

# Identity Testing for Circuits with Exponentiation Gates

Jiatu Li  

Massachusetts Institute of Technology, Cambridge, MA, USA

Mengdi Wu  

Carnegie Mellon University, Pittsburgh, PA, USA

---

## Abstract

Motivated by practical applications in the design of optimization compilers for neural networks, we initiated the study of identity testing problems for arithmetic circuits augmented with *exponentiation gates* that compute the real function  $x \mapsto e^x$ . These circuits compute real functions of form  $P(\vec{x})/P'(\vec{x})$ , where both  $P(\vec{x})$  and  $P'(\vec{x})$  are *exponential polynomials*

$$\sum_{i=1}^k f_i(\vec{x}) \cdot \exp\left(\frac{g_i(\vec{x})}{h_i(\vec{x})}\right),$$

for polynomials  $f_i(\vec{x})$ ,  $g_i(\vec{x})$ , and  $h_i(\vec{x})$ .

We formalize a black-box query model over finite fields for this class of circuits, which is mathematically simple and reflects constraints faced by real-world neural network compilers. We proved that a simple and efficient randomized identity testing algorithm achieves perfect completeness and non-trivial soundness. Concurrent with our work, the algorithm has been implemented in the optimization compiler Mirage by Wu et al. (OSDI 2025), demonstrating promising empirical performance in both efficiency and soundness error. Finally, we propose a number-theoretic conjecture under which our algorithm is sound with high probability.

**2012 ACM Subject Classification** Theory of computation → Circuit complexity

**Keywords and phrases** Polynomial Identity Testing, Exponential Polynomials

**Digital Object Identifier** 10.4230/LIPIcs.ITCS.2026.95

**Related Version** *Full Version*: <https://arxiv.org/abs/2506.04529>

**Funding** *Jiatu Li*: received support from the National Science Foundation under Grant CCF-2127597. *Mengdi Wu*: received support from NSF awards CNS-2147909, CNS-2211882, and CNS-2239351 and research awards from Amazon, Cisco, Google, Meta, NVIDIA, Oracle, Qualcomm, and Samsung.

**Acknowledgements** We are grateful to Seyoon Ragavan for mentioning the connection of our results to [11] and related works. We also thank Ryan O'Donnell and Ryan Williams for discussions about the identity testing algorithm and its analysis, and Zhihao Jia and Oded Padon for discussions about modeling the real-world problem.

## 1 Introduction

Polynomial Identity Testing (PIT) is a central problem of theoretical computer science. Given a multi-variate polynomial  $f : \mathbb{F}^n \rightarrow \mathbb{F}$  with certain properties, the goal of the problem is to verify whether  $f$  is identically zero.

Early results on identity testing of polynomials date back to DeMillo and Lipton [8], Zippel [26], and Schwartz [19] (see also [14] for a special case on finite fields). It is proved that a black-box probabilistic testing, namely checking whether  $f(\vec{x}) = 0$  for a random element  $\vec{x} \in \mathbb{F}^n$ , works well for low-degree polynomials. This simple algorithm serves as a key step in many randomized algorithms (see, e.g., [22, 12, 1]) as well as probabilistic proof



© Jiatu Li and Mengdi Wu;

licensed under Creative Commons License CC-BY 4.0

17th Innovations in Theoretical Computer Science Conference (ITCS 2026).

Editor: Shubhangi Saraf; Article No. 95; pp. 95:1–95:22

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

systems, including the celebrated interactive proof system for PSPACE [20] and the PCP theorem [3]. Subsequently, there has been a rich literature on identity testing of various types of polynomials, including sparse polynomials and polynomials computed by depth-3 algebraic circuits (see, e.g., [17, 18] and the references therein).

Perhaps surprisingly, the probabilistic identity testing algorithm for low-degree polynomials has recently been found useful in developing optimization compilers for deep neural networks. Loosely speaking, deep neural networks are represented as a high-level programming language, and the optimization compiler is designed to automatically identify redundancy in programs to improve the efficiency of the neural network. In a recent optimization compiler PET [24], Wang et al. observed that redundancy detection in partial neural network computation *without* non-linear activation can be naturally modeled as black-box identity testing of low-degree polynomials. By applying the classical probabilistic testing algorithm [8, 26, 19], PET achieves a significant speed-up over its benchmark.

► **Remark 1 (black-box model).** In the black-box setting, the algorithm is given oracle access to the polynomial rather than its description. For the optimization compilers of neural networks (e.g., [24]), the neural networks are gigantic and are usually loaded into optimization devices such as GPUs or TPUs. Black-box identity testing algorithms are favorable, as *evaluation* on (partial) neural networks exploits the optimization devices and is therefore efficient in practice. In contrast, algorithms analyzing the circuit in white-box, even when they have lower computational complexity in theory, may still be less efficient if they cannot make full use of optimization devices.<sup>1</sup>

A natural question is, therefore, whether this approach can be extended to neural networks *with* non-linear activation. In a subsequent work, Wu et al. [25] introduced a framework, Mirage, that could benefit from extending the redundancy detection algorithm in PET [24] to neural network components with *exponentiation* operators. This enables the modeling of non-linear activation functions such as  $\text{softmax}(\cdot)$ . In the language of circuit complexity, the program representation in Mirage can be modeled as integer-coefficient arithmetic circuits with “exponentiation gates”, where there is at most one exponentiation gate in each path from output gates to input gates (see Section 3.1). The goal of Mirage is to “efficiently” check whether a circuit  $C : \mathbb{R}^n \rightarrow \mathbb{R}$  of such form is identical to 0 on  $\mathbb{R}$  in a black-box query model that is practically satisfactory.

Compared to polynomials and standard arithmetic circuits computing polynomials, the behavior of circuits with exponentiation gates is not well studied. Integer-coefficient polynomials can be naturally evaluated over  $\mathbb{F}_p$  with modular arithmetic. However, it is not a-prior clear what is the correct way to define the evaluation of circuits with exponentiation gates over finite fields, as there is no standard analogy of the exponentiation function in finite fields. This makes it hard to even define a suitable “black-box query model” for the identity testing problem that is easy to analyze and captures the real-world constraints. Moreover, it is unclear whether the standard randomized algorithm [8, 19, 26] works after we define the evaluation of such circuits.

**Our Contribution.** Motivated by the practical applications, we initiate a theoretical analysis of the identity testing of *circuit with exponentiation gates*.

- We introduce a natural circuit model that formalizes circuits with exponentiation gates  $x \mapsto \exp(x)$ , which suffices to capture components of modern neural networks such as Attention [23] with certain non-linear activation functions. The circuit model generalizes the standard algebraic circuit model by allowing exponentiation gates.

---

<sup>1</sup> Moreover, black-box algorithms are much easier to implement, as one can reuse the existing software infrastructure for evaluating neural networks, making them more reliable while reducing human efforts.

- We prove that in the *idealized model* where the algorithm could query the circuit on any *real number*, a simple probabilistic testing is correct with high probability.
- We introduce an *algebraic query model* that captures the real-world constraints in optimization compilers for deep neural networks such as Mirage [25], and design a simple randomized algorithm that is perfectly complete and *non-trivially* sound. Concurrent with our work, this algorithm is implemented in Mirage as one of its two redundancy detection algorithms. We also introduce a conjecture about sparse polynomials in finite field, under which our randomized algorithm in algebraic query model is sound *with high probability*.

### 1.1 Main Results: A Simplified Setting

Before formally describing our circuit and query models in Section 3, we try to describe a simplified version of the main mathematical problem. Let  $k \in \mathbb{N}$ ,  $f_i, g_i, h_i$  be  $n$ -variate degree- $d$  polynomials with integer coefficients in  $[-w, w]$ , and  $P : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\perp\}$  be a partial real function defined as

$$P(\vec{x}) \triangleq \sum_{i=1}^k f_i(\vec{x}) \exp\left(\frac{g_i(\vec{x})}{h_i(\vec{x})}\right),$$

where  $\exp(x) \triangleq e^x$ . This function is similar to the *exponential polynomials* that are used in, e.g., transcendental number theory [4, Chapter 12]. For simplicity, we will call  $P(\vec{x})$  a degree- $d$  exponential polynomial. The number of terms  $k$  in  $P(\vec{x})$  is said to be the *width* of  $P(\vec{x})$ . We call  $\{f_i(\vec{x})\}_{i \in [k]}$  the coefficient polynomials of  $P(\vec{x})$ , and  $\{g_i(\vec{x})/h_i(\vec{x})\}_{i \in [k]}$  the exponent fractions of it.

The algorithmic problem we are interested in is to check whether  $P(\vec{x})$  is identically zero on  $\mathbb{R}^n$  with queries of one of the following two forms:

- (*Real*): Given  $\vec{x} \in \mathbb{Q}^n$ , the oracle outputs  $\{0, 1, \perp\}$  indicating that  $P(\vec{x}) = 0$ ,  $P(\vec{x}) \in \mathbb{R} \setminus \{0\}$ , and  $P(\vec{x}) = \perp$  (i.e. undefined due to division-by-zero), respectively.
- (*Algebraic*): Let  $p, q$  be prime numbers such that  $q \mid p - 1$ ,  $G = \langle g \rangle$  be the unique multiplicative subgroup of  $\mathbb{F}_p^*$  of order  $q$ . Given  $\vec{u} \in \mathbb{F}_p^n, \vec{v} \in \mathbb{F}_q^n$ , and  $a \in G$ , the oracle outputs

$$P_a(\vec{u}, \vec{v}) \triangleq \left( \sum_{i=1}^k f_i(\vec{u}) \cdot a^{g_i(\vec{v}) \cdot (h_i(\vec{v}))^{-1} \bmod q} \right) \bmod p,$$

where  $(\cdot)^{-1}$  on the exponent denotes the multiplicative inverse in  $\mathbb{F}_q$ . The oracle returns  $\perp$  if  $h_i(\vec{v}) \bmod q = 0$  for any  $i \in [k]$ .

Before introducing our results, we make a few short remarks on our modeling of the identity testing problem and its connection to real world engineering practice.

► **Remark 2 (numerical stability issues).** Testing whether  $P(\vec{x})$  (modeling a program) is identically zero on  $\mathbb{R}^n$  is motivated by the following compilation task – compilers may replace a (sub)program by another optimized one provided that they compute the same function over real numbers. As real numbers are only an idealized model for, say, floating point numbers, this optimization strategy may cause numerical stability issues. Nevertheless, most modern classical and neural network compilers offer the option for users to enable such optimizations as it may significantly improve performance, e.g., the `-Ofast` option of GCC [21, Chapter 3] and the FlashAttention [7] implementation in PyTorch [15, 16]; see also [10, 24, 25].

► **Remark 3** (query models). The former type of query is an idealized model. In practical applications (e.g. [24, 25]), there is no efficient algorithm to implement the oracle that checks whether  $P(\vec{x}) = 0$  *precisely* given  $\vec{x}$ , and the approximation of real numbers using floating-point numbers is also unsatisfactory in practice. The latter type of query is more realistic; indeed, it is implemented in the optimization compiler Mirage [25] where the explicit representation of  $P(\vec{x})$  is a (partial) neural network.<sup>2</sup> See Section 1.2 and 3 for more detailed discussions.

