Multivariate Resultants: Complexity and Applications to Real Feasibility

C. S. Bhargav[1], Radu Curticapean[1,2], Gorav Jindal[1], and Himanshu Shukla[1]

[1] {bhargav.cs,radu.curticapean,gorav.jindal,himanshu.shukla}@ur.de
[1]University of Regensburg, Germany*
[2]IT University Copenhagen, Denmark

Abstract

The multivariate (or *Macaulay*) resultant is a polynomial in the coefficients of homogeneous polynomials $p_1, \ldots, p_n \in \mathbb{Z}[x_1, \ldots, x_n]$. It vanishes precisely when these polynomials have a nonzero common *complex* root, which makes the resultant a fundamental object in computational algebra and algebraic geometry. The degree of a resultant can be exponential in n, even when the input forms are quadratic. A recent breakthrough of Andrews, Garg, and Schost (FOCS 2026) showed that multivariate resultants can nevertheless be evaluated in the counting hierarchy. On the complexity side, testing whether a resultant vanishes is only known to be NP-hard. We significantly narrow down the complexity of multivariate resultants by proving the following:

1. Polynomial-size arithmetic circuits for multivariate resultants imply VP = VNP.
2. Evaluation of multivariate resultants modulo powers of two is #P-hard, and determining a specified binary digit of their absolute value is PP-hard under many-one reductions.
3. Deciding the sign of a resultant is $\oplus$P-hard, even when it is guaranteed to be non-zero.

These results imply that even determining the sign or a specified bit of a resultant lies outside the polynomial hierarchy, unless the hierarchy collapses. The hardness results already hold for the resultants of homogeneous quadratic forms defined by symmetric tensors of order three. Such resultants are also known as *tensor determinants* (or symmetric hyperdeterminants), and are, up to normalization, discriminants of the associated cubic forms.

We also apply resultants to real feasibility, proving that the existential theory of the reals belongs to the counting hierarchy (ETR $\in$ CH). Using the critical point method and resultant-based elimination, we reduce real feasibility to real-root tests for univariate polynomials of exponential degree. We then guess polynomial-length certificates for univariate real-root existence from short partial Thom encodings. We verify these certificates by encoding the parity of the number of matching real roots using the sign (of a coefficient) of a parameterized resultant. This extends the parity encoding used in our sign-hardness proof. Crucially, we use the constructions of Andrews, Garg, and Schost to preserve CH-access to the coefficient bits throughout, allowing the entire procedure to remain in the counting hierarchy.

*All authors are funded by the European Union (ERC, CountHom, 101077083). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency.



Contents

1 Introduction **3**
- 1.1 Our Results . . . 6
- 1.2 Proof Outline . . . 8
  - 1.2.1 Hardness of Resultant . . . 8
  - 1.2.2 $\exists\mathbb{R}$ in the counting hierarchy . . . 9
- 1.3 Background and AI use . . . 11

2 Preliminaries **11**
- 2.1 Notation . . . 11
- 2.2 Resultants . . . 11
- 2.3 Tensor determinants . . . 13

3 Algebraic hardness of the resultant **14**
- 3.1 Algebraic complexity . . . 14
- 3.2 Grid assignments . . . 14
- 3.3 Coefficients of grid assignments . . . 15
- 3.4 Hardness of the coefficient family . . . 17
- 3.5 Discriminants of cubic forms . . . 18

4 Modular evaluation of the resultant **18**
- 4.1 Dyadic determinant evaluations . . . 19
- 4.2 Addressed bits . . . 21

5 Hardness of resultant signs **22**
- 5.1 Encoding parities algebraically . . . 22

6 $\exists\mathbb{R}$ in the counting hierarchy **24**
- 6.1 Polynomial arithmetic in CH . . . 24
- 6.2 Constructing univariate families preserving feasibility . . . 26
  - 6.2.1 Constructing a finite set witnessing feasibility . . . 26
  - 6.2.2 Constructing a family of univariate polynomials preserving feasibility . . . 28
- 6.3 Univariate real-root existence testing . . . 29
  - 6.3.1 Short description of roots . . . 29
  - 6.3.2 Rational approximation and combined sign tests . . . 30
- 6.4 Completing the proof . . . 32

A Proofs omitted from Section 3 **37**

B Proofs omitted from Section 4 **38**

C Proofs omitted from Section 5 **40**

D Proofs omitted from Section 6 **40**

E Parity hardness of univariate resultant signs **45**

F Hilbert Nullstellensatz in AM **47**

1 Introduction

Arguably the most fundamental problem in computational algebra is to decide whether a system of polynomial equations over the complex numbers $\mathbb{C}$ has a solution. Due to its role in a famous connection between algebraic geometry and algebra, this is also known as the *Hilbert Nullstellensatz problem*.

Problem 1.1 (Polynomial satisfiability over $\mathbb{C}$)**.** Given a list of polynomials $p_1, \ldots, p_k \in \mathbb{Z}[x_1, \ldots, x_n]$, each specified as a list of monomials and coefficients, decide whether there is a point $\boldsymbol{a} \in \mathbb{C}^n$ such that $p_1(\boldsymbol{a}) = \cdots = p_k(\boldsymbol{a}) = 0$.

When all polynomials are of degree at most 1, this problem amounts to solving a linear system of equations and therefore admits efficient algorithms. Even for polynomials of degree 2, the problem is NP-hard, for example, because it can be used to decide whether a graph $G = (V, E)$ contains an independent set of size k: Introduce a variable x_v for each $v \in V$ and consider the $|V| + |E| + 1$ equations

$$x_v(x_v - 1) = 0 \text{ for all } v \in V, \qquad x_u x_v = 0 \text{ for all } uv \in E, \qquad \sum_{v \in V} x_v = k.$$

The joint solutions correspond precisely to independent sets of size k: The first equations force $x_v \in \{0, 1\}$, the second ensure non-adjacency, and the last equation checks that exactly k vertices are selected.

Upper bounds for polynomial system satisfiability. In contrast to NP-hardness, proving that polynomial satisfiability over $\mathbb{C}$ is *contained* in NP seems much more difficult. Unlike over small finite fields, a solution over the complex numbers need not have a polynomial-length representation and hence cannot simply be guessed as a witness. For a long time, the best known unconditional upper bound for polynomial satisfiability over $\mathbb{C}$ was PSPACE, due to Canny [Can88].

Koiran [Koi96] showed that, conditional on the Generalized Riemann Hypothesis (GRH), polynomial satisfiability over $\mathbb{C}$ lies in the polynomial-time hierarchy, even in AM. His algorithm is essentially a randomized reduction to satisfiability over $\mathbb{Z}/p\mathbb{Z}$: Once a prime p of polynomial bitlength is fixed, satisfiability admits a polynomial-length witness, as the variables range over $\mathbb{Z}/p\mathbb{Z}$. Koiran proves that an unsatisfiable system can become satisfiable modulo only comparatively few primes, whereas GRH guarantees that a satisfiable system is satisfiable modulo sufficiently many small primes. This gap can be detected by an Arthur–Merlin protocol.

More recently, Andrews, Garg, and Schost [AGS26] discovered a different route and showed that polynomial satisfiability over $\mathbb{C}$ lies in the counting hierarchy CH *without* assuming GRH: They reduce general systems to resultant computations and then transform uniform constant-depth arithmetic circuits of exponential size for resultants into algorithms in CH. Resultants are among the central algebraic tools for studying polynomial systems. The CH algorithm gives an efficient connection between the computational complexity of resultants and that of polynomial satisfiability, providing a strong motivation for understanding resultant computation.

Remark 1.2. The full GRH is not needed for this upper bound. Assuming the zero-free region for Dedekind zeta functions claimed by Cureton and Thorner [CT26], the prime-ideal estimate of Frei, Sofos, and Widmer [FSW26] implies that polynomial satisfiability over $\mathbb{C}$ lies in AM; see Appendix F. For related earlier claims, see OpenAI [Ope26c].

Hilbert's tenth problem. For a subring $A \subseteq \mathbb{C}$, let HN_A denote the problem of deciding whether a finite system of polynomial equations with integer coefficients has a solution in A. In particular, $\mathrm{HN}_{\mathbb{C}}$ is the problem in Problem 1.1.

Hilbert's tenth problem asks for a uniform algorithm that, given a Diophantine equation (a polynomial equation with integer coefficients), decides whether it has a solution in integers. A finite system of such equations has an integer solution exactly when the sum of the squares of its constituent polynomials has one. Thus Hilbert's tenth problem is equivalent to $\mathrm{HN}_{\mathbb{Z}}$. The MRDP theorem, proved by Matiyasevich building on work of Davis, Putnam, and Robinson, shows that no such algorithm exists [Mat70; Dav73].

The analogous problem $\mathrm{HN}_{\mathbb{Q}}$, often called Hilbert's tenth problem over $\mathbb{Q}$, asks whether an integer polynomial system has a rational solution. Its decidability is a well-known open problem; see [Poo08]. OpenAI has released a preprint claiming that this problem is undecidable [Ope26b].

Real feasibility. As defined above, $\mathrm{HN}_{\mathbb{R}}$ asks whether a finite system of polynomial equations with integer coefficients has a real solution. It is related to the *existential theory of the reals*, denoted ETR, the set of all true sentences of the form

$$\exists x_1, \ldots, x_n \in \mathbb{R} : \ \phi(x_1, \ldots, x_n),$$

where ϕ is a quantifier-free Boolean combination of polynomial equations and inequalities with integer coefficients. The class $\exists\mathbb{R}$ consists of the decision problems that reduce to this theory under polynomial-time many-one reductions. Although $\mathrm{HN}_{\mathbb{R}}$ only asks about equations, inequalities and Boolean combinations can be encoded using equations and additional variables. This implies that $\mathrm{HN}_{\mathbb{R}}$ is complete for $\exists\mathbb{R}$ [SCM24, Section 1.1]. Thus $\exists\mathbb{R}$ also consists of the problems that reduce to $\mathrm{HN}_{\mathbb{R}}$ in polynomial time, just as NP consists of the problems that reduce to Boolean satisfiability.

The class $\exists\mathbb{R}$ captures many problems in geometry and optimization, including geometric realization and graph drawing problems, and has applications in robotics and control theory. It was known to be contained in PSPACE [Can88]; see [Sch13, Section 1] and [SCM24] for background.

The class $\exists\mathbb{R}$ also has a connection to computation over the reals. In the Blum–Shub–Smale (BSS) model, registers hold arbitrary real numbers, and arithmetic and comparison operations take unit time [BSS89]. The classes $\mathsf{P}_{\mathbb{R}}$ and $\mathsf{NP}_{\mathbb{R}}$ are the deterministic and nondeterministic polynomial-time classes in this model. To connect the BSS model to Boolean complexity classes, we restrict inputs to Boolean strings and machines to the constants 0 and 1; write $\mathsf{BP}(\mathsf{P}^0_{\mathbb{R}})$ and $\mathsf{BP}(\mathsf{NP}^0_{\mathbb{R}})$ for the resulting classes. Let PosSLP be the problem of deciding whether a division-free straight-line program computes a positive integer. The relationships between these classes and Boolean computation are

$$\mathsf{BP}(\mathsf{P}^0_{\mathbb{R}}) = \mathsf{P}^{\mathsf{PosSLP}} \qquad \text{[ABKM09]},$$
$$\mathsf{BP}(\mathsf{NP}^0_{\mathbb{R}}) = \exists\mathbb{R} \qquad \text{[SCM24, Section 1.1]}.$$

The first equality motivates the conjecture that its nondeterministic counterpart also holds: $\exists\mathbb{R} = \mathsf{NP}^{\mathsf{PosSLP}}$ [SCM24, Section 1.2]. Allender et al. also showed that $\mathsf{PosSLP} \in \mathsf{CH}$ [ABKM09], although this bound alone does not establish that $\exists\mathbb{R} \subseteq \mathsf{CH}$.

Resultants. For a square matrix M, the system $M\boldsymbol{x} = 0$ has a nonzero solution exactly when $\det M = 0$. Resultants extend this test to homogeneous polynomials, whose monomials all have the same total degree. For each choice of positive degrees $d_1, \ldots, d_n$, the multivariate resultant is a fixed polynomial in the coefficients of n homogeneous forms $f_1, \ldots, f_n$ in n variables. It vanishes exactly when the forms have a common nonzero solution over $\mathbb{C}$. We write $\mathrm{Res}_n^{(d)}$ when all degrees are d; in particular, $\mathrm{Res}_n^{(2)}$ is the resultant of n quadratic forms. The linear case is the ordinary determinant.

Multivariate resultants in general have determinantal representations: classical Macaulay formulas express them as quotients of determinants, and D'Andrea and Dickenstein [DD01] give further explicit formulas. For sparse resultants, D'Andrea, Jeronimo, and Sombra [DJS23] obtain analogous quotient formulas under suitable hypotheses. These formulas are useful for computation, although the matrix sizes can grow exponentially with the number of variables.

As a more familiar example, consider univariate polynomials $f, g \in \mathbb{C}[t]$ of positive degrees. Their resultant is a polynomial in the coefficients of f and g, and it vanishes exactly when they have a common root. It is also the determinant of their Sylvester matrix. Suppose f and g have degrees r, s and leading coefficients a, b, respectively. Write their roots, with multiplicity, as $\alpha_1, \ldots, \alpha_r$ and $\beta_1, \ldots, \beta_s$. By the Poisson formula [CLO05, Chapter 3], their resultant is given by

$$\mathrm{Res}(f, g) = a^s b^r \prod_{i=1}^{r} \prod_{j=1}^{s} (\alpha_i - \beta_j) = a^s \prod_{i=1}^{r} g(\alpha_i).$$

To connect this example with homogeneous systems, introduce $F(x, y) = y^r f(x/y)$ and $G(x, y) = y^s g(x/y)$. Since the leading coefficients a, b are nonzero, F and G have no common nonzero solution with $y = 0$. Their common nonzero solutions therefore correspond to common roots of f and g by $t = x/y$, and $\mathrm{Res}_{r,s}(F, G) = \mathrm{Res}(f, g)$.

We briefly discuss a geometric motivation for computing a resultant polynomial. Consider n homogeneous polynomials $\boldsymbol{f} = (f_1, \ldots, f_n)$ of positive degrees in variables $x_0, \ldots, x_n$ over an algebraically closed field K. If $\boldsymbol{f}$ defines a finite projective set $X \subset \mathbb{P}^n$, then computing the intersection multiplicity m_x of $f_1, \ldots, f_n$ at a point $x \in X$ (which intuitively measures how 'deeply' the polynomials intersect at x) is a fundamental problem in algebraic geometry. One can use multivariate resultants for this as follows. Take a generic linear form $\ell = \sum_{i=0}^{n} u_i x_i$ defined over $K(u_0, \ldots, u_n)$, and choose a nonzero representative $\widetilde{\boldsymbol{x}} \in K^{n+1}$ of each $x \in X$. By the multivariate version of the Poisson formula [CLO05, Chapter 3] there is a scalar $c \in K^*$, independent of $u_0, \ldots, u_n$, such that

$$\mathrm{Res}_{x_0,\ldots,x_n}(f_1, \ldots, f_n, \ell) = c \prod_{x \in X} \ell(\widetilde{\boldsymbol{x}})^{m_x}.$$

Thus factorization of the resultant polynomial in $K[u_0, \ldots, u_n]$ gives the intersection multiplicities at $x \in X$. This illustrates an application of computing the resultant polynomial beyond deciding its vanishing. We encourage readers to consult [CLO05, Chapter 3] and [GKZ94, Chapter 13] for a comprehensive treatment of resultants. We give the definition and properties in Section 2.

For integer-coefficient systems, one can also study the resultant's value, its sign, and the bits of its absolute value. This raises the following questions about the computational complexity of resultants.

Problem 1.3 (Complexity of the multivariate resultant)**.** How efficiently can one evaluate a multivariate resultant or decide whether it vanishes? As n and the degrees vary, how small can arithmetic circuits over $\mathbb{C}$ for these polynomials be? In particular, does the family $\{\mathrm{Res}_n^{(2)}\}_{n \geq 1}$ have circuits of size polynomial in n?

Grenet, Koiran, and Portier [GKP13] showed that deciding whether a resultant is zero is NP-hard for low-degree systems, over fields of *any* characteristic. For systems with integer coefficients, they also observed that deciding whether the resultant vanishes lies in AM under GRH, by reduction to the Hilbert Nullstellensatz problem [GKP13, Proposition 1 and Corollary 1]. They also examined the large matrices used in classical resultant formulas. Although these matrices can have exponential size, a small Boolean circuit can compute any requested entry in binary from its binary row and column indices. For general integer matrices given by such circuits, deciding whether the determinant is zero is PSPACE-complete [GKP13, Section 4.2, Theorem 7]; the problem is already hard for $0/1$ matrices whose determinant is promised to be 0, 1, or -1. This does not establish the same hardness for Macaulay matrices, but it shows that their short descriptions alone do not make determinant computation easy.

Andrews, Garg, and Schost [AGS26] subsequently found a way to exploit the structure of resultants: they constructed uniform constant-depth arithmetic circuits of exponential size and obtained counting-hierarchy algorithms for resultant computation. We study the arithmetic circuit complexity of the resultant as a polynomial, as asked in Problem 1.3; our result is stated in Theorem 1.4. We also study the complexity of evaluating it modulo an integer (Theorem 1.8) and determining its sign (Theorem 1.11).

Tensor determinants. To prove our hardness results in Section 1.1, we study resultants arising from symmetric cubic tensors. Such a tensor $A = (a_{ijk})$ is an array whose entries are unchanged by permutations of their indices. It defines n quadratic forms $(A\boldsymbol{x}^2)_i = \sum_{j,k=1}^{n} a_{ijk} x_j x_k$. Their resultant is the *tensor determinant* $\mathrm{Det}_n(A)$ [HHLQ13, Definition 1.2]. Qi called its restriction to symmetric tensors the symmetric hyperdeterminant [Qi05, Section 2, Proposition 1]. We use tensor determinant throughout. It vanishes exactly when these quadratic forms have a common nonzero solution. We give the formal definition in Section 2.

The word *hyperdeterminant* also refers to a different construction. The geometric hyperdeterminant of Gelfand, Kapranov, and Zelevinsky detects degeneracy of a multilinear form on a product of projective

spaces [GKZ94, Chapter 14, Section 1]. Its defining polynomial exists when the corresponding dual variety is a hypersurface. The tensor determinant instead tests whether the homogeneous system $A\boldsymbol{x}^2 = 0$ has a nonzero solution. The two constructions agree for symmetric $2\times2\times2$ tensors, but differ in general: for $3 \times 3 \times 3$ tensors, their degrees are respectively 36 and 12 [BHO14, Section 1]. The VNP-hardness conjecture of Hillar and Lim [HL13, Conjecture 13.1], discussed also by Joux and Narayanan [JN25, Section 1.2] and Narayanan [Nar25, Section 1.4], concerns the geometric hyperdeterminant. Our results concern the tensor determinant and do not settle that conjecture.

There is also a *combinatorial hyperdeterminant*, defined by extending the permutation formula for matrix determinants to even-order tensors [HL13, Section 4]. Barvinok proved that testing whether it is zero is NP-hard under polynomial-time Turing reductions for integer tensors of order four [Bar95, Corollary 5.5.2]; see also [HL13, Sections 1.4 and 4] for this reduction convention. Gurvits proved #P-hardness of its evaluation, and Hillar and Lim observed that the same reduction gives VNP-hardness [HL13, Section 4, Theorem 4.2 and Corollary 4.3]. This combinatorial construction is distinct from the tensor determinant studied here.

1.1 Our Results

The resultant $\mathrm{Res}_n^{(2)}$ of quadratic forms has $n\binom{n+1}{2}$ coefficient variables and degree $n2^{n-1}$. Its degree is too large for it to belong to the usual polynomial-degree classes VP or VNP. Nonetheless, it could have circuits of polynomial size and exponential degree: for example, x^{2^n} has an $O(n)$-size circuit by repeated squaring. Our first result is that such small circuits for the resultant would falsify Valiant's hypothesis.

Theorem 1.4 (Algebraic hardness of resultants)**.** *If the polynomial family* $\{\mathrm{Res}_n^{(2)}\}_{n\geq1}$ *has* $\mathsf{poly}(n)$*-size arithmetic circuits over* $\mathbb{C}$*, then* $\mathsf{VP} = \mathsf{VNP}$.

In fact, the same conclusion already holds if $\mathsf{poly}(n)$*-size circuits exist for the family of tensor determinants* $\{\mathrm{Det}_n(A)\}_{n\geq1}$ *restricted to symmetric tensors A of order three.*

Remark 1.5. Although the hardness result is stated over $\mathbb{C}$, the same proof works over any field of characteristic zero. If $\mathrm{Res}_n^{(2)}$ has quasi-polynomial size circuits, the proof of the above theorem instead gives $\mathsf{VNP} \subseteq \mathsf{VQP}$, which would be contrary to Valiant's *extended hypothesis* [Bür00, Corollary 2.31]. We get an inclusion but not an equality, since VQP is not contained in VNP [Bür00, Corollary 8.9].

Since discriminants and symmetric tensor determinants are the same up to a natural change of coordinates and an explicit scalar, we also obtain the hardness of the (cubic) discriminant as a corollary.

Corollary 1.6 (Algebraic hardness of discriminants)**.** *If the family of discriminants* $\{\Delta_n(f)\}_{n\geq1}$ *of cubic forms f in n variables has arithmetic circuits of size* $\mathsf{poly}(n)$ *over* $\mathbb{C}$*, then* $\mathsf{VP} = \mathsf{VNP}$.

Just as for matrices, one can study the spectral theory of (cubic) tensors [Qi05; HHLQ13; QL17]. For an indeterminate λ, one can consider the characteristic polynomial $\mathrm{Det}_n(\lambda I - A)$ where I is the (cubic) identity tensor. And just as for matrices, the coefficients of this polynomial give fundamental invariants of the tensor. To prove our hardness results, we study a closely related *characteristic coefficient polynomial*

$$\Phi_{n,s}(A) := [z^s]\,\mathrm{Det}_n(I + zA),$$

the coefficient of z^s in the polynomial $\mathrm{Det}_n(I + zA)$. It is a homogeneous polynomial of degree s in the entries of A, and we show that a particular family of these coefficients is VNP-complete.

Theorem 1.7 (VNP-completeness of the coefficient family)**.** *Let A be a generic symmetric tensor of order three. The family*

$$\left\{\Phi_{5m^2,m^2}(A)\right\}_{m\geq3}$$

is VNP*-complete under p-projections over* $\mathbb{C}$.

For symmetric integer tensors A with binary-encoded entries, exact computation of $\Phi_{n,s}(A)$, with $1 \le s \le n$, is also #P-hard under polynomial-time counting many-one reductions (Lemma 4.5). These low-degree coefficients have polynomial bit length, by Lemma 4.2.

We now turn to the problem of evaluating the resultant. Even for quadratic forms with coefficients of constant bit length, the resultant can have doubly exponential magnitude, and therefore, exponentially many bits. So we instead ask for the value modulo 2^k, with k given in unary, so the output is polynomial in the input size.

Theorem 1.8 (Hardness of modular evaluation)**.** *Given quadratic forms $f_1, \dots, f_n \in \mathbb{Z}[\boldsymbol{x}]$ with coefficients encoded in binary, and any integer $k \ge 1$ specified in unary, the problem of computing*

$$\mathrm{Res}_n^{(2)}(f_1, \dots, f_n) \bmod 2^k$$

is #P-hard under polynomial-time many-one (counting) reductions.

The modular hardness already holds for $\mathrm{Det}_n(A)$ with A an integer symmetric tensor of order three. Computing an addressed bit $\lfloor |\mathrm{Det}_n(A)|/2^j \rfloor \bmod 2$, with $j \ge 0$ supplied in unary, is PP-hard under polynomial-time many-one reductions, even when $\mathrm{Det}_n(A) > 0$.

Remark 1.9. By Toda's theorem [AB09, Theorem 17.14], we know that $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$. Thus, Theorem 1.8 makes it unlikely that modular resultant evaluation is in PH, as that would collapse the polynomial hierarchy.

For the problem of modular evaluation of the tensor determinant, the proof of the above theorem also gives a matching upper bound.

Corollary 1.10. *For an integer symmetric cubic tensor A of dimension n, and integer $k \ge 1$ specified in unary, computing $\mathrm{Det}_n(A) \bmod 2^k$ is $\mathsf{FP}^{\mathsf{PP}}$-complete under polynomial-time Turing reductions.*

One could then wonder about the complexity of computing the sign of the resultant. We show that even this is ⊕P-hard, already for resultants of quadratic forms.

Theorem 1.11 (Hardness of sign computation)**.** *Given quadratic forms $f_1, \dots, f_n \in \mathbb{Z}[\boldsymbol{x}]$, deciding whether $\mathrm{Res}_n^{(2)}(f_1, \dots, f_n) > 0$ is ⊕P-hard under polynomial time many-one reductions.*

The hardness in fact holds for deciding whether $\mathrm{Det}_n(A) > 0$, where A is an integer symmetric cubic tensor of dimension n with $O(n)$ nonzero entries and $\mathrm{Det}_n(A) \neq 0$.

Remark 1.12. The class ⊕P can be seen as decision problems corresponding to the least significant bit of a #P problem. Even though we do not know if NP is contained in $\mathsf{P}^{\oplus\mathsf{P}}$, the first half of the proof of Toda's theorem shows that $\mathsf{PH} \subseteq \mathsf{BPP}^{\oplus\mathsf{P}}$ (see, for example, [For09]). Combined with the Sipser-Gács-Lautemann theorem [AB09, Theorem 7.15] that $\mathsf{BPP} \subseteq \mathsf{PH}$, the hardness of sign computation implies a collapse of the polynomial hierarchy if the sign of the resultant could be computed in PH.

We can also show that computing the sign of a *univariate* resultant (of exponential degree) is ⊕P-hard, even when the polynomials are given as constant free arithmetic circuits. We include a proof in Appendix E.

Our next result is an application of resultants to the study of real feasibility. Using standard techniques in the field of real algebraic geometry and the work of Andrews, Garg and Schost [AGS26] on resultant computations, we show that the existential theory of the reals is in the counting hierarchy. The same technique of encoding parity by resultant signs used in our previous sign-hardness result plays a central role here.

Theorem 1.13. $\exists\mathbb{R} \subseteq \mathsf{CH}$.

Remark 1.14. A few observations are in order. First, our proof of $\exists\mathbb{R} \subseteq \mathsf{CH}$ uses resultant-sign computations. Placing unrestricted resultant-sign computation in PH would collapse the polynomial hierarchy by our hardness result. This presents a barrier to improving the bound through a general algorithm

for computing resultant signs. It does not rule out faster algorithms for the restricted resultant families arising in real feasibility. Concurrently and independently, OpenAI [Ope26a] and Meiburg [Mei26] also announced that $\exists\mathbb{R} \subseteq \mathsf{CH}$. As far as we can tell, their proofs also seem to need resultants, albeit implicitly.

Second, we note that Guo, Saxena, and Sinhababu [GSS19] introduced the Approximate Polynomial Satisfiability (APS) problem, where we are given polynomials $p_1, \ldots, p_m \in \mathbb{Z}[\boldsymbol{x}]$, and asked to decide if there is a point $\boldsymbol{\alpha} \in \mathbb{C}(\varepsilon)^n$ such that for all i, we have $\lim_{\varepsilon \to 0} p_i(\boldsymbol{\alpha}) = 0$. Recently, Balaji, Shirmohammadi, Tavenas, and Worrell [BSTW26] show that APS reduces to ETR in polynomial time. Since CH is closed under polynomial-time reductions, Theorem 1.13 implies that $\mathsf{APS} \in \mathsf{CH}$, an independent route to the direct bounds of [AHPS26; BSTW26].

1.2 Proof Outline

1.2.1 Hardness of Resultant

The starting point for the VNP-hardness and #P-hardness results is a grid encoding of the permanent by Brand et al. [BCKLOSW26, Lemma 25]. See also Curticapean and Wang [CW26]. This standard construction gives us a symmetric tensor $A = A_m(Z)$ of dimension N whose indices are partitioned into groups Ω_{ij} corresponding to positions $(i, j) \in \mathbb{Z}_m^2$ of a toroidal grid. For every possible way of assigning one index $\nu_{ij} \in \Omega_{ij}$ at each position, we take the product of the tensor entries on all grid triangles, and we sum these weights:

$$H_m(A) := \sum_{\nu_{ij} \in \Omega_{ij}} \prod_{(i,j) \in \mathbb{Z}_m^2} a_{\nu_{ij}, \nu_{i+1,j}, \nu_{i,j+1}}.$$

Here grid coordinates are taken modulo m. The construction ensures that $H_m(A_m(Z)) = \mathrm{per}(Z)$ is hard. Related grid encodings have been used before in the context of counting patterns by Curticapean and Marx [CM14] and Curticapean [Cur24], and in algebraic complexity by Dwivedi, Pago, and Seppelt [DPS26, Lemma 8.3].

We make the curious observation that, for $m = 3^r + 1$ and tensors supported on the grid triangles, the characteristic coefficient polynomial agrees with the grid assignment polynomial up to a constant factor.

$$\Phi_{N,m^2}(A) = \kappa_{N,m} H_m(A), \qquad \kappa_{N,m} \neq 0.$$

Methods for recovering individual graph counts from linear combinations [Cur24; DPS26], combined with insights from combinatorial expansions of tensor determinants [HHLQ13; SQH15] already suggest that one could extract a hard grid term from $\Phi_{N,m^2}(A)$. Indeed an earlier version of the paper invoked these techniques. The surprising feature of our new reduction is how direct the extraction is: for the chosen grid sizes, restricting the support already suffices to obtain the displayed identity.

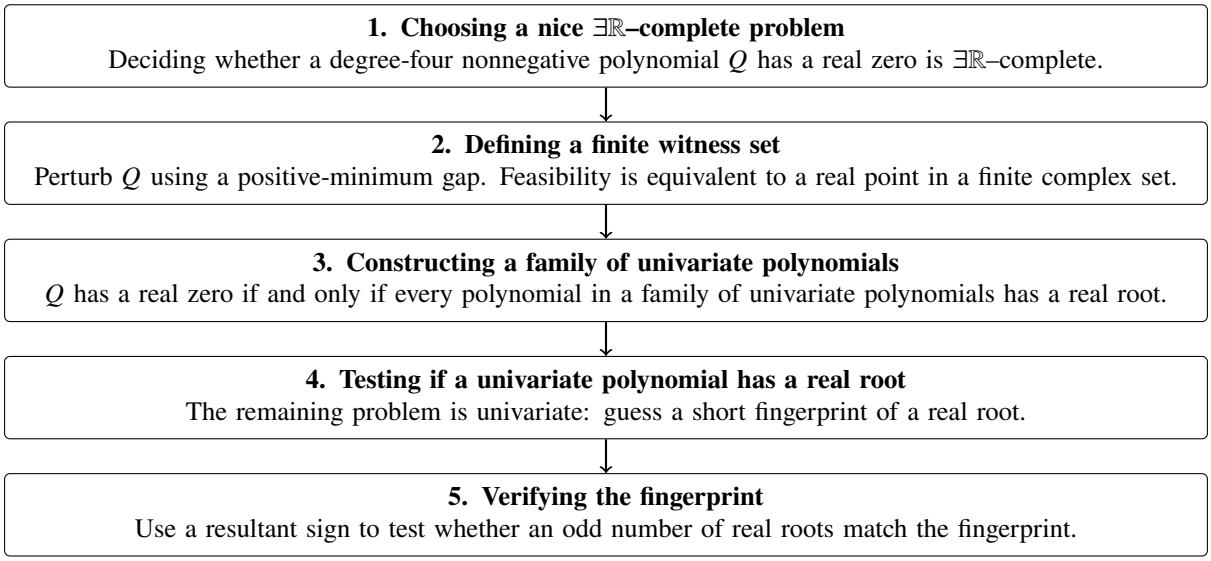
For the algebraic reduction, one extra diagonal entry of the permanent instance cancels $\kappa_{N,m}$. This gives the p-projection in Section 3.4. For the counting reduction, we recover the normalization from the same answer as the count by a more involved padding argument. Finally, evaluating the shifted tensor determinant at a sufficiently large power of two recovers a large characteristic coefficient as a signed digit. The construction and the decoding have polynomial size, giving a single-query reduction.

The sign-hardness proof (Theorem 1.11) proceeds in a different way. It reduces from counting (modulo 2) integer subsets summing up to at most T by encoding subsets $S \subseteq [n]$ as the simple real zeros of $x_i(x_i + 1) = 0$ for $i \in [n]$. Given positive integer weights w_i and an integer threshold T, a further quadratic takes the value $2\sum_{i \in S} w_i - (2T + 1)$ at the zero for S. These values are nonzero and negative exactly for subsets of weight at most T. The Poisson product formula expresses the resultant, up to a positive factor, as their product, so its sign records the parity of the number of such subsets. A suitable scaling realizes the homogenized equations as a sparse symmetric tensor, proving $\oplus$P-hardness. This connection between resultant signs and the parity of real solutions also underlies our real-feasibility algorithm, outlined next in Section 1.2.2.

1.2.2 $\exists\mathbb{R}$ in the counting hierarchy

At a very high level, the algorithm proceeds through the following steps. Starting from a nonnegative degree-four polynomial encoding the instance, we define a *finite* complex set whose real points witness feasibility. We then use multivariate resultants to project this set to a family of univariate polynomials: the original instance is feasible if and only if every polynomial in the family has a real root. To test each polynomial for a real root, we guess a short fingerprint for some root, and use the sign of (a coefficient of) a univariate resultant to detect whether an odd number of real roots match it. A short fingerprint is guaranteed to exist whenever a real root exists.

Now both resultant constructions may produce polynomials of exponential size. We crucially rely on the constructions of Andrews, Garg, and Schost [AGS26] because they let us maintain compact access in CH to coefficient bits through these computations. Together with the closure properties of CH, this lets us test fingerprints and quantify over projections within CH. We give an intuitive explanation of each step below.



Step 1: Choosing a nice $\exists\mathbb{R}$–complete problem. We start with the $\exists\mathbb{R}$–complete problem of deciding whether homogeneous integral quadratics $q_1, \ldots, q_m$ have a common nonzero real solution [SCM24, Entry A6]. Homogeneity guarantees a solution (if it exists) on the unit sphere. We combine the equations and the sphere condition into the nonnegative quartic polynomial

$$Q(\boldsymbol{x}) = \sum_{j=1}^{m} q_j(\boldsymbol{x})^2 + \left(\sum_{i=1}^{n} x_i^2 - 1\right)^2.$$

It can be shown that the original instance is feasible exactly when Q has a real zero (see Lemma 6.10). The sphere term also makes Q tend to infinity as $\|\boldsymbol{x}\| \to \infty$, so Q attains its minimum. Now feasibility is equivalent to asking whether this minimum is zero.

Step 2: Defining a finite witness set. We would like to use the critical point method [Bas17, Section 2.2] to find this minimum. However, every zero of Q (of which there could be infinitely many) is a global minimum and satisfies the critical point equations $\partial_i Q(\boldsymbol{x}) = 0$ for $i \in [n]$. We need a finite witness set, and we use the fact that if the instance is infeasible, there is an effective number $\gamma > 0$ such that $Q \geq \gamma$ everywhere. Although γ can have an exponentially long representation, its bits are efficiently accessible (see Lemma 6.11).

Therefore, we add a small degree-six (smallest even number greater than $\deg Q$) perturbation and set

$$H(\boldsymbol{x}) = Q(\boldsymbol{x}) + \frac{\gamma}{4} \sum_{i=1}^{n} x_i^6.$$

We define the complex set

$$Z = \{z \in \mathbb{C}^n : H(z) = \gamma/2,\ \partial_i H(z) = 0 \text{ for } 2 \le i \le n\}.$$

The perturbation makes Z finite, and the gap ensures that Z has a real point exactly when the original instance is feasible (Lemmas 6.12 and 6.13). The algorithm does not actually construct or enumerate Z, it works with its defining equations instead.

Step 3: Constructing a family of univariate polynomials. We use multivariate resultants to construct a family of univariate projection polynomials. The finite set Z has a real point if and only if every polynomial in the family has a real root (Lemma 6.14). Such univariate representations are standard [BPR06, Section 12.4], but the polynomials here may have exponential degree and coefficient bit lengths. We therefore maintain *compact coefficient access*: the bit of a coefficient at a given index can be computed in CH. The resultant constructions of Andrews, Garg, and Schost [AGS26] help preserve this access through projection (see Lemma 6.15).

Step 4: Testing if a univariate polynomial has a real root. We now need to decide whether a univariate polynomial R given by compact coefficient access has a real root. We replace R by its monic squarefree part p, which has the same roots as R but without multiplicities. A standard way to count the number of real roots in an interval is via the Sturm sequence [BPR06, Section 2.2.2]. This sequence can have exponentially many members, so explicitly examining it does not give the required CH bound.

Instead, the key idea is to guess the description of some real root and verify it. Now each real root of p has a distinct *Thom encoding* – the list of signs of the derivatives of p at that root [CR88] – which could again be exponentially long. The closure properties of CH only allow a polynomial length guess. Nonetheless, a halving argument (Lemma 6.18) shows that, if a real root exists, some root is uniquely identified among the real roots by at most $\lceil \log_2 \deg p \rceil$ derivative-sign conditions, which is polynomial in the input size. We call this short list a *fingerprint* and guess it. Figure 1 illustrates how two derivative signs can isolate a root.

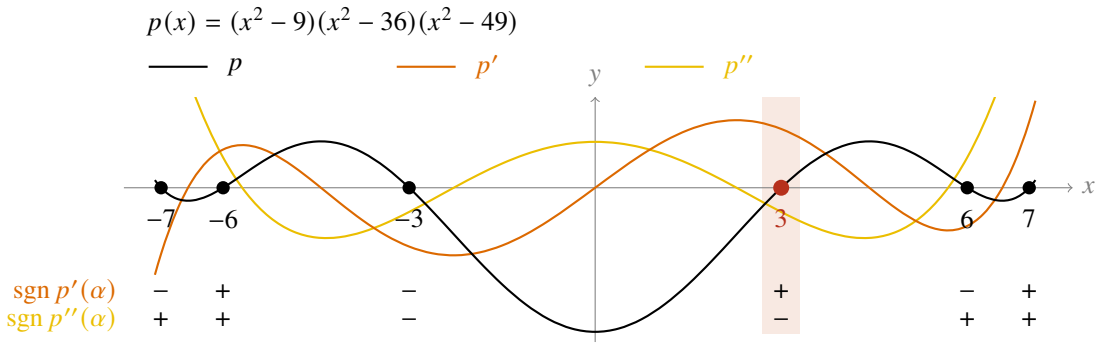


Figure 1: A fingerprint formed from the signs of the first two derivatives. Some roots share the same two signs, but there is a fingerprint $(+, -)$ that isolates the root 3.

Step 5: Verifying the fingerprint. For a guessed fingerprint, we construct a polynomial Ψ that is nonzero at every real root of p and negative exactly at the roots matching the fingerprint. By the Poisson formula, $\mathrm{Res}_x(p, \Psi + t)$ is the product of $\Psi(\alpha) + t$ over the roots α of p. Its lowest-degree nonzero coefficient is negative exactly when an odd number of real roots match the fingerprint: shared nonreal roots contribute powers of t, and all remaining nonreal conjugate pairs contribute positive factors. This parity test is established in Corollary 6.23.

If p has a real root, an isolating fingerprint selects exactly one real root, so some guess is accepted. Conversely, an accepted guess selects an odd number of real roots, and hence certifies that a real root

exists. The arithmetic and resultant constructions preserve compact coefficient access, so the polynomial-length fingerprint guess, and the quantification over projection polynomials remain in CH.

1.3 Background and AI use

In May 2026, the authors developed the initial proof strategy for the hardness results, based on the trace expansion of the tensor determinant. ChatGPT assisted in developing initial proofs of the algebraic and counting hardness results in Sections 3 and 4, and later helped us simplify and polish those proofs and write the text. In September 2026, ChatGPT obtained the sign-hardness result in Section 5 and a proof of $\exists\mathbb{R} \subseteq \mathsf{CH}$. The authors independently verified these results and subsequently developed the proof in Section 6 with ChatGPT's assistance. The authors take full responsibility for the results and their presentation.

2 Preliminaries

2.1 Notation

We collect some notation that will be used throughout the paper.

- The set $\{1, 2, \ldots, n\}$ is denoted by $[n]$.
- We use bold lowercase symbols for tuples, including Boolean assignments, and ordinary lowercase symbols for individual variables and input strings.
- For a tuple $\boldsymbol{e} = (e_1, \ldots, e_n)$, the quantity $|\boldsymbol{e}|$ denotes the sum $\sum_{i=1}^{n} e_i$. For a string $w \in \{0,1\}^*$, we use $|w|$ to denote its length.
- For a variable tuple $\boldsymbol{x} = (x_1, \ldots, x_n)$ and an exponent vector $\boldsymbol{e} = (e_1, \ldots, e_n)$, we use $\boldsymbol{x}^{\boldsymbol{e}}$ to denote the monomial $\prod_{i=1}^{n} x_i^{e_i}$. If $f = \sum_{\boldsymbol{e}} c_{\boldsymbol{e}} \boldsymbol{x}^{\boldsymbol{e}}$ is a polynomial, then $[\boldsymbol{x}^{\boldsymbol{e}}]f$ denotes the coefficient $c_{\boldsymbol{e}}$.

We first give a set of preliminaries shared by the entire paper. The individual sections introduce additional section-specific preliminaries.

2.2 Resultants

We begin by recalling the definition of the multivariate resultant, which is a classical object in algebraic geometry [GKZ94; CLO05].

Definition 2.1 (Multivariate Resultant)**.** Let $d_1, \ldots, d_n$ be positive integers, and for $i \in [n]$, let $f_i(\boldsymbol{x}) = \sum_{|\boldsymbol{e}|=d_i} c_{i,\boldsymbol{e}} \boldsymbol{x}^{\boldsymbol{e}}$ be a homogeneous polynomial of degree d_i in n variables $\boldsymbol{x} = (x_1, \ldots, x_n)$ over $\mathbb{C}$. Introduce the universal coefficient variables $\boldsymbol{u} = (u_{i,\boldsymbol{e}})$, one for each possible pair $(i, \boldsymbol{e})$. The multivariate resultant $\mathrm{Res} \in \mathbb{Z}[\boldsymbol{u}]$ is the unique integral polynomial that satisfies the following properties:

1. For every choice of coefficients $c_{i,\boldsymbol{e}} \in \mathbb{C}$, $\mathrm{Res}(f_1, \ldots, f_n) = 0$ if and only if $f_1(\boldsymbol{x}) = \cdots = f_n(\boldsymbol{x}) = 0$ has a nontrivial solution $\boldsymbol{x} \in \mathbb{C}^n \setminus \{0\}$. Here, the notation $\mathrm{Res}(f_1, \ldots, f_n)$ means the evaluation of Res by replacing each variable $u_{i,\boldsymbol{e}}$ with the corresponding coefficient $c_{i,\boldsymbol{e}}$ of f_i.
2. $\mathrm{Res}(x_1^{d_1}, \ldots, x_n^{d_n}) = 1$.
3. Res is irreducible, even when viewed as a polynomial in $\mathbb{C}[\boldsymbol{u}]$.

When we want to emphasize the degrees of the polynomials, we denote the resultant by $\mathrm{Res}_{d_1,\ldots,d_n}$.

The existence and uniqueness of the multivariate resultant is a nontrivial fact; see, for example, [GKZ94, Chapter 13]. The resultant is a *homogeneous* polynomial with the following degree bound [CLO05, Theorem 3.1]:

Lemma 2.2 (Degree of the Multivariate Resultant)**.** *For a fixed $i \in [n]$, $\mathrm{Res}_{d_1,\ldots,d_n}$ is homogeneous in the variables $u_{i,\boldsymbol{e}}$, $|\boldsymbol{e}| = d_i$, of degree $D_i = \prod_{j\in[n], j\neq i} d_j$. In particular, this implies that for any $\lambda \in \mathbb{C}$,*

$$\mathrm{Res}(f_1, \ldots, \lambda f_i, \ldots, f_n) = \lambda^{D_i} \mathrm{Res}(f_1, \ldots, f_i, \ldots, f_n).$$

Furthermore, the total degree of $\mathrm{Res}_{d_1,\dots,d_n}$ *is* $\sum_{i=1}^{n} D_i$. *In particular, the resultant of quadratic forms, denoted* $\mathrm{Res}_n^{(2)}$, *has total degree* $n2^{n-1}$.

We will usually use homogenized variants of polynomials.

Definition 2.3 (Homogenization)**.** Let $f \in \mathbb{C}[\boldsymbol{x}]$ be a polynomial in n variables with degree $\deg f \le d$. We write

$$f^h(\boldsymbol{x}, z) \coloneqq z^d f(x_1/z, \dots, x_n/z)$$

for its homogenization to degree d in the additional variable z. Unless otherwise specified, $d = \deg f$.

Remark 2.4. Throughout, the variables are ordered as they are listed. When homogenizing polynomials in $x_1, \dots, x_n$ with a new variable z, we use the order $(x_1, \dots, x_n, z)$. Together with the normalization in Definition 2.1, this fixes the sign convention for the resultant.

The univariate Poisson product formula [CLO05, Chapter 3] can be generalized to multivariate resultants. The version we state here is from [AGS26, Theorem 4.3].

Lemma 2.5 (Poisson formula)**.** *Let* $f_1, \dots, f_n, g \in \mathbb{C}[\boldsymbol{x}, z]$ *be homogeneous polynomials in* $n+1$ *variables of positive degrees* $d_1, \dots, d_n, e$, *respectively. Define*

$$F_i(\boldsymbol{x}) \coloneqq f_i(\boldsymbol{x}, 1) \quad (i \in [n]), \qquad G(\boldsymbol{x}) \coloneqq g(\boldsymbol{x}, 1).$$

Suppose that

$$\mathrm{Res}_{d_1,\dots,d_n}\big(f_1(\boldsymbol{x}, 0), \dots, f_n(\boldsymbol{x}, 0)\big) \neq 0.$$

Then the common zero set

$$Z \coloneqq \{\boldsymbol{\alpha} \in \mathbb{C}^n : F_1(\boldsymbol{\alpha}) = \cdots = F_n(\boldsymbol{\alpha}) = 0\}$$

is finite, and

$$\mathrm{Res}_{d_1,\dots,d_n,e}(f_1, \dots, f_n, g) = \mathrm{Res}_{d_1,\dots,d_n}\big(f_1(\boldsymbol{x}, 0), \dots, f_n(\boldsymbol{x}, 0)\big)^e \prod_{\boldsymbol{\alpha} \in Z} G(\boldsymbol{\alpha})^{m_{\boldsymbol{\alpha}}},$$

where $m_{\boldsymbol{\alpha}}$ *is the local multiplicity of* $\boldsymbol{\alpha}$ *in the system* $F_1 = \cdots = F_n = 0$.

We will now collect some more standard properties of the resultant for later use. For proofs, see, for example, [Dem12, Section 4] or [CD05, Section 1.6.1].

Lemma 2.6. *The normalized resultant* Res *has the following properties.*

1. (Multiplicativity). *It is multiplicative in each form:*

$$\mathrm{Res}(\dots, gh, \dots) = \mathrm{Res}(\dots, g, \dots)\,\mathrm{Res}(\dots, h, \dots),$$

   with the appropriate degrees.
2. (Linear case). *The resultant of linear forms is the determinant of their coefficient matrix. For any* $n \times n$ *matrix* $A = (a_{ij})$,

$$\mathrm{Res}_{1,\dots,1}\left(\sum_{j=1}^{n} a_{1j}x_j, \dots, \sum_{j=1}^{n} a_{nj}x_j\right) = \det A.$$

3. (Variable scaling). *Let* $f_1, \dots, f_n$ *be homogeneous forms of degrees* $d_1, \dots, d_n$, *respectively. For any* $n \times n$ *matrix* Q *and* $\boldsymbol{x} = (x_1, \dots, x_n)^T$,

$$\mathrm{Res}\big(f_1(Q\boldsymbol{x}), \dots, f_n(Q\boldsymbol{x})\big) = (\det Q)^{\prod_{j=1}^{n} d_j}\,\mathrm{Res}(f_1, \dots, f_n).$$

4. (Simultaneous permutation). *Let $f_1, \dots, f_n$ be homogeneous forms of equal degree. For any permutation $\sigma \in S_n$, write $\boldsymbol{x}_\sigma = (x_{\sigma(1)}, \dots, x_{\sigma(n)})$. Then*

$$\mathrm{Res}\big(f_{\sigma(1)}(\boldsymbol{x}_\sigma), \dots, f_{\sigma(n)}(\boldsymbol{x}_\sigma)\big) = \mathrm{Res}\big(f_1(\boldsymbol{x}), \dots, f_n(\boldsymbol{x})\big).$$

5. (Padding). *If $f_1, \dots, f_k$ are quadratic forms in $x_1, \dots, x_k$ and $n \geq k$, then*

$$\mathrm{Res}_n^{(2)}(f_1, \dots, f_k, x_{k+1}^2, \dots, x_n^2) = \mathrm{Res}_k^{(2)}(f_1, \dots, f_k)^{2^{n-k}}.$$

6. (Evaluation). *For a homogeneous form $h(\boldsymbol{x}, z)$ of positive degree and constants $\boldsymbol{c}$,*

$$\mathrm{Res}(x_1 + c_1 z, \dots, x_n + c_n z, h) = h(-c_1, \dots, -c_n, 1).$$

2.3 Tensor determinants

A cubic tensor of dimension n is an array $A = (a_{ijk})_{i,j,k \in [n]}$ with entries from $\mathbb{C}$. If the tensor is *generic*, we treat its entries a_{ijk} as independent variables. The tensor A is *symmetric* if its entries are unchanged by permuting the index positions, i.e., if $a_{ijk} = a_{i'j'k'}$ whenever i', j', k' are a permutation of i, j, k. Then it is specified by the entries a_{ijk} for $1 \leq i \leq j \leq k \leq n$, and all other entries are determined by symmetry.

Definition 2.7 (Tensor Determinant)**.** The tensor A defines the quadratic map

$$(A\boldsymbol{x}^2)_i = \sum_{j,k=1}^{n} a_{ijk} x_j x_k$$

whose tensor determinant Det_n is the resultant of these quadratic forms:

$$\mathrm{Det}_n(A) = \mathrm{Res}_n^{(2)}((A\boldsymbol{x}^2)_1, \dots, (A\boldsymbol{x}^2)_n). \tag{1}$$

The following lemma follows from standard properties of the resultant [Dem12, Section 4]. The degree of the tensor determinant also appears in [HHLQ13].

Lemma 2.8 (Degree, symmetry, and scaling)**.** *Let A be a cubic tensor of dimension n, and let $q = 2^{n-1}$ and $D = nq$. The polynomial Det_n is homogeneous of degree D and invariant under simultaneous permutation of its three index ranges: for every $\pi \in S_n$, the substitution $a_{ijk} \mapsto a_{\pi(i)\pi(j)\pi(k)}$ leaves Det_n unchanged. For arbitrary scalars $\lambda_1, \dots, \lambda_n$,*

$$\mathrm{Det}_n\big((\lambda_i \lambda_j \lambda_k a_{ijk})_{ijk}\big) = \left(\prod_{i=1}^{n} \lambda_i^{3q}\right) \mathrm{Det}_n(A). \tag{2}$$

Proof. By Lemma 2.2, the resultant is homogeneous of degree q in the coefficients of each of its n quadratic forms, giving total degree nq. Simultaneously permuting the tensor indices permutes the forms and their variable indices, so invariance follows from Lemma 2.6(4). For arbitrary λ_i, the transformed quadratic forms are $\lambda_i (A(Q\boldsymbol{x})^2)_i$, where Q is the diagonal matrix with $(\lambda_1, \dots, \lambda_n)$ on the diagonal. Multihomogeneity (Lemma 2.2) contributes $\prod_i \lambda_i^q$, and diagonal variable substitution contributes $\prod_i \lambda_i^{2^n}$, by Lemma 2.6(3). Their product is $\prod_i \lambda_i^{q+2^n} = \prod_i \lambda_i^{3q}$. □

We will formulate our algebraic results over the complex numbers $\mathbb{C}$. The Boolean complexity results will use the input representations as specified in their statements. In the real feasibility arguments, the polynomials have rational coefficients, