ for $n$ written in binary.
1:	Set $d := \max_{i=1}^s d_i$ .
2:	Compute the orbit $\text{Orb}(g(\zeta_n))$ of $g(\zeta_n)$ w.r.t. the set $\{k \in \mathbb{Z}_n^* : k \leq G(n)\}$ .
3:	If $ \text{Orb}(g(\zeta_n))  > d$ then return 'Non-Zero'.
4:	If $ \text{Orb}(g(\zeta_n))  \leq d$ then compute $\alpha \in \mathbb{Q}(i)$ such that $ \alpha - f(\zeta_n)  < \frac{\epsilon(f)}{3}$ and return 'Zero' if $\alpha < \frac{\epsilon(f)}{3}$ and return 'Non-Zero' otherwise.

**Figure 3: Algorithm for Restricted Diagonal Circuits**

of the  $n$ -th cyclotomic polynomial lie in  $\{-1, 0, 1\}$  in case  $n$  has at most two odd prime divisors.

## 6 DIAGONAL CIRCUITS

In this section we study CIT for the class of diagonal circuits [36]. In the multivariate formulation of CIT these compute polynomials of the form  $f = \sum_{i=1}^s g_i^{d_i}$ , where the  $g_i$  are linear forms and the  $d_i$  are integers in unary. The resulting CIT problem is a special case of Bounded-CIT and generalises Sparse-CIT. Note that in our univariate formulation the  $g_i$  become polynomials in sparse representation. We start with the following hardness result.

**THEOREM 19.** *If CIT for diagonal circuits is solvable in polynomial time then PIT for algebraic circuits of size  $s$  and degree  $d \leq s$  can be solved in  $s^{O(\sqrt{d})}$  time.*

Building on the proof of Theorem 19, we can show that efficient algorithms for another simple variant of sparse-CIT, namely evaluating low-degree sparse multivariate polynomials  $f(x_1, \dots, x_k)$  at translations of roots of unity, will yield a sub-exponential-time algorithm for PIT. More formally, if evaluating  $f(a_1 + \zeta_n^{e_1}, \dots, a_k + \zeta_n^{e_k})$  where  $f$  is a  $k$ -variate,  $k^{O(1)}$ -sparse polynomial and  $a_i \in \mathbb{Q}$ ,  $e_i \in \mathbb{N}$  are specified in binary is decidable in  $\text{poly}(k, \log n)$  time, then PIT for algebraic circuits of size  $s$  and degree  $d$  can be decided in  $s^{O(\sqrt{d})}$ -time.

We now specialise to consider the subclass of diagonal circuits that compute polynomials of the form  $f = \sum_{i=1}^s g^{d_i}$  with  $g$  a single univariate polynomial in sparse representation. We give an algorithm that solves the CIT for this class of circuits in polynomial time assuming GRH.

**THEOREM 4.** *Assuming GRH, CIT can be solved in polynomial time on the class of polynomials of the form  $f = \sum_{i=1}^m g^{d_i}$  with  $g$  in sparse representation and  $d_i$  an integer in unary.*

PROOF. The algorithm is given in Figure 3. It involves an integer parameter  $G(n)$  and a rational parameter  $\epsilon(f)$  that are both functions of the input. We will say more about both parameters shortly, suffice to say for now that  $G(n)$  is chosen such that  $\{k \in \mathbb{Z}_n^* : 1 \leq k \leq G(n)\}$  generates  $\mathbb{Z}_n^*$ .

We first argue correctness and then move to analysing the complexity. Line 2 refers to the action of the group  $\mathbb{Z}_n^*$  on the

field  $\mathbb{Q}(\zeta_n)$ , obtained by associating with  $k \in \mathbb{Z}_n^*$  the automorphism  $\sigma_k$  of  $\mathbb{Q}(\zeta_n)$  defined by  $\sigma_k(\zeta_n) = \zeta_n^k$ . Specifically, Line 2 computes the orbit  $\text{Orb}(g(\zeta_n))$  of  $g(\zeta_n)$  under the subgroup of  $\mathbb{Z}_n^*$  generated by the set  $\{k \in \mathbb{Z}_n^* : k \leq G(n)\}$ , that is, the smallest set that contains  $g(\zeta_n)$  and is closed under the action of the aforementioned subgroup.

Observe that when the algorithm halts in Line 3 the output is correct: if  $g(\zeta_n)$  has more than  $d$  distinct conjugates then it cannot be that  $f(\zeta_n) = \sum_{i=1}^s g(\zeta_n)^{d_i} = 0$ .

Now suppose that  $|\text{Orb}(g(\zeta_n))| \leq d$  in Line 3. We will use this assumption to bound the degree and height of  $f(\zeta_n)$ . By the assumption that  $\{k \in \mathbb{Z}_n^* : 1 \leq k \leq G(n)\}$  generates  $\mathbb{Z}_n^*$ , we have that  $\text{Orb}(g(\zeta_n))$  consists of all Galois conjugates of  $g(\zeta_n)$ . Since  $|\text{Orb}(g(\zeta_n))| \leq d$  it follows that  $g(\zeta_n)$ , and hence also  $f(\zeta_n)$ , have degree at most  $d$ . Furthermore, for every  $\ell \in \mathbb{Z}_n^*$  we have  $|f(\zeta_n^\ell)| \leq sM^d$ , where  $M$  is the sum of the absolute value of all coefficients of  $g$ . By writing the coefficients of the minimal polynomial of  $f(\zeta_n)$  in terms of the Galois conjugates of  $f(\zeta_n)$ , we have that  $f(\zeta_n)$  has height at most  $H := 2^d sM^d$ .

But a non-zero algebraic number of degree  $d$  and height  $H$  has magnitude at least  $\frac{2}{d^{d+1}H^d}$  [27]. Thus if we define

$$\varepsilon(f) := \frac{2}{d^{d+1}H^d}, \quad (4)$$

we have that if  $f(\zeta_n) \neq 0$  then  $|f(\zeta_n)| > \varepsilon(f)$ : hence for the number  $\alpha$  computed in Line 4 we have  $|\alpha| > \frac{2\varepsilon(f)}{3}$ . On the other hand, if  $f(\zeta_n) = 0$  then  $|\alpha| < \frac{\varepsilon(f)}{3}$ . Thus the output produced in Line 4 is correct. This completes the proof that the algorithm gives the correct output.

We turn now to the complexity. Note that we can use the procedure presented in the previous section to determine in polynomial time the equality of two conjugates  $g(\zeta_n^\ell)$  and  $g(\zeta_n^j)$  of  $g(\zeta_n)$ . Since the computation of  $\text{Orb}(g(\zeta_n))$  terminates as soon as  $|\text{Orb}(g(\zeta_n))| > d$ , we see that Line 2 can be executed in time polynomial in the size of the input and the parameter  $G(n)$ . Now it was shown in [30] that under GRH there is a function  $G(n) = O(\log^2 n)$  such that  $\{k \in \mathbb{Z}_n^* : 1 \leq k \leq G(n)\}$  generates  $\mathbb{Z}_n^*$ .<sup>1</sup> will yield a set of generators. It follows that Line 2 of the procedure can be executed in polynomial time, assuming GRH. Finally, from Expression (4) we see that  $|\log(\varepsilon(f))|$  is polynomially bounded in the input size. Thus  $f(\zeta_n)$  can be computed to within precision  $\frac{\varepsilon(f)}{3}$  in polynomial time, e.g., using the approach via the Taylor expansion as described in Section 4.  $\square$

## ACKNOWLEDGMENTS

Nikhil Balaji and James Worrell were supported by EPSRC grant EP/N008197/1.

## REFERENCES

- [1] M. Agrawal and S. Biswas. 2003. Primality and identity testing via Chinese remaindering. *J. ACM* 50, 4 (2003), 429–443.
- [2] E. Allender, P. Bürgisser, P. Kjeldgaard-Pedersen, and P. B. Miltersen. 2009. On the complexity of numerical analysis. *SIAM J. Comput.* 38, 5 (2009), 1987–2006.

- [3] E. Allender, J. Jiao, M. Mahajan, and V. Vinay. 1998. Non-commutative arithmetic circuits: depth reduction and size lower bounds. *TCS* 209, 1-2 (1998), 47–86.
- [4] E. Bach and L. Huelsbergen. 1993. Statistical Evidence for Small Generating Sets. *Math. Comp.* 61, 203 (1993), 69–82.
- [5] N. Balaji, S. Perifel, M. Shirmohammadi, and J. Worrell. 2020. Cyclotomic Identity Testing and Applications. *CoRR* abs/2007.13179 (2020).
- [6] AS Bang. 1895. Om Ligningen  $\Phi_n(x) = 0$ . *Nyt tidsskrift for matematik* 6 (1895), 6–12.
- [7] P. W. Beame, S. A. Cook, and H. J. Hoover. 1986. Log depth circuits for division and related problems. *SIAM J. Comput.* 15, 4 (1986), 994–1003.
- [8] J. Blömer. 1998. A probabilistic zero-test for expressions involving roots of rational numbers. In *ESA'98*. Springer, 151–162.
- [9] R. P. Brent. 1976. Fast multiple-precision evaluation of elementary functions. *J. ACM* 23, 2 (1976), 242–251.
- [10] Z.Z. Chen and M.Y. Kao. 2000. Reducing randomness via irrational numbers. *SIAM J. Comput.* 29, 4 (2000), 1247–1256.
- [11] Q. Cheng. 2007. Derandomization of sparse cyclotomic integer zero testing. In *FOCS'07*. IEEE, 74–80.
- [12] Q. Cheng, S. P. Tarasov, and M. N. Vyalyi. 2010. Efficient algorithms for sparse cyclotomic integer zero testing. *Theory of Computing Systems* 46, 1 (2010), 120–142.
- [13] A. L. Chistov and D. Grigoriev. 1984. Complexity of Quantifier Elimination in the Theory of Algebraically Closed Fields. In *MFCS (LNCS, Vol. 176)*. Springer, 17–31.
- [14] S. Cook. 1985. A taxonomy of problems with fast parallel algorithms. *Information and control* 64, 1-3 (1985), 2–22.
- [15] H. Davenport and H.L. Montgomery. 2013. *Multiplicative Number Theory*. Springer New York.
- [16] F. E. Fich and M. Tompa. 1988. The parallel complexity of exponentiating polynomials over finite fields. *J. ACM* 35, 3 (1988), 651–667.
- [17] G. Ge. 1993. Testing equalities of multiplicative representations in polynomial time. In *FOCS'93*. IEEE, 422–426.
- [18] Y. Hirshfeld, M. Jerrum, and F. Moller. 1994. A Polynomial-time Algorithm for Deciding Equivalence of Normed Context-free Processes. In *FOCS'94*. IEEE Computer Society, 623–631.
- [19] H. Iwaniec and E. Kowalski. 2004. *Analytic number theory*. Vol. 53. AMS.
- [20] A. Jež. 2012. Faster Fully Compressed Pattern Matching by Recompression. In *ICALP 2012, Part I (LNCS, Vol. 7391)*. Springer, 533–544.
- [21] E. Kaltofen and P. Koiran. 2006. Finding small degree factors of multivariate supersparse (lacunary) polynomials over algebraic number fields. In *ISSAC'06*. 162–168.
- [22] P. Koiran. 1996. Hilbert's Nullstellensatz is in the polynomial hierarchy. *J. complexity* 12, 4 (1996), 273–286.
- [23] Daniel König and Markus Lohrey. 2015. Parallel Identity Testing for Skew Circuits with Big Powers and Applications. In *MFCS'15*. Springer, 445–458.
- [24] R. E. Ladner and M. J. Fischer. 1980. Parallel Prefix Computation. *J. ACM* 27, 4 (1980), 831–838.
- [25] H. W. Lenstra Jr. 1999. Finding small degree factors of lacunary polynomials. *Number theory in progress* 1 (1999), 267–276.
- [26] K. Mehlhorn, R. Sundar, and C. Uhrig. 1997. Maintaining Dynamic Sequences under Equality Tests in Polylogarithmic Time. *Algorithmica* 17, 2 (1997), 183–198.
- [27] M. Mignotte. 1983. *Some Useful Bounds*. Springer, 259–263.
- [28] A. Migotti. 1883. Zur Theorie der Kreisteilungs-gleichung. *S.-B. der Math.-Naturwiss. Class der Kaiser. Akad. Der Wiss., Wien* 87 (1883), 7–14.
- [29] J. Mittmann. 2013. *Independence in algebraic complexity theory*. Ph.D. Dissertation. Universitäts- und Landesbibliothek Bonn.
- [30] H. L. Montgomery. 1971. *Multiplicative Number Theory*. 227 (1971).
- [31] K. Mulmuley. 1987. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. *Comb.* 7, 1 (1987), 101–104.
- [32] D. A. Plaisted. 1984. New NP-hard and NP-complete polynomial and integer divisibility problems. *TCS* 31, 1-2 (1984), 125–138.
- [33] W. Plandowski. 1994. Testing Equivalence of Morphisms on Context-Free Languages. In *ESA'94 (LNCS, Vol. 855)*. Springer, 460–470.
- [34] J. H. Reif and S. R. Tate. 1992. On threshold circuits and polynomial computation. *SIAM J. Comput.* 21, 5 (1992), 896–908.
- [35] J.M. Rojas. 2007. Efficiently detecting subtori and torsion points. 448 (2007), 213–233.
- [36] N. Saxena. 2008. Diagonal circuit identity testing and lower bounds. In *ICALP'08*. Springer, 60–71.
- [37] Arnold Schönhage. 1979. On the power of random access machines. In *ICALP'79*. Springer, 520–529.
- [38] L. G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff. 1983. Fast Parallel Computation of Polynomials Using Few Processors. *SIAM J. Comput.* 12, 4 (1983), 641–644.
- [39] M. Wetherfield. 1996. The Enhancement of Machin's Formula by Todd's Process. *The Mathematical Gazette* 80, 488 (1996), 333–344.

<sup>1</sup>The paper [4] gives heuristic arguments and experimental data suggesting that the choice  $G(n) = (\ln 2)^{-1} \ln n \ln \ln n$