**Identity Testing Algorithms in the Simplified Setting.** Now we describe our identity testing algorithms in the simplified setting. First, we show that a simple randomized algorithm with *one* query works in the real query model. The algorithm is to randomly sample  $\vec{x} \in [B]^n$  for  $B = 20 \cdot d \cdot k^2$  and accepts if  $P(\vec{x}) \in \{0, \perp\}$ . It is clear that the algorithm is perfectly complete, and the soundness is given by the following theorem:

► **Theorem 4** (Theorem 22, simplified). *Let  $P(\vec{x})$  be an  $n$ -variate degree- $d$  exponential polynomial of width  $k$  that is not identically zero on its domain, and  $S \subseteq \mathbb{Q}$  be any finite non-empty set. For  $\vec{x} \in S^n$  sampled uniformly at random,  $\Pr[P(\vec{x}) \in \{0, \perp\}] \leq 8dk^2/|S|$ .*

As the second result, we analyze the following simple randomized identity testing algorithm in the algebraic query model: Let  $p, q, G$  be defined as above, the algorithm randomly samples  $\vec{u} \in \mathbb{F}_p^n, \vec{v} \in \mathbb{F}_q^n$ , and  $a \in G$ , and accepts if  $P_a(\vec{u}, \vec{v}) \in \{0, \perp\}$ . The following theorem formalizes the completeness and soundness of the algorithm:

► **Theorem 5** (Theorem 7, simplified). *Let  $P(\vec{x})$  be an  $n$ -variate degree- $d$  exponential polynomial of width  $k$ . Let  $p, q$  be prime numbers,  $q \mid p - 1$ , and  $G = \langle g \rangle$  be the unique order- $q$  multiplicative subgroup of  $\mathbb{F}_p^*$ . Suppose that all integer coefficients in  $P(\vec{x})$  are within  $[-w, w]$ ,  $q > 2w$ , then:*

- (Completeness). *If  $P(\vec{x})$  is identically zero on its domain as a partial real function, for any  $a \in G, \vec{u} \in \mathbb{F}_p^n$ , and  $\vec{v} \in \mathbb{F}_q^n$ ,  $P_a(\vec{u}, \vec{v}) \in \{0, \perp\}$ .*
- (Soundness). *If  $P(\vec{x})$  is not identically zero on its domain as a partial real function,  $p, q \geq 2kw$ , then for uniformly random  $a \in G, \vec{u} \in \mathbb{F}_p^n$ , and  $\vec{v} \in \mathbb{F}_q^n$ , the probability that  $P_a(\vec{u}, \vec{v}) \in \{0, \perp\}$  is at most  $q^{-\frac{1}{k-1}} + O(dk^2/q)$ .*

Note that for sufficiently large  $k, q \in \mathbb{N}$  such that  $\ln q \leq k - 1$ , we have

$$1 - \frac{2 \ln q}{k - 1} \leq q^{-\frac{1}{k-1}} \leq 1 - \frac{\ln q}{2(k - 1)}.$$

Thus in a typical setting that  $dk^3 \leq q \leq k^{O(1)}$ , the soundness error is  $1 - \Theta(\log k/(k - 1))$ .<sup>3</sup> By parallel repetition of the randomized algorithm for  $O(k \log k / \log q)$  times, we can boost the error probability to  $1/k^{O(1)}$ . This leads to an efficient randomized identity testing algorithm when  $k$  is relatively small and the evaluation is much more efficient than obtaining the description of  $P(\vec{x})$ , which, in particular, captures the real-world constraints for the optimization compilers such as Mirage [25].

<sup>2</sup> The special case of the algebraic query model without exponentiation functions is implemented in [24].

<sup>3</sup> The error probability is constant if  $q$  is exponential in  $k$ . However, this setting is not meaningful: The width  $k$  is usually comparable to the input length, and thus the arithmetic operations over  $q$  would be extremely inefficient.

## 1.2 Connection to the Circuit Models

We stress that the abstraction in previous subsection is mathematically clean but omits crucial details in modeling the real-world problem. Specifically:

- It is not immediately clear how the identity testing results for exponential polynomials apply to our motivating real-world application, namely identity testing of neural networks formalized by “circuits with exponentiation gates” [24, 25].
- It is also not clear how the function  $P_a(\vec{u}, \vec{v})$  in Theorem 7 relates to the circuits, and why it can be computed efficiently when  $P(\vec{x})$  is explicitly given by a “circuit with exponentiation gates”. This is important as in real-world applications, see [25], the function  $P_a(\vec{u}, \vec{v})$  is implemented on specialized devices such as GPU or TPU, which are optimized for specific computation patterns. Moreover, it is unclear where the restrictions in the algebraic query model (e.g.,  $a$  must be in  $G$ ) come from.

**Circuit Model.** We briefly explain our circuit model (called  $\text{AExp}^1$  circuits) and refer readers to Sections 3 and 4 for more details. We work with (arithmetic) circuits with integer coefficients, unbounded fan-in addition and multiplication gates, fan-in two division gates, and fan-in one exponentiation gates. In addition, we impose the restriction that on each path from input variables to output gates, there is at most *one* exponentiation gates – the circuit cannot compute double exponential  $x \mapsto e^{e^x}$  by composing exponentiation gates.

The evaluation of circuits over real numbers is defined by a standard gate-by-gate evaluation algorithm, where the exponentiation gate is interpreted as the function  $x \mapsto e^x$ . However, it is unclear how to define the evaluation of such circuits over finite fields, as there is no standard interpretation of the exponential function.

Let  $p$  be a prime number. To define the evaluation of an  $\text{AExp}^1$  circuit  $C$  over  $\mathbb{F}_p$ , a natural idea is to replace the exponential function by  $x \mapsto a^x$  for some element  $a \in \mathbb{F}_p$ . This is not ideal as such an interpretation is *inconsistent* with the evaluation over  $\mathbb{R}$ : It could be the case that

$$a^{x_1+x_2 \bmod p} \not\equiv a^{x_1} a^{x_2} \pmod{p}$$

for  $x_1, x_2 \in \mathbb{F}_p$ , while  $e^{x_1+x_2} = e^{x_1} e^{x_2}$  for any  $x_1, x_2 \in \mathbb{R}$ . As a result, this definition cannot be used for the identity testing of  $\text{AExp}^1$  circuits.

To address the issue, we exploit the algebraic structure of finite fields by using different moduli over and under the exponents. Let  $p, q$  be prime numbers such that  $q \mid p-1$ ,  $G \subseteq \mathbb{F}_p^*$  be the unique order- $q$  multiplicative subgroup, and  $a \in G$ . It follows that

$$a^{x_1+x_2 \bmod q} \equiv a^{x_1} \cdot a^{x_2} \pmod{p}$$

for any  $x_1, x_2 \in \mathbb{F}_q$ . If we use  $q$  as the modulus “over the exponent” and  $p$  as the modulus “under the exponent”, the function  $x \mapsto a^x$  will be consistent with the arithmetic law  $e^{x_1+x_2} = e^{x_1} \cdot e^{x_2}$ . Following the intuition, we define the evaluation of  $C$  with respect to  $(p, q, a)$  using a gate-by-gate evaluation algorithm such that:

- Each input variable or wire carries a pair  $(u, v) \in (\mathbb{F}_p \cup \{\perp\}) \times (\mathbb{F}_q \cup \{\perp\})$ .
- Addition, multiplication, and division gates are implemented coordinate-wisely.
- The exponentiation gate is implemented by  $(u, v) \mapsto (a^v \bmod p, \perp)$ .

This evaluation algorithm can be implemented efficiently given the description of the circuit  $C$ . We refer readers to Section 3 for more details.

**A Structural Lemma.** Let  $\vec{u} = (u_1, \dots, u_n) \in \mathbb{F}_p^n$ ,  $\vec{v} = (v_1, \dots, v_n) \in \mathbb{F}_q^n$ , we use  $C_a(\vec{u}, \vec{v})$  to denote the output of the evaluation algorithm where the  $i$ -th input variable is assigned to  $(u_i, v_i)$ . Similar to the standard results that algebraic circuits compute polynomials [2, Section 16], we can then prove that  $\text{AExp}^1$  circuits can be converted to an equivalent *fraction of exponential polynomials*:

► **Lemma 6** (Lemma 20, informal). *For every  $n$ -input  $\text{AExp}^1$  circuit  $C$ , there are  $n$ -variate integer-coefficient exponential polynomials  $P(\vec{x})$  and  $P'(\vec{x})$  such that the following holds:*

- *For each  $\vec{u} \in \mathbb{R}$ ,  $C(\vec{u}) = P(\vec{u})/P'(\vec{u})$ .*
- *Let  $p, q$  be prime numbers such that  $q \mid p - 1$ ,  $G_{p,q}$  be the multiplicative subgroup of  $\mathbb{F}_p^*$  of order  $q$ , and  $a \in G_{p,q}$ . Then for every  $(\vec{u}, \vec{v}) \in \mathbb{F}_p^n \times \mathbb{F}_q^n$ ,  $C_a(\vec{u}, \vec{v}) \equiv P_a(\vec{u}, \vec{v}) \cdot (P'_a(\vec{u}, \vec{v}))^{-1} \pmod{p}$ , where  $(a)^{-1}$  denotes the multiplicative inverse of  $a$  modulo  $p$ .*

We stress that the structural lemma *does not* provide a polynomial upper bound on the degree, width, or integer weight of the integer-coefficient exponential polynomials  $P, P'$ . Nevertheless, it can be verified that the exponential polynomials  $P, P'$  for  $\text{AExp}^1$  circuits from real-world neural network applications, such as  $\text{softmax}(\cdot)$  and Attention with softmax activations [23], tend to have relatively small degree, width, and weight; interested readers are referred to Section 4.5 for more discussion, and [25] for experimental results.

**The Main Theorem.** Let  $\text{dom}_{\mathbb{R}}(C)$  be the domain of the circuit  $C$ , i.e., the set of  $\vec{x} \in \mathbb{R}^n$  such that  $C(\vec{x}) \neq \perp$ . An  $\text{AExp}^1$  circuit  $C$  is said to have degree  $d$ , width  $k$ , and weight  $w$  if there are integer-coefficient degree- $d$  width- $k$  exponential polynomials  $P(\vec{x})$  and  $P'(\vec{x})$  with all integer coefficients within  $[-w, w]$  that satisfy Lemma 6.

By generalizing Theorems 4 and 5 to fractions of exponential polynomials and combining them with Lemma 6, we can obtain the final result:

► **Theorem 7.** *Let  $C : \mathbb{R}^n \rightarrow \mathbb{R}$  be an  $\text{AExp}^1$  circuit of width  $k$ , degree  $d$  and weight  $w$ ,  $p$  and  $q$  be prime numbers such that  $q \mid p - 1$  and  $q > 2(kw)^2$ . Let  $G_{p,q}$  be the unique multiplicative subgroup of  $\mathbb{F}_p^*$  of order  $q$ . The following hold:*

- *(Completeness). If  $C$  is identically zero on  $\text{dom}_{\mathbb{R}}(C)$ , then for every  $a \in G_{p,q}$ ,  $C_a(\vec{u}, \vec{v}) \in \{0, \perp\}$  for every  $(\vec{u}, \vec{v}) \in \mathbb{F}_p^n \times \mathbb{F}_q^n$ .*
- *(Soundness). If  $C$  is not identically zero on  $\text{dom}_{\mathbb{R}}(C)$ , then for uniformly random  $\vec{u} \leftarrow \mathbb{F}_p^n, \vec{v} \leftarrow \mathbb{F}_q^n, a \leftarrow G_{p,q}$ ,  $\Pr[C_a(\vec{u}, \vec{v}) \notin \{0, \perp\}] \geq 1 - 8dk^4 \cdot q^{-1} - q^{-1/(k^2-1)}$ .*

### 1.3 Technical Overview

We briefly explain the proof of our simplified technical results Theorem 4 and Theorem 5 in the simplified setting, as well as the main theorem (see Theorem 7) that generalizes the results to the circuit setting.

**Real Queries.** The idea behind Theorem 4 is very simple: We manage to combine the Schwartz-Zippel lemma and Lindemann-Weierstrass theorem, i.e.,  $e$  is *transcendental* (see Theorem 13).

To avoid technical subtlety, we assume that  $P(\vec{x})$  is a degree- $d$  exponential polynomial where the exponent fractions  $g_i(\vec{x})/h_i(\vec{x})$  are pairwise distinct; that is, for every pair of indices  $i, j \in [k]$ ,  $i \neq j$ ,  $g_i(\vec{x})h_j(\vec{x}) - g_j(\vec{x})h_i(\vec{x})$  is not a zero polynomial. Such an exponential polynomial is said to be *condensed*, and the general case can be reduced to the condensed case by considering another exponential polynomial  $\hat{P}$  that “groups” coefficients based on the exponent fraction; see Section 4.3. Moreover, we assume without loss of generality that  $f_i \neq 0$  and  $h_i \neq 0$  for each  $i \in [k]$ .

Let  $S \subseteq \mathbb{Q}$  be a non-empty finite set, we know by the Schwartz-Zippel lemma that for any  $i \in [k]$ ,  $j \in [k]$ ,  $i \neq j$ , each of the following non-zero polynomials of degree at least  $2d$

$$f_i(\vec{x}), \quad h_i(\vec{x}), \quad g_i(\vec{x})h_j(\vec{x}) - g_j(\vec{x})h_i(\vec{x})$$

evaluates to zero with probability at most  $2d/|S|$  for a uniformly random  $\vec{x} \in S^n$ . By the union bound, we can further prove that with probability at least  $1 - 3dk^2/|S|$ , none of the polynomials evaluate to 0. For each of such  $\vec{x}$ , we can see that for the rational numbers  $\alpha_i \triangleq g_i(\vec{x})/h_i(\vec{x})$  and  $\beta_i \triangleq f_i(\vec{x})$ , we have

$$P(\vec{x}) = \beta_1 e^{\alpha_1} + \beta_2 e^{\alpha_2} + \dots + \beta_k e^{\alpha_k},$$

which must be non-zero as  $e$  is transcendental.

**Connection to Algebraic Queries.** Unwinding the proof of Theorem 4, we could actually give a necessary and sufficient condition for an exponential polynomial with integer weights to be identically zero over real evaluations. Let  $P(\vec{x})$  be a condensed degree- $d$  exponential polynomial

$$P(\vec{x}) = \sum_{i=1}^k f_i(\vec{x}) \exp\left(\frac{g_i(\vec{x})}{h_i(\vec{x})}\right)$$

with integer coefficients, then  $P(\vec{x})$  is identically zero on its domain (over real evaluation) if and only if for every  $i \in [k]$ ,  $f_i = 0$ . Subsequently, if  $p, q$  are chosen to be larger than the integer coefficients, the identity testing of a condensed exponential function  $P(\vec{x})$  is equivalent to testing whether  $f_i = 0 \pmod{p}$  for every  $i \in [k]$ .

With the characterization above, the completeness of Theorem 5 is relatively simple, so we will focus on the soundness property.

We try to mimic the strategy in the proof of Theorem 4. Suppose that  $P(\vec{x})$  is not identically zero on real evaluations, we know by the discussion above that when  $p$  are sufficiently large, there is at least one  $f_i \neq 0 \pmod{p}$  for any  $i \in [k]$ . Moreover, if  $q$  is sufficiently large, we know that for any  $i \neq j$ ,  $g_i h_j - g_j h_i \neq 0 \pmod{q}$ . Assume without loss of generality that  $f_i \neq 0$  for all  $i \in [k]$ . For  $\vec{u} \in \mathbb{F}_p^n$  and  $\vec{v} \in \mathbb{F}_q^n$  sampled uniformly at random, we still know by the Schwartz-Zippel lemma and the union bound that with probability at least  $1 - 3dk^2/q$ , none of

$$f_i(\vec{x}) \pmod{p}, \quad h_i(\vec{x}) \pmod{q}, \quad g_i(\vec{x})h_j(\vec{x}) - g_j(\vec{x})h_i(\vec{x}) \pmod{q}$$

evaluates to zero for  $i, j \in [k]$ ,  $i \neq j$ . We can then define  $\alpha_i \triangleq g_i(\vec{x})/h_i(\vec{x}) \pmod{q}$  and  $\beta_i \triangleq f_i(\vec{x}) \pmod{p}$  for  $i \in [k]$  such that

$$P_a(\vec{u}, \vec{v}) = \beta_1 a^{\alpha_1} + \dots + \beta_k a^{\alpha_k} \pmod{p},$$

where  $\beta_1, \dots, \beta_k$  are non-zero and  $\alpha_1, \dots, \alpha_k$  are pairwise distinct.

**A Weak Descartes' Rule in Finite Fields.** It then suffices to prove that for any non-zero  $\beta_1, \dots, \beta_k \in \mathbb{F}_p$  and distinct  $\alpha_1, \dots, \alpha_k \in \mathbb{F}_q$ ,  $k \geq 1$ , if we sample  $a \in G$  uniformly at random, the probability that  $\beta_1 a^{\alpha_1} + \dots + \beta_k a^{\alpha_k} \pmod{p} = 0$  is at most  $q^{-1/(k-1)}$ . We note that this can be viewed as a generalization of the *Descartes' Rule* (see also [11]), which asserts that any univariate polynomial with  $k$  monomials has at most  $2k$  roots.

We provide an elementary proof of a weaker result: The probability that  $\beta_1 a^{\alpha_1} + \dots + \beta_k a^{\alpha_k} \pmod p = 0$  is at most  $1 - 1/k$ . Theorem 5 is proved using a lemma from [11], which employs a similar but slightly more complicated argument.

Recall that the order- $q$  multiplicative subgroup  $G \subseteq \mathbb{F}_p^*$  is a cyclic group. For any fixed  $i \in \mathbb{F}_q$ , a random element  $a \in G$  can be viewed as generated from randomly sampling  $g \in G$  and outputting  $g^i$ . Our idea is to choose a good  $i \in \mathbb{F}_q$  such that at least a  $1/k$  fraction of  $g$  satisfies that  $\beta_1 g^{i \cdot \alpha_1} + \dots + \beta_k g^{i \cdot \alpha_k} \pmod p \neq 0$ .

We say that an index  $i \in \mathbb{F}_q$  is *good* for  $\alpha \in \mathbb{F}_q$  if  $i \cdot \alpha \pmod q \leq (1 - 1/k) \cdot q$ , and is said to be *good* if for every  $j \in [k]$ ,  $i$  is good for  $\alpha_j$ . It turns out that if there is a good  $i \in \mathbb{F}_q$ ,

$$\beta_1 g^{i \cdot \alpha_1} + \dots + \beta_k g^{i \cdot \alpha_k}$$

can be viewed as a polynomial in  $\mathbb{F}_p[g]$  of degree at most  $(1 - 1/k) \cdot q$ ; in that case, there are at most  $(1 - 1/k) \cdot q$  roots in  $\mathbb{F}_p$  (and also in  $G$ ).

Therefore, it remains to prove the existence of a good  $i$ . Since  $i \mapsto i \cdot \alpha$  is a bijection in  $\mathbb{F}_q$ , we know that for any fixed  $\alpha$ , the probability that a random  $i \in \mathbb{F}_q$  is good for  $\alpha$  is at least  $1 - 1/k$ . By the union bound, we know that a random  $i \in \mathbb{F}_q$  is good for  $\alpha_1, \dots, \alpha_k$  with non-zero probability. This implies that there must be a good  $i \in \mathbb{F}_q$ , which completes the proof.

► **Remark 8 (related results).** The analogy of Descartes’ Rule over finite fields has been studied prior to our work. Motivated by understanding the security of the Diffie-Hellman cryptosystem, Canetti et al. [5] proved an upper bound on the number of roots of sparse univariate polynomials. This upper bound was later improved by Kelley [11]. We use the techniques from [5, 11] and obtain similar upper bounds. Note that [5, 11] considers the number of roots in  $\mathbb{F}_p$ , while we consider the number of roots in a multiplicative subgroup  $G \subseteq \mathbb{F}_p^*$ ; as a result, our upper bound is cleaner and easier to prove.

**Generalization to the Circuit Setting.** There are a few technical issues to obtain the main theorem (see Theorem 7).

First, the proof overview above assumes that the exponential polynomial is *condensed*, i.e., its exponent fractions are pairwise distinct. For exponential polynomials that are not condensed, we need to first *condense* the polynomial by merging terms with equivalent exponent fractions. For instance, the following exponential polynomial  $P(\vec{x}) = \exp\left(\frac{x^2-3x}{x-3}\right) + \exp\left(\frac{x^2-2x}{x-2}\right)$  may be condensed to  $\hat{P}(\vec{x}) = 2 \exp\left(\frac{x^2-3x}{x-3}\right)$ . In general, such condensation procedure results into another exponential polynomial  $\hat{P}$  that has larger domain and agree with  $P$  on the domain of  $P$  (see Proposition 19). We need to bridge the gap between  $P$  and  $\hat{P}$  with standard probability analysis.

Second, as shown in Section 1.2,  $\text{AExp}^1$  circuits are converted to fractions  $P/P'$  of exponential polynomials rather than exponential polynomials. This requires a careful (but straightforward) adaption of the techniques above. In particular, in the soundness analysis, we use the observation that  $P/P'$  is identically zero on its domain (over  $\mathbb{R}$ ) if and only if the exponential polynomial  $P \cdot P'$  is identically zero on its domain (over  $\mathbb{R}$ ). This leads to a quadratic overhead (in  $k$ ) in the soundness error of Theorem 7 compared to Theorem 5, as  $P \cdot P'$  may have width up to  $k^2$  when both  $P$  and  $P'$  are of width  $k$ .

**Organization of the Paper.** We review basic definitions and classical results that we will need in Section 2. In Section 3, we formally describe our circuit model as well as the query models we considered in our paper – the idealized real query model and the algebraic query

model; we also briefly explain why the algebraic query model is a better abstraction in the application of optimization compilers for neural networks. In Section 4, we define exponential polynomials, discuss its basic properties, and prove the structural lemma that converts circuits to fractions of exponential polynomials. In Section 5, we prove the correctness of our probabilistic testing algorithm in the real and algebraic query models.

## 1.4 Discussion and Open Problems

The most interesting open problem is to improve the soundness error of Theorem 5 and Theorem 7. We conjecture that the actual soundness error is  $1 - \Omega(1)$ . In particular, we propose the following number-theoretic conjecture under which the soundness error is indeed  $1 - \Omega(1)$ :

► **Conjecture 9** (Strong Descartes' Rule Conjecture over Finite Fields). *There are constants  $\varepsilon, \delta < 1$  such that the following holds. Let  $p, q$  be sufficiently large prime numbers such that  $q \mid p - 1$ ,  $g \in \mathbb{F}_q$  be an element of order  $q$ , and  $G \triangleq \langle g \rangle$  be the unique multiplicative subgroup of  $\mathbb{F}_p^*$  of order  $q$ . For any  $k \leq q^\delta$ , distinct  $\alpha_1, \dots, \alpha_k \in \mathbb{F}_q$ , and  $\beta_1, \dots, \beta_k \in \mathbb{F}_p \setminus \{0\}$ , the non-zero univariate polynomial*

$$f(z) \triangleq \beta_1 z^{\alpha_1} + \beta_2 z^{\alpha_2} + \dots + \beta_k z^{\alpha_k}$$

*has at most  $\varepsilon \cdot q$  roots in  $G$ .*

This conjecture can be interpreted as a property of the isomorphism mapping

$$I : [f] \mapsto (f(1), f(g), \dots, f(g^{q-1}))$$

from  $\mathbb{F}_p[x]/(x^q - 1)$  onto  $\mathbb{F}_p^q$ . It states that there is no non-zero polynomial  $[f] \in \mathbb{F}_p[x]/(x^q - 1)$  such that both  $f$  and  $I([f])$  are very sparse. This seems to be an analogy of the ‘‘uncertainty principle’’ in Fourier analysis (see, e.g., [13, Exercise 3.15]). We also note that numerical experiments (see, e.g., [11, 6]) suggest that it is hard to construct sparse polynomials over  $\mathbb{F}_p$  with many roots for prime  $p$ .

**Complexity-theoretic Perspectives.** It is also interesting to consider whether we can design better identity testing algorithms for some restricted classes of  $\text{AExp}^1$  circuits (e.g. of small constant depth). This would potentially lead to real-world applications, as the structure of circuits from neural networks is relatively simple. To start with, one may consider adapting existing techniques for identity testing of algebraic circuits (see, e.g., [17, 18]) to  $\text{AExp}^1$  circuits.

On the other hand, it is interesting to consider whether there are conditional or unconditional *lower bounds* for identity testing of  $\text{AExp}^1$  circuits in black-box models, such as the algebraic query models that we introduced in Section 1.2.

**Other Modeling of the Problem.** We note the the algebraic query model in Section 1.2 is not necessarily the only reasonable formalization of the real world problem. Recall that in optimization compilers for neural networks [24, 25], the neural network (modeled by an  $\text{AExp}^1$  circuit) is loaded into specialized devices (such as GPU or TPU) that are optimized for specific computation patterns and have high communication overhead with the CPU. Algorithms in other black-box models are potentially useful for real-world applications if:

1. the queries can be efficiently implemented on those specialized devices; and
2. the communication overhead is small.

We are not aware of other natural models that satisfies these constraints for identity testing of  $\text{AExp}^1$  circuits, and it remains an interesting open problem that may require joint efforts of system and theory communities. Note that it may make it possible to design better identity testing algorithms if we work with another black-box query model that has better mathematical properties.

## 2 Preliminaries

### 2.1 Abstract Algebra

We assume basic familiarity to elementary ring and field theory (see, e.g., [9]). We will use the standard notation:  $R[x_1, \dots, x_m]$  denotes the ring of  $m$ -variate polynomials with coefficients in  $R$ ; for any ideal  $I$  in  $R$ ,  $R/I$  denotes the quotient ring of  $R$  modulo  $I$ ;  $\text{Quot}(R)$  denotes the quotient field (i.e. the field of fraction) extending an integral domain  $R$ . For any prime  $p$  and  $u \in \mathbb{F}_p \setminus \{0\}$ , we use  $\text{Inv}_p(u)$  to denote the multiplicative inverse of  $u$  modulo  $p$ .

Recall that an *integral domain* is a non-zero commutative ring where the multiplication of two non-zero elements is non-zero. We will need the following result for polynomials.

► **Lemma 10** ([9, Proposition 1 of Section 9.1]). *For any integral domain  $R$ , the ring of  $R$ -coefficient multi-variate polynomials  $R[x_1, \dots, x_m]$  is also an integral domain.*

### 2.2 Schwartz-Zippel Lemma

► **Lemma 11** ([8, 26, 19]). *Let  $R$  be an integral domain and  $S \subseteq R$  be a finite subset of  $R$ . For any  $m, d \in \mathbb{N}$  and any non-zero  $m$ -variate polynomial  $f : R^m \rightarrow R$  of total degree  $d$*

$$\Pr_{\vec{x} \in S^m} [f(\vec{x}) = 0] \leq \frac{d}{|S|},$$

where  $\vec{x} = (x_1, \dots, x_m)$  is uniformly sampled from  $S^m$ .

### 2.3 Transcendental Number Theory

► **Definition 12.** *A complex number  $\alpha \in \mathbb{C}$  is called algebraic if it is the root of a non-zero integer-coefficient polynomial of finite degree, and called transcendental if it is not algebraic.*

Algebraic numbers, denoted by  $\overline{\mathbb{Q}}$ , is a sub-field of  $\mathbb{C}$ . In particular, any rational number  $\alpha = p/q$  is algebraic as it is the root of the degree-1 integer-coefficient polynomial  $qx - p$ .

► **Theorem 13** (Lindemann–Weierstrass Theorem [4, Theorem 1.4]). *If  $\alpha_1, \dots, \alpha_n \in \overline{\mathbb{Q}}$  are distinct algebraic numbers,  $e^{\alpha_1}, \dots, e^{\alpha_n} \in \mathbb{C}$  are linearly independent over the field  $\overline{\mathbb{Q}}$  of algebraic numbers. In particular,  $e$  is transcendental.*

## 3 Circuit and Query Models

### 3.1 Definition of the Circuit Model

The circuit model we introduce next extends the standard *arithmetic circuit model* by *exponentiation gates* that is intended to model the exponential function  $\exp(x) \triangleq e^x$  over real numbers. Formally, an  $\text{AExp}^1$  circuit is a DAG consisting of vertices that are either an input variable or a gate of the following types (all gates are of fan-out 1):

1. constant gates  $C_a$  of fan-in 0 intended to model a fixed integer  $a$ ;
2. addition gates  $\Sigma(x_1, \dots, x_m)$  of unbounded fan-in that are intended to model the addition of real numbers, i.e.,  $x_1 + \dots + x_m$ ;
3. multiplication gates  $\Pi(x_1, \dots, x_m)$  of unbounded fan-in that are intended to model the multiplication of real numbers, i.e.,  $x_1 x_2 \dots x_m$ .
4. division gates  $\text{Frac}(x, y)$  of fan-in 2 intended to model the division of real numbers, i.e.,  $x/y$ ;
5. exponentiation gates  $\text{Exp}(x)$  of fan-in 1 intended to model the exponential function, i.e.,  $e^x$ .

In addition, we do not allow compositions of exponentiation gates in  $\text{AExp}^1$  circuits; that is, for any  $\text{AExp}^1$  circuit, there is at most *one* exponentiation gate along any path from an input variable to an output gate.<sup>4</sup> We assume by default that an  $\text{AExp}^1$  circuit has only one output gate, though the definitions and our results can be easily generalized to multi-output circuits.

**Evaluation of  $\text{AExp}^1$  Circuits over Real Numbers.** The evaluation of  $\text{AExp}^1$  circuits over real numbers should be clear through the definition. Let  $C$  be any  $n$ -input  $\text{AExp}^1$  circuit and  $\vec{x} \in \mathbb{R}^n$ , the evaluation of  $C(\vec{x})$  is defined as the output of the output gate, where the output values of gates are defined by gate-by-gate evaluation following a topological order.

In particular, if the divisor of a division gate is zero,  $C(\vec{x})$  is undefined. Therefore, an  $\text{AExp}^1$  circuit may compute a partial function  $C : \mathbb{R}^n \rightarrow \mathbb{R}$ . We define the *domain* of a  $C(\vec{x})$  circuit on  $\mathbb{R}$ , denoted by  $\text{dom}_{\mathbb{R}}(C)$ , as the set of  $\vec{x} \in \mathbb{R}^n$  such that  $C(\vec{x})$  is defined. For any  $\vec{x} \notin \text{dom}_{\mathbb{R}}(C)$ , we may also say  $C(\vec{x}) = \perp$ .

**Practical Motivation: Modeling Components in Neural Networks.** Our circuit model is a straightforward formalization of the program representation in Mirage [25]. Intuitively, the motivation to introduce  $\text{AExp}^1$  circuit is to model components that are widely used in model artificial neural networks that consist of non-linear activation functions.

### 3.2 Query Model over Reals, and Why it is not Satisfactory

Since  $\text{AExp}^1$  circuits compute functions over  $\mathbb{R}$ , a natural idealized model for the identity testing problem is that the algorithm can evaluate the circuit over any real input. Formally, a identity testing algorithm for  $\text{AExp}^1$  circuits  $C$  in *real query model* is allowed to query the following oracle:

- (*Evaluation*). Given an input  $\vec{x} \in \mathbb{Q}^n$ , the evaluation oracle reports whether  $C(\vec{x})$  is undefined,  $C(\vec{x}) = 0$ , or  $C(\vec{x}) \in \mathbb{R} \setminus \{0\}$ .

The main problem of the real query model is that computers cannot deal with real numbers. For real-world applications, real numbers are usually approximated by *floating-point numbers*, which does not satisfy arithmetic laws such as the commutativity and associativity of addition. This leads *two-sided errors* in the implementation of a testing algorithm in real query model:

- (*Completeness*). Even if  $C$  is identically zero over  $\mathbb{R}$ , it may evaluate to a non-zero value on some inputs when the evaluation queries are implemented in float-point numbers.

<sup>4</sup> Similarly, one can define  $\text{AExp}^k$  circuits where there are at most  $k$  exponential gates along any such path. In this work, we focus on  $\text{AExp}^1$  circuits as it is natural and is more relevant to the practical motivation of this work.

- (*Soundness*). Even if  $C$  is not identically zero, it may evaluate to 0 on any input when evaluation queries are implemented in float-point numbers, either because of the precision issue or because of the failure of arithmetic laws.

The occurrences of errors on both sides reduce the consistency and reliability of the testing algorithms. Take the example of optimization compilers PET [24] or Mirage [25] that aim to detect redundancy in neural networks modeled as  $\text{AExp}^1$  circuits. Two sided error will make the algorithms less predictable and reliable to users; indeed, it could be possible that the oracle will hardly ever report  $C(\vec{x}) \in \mathbb{R} \setminus \{0\}$  even if  $C(\vec{x})$  is defined and non-zero for most or all  $\vec{x}$  due to the floating-point issue. In that case, the optimization compiler will barely find any redundancy. It is worth noting that the error is *on top of* the completeness and soundness error of the identity testing algorithm, so it cannot be resolved by a better (e.g., deterministic) testing algorithm.

### 3.3 Query Model over Finite Fields

Next, we introduce a natural query model for the identity testing problem of  $\text{AExp}^1$  circuits over finite fields that can be implemented in real-world applications *without* the precision issue.

**Evaluation over Finite Fields.** Let  $p, q$  be two primes such that  $q \mid p - 1$ , and  $G_{p,q}$  be the (unique) multiplicative subgroup of  $\mathbb{F}_p^*$  of order  $q$ . Equivalently,  $G_{p,q}$  contains the roots of the univariate polynomial  $z^q - 1$  over  $\mathbb{F}_p$ . Loosely speaking, we will define the evaluation of  $C$  on the input  $(\vec{u}, \vec{v}) \in \mathbb{F}_p^n \times \mathbb{F}_q^n$  and  $a \in G_{p,q}$  (modulo  $(p, q)$ ) by

- interpreting the exponentiation gate by the function  $x \mapsto a^x \bmod p$ ,
- implementing all computation “on the exponent” in  $\mathbb{F}_q$ ,
- implementing all computation “under the exponent” in  $\mathbb{F}_p$ ,

and evaluating the circuit gate by gate.

We use  $C_a(\vec{u}, \vec{v})$  to denote the evaluation above; see the full version for its formal definition.

Similar to the evaluation of  $\text{AExp}^1$  circuits on  $\mathbb{R}$ ,  $C_a(\vec{u}, \vec{v})$  may be undefined, denoted by  $C_a(\vec{u}, \vec{v}) = \perp$ , due to division-by-zero. For each  $a \in G_{p,q}$ , we define the domain of  $C$  with respect to  $a$  modulo  $(p, q)$ , denoted by  $\text{dom}_{p,q,a}(C)$ , as  $\{(\vec{u}, \vec{v}) \in \mathbb{F}_p^n \times \mathbb{F}_q^n \mid C_a(\vec{u}, \vec{v}) \neq \perp\}$ .

**Algebraic Query Model.** Subsequently, we define the identity testing problem for an  $\text{AExp}^1$  circuit  $C$  over the algebraic query model as follows. Let  $p, q, G_{p,q}$  be specified above. The algorithm for identity testing is allowed to query the following oracle:

- (*Evaluation*). Given  $(\vec{u}, \vec{v}) \in \mathbb{F}_p^n \times \mathbb{F}_q^n$  and  $a \in G_{p,q}$ , the oracle reports  $C_a(\vec{u}, \vec{v}) \in \mathbb{F}_p \cup \{\perp\}$ .

This query model is efficient both theoretically and practically. It can be observed, for instance, that the oracle in the algebraic query model can be implemented by algorithms in  $O(s \cdot (\log p + \log q))$  space and polynomial time. In practice, the simplicity of the evaluation model makes it possible to implement it in the neural network compiler scenario (see [24, 25] for more details).

## 4 Exponential Polynomials

To understand the functions computed by  $\text{AExp}^1$  circuits, we will define *exponential polynomials* that, intuitively, generalize (multi-variate) polynomials by allowing terms of form  $\exp(\cdot)$ . Formally:

► **Definition 14** (exponential polynomial). *Let  $R$  be an integral domain and  $d, m$  be integers,  $d \geq 0$ ,  $m \geq 1$ . An  $m$ -variate  $R$ -coefficient exponential polynomial  $P(\vec{x})$  of degree  $d$  is defined as*

$$P(\vec{x}) \triangleq f_1(\vec{x}) \exp\left(\frac{g_1(\vec{x})}{h_1(\vec{x})}\right) + f_2(\vec{x}) \exp\left(\frac{g_2(\vec{x})}{h_2(\vec{x})}\right) + \cdots + f_k(\vec{x}) \exp\left(\frac{g_k(\vec{x})}{h_k(\vec{x})}\right),$$

where  $\vec{x} = (x_1, \dots, x_m)$  denotes the indeterminates,  $f_i, g_i$  and  $h_i$  are  $m$ -variate polynomials of degree  $d$  with coefficients from  $R$ .

The number of terms  $k$  is said to be the width of  $P(\vec{x})$ ,  $\{f_i(\vec{x})\}_{i \in [k]}$  are said to be the coefficient polynomials of  $P(\vec{x})$ , and  $\{g_i(\vec{x})/h_i(\vec{x})\}_{i \in [k]}$  are said to be the exponent fractions of  $P(\vec{x})$ .

In this paper, we only use the special case where  $R = \mathbb{Z}$ , i.e., integer-coefficient exponential polynomials. Nevertheless, we will develop the elementary arithmetic of exponential polynomials with respect to an arbitrary integral domain  $R$  for coefficients.

## 4.1 Basic Arithmetic Properties

We stress that an exponential polynomial should be considered as an abstract expression rather than a function. In particular,  $\exp(\cdot)$  should be considered as a symbol instead of the exponential function over  $\mathbb{R}$ . For simplicity, we will also use the summation symbol  $\sum$  to define an exponential polynomial, i.e.,

$$P(\vec{x}) \triangleq \sum_{i=1}^k f_i(\vec{x}) \exp\left(\frac{g_i(\vec{x})}{h_i(\vec{x})}\right),$$

where the summation symbol is a shorthand of the  $k$ -term summation in Definition 14.

We first define the addition and multiplication of exponential polynomials. Let  $R$  be a ring and  $P(\vec{x}), P'(\vec{x})$  be  $R$ -coefficient exponential polynomials defined as

$$P(\vec{x}) \triangleq \sum_{i=1}^k f_i(\vec{x}) \exp\left(\frac{g_i(\vec{x})}{h_i(\vec{x})}\right), \quad P'(\vec{x}) \triangleq \sum_{i=1}^{k'} f'_i(\vec{x}) \exp\left(\frac{g'_i(\vec{x})}{h'_i(\vec{x})}\right). \quad (1)$$

We can naturally define the addition and multiplication of  $P(\vec{x})$  and  $P'(\vec{x})$  as:

$$P(\vec{x}) + P'(\vec{x}) \triangleq \sum_{i=1}^k f_i(\vec{x}) \exp\left(\frac{g_i(\vec{x})}{h_i(\vec{x})}\right) + \sum_{i=1}^{k'} f'_i(\vec{x}) \exp\left(\frac{g'_i(\vec{x})}{h'_i(\vec{x})}\right). \quad (2)$$

$$P(\vec{x}) \cdot P'(\vec{x}) \triangleq \sum_{i=1}^k \sum_{j=1}^{k'} f_i(\vec{x}) f'_j(\vec{x}) \exp\left(\frac{g_i(\vec{x}) h'_j(\vec{x}) + g'_j(\vec{x}) h_i(\vec{x})}{h_i(\vec{x}) h'_j(\vec{x})}\right). \quad (3)$$

Next, we consider the arithmetic laws for exponential polynomials. As hinted at in the definitions of addition and multiplication, both operations are *commutative* and *associative*, and multiplication is *distributive* over addition. Moreover, it is implicit in the definition of multiplication that exponentiation symbol satisfies

$$\exp\left(\frac{g(\vec{x})}{h(\vec{x})}\right) \cdot \exp\left(\frac{g'(\vec{x})}{h'(\vec{x})}\right) = \exp\left(\frac{g(\vec{x})h'(\vec{x}) + g'(\vec{x})h(\vec{x})}{h(\vec{x})h'(\vec{x})}\right). \quad (4)$$

Furthermore, we impose the following axiom that allows merging terms with the same exponent fraction:

$$f(\vec{x}) \exp\left(\frac{g(\vec{x})}{h(\vec{x})}\right) + f'(\vec{x}) \exp\left(\frac{g(\vec{x})}{h(\vec{x})}\right) = (f(\vec{x}) + f'(\vec{x})) \exp\left(\frac{g(\vec{x})}{h(\vec{x})}\right). \quad (5)$$

## 95:14 Identity Testing for Circuits with Exponentiation Gates

We say that  $P(\vec{x})$  and  $P'(\vec{x})$  are *identical*, denoted by  $P(\vec{x}) = P'(\vec{x})$ , if they can be transformed to each other using the arithmetic laws above.

We note that the following two exponential polynomials

$$P_1(\vec{x}) = \exp(x); \quad P_2(\vec{x}) = \exp\left(\frac{x^2 - 2x}{x - 2}\right)$$

are not identical as the division law is *not* allowed in the exponentiation symbol. This is intentional as in the evaluation of  $\text{AExp}^1$  circuits, we will only evaluate the circuit gate by gate without trying to simplify the circuit using the division law.

### 4.2 Evaluation of Exponential Polynomials

Similar to standard multivariate polynomials, we can define the evaluation of exponential polynomials that explains how to view such abstract expressions as functions. In particular, we will introduce two definitions corresponding to the *real evaluation model* and *algebraic evaluation model* of  $\text{AExp}^1$  circuits.

Let  $P(\vec{x})$  be an  $n$ -variate integer-coefficient exponential polynomial defined by the coefficient polynomials  $\{f_i\}_{i \in [k]}$  and exponent fractions  $\{g_i/h_i\}_{i \in [k]}$ .

**Evaluation of Exponential Polynomials on Real Numbers.** We can view  $P(\vec{x})$  as a function  $P(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$  as follows. Given an input  $\vec{u} \in \mathbb{R}^n$ , the evaluation of  $P(\vec{x})$  on the input  $\vec{u}$  is defined as

$$P(\vec{u}) \triangleq \sum_{i=1}^k f_i(\vec{u}) \cdot \exp\left(\frac{g_i(\vec{u})}{h_i(\vec{u})}\right)$$

where all the operators (i.e., additions, divisions, multiplications, and exponentiations) are interpreted as corresponding functions in  $\mathbb{R}$ . If for some  $i \in [k]$ ,  $h_i(\vec{u}) = 0$ , the exponential polynomial is said to be undefined on  $\vec{u}$ . The domain of  $P(\vec{x})$  over  $\mathbb{R}$ , denoted by  $\text{dom}_{\mathbb{R}}(P)$ , is defined as  $\text{dom}_{\mathbb{R}}(P) \triangleq \{\vec{u} \in \mathbb{R}^n \mid h_i(\vec{u}) \neq 0 \forall i \in [k]\}$ .

**Evaluation of Exponential Polynomials on Finite Fields.** Similar to the algebraic query model for  $\text{AExp}^1$  circuits, we require two finite fields for the computation under and over exponents to define the evaluation of  $P$  over finite fields. Let  $p, q$  be two primes such that  $q \mid p - 1$ , and  $G_{p,q}$  be the multiplicative subgroup of  $\mathbb{F}_p^*$  of order  $q$ . Given inputs  $\vec{u} \in \mathbb{F}_p^n$ ,  $\vec{v} \in \mathbb{F}_q^n$ , and  $a \in G_{p,q}$ , the evaluation of  $P$  on  $\vec{u}$  and  $\vec{v}$ , denoted by  $P_a(\vec{u}, \vec{v})$ , is defined as:

$$P_a(\vec{u}, \vec{v}) \triangleq \left( \sum_{i=1}^k f_i(\vec{u}) \cdot a^{g_i(\vec{u}) \cdot (h_i(\vec{v}))^{-1} \bmod q} \right) \bmod p.$$

Note that  $P_a(\vec{u}, \vec{v})$  is undefined, denoted by  $P_a(\vec{u}, \vec{v}) = \perp$ , if  $h_i(\vec{v}) \bmod q = 0$  for some  $i \in [k]$ . The domain of  $P(\vec{x})$  over  $(p, q, a)$  is defined as

$$\text{dom}_{p,q,a}(P(\vec{x})) \triangleq \{(\vec{u}, \vec{v}) \in \mathbb{F}_p^n \times \mathbb{F}_q^n \mid h_i(\vec{u}) \neq 0 \bmod q \forall i \in [k]\}.$$

► **Proposition 15.** *Let  $P(\vec{x})$  and  $P'(\vec{x})$  be identical integer-coefficient exponential polynomials,  $p, q$  be two primes such that  $q \mid p - 1$ . For any  $a$  in the multiplicative subgroup of  $\mathbb{F}_p^*$  of order  $q$ , we have that  $\text{dom}_{p,q,a}(P) = \text{dom}_{p,q,a}(P')$  and  $P_a(\vec{u}, \vec{v}) = P'_a(\vec{u}, \vec{v})$  for any  $\vec{u} \in \mathbb{F}_p^n$ ,  $\vec{v} \in \mathbb{F}_q^n$ .*

### 4.3 Condensation of Exponential Polynomials

Recall that two exponential polynomials are said to be identical if they can be transformed to each other by arithmetic laws. As there is no division law for the exponent fractions, the exponential polynomials

$$P_1(x) \triangleq \exp(x); \quad P_2(x) \triangleq \exp\left(\frac{x^2 - 2x}{x - 2}\right)$$

are not considered to be the same polynomial. In particular, we can notice that  $\text{dom}_{\mathbb{R}}(P_1) \neq \text{dom}_{\mathbb{R}}(P_2)$ . Nevertheless, these two exponential polynomials as functions are essentially the same over all but the input  $x = 2$ .

We now introduce the *condensation* of integer-coefficient exponential polynomials that simplifies an exponential polynomial by merging terms with “essentially the same” exponent fractions together, which will be useful in proving our main results.

**Equivalent Exponential Fractions.** Let  $P(\vec{x})$  be a multi-variate integer-coefficient exponential polynomial of width  $k$ ,  $\{g_i(\vec{x})/h_i(\vec{x})\}_{i \in [k]}$  be the exponent fractions of  $P(\vec{x})$ . Suppose that for each  $i \in [k]$ ,  $h_i$  is not a zero polynomial. We define  $\sim_P$  to be the relation over  $[k]$  such that

$$i \sim_P j \quad \text{iff} \quad g_i(\vec{x})h_j(\vec{x}) - g_j(\vec{x})h_i(\vec{x}) = 0.$$

Note that  $g_i(\vec{x})h_j(\vec{x}) - g_j(\vec{x})h_i(\vec{x}) = 0$  means that it is a zero integer-coefficient polynomial, or equivalently,  $g_i(\vec{u})h_j(\vec{u}) - g_j(\vec{u})h_i(\vec{u}) = 0$  for any  $\vec{x} \in \overline{\mathbb{R}}$ .

► **Lemma 16.** *Suppose that  $h_i(\vec{x})$  is not a zero polynomial for any  $i \in [k]$ , then  $\sim_P$  is an equivalence relation over  $[k]$ .*

**Condensation of Exponential Polynomials.** Subsequently, we define a condensation  $\hat{P}$  of an exponential polynomial  $P$  as obtained by grouping the coefficient polynomials according to the relation  $\sim_P$ . Formally:

► **Definition 17** (Condensation of exponential polynomials). *Let  $P(\vec{x})$  be an integer-coefficient exponential polynomial of degree  $d$  and width  $k$ ,  $\{f_i(\vec{x})\}_{i \in [k]}$  be the coefficient polynomials of  $P(\vec{x})$ ,  $\{g_i(\vec{x})/h_i(\vec{x})\}$  be the exponent fractions of  $P(\vec{x})$  such that  $h_i(\vec{x})$  is not a zero polynomial for each  $i \in [k]$ . Let  $\pi = \{[i_1]_{\pi}, [i_2]_{\pi}, \dots, [i_t]_{\pi}\}$  be the partition of  $[k]$  induced by  $\sim_P$ , and  $i_1, \dots, i_t$  be arbitrary representative elements. We say that  $\hat{P}(\vec{x})$  is a condensation of  $P(\vec{x})$  if it is of form*

$$\tilde{P}(\vec{x}) \triangleq F_1(\vec{x}) \cdot \exp\left(\frac{g_{i_1}(\vec{x})}{h_{i_1}(\vec{x})}\right) + \dots + F_t(\vec{x}) \cdot \exp\left(\frac{g_{i_t}(\vec{x})}{h_{i_t}(\vec{x})}\right)$$

where  $F_j(\vec{x}) \triangleq \sum_{i \in [i_j]_{\pi}} f_i(\vec{x})$  is an integer-coefficient polynomial of degree at most  $\hat{d}$ .

► **Definition 18** (Condensed exponential polynomials). *Let  $P(\vec{x})$  be an exponential polynomial with exponent fractions  $\{g_i(\vec{x})/h_i(\vec{x})\}_{i \in [k]}$ .  $P(\vec{x})$  is a condensed exponential polynomial if  $h_i$ 's are not zero polynomials and  $g_i(\vec{x})h_j(\vec{x}) \neq h_i(\vec{x})g_j(\vec{x})$  for  $i, j \in [k]$  and  $i \neq j$ .*

We stress that the condensation of an exponential polynomial is not unique as we can choose the representative elements  $i_1, \dots, i_t$  arbitrarily from their equivalent classes. The following proposition shows that  $\hat{P}$  may have a larger domain compared to  $P$ , but they agree on the domain of  $\hat{P}$ .

► **Proposition 19.** *Let  $\hat{P}$  be a condensation of an integer-coefficient exponential polynomial  $P$ . Then:*

- $\text{dom}_{\mathbb{R}}(P) \subseteq \text{dom}_{\mathbb{R}}(\hat{P})$ , and  $P, \hat{P}$  agree on  $\text{dom}_{\mathbb{R}}(P)$ .
- Let  $p, q$  be prime numbers such that  $q \mid p - 1$ ,  $G_{p,q}$  be the multiplicative subgroup of  $\mathbb{F}_p^*$  of order  $q$ , and  $a \in G_{p,q}$ . Then  $\text{dom}_{p,q,a}(P) \subseteq \text{dom}_{p,q,a}(\hat{P})$ , and  $P, \hat{P}$  agree on  $\text{dom}_{p,q,a}(P)$ .

#### 4.4 Structural Lemma for AExp<sup>1</sup> Circuits

Now we are ready to prove a structural lemma showing that AExp<sup>1</sup> circuits can be seen as fractions of exponential polynomials.

For simplicity, we introduce the following notation. Let  $P, P'$  be  $n$ -variate exponential polynomials, we define  $\text{dom}_{\mathbb{R}}(P/P')$  be the set  $\{\vec{u} \in \text{dom}_{\mathbb{R}}(P) \cap \text{dom}_{\mathbb{R}}(P') \mid P'(\vec{u}) \neq 0\}$ , i.e., the domain of the fraction  $P(\vec{x})/P'(\vec{x})$ . Similarly, let  $p, q$  be prime numbers such that  $q \mid p - 1$ ,  $G_{p,q}$  be the multiplicative subgroup of  $\mathbb{F}_p^*$  of order  $q$ , and  $a \in G_{p,q}$ , we define  $\text{dom}_{p,q,a}(P/P')$  as the set  $\{\vec{u} \in \text{dom}_{p,q,a}(P) \cap \text{dom}_{p,q,a}(P') \mid P'(\vec{u}) \neq 0\}$ .

► **Lemma 20.** *For every  $n$ -input AExp<sup>1</sup> circuit  $C$ , there are  $n$ -variate integer-coefficient exponential polynomials  $P(\vec{x})$  and  $P'(\vec{x})$  such that the following holds:*

- $\text{dom}_{\mathbb{R}}(C) = \text{dom}_{\mathbb{R}}(P/P')$ , and for each  $\vec{u} \in \text{dom}_{\mathbb{R}}(C)$ ,  $C(\vec{u}) = P(\vec{u})/P'(\vec{u})$ .
- Let  $p, q$  be prime numbers such that  $q \mid p - 1$ ,  $G_{p,q}$  be the multiplicative subgroup of  $\mathbb{F}_p^*$  of order  $q$ , and  $a \in G_{p,q}$ . Then  $\text{dom}_{p,q,a}(C) = \text{dom}_{p,q,a}(P/P')$  and for every  $(\vec{u}, \vec{v}) \in \text{dom}_{p,q,a}(C)$ ,  $C_a(\vec{u}, \vec{v}) \equiv P_a(\vec{u}, \vec{v}) \cdot \text{Inv}_p(P'_a(\vec{u}, \vec{v})) \pmod{p}$ .

*In particular, if  $C$  does not contain exponentiation gates, both  $P$  and  $P'$  do not contain the exponentiation symbol, i.e.,  $P$  and  $P'$  are integer-coefficient polynomials.*

The proof is left to the full version of the paper.

#### 4.5 Width, Degree, and Weight of Concrete Neural Network Components

It is proved in Lemma 20 that any AExp<sup>1</sup> circuit can be transformed to an equivalent fraction of exponential polynomials. We can therefore define the *width*, *degree*, and *weight* of an AExp<sup>1</sup> circuit.

► **Definition 21** (width, degree, and weight of AExp<sup>1</sup> circuits). *An AExp<sup>1</sup> circuit  $C$  is said to have width  $k$ , degree  $d$ , and weight  $w$  if there are integer-coefficient degree- $d$  width- $k$  exponential polynomials  $P(\vec{x})$  and  $P'(\vec{x})$  that satisfy both conditions in Lemma 20.*

We note that the transformation in Lemma 20 may not be efficient – the width, degree, and the bit-length of the integer coefficients may grow exponentially with respect to the size of the circuit. Nevertheless, it can be verified that many neural network components can be simulated by fractions of exponential polynomials with relatively small width, degree, and coefficients. We refer readers to the full version for more discussions.

## 5 Algorithms for Identity Testing

Now we are ready to describe our identity testing algorithms in real and algebraic query models. Indeed, our algorithms are essentially the same one: Randomly sample an input  $x$  (in corresponding models) and check whether the circuit evaluates to zero or  $\perp$  on the input  $\vec{x}$ . We will first prove the correctness of the algorithm in real query model (see Section 5.1), and generalize the proof to the algebraic query models in subsequent subsections.

## 5.1 Identity Testing in Real Query Model

Formally, the identity testing algorithm in real query model works as follows: Suppose that  $C : \mathbb{R}^n \rightarrow \mathbb{R}$  is an  $\text{AExp}^1$  circuit of width  $k$  and degree  $d$ , let  $B = 20dk^2$  be sufficiently large and  $S = \{1, 2, \dots, B\}$ , the algorithm uniformly sample  $x_1, x_2, \dots, x_n \in S$ , and accept if  $C(x_1, \dots, x_n) \neq 0$ .

It is clear that the algorithm is perfectly complete, and the soundness can be formalized as the following theorem.

► **Theorem 22.** *Let  $C : \mathbb{R}^n \rightarrow \mathbb{R}$  be an  $\text{AExp}^1$  circuit of width  $k$  and degree  $d$  that is not identically zero on  $\text{dom}_{\mathbb{R}}(C)$ . Then for any non-empty finite subset  $S \subseteq \mathbb{Q}$ , if  $\vec{x} \in S^n$  is sampled uniformly at random,  $\Pr[C(\vec{x}) \in \{0, \perp\}] \leq 8dk^2/|S|$ .*

The key step for proving Theorem 22 is a necessary and sufficient condition for an exponential polynomial  $P$  to be identically zero. Intuitively, if the exponent fractions  $g_1/h_1, \dots, g_k/h_k$  of  $P$  are pairwise distinct, then  $P$  is identically zero on  $\mathbb{R}^n$  if and only if all coefficient polynomials  $f_1, \dots, f_k$  are all identically zero. Formally:

► **Lemma 23.** *Let  $P : \mathbb{R}^n \rightarrow \mathbb{R}$  be a condensed exponential polynomial of form*

$$P(\vec{x}) = \sum_{i=1}^k f_i(\vec{x}) \cdot \exp\left(\frac{g_i(\vec{x})}{h_i(\vec{x})}\right),$$

where  $f_i, g_i, h_i$  are integer-coefficient polynomials of total degree  $d$  and  $h_i(\vec{x})$  is not identically zero on  $\mathbb{R}^n$  for every  $i \in [k]$ . Then the following statements are equivalent:

1.  $P$  is identically zero on  $\text{dom}_{\mathbb{R}}(P)$ ;
2.  $\Pr[P(\vec{x}) \in \{0, \perp\}] > 3dk^2/|S|$  for  $\vec{x} \in S^n$  sampled uniformly at random for any non-empty finite subset  $S \subseteq \mathbb{Q}$ .
3. For every  $i \in [k]$ ,  $f_i$  is identically zero on  $\mathbb{R}^n$ .

The proof is omitted; see the full version of the paper.

We are now ready to prove Theorem 22.

**Proof of Theorem 22.** Let  $C : \mathbb{R}^n \rightarrow \mathbb{R}$  be an  $\text{AExp}^1$  circuit of width  $k$  and degree  $d$  that is not identically zero on  $\text{dom}_{\mathbb{R}}(C)$ . Let  $S \subseteq \mathbb{Q}$  be a non-empty finite set. Then by the definition of degree and width of  $C$ , there are exponential polynomials  $P, P'$  defined by

$$P(\vec{x}) \triangleq \sum_{i \in [k]} f_i(\vec{x}) \cdot \exp\left(\frac{g_i(\vec{x})}{h_i(\vec{x})}\right), \quad P'(\vec{x}) \triangleq \sum_{j \in [k]} f'_j(\vec{x}) \cdot \exp\left(\frac{g'_j(\vec{x})}{h'_j(\vec{x})}\right).$$

such that  $C(\vec{x}) = P(\vec{x})/P'(\vec{x})$  for every  $\vec{x} \in \mathbb{R}^n$ , and for each  $i \in [k], j \in [k]$ ,  $f_i, g_i, h_i, f'_j, g'_j, h'_j$  are integer-coefficient  $n$ -variate polynomials of total degree at most  $d$ .

Note that for each  $i \in [k], j \in [k]$ , we have that  $h_i(\vec{x}), h'_j(\vec{x})$  are not identically zero on  $\mathbb{R}^n$ . This is because otherwise at least one of  $P(\vec{x}), P'(\vec{x})$  is undefined on every  $\vec{x} \in \mathbb{R}^n$ , which implies that  $\text{dom}_{\mathbb{R}}(C) = \emptyset$ . Moreover, we know that  $P$  and  $P'$  are not identically zero on their domains, respectively, as otherwise  $C$  must be identically zero on its domain.

Let  $\hat{P}(\vec{x})$  be a condensation of  $P(\vec{x})$ . As  $P$  is not identically zero on its domain, we know by Proposition 19 that  $\hat{P}(\vec{x})$  is not identically zero on its domain. Consider the random variable  $\vec{x} \in S^n$  sampled uniformly at random. We calculate the probability of the following events:

- Let  $\mathcal{E}_{\hat{P}}$  be the event that  $\hat{P}(\vec{x}) \notin \{0, \perp\}$ , i.e.,  $\hat{P}(\vec{x})$  is defined and  $\hat{P}(\vec{x}) \neq 0$ . By Lemma 23, we can see that  $\Pr[\mathcal{E}_{\hat{P}}] \geq 1 - 3dk^2/|S|$ .

## 95:18 Identity Testing for Circuits with Exponentiation Gates

- Let  $\mathcal{E}_\perp$  be the event that  $P(\vec{x})$  is defined, i.e.,  $h_i(\vec{x}) \neq 0$  for every  $i \in [k]$ . Note that  $\Pr[h_i(\vec{x}) = 0] \leq d/|S|$  for every fixed  $i \in [k]$  by Lemma 11. By the union bound, we know that  $\Pr[\mathcal{E}_\perp] = 1 - \Pr[\neg\mathcal{E}_\perp] \geq 1 - dk/|S|$ .
- Let  $\mathcal{E}_P$  be the event that  $P(\vec{x}) \notin \{0, \perp\}$ . Notice that

$$\Pr[\mathcal{E}_P] = \Pr[\mathcal{E}_P \wedge \mathcal{E}_\perp] + \Pr[\mathcal{E}_P \wedge \neg\mathcal{E}_\perp] \geq 1 - 3dk^2/|S|,$$

where  $\Pr[\mathcal{E}_P \wedge \mathcal{E}_\perp] = \Pr[\mathcal{E}_P]$  by Proposition 19 and  $\Pr[\mathcal{E}_P \wedge \neg\mathcal{E}_\perp] \leq \Pr[\neg\mathcal{E}_\perp] \leq dk/|S|$ . Therefore,  $\Pr[\mathcal{E}_P] \geq 1 - 3dk^2/|S| - dk/|S| \geq 1 - 4dk^2/|S|$ .

Following the same argument, we can prove that  $\Pr[P'(\vec{x}) \notin \{0, \perp\}] \geq 1 - 4dk^2/|S|$  over uniformly random  $\vec{x} \in S^n$ . It follows that

$$\begin{aligned} \Pr[C(\vec{x}) \in \{0, \perp\}] &= \Pr[P(\vec{x}) \in \{0, \perp\} \vee P'(\vec{x}) \in \{0, \perp\}] \\ &\leq \Pr[P(\vec{x}) \in \{0, \perp\}] + \Pr[P'(\vec{x}) \in \{0, \perp\}] \leq 8dk^2/|S|. \end{aligned}$$

This completes the proof. ◀

### 5.2 A Weak Descartes' Rule over Finite Fields

► **Theorem 24.** *Let  $p, q$  be prime numbers such that  $q \mid p - 1$ ,  $g \in \mathbb{F}_p$  be an element of order  $q$ , and  $G \triangleq \langle g \rangle$  be the unique multiplicative subgroup of  $\mathbb{F}_p^*$  of order  $q$ . Let  $k \leq q$ ,  $\alpha_1, \dots, \alpha_k \in \mathbb{F}_q$  be distinct, and  $\beta_1, \dots, \beta_k \in \mathbb{F}_p$ . Then the univariate polynomial*

$$f(z) \triangleq \beta_1 z^{\alpha_1} + \beta_2 z^{\alpha_2} + \dots + \beta_k z^{\alpha_k}$$

has at most  $q^{1-1/(k-1)}$  roots in  $G$ .

The key idea of the proof (which follows from [5, 11]) is to find a low-degree polynomial that has as many roots as  $f(z)$  in  $G$ . Let  $p, q$  be prime numbers such that  $q \mid p - 1$ , and  $G = \langle g \rangle$  be the unique multiplicative subgroup of  $\mathbb{F}_p^*$  of order  $q$ . Formally:

► **Proposition 25.** *Let  $c \in \{1, 2, \dots, q - 1\}$  and  $f_c(z)$  be the polynomial*

$$f_c(z) \triangleq \beta_1 z^{c\alpha_1 \bmod q} + \beta_2 z^{c\alpha_2 \bmod q} + \dots + \beta_k z^{c\alpha_k \bmod q}.$$

Then the polynomials  $\{f_c(z)\}_{c \in [q-1]}$  have the same number of roots in  $G$ .

► **Lemma 26** (Lemma 4.1 of [11]). *For  $\alpha_1, \dots, \alpha_t, N \in \mathbb{N}$  and  $n \leq N/\gcd(\alpha_1, \dots, \alpha_t, N)$ , there is a  $c \in [n - 1]$  such that*

$$\max_{i \in [t]} \{\alpha_i \cdot c \bmod N\} \leq \frac{N}{n^{1/t}}.$$

**Proof of Theorem 24.** Fix  $p, q \in \mathbb{N}$ ,  $g \in \mathbb{F}_p$ ,  $k$ , elements  $\alpha_1, \dots, \alpha_k \in \mathbb{F}_q$  and  $\beta_1, \dots, \beta_k \in \mathbb{F}_p$ . Let  $f_c(z) \in \mathbb{F}_p[z]$  be the univariate polynomial

$$f_c(z) \triangleq \beta_1 z^{c\alpha_1 \bmod q} + \beta_2 z^{c\alpha_2 \bmod q} + \dots + \beta_k z^{c\alpha_k \bmod q}$$

and  $f(z) = f_1(z)$ . Assume that  $\alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_k$ . We may further assume without loss of generality that  $\alpha_1 = 0$ , as otherwise we can consider the polynomial  $f(z)/z^{\alpha_1}$  that has the same number of roots in  $G$ .

Let  $N \triangleq q$  and  $n = q$ , we have  $\gcd(\alpha_2, \dots, \alpha_k, N) = 1$  and thus  $n \leq N/\gcd(\alpha_2, \dots, \alpha_k, N)$ . By Lemma 26, there exists some  $c \in \{1, 2, \dots, q - 1\}$  such that for every  $i \in \{2, 3, \dots, k\}$ ,

$$\alpha_i \cdot c \bmod q \leq q^{1-1/(k-1)},$$

which further implies that the polynomial  $f_c(z)$  is of degree at most  $q^{1-1/(k-1)}$ . Subsequently,  $f_c(z)$  has at most  $q^{1-1/(k-1)}$  roots in  $G$ . This completes the proof as  $f(z)$  and  $f_c(z)$  have the same number of roots in  $G$  by Proposition 25. ◀

### 5.3 Identity Testing in Algebraic Query Model

► **Theorem 7.** Let  $C : \mathbb{R}^n \rightarrow \mathbb{R}$  be an  $\text{AExp}^1$  circuit of width  $k$ , degree  $d$  and weight  $w$ ,  $p$  and  $q$  be prime numbers such that  $q \mid p-1$  and  $q > 2(kw)^2$ . Let  $G_{p,q}$  be the unique multiplicative subgroup of  $\mathbb{F}_p^*$  of order  $q$ . The following hold:

- (Completeness). If  $C$  is identically zero on  $\text{dom}_{\mathbb{R}}(C)$ , then for every  $a \in G_{p,q}$ ,  $C_a(\vec{u}, \vec{v}) \in \{0, \perp\}$  for every  $(\vec{u}, \vec{v}) \in \mathbb{F}_p^n \times \mathbb{F}_q^n$ .
- (Soundness). If  $C$  is not identically zero on  $\text{dom}_{\mathbb{R}}(C)$ , then for uniformly random  $\vec{u} \leftarrow \mathbb{F}_p^n, \vec{v} \leftarrow \mathbb{F}_q^n, a \leftarrow G_{p,q}$ ,  $\Pr[C_a(\vec{u}, \vec{v}) \notin \{0, \perp\}] \geq 1 - 8dk^4 \cdot q^{-1} - q^{-1/(k^2-1)}$ .

We first prove two lemmas: Lemma 27 is used to prove the completeness property, and Lemma 28 is used to prove the soundness property.

► **Lemma 27.** Let  $\hat{P}$  be a condensation of an integer-coefficient exponential polynomial  $P$ . If  $P$  is identically zero on  $\text{dom}_{\mathbb{R}}(P)$  and  $\text{dom}_{\mathbb{R}}(P) \neq \emptyset$ , then  $\hat{P}$  is identically zero on  $\text{dom}_{\mathbb{R}}(\hat{P})$ .

The proof is omitted; see the full version of the paper.

► **Lemma 28.** Let  $p, q$  be prime numbers such that  $q \mid p-1$ ,  $G_{p,q}$  be the unique multiplicative subgroup of  $\mathbb{F}_p^*$  of order  $q$ . Let  $P$  be a condensed integer-coefficient exponential polynomial

$$P(\vec{x}) = \sum_{i \in [k]} f_i(\vec{x}) \cdot \exp\left(\frac{g_i(\vec{x})}{h_i(\vec{x})}\right)$$

of degree  $d$  and width  $k$ . Suppose that for every  $i \in [k]$ ,  $f_i(\vec{x})$  is not identically zero on  $\mathbb{F}_p$  and  $h_i(\vec{x})$  is not identically zero on  $\mathbb{F}_q$ , then for  $(\vec{u}, \vec{v}, a)$  uniformly sampled from  $\mathbb{F}_p^n \times \mathbb{F}_q^n \times G_{p,q}$ ,

$$\Pr[P_a(\vec{u}, \vec{v}) \in \{0, \perp\}] \leq 3dk^2 \cdot q^{-1} + q^{-1/(k-1)}.$$

The proof is similar to that of Lemma 23; see the full version of the paper.

**Proof of Theorem 7.** By the definition of width, degree of  $\text{AExp}$  circuits, there are exponential polynomials

$$P(\vec{x}) = \sum_{i \in [k]} f_i(\vec{x}) \cdot \exp\left(\frac{g_i(\vec{x})}{h_i(\vec{x})}\right), P'(\vec{x}) = \sum_{i \in [k]} f'_i(\vec{x}) \cdot \exp\left(\frac{g'_i(\vec{x})}{h'_i(\vec{x})}\right)$$

such that the following hold:

- $f_i, g_i, h_i, f'_i, g'_i, h'_i$  are  $d$ -degree polynomials with coefficients in  $[-w, w] \cap \mathbb{Z}$ ,
- $\text{dom}_{\mathbb{R}}(C) = \text{dom}_{\mathbb{R}}(P/P')$ , and  $C(\vec{u}) = P(\vec{u})/P'(\vec{u})$  for  $\vec{u} \in \text{dom}_{\mathbb{R}}(C)$ ,
- $\text{dom}_{p,q,a}(C) = \text{dom}_{p,q,a}(P/P')$ , and  $C_a(\vec{u}, \vec{v}) = P(\vec{u}, \vec{v}) \cdot \text{Inv}_p(P'(\vec{u}, \vec{v}))$  for any  $a \in G_{p,q}$  and  $(\vec{u}, \vec{v}) \in \text{dom}_{p,q,a}(C)$ .

**Proof of Completeness.** Let  $C$  be an  $\text{AExp}^1$  circuit that is identically zero on  $\text{dom}_{\mathbb{R}}(C)$ . We first prove that either  $P(\vec{x})$  is identically zero on  $\text{dom}_{\mathbb{R}}(P)$ , or  $P'(\vec{x})$  is identically zero on  $\text{dom}_{\mathbb{R}}(P')$ .

Towards a contradiction, we assume that  $P$  is not identically zero on  $\text{dom}_{\mathbb{R}}(P)$  and  $P'$  is not identically zero on  $\text{dom}_{\mathbb{R}}(P')$ . Let  $S \subseteq \mathbb{Z}$  be a set of size  $30dk^2$ . By Theorem 22, we know that for  $\vec{x} \in S^n$  sampled uniformly at random,

$$\Pr[P(\vec{x}) \in \{0, \perp\}], \Pr[P'(\vec{x}) \in \{0, \perp\}] < \frac{1}{3}.$$

By the union bound, we know that for  $\vec{x} \in S^n$  sampled uniformly at random, with probability at least  $1/3$ ,  $P(\vec{x}), P'(\vec{x}) \notin \{0, \perp\}$ . For any such  $\vec{x} \in S^n$ , we have that  $\vec{x} \in \text{dom}_{\mathbb{R}}(C)$  and  $C(\vec{x}) = P(\vec{x})/P'(\vec{x}) \neq 0$ , which is impossible as  $C$  is identically zero on its domain.

Suppose that  $P(\vec{x})$  is identically zero on  $\text{dom}_{\mathbb{R}}(P)$ , we know by Lemma 27 that the condensation  $\hat{P}(\vec{x})$  of  $P(\vec{x})$  is also identically zero on  $\text{dom}_{\mathbb{R}}(\hat{P})$ . Let  $k'$  be the width of  $\hat{P}$  and

$$\hat{P}(\vec{x}) = \sum_{i \in [k']} \hat{f}_i(\vec{x}) \cdot \exp\left(\frac{\hat{g}_i(\vec{x})}{\hat{h}_i(\vec{x})}\right).$$

By Lemma 23, we know that for every  $i \in [k']$ ,  $\hat{f}_i(\vec{x})$  is identically zero on  $\mathbb{R}^n$ , or equivalently,  $\hat{f}_i(\vec{x})$  is a zero polynomial. In that case, we must have  $\hat{f}_i(\vec{x}) \equiv 0 \pmod{p}$  for every  $p$ . This implies that

$$\hat{P}_a(\vec{u}, \vec{v}) = \sum_{i \in [k']} \hat{f}_i(\vec{u}) \cdot \exp\left(\frac{\hat{g}_i(\vec{v})}{\hat{h}_i(\vec{v})}\right) \in \{0, \perp\}.$$

Subsequently,  $P_a(\vec{u}, \vec{v}) \in \{0, \perp\}$  as  $\hat{P}$  and  $P$  agree on  $\text{dom}_{p,q,a}(P)$  (see Proposition 19), which further implies that  $C_a(\vec{u}, \vec{v}) \in \{0, \perp\}$ .

Similarly, if  $P'(\vec{x})$  is identically zero on  $\text{dom}_{\mathbb{R}}(P')$ , then  $P'_a(\vec{u}, \vec{v}) \in \{0, \perp\}$ , which implies that  $C_a(\vec{u}, \vec{v}) = \perp$ . This concludes the completeness of the algorithm.

**Proof of Soundness.** Let  $R(\vec{x}) = P(\vec{x}) \cdot P'(\vec{x})$ . It can be verified that it is an exponential polynomial with width  $k^2$ , degree  $2d$  and weight  $w^2$ . It follows that:

- For any  $\vec{x} \in \mathbb{R}^n$ ,  $C(\vec{x}) = P(\vec{x})/P'(\vec{x}) \in \{0, \perp\}$  if and only if  $R(\vec{x}) \in \{0, \perp\}$ .
- For any  $\vec{x} \in \mathbb{F}_p^n \times \mathbb{F}_q^n, \vec{a} \in G_{p,q}$ ,  $C_a(\vec{x}) = P_a(\vec{x})/P'_a(\vec{x}) \in \{0, \perp\}$  if and only if  $R_a(\vec{x}) \in \{0, \perp\}$ .

Since  $C$  is not identically zero on  $\text{dom}_{\mathbb{R}}(C)$ , i.e.,  $C(x) \notin \{0, \perp\}$  for some  $\vec{x} \in \mathbb{R}^n$ ,  $R$  is not identically zero on  $\text{dom}_{\mathbb{R}}(R)$ . Let  $\hat{R}$  be a condensation of  $R$ . By Proposition 19,  $\hat{R}$  is not identically zero on  $\text{dom}_{\mathbb{R}}(\hat{R})$ .

Let  $\{f''_i\}_{i \in [k^2]}$  be the coefficient polynomials of  $\hat{R}$ , and  $\{g''_i/h''_i\}_{i \in [k^2]}$  be the exponent fractions of  $\hat{R}$ . Note that  $h''_i$  is non-zero for every  $i \in [k^2]$ , as otherwise  $\text{dom}_{\mathbb{R}}(R) = \emptyset$ . By Lemma 23, we know that  $f''_i$  is non-zero for some  $i \in [k^2]$  and, without loss of generality, we may assume that  $f''_i$  is non-zero for every  $i \in [k^2]$ .

It can be verified that the integer weights in  $h''_i$  are within  $[-w^2, w^2]$ . Moreover, the integer weights in  $f''_i$  are within  $[-(kw)^2, (kw)^2]$ , as each  $f''_i$  is a summation of at most  $k^2$  polynomials that have integer weights within  $[-w^2, w^2]$ . As  $p > q > 2(kw)^2$ , we know that for every  $i \in [k^2]$ ,  $f''_i$  is not identically zero on  $\mathbb{F}_p^n$  and  $h''_i$  is not identically zero on  $\mathbb{F}_q^n$ .

Let  $\vec{x} = (\vec{u}, \vec{v}) \in \mathbb{F}_p^n \times \mathbb{F}_q^n$  and  $a \in G_{p,q}$  be random variables sampled uniformly at random. We calculate the probability of the following events:

- Let  $\mathcal{E}_{\hat{R}}$  be the event that  $\hat{R}(\vec{x}) \notin \{0, \perp\}$ . By Lemma 28,  $\Pr[\mathcal{E}_{\hat{R}}] \geq 1 - 6dk^4/q - q^{-1/(k^2-1)}$ .
- Let  $\mathcal{E}_{\perp}$  be the event that  $h''_i(\vec{v}) \neq 0$  for every  $i \in [k^2]$ . For every fixed  $i \in [k^2]$ , by Lemma 11,  $\Pr[h''_i(\vec{v}) = 0] \leq 2d/q$ . Then by the union bound,  $\Pr[\mathcal{E}_{\perp}] = 1 - \Pr[\neg \mathcal{E}_{\perp}] \geq 1 - 2dk^2/q$ .
- Let  $\mathcal{E}_R$  be the event that  $R(\vec{x}) \notin \{0, \perp\}$ . Notice that

$$\Pr[\mathcal{E}_{\hat{R}}] = \Pr[\mathcal{E}_{\hat{R}} \wedge \mathcal{E}_{\perp}] + \Pr[\mathcal{E}_{\hat{R}} \wedge \neg \mathcal{E}_{\perp}] \geq 1 - 6dk^4/q - q^{-1/(k^2-1)},$$

where  $\Pr[\mathcal{E}_{\hat{R}} \wedge \mathcal{E}_{\perp}] = \Pr[\mathcal{E}_R]$  and  $\Pr[\mathcal{E}_{\hat{R}} \wedge \neg \mathcal{E}_{\perp}] \leq 2dk^2/q$ . Therefore,

$$\Pr[\mathcal{E}_R] \geq 1 - 6dk^4/q - q^{-1/(k^2-1)} - 2dk^2/q \geq 1 - 8dk^4/q - q^{-1/(k^2-1)}.$$

This completes the proof, as  $C_a(\vec{x}) \in \{0, \perp\}$  if and only if  $R_a(\vec{x}) \in \{0, \perp\}$ . ◀

## References

- 1 Manindra Agrawal and Somenath Biswas. Primality and identity testing via chinese remaindering. *J. ACM*, 50(4):429–443, 2003. doi:10.1145/792538.792540.
- 2 Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. URL: <http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264>.
- 3 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998. doi:10.1145/273865.273901.
- 4 Alan Baker. *Transcendental Number Theory*. Cambridge Mathematical Library. Cambridge University Press, 2022.
- 5 Ran Canetti, John Friedlander, Sergei Konyagin, Michael Larsen, Daniel Lieman, and Igor Shparlinski. On the statistical properties of diffie-hellman distributions. *Israel Journal of Mathematics*, 120:23–46, 2000.
- 6 Qi Cheng, Shuhong Gao, J. Maurice Rojas, and Daqing Wan. Sparse univariate polynomials with many roots over finite fields. *Finite Fields Their Appl.*, 46:235–246, 2017. doi:10.1016/J.FFA.2017.03.006.
- 7 Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.
- 8 Richard A. DeMillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Lett.*, 7(4):193–195, 1978. doi:10.1016/0020-0190(78)90067-4.
- 9 David Steven Dummit and Richard M Foote. *Abstract algebra*, volume 3. Wiley Hoboken, 2004.
- 10 Zhihao Jia, Oded Padon, James Thomas, Todd Warszawski, Matei Zaharia, and Alex Aiken. Taso: optimizing deep learning computation with automatic generation of graph substitutions. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles, SOSP '19*, pages 47–62, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3341301.3359630.
- 11 Zander Kelley. Roots of sparse polynomials over a finite field. *LMS Journal of Computation and Mathematics*, 19(A):196–204, 2016. doi:10.1112/S1461157016000334.
- 12 László Lovász. On determinants, matchings, and random algorithms. In Lothar Budach, editor, *Fundamentals of Computation Theory, FCT 1979, Proceedings of the Conference on Algebraic, Arithmetic, and Categorical Methods in Computation Theory, Berlin/Wendisch-Rietz, Germany, September 17-21, 1979*, pages 565–574. Akademie-Verlag, Berlin, 1979.
- 13 Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. URL: <http://www.cambridge.org/de/academic/subjects/computer-science/algorithmics-complexity-computer-algebra-and-computational-g-analysis-boolean-functions>.
- 14 Øystein Ore. *Über höhere kongruenzen*. Grøndahl, 1921.
- 15 Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Naresh Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Liang, Mengli Liu, Ignasi Mattei, Suriya Singh, Jonathan Smith, Nikolaus Sygnowski, Agata Wu, Yongfei Yao, Zachary Bell, Gregory Devito, Jie Yuan, Brian Zhu, Michael Zick, Luke Jia, Tero Maggioni, Soumith Chintala, and Gavin Stevens. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.
- 16 PyTorch. PyTorch Documentation, entry: torch.nn.functional.scaled\_dot\_product\_attention. [https://docs.pytorch.org/docs/2.2/generated/torch.nn.functional.scaled\\_dot\\_product\\_attention.html](https://docs.pytorch.org/docs/2.2/generated/torch.nn.functional.scaled_dot_product_attention.html), 2025. Accessed: 2025-11-27.
- 17 Nitin Saxena. Progress on polynomial identity testing. *Bull. EATCS*, 99:49–79, 2009.
- 18 Nitin Saxena. Progress on polynomial identity testing-II. In *Perspectives in Computational Complexity: The Somenath Biswas Anniversary Volume*, pages 131–146. Springer, 2014.

- 19 Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. doi:10.1145/322217.322225.
- 20 Adi Shamir.  $IP = PSPACE$ . *J. ACM*, 39(4):869–877, 1992. doi:10.1145/146585.146609.
- 21 Richard M. Stallman and the GCC Developer Community. *Using the GNU Compiler Collection*. GNU Press, Boston, MA, USA, 2025. For GCC version 15.2.0. URL: <https://gcc.gnu.org/onlinedocs/gcc-15.2.0/gcc.pdf>.
- 22 William T Tutte. The factorization of linear graphs. *Journal of the London Mathematical Society*, 1(2):107–111, 1947.
- 23 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf).
- 24 Haojie Wang, Jidong Zhai, Mingyu Gao, Zixuan Ma, Shizhi Tang, Liyan Zheng, Yuanzhi Li, Kaiyuan Rong, Yuanyong Chen, and Zhihao Jia. PET: Optimizing tensor programs with partially equivalent transformations and automated corrections. In *15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21)*, pages 37–54. USENIX Association, July 2021. URL: <https://www.usenix.org/conference/osdi21/presentation/wang>.
- 25 Mengdi Wu, Xinhao Cheng, Shengyu Liu, Chunan Shi, Jianan Ji, Kit Ao, Praveen Velliengiri, Xupeng Miao, Oded Padon, and Zhihao Jia. Mirage: A multi-level superoptimizer for tensor programs. In *19th USENIX Symposium on Operating Systems Design and Implementation (OSDI 25)*, Boston, MA, July 2025. USENIX Association. URL: <https://www.usenix.org/conference/osdi25/presentation/wu-mengdi>.
- 26 Richard Zippel. Probabilistic algorithms for sparse polynomials. In Edward W. Ng, editor, *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposium on Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979. doi:10.1007/3-540-09519-5\_73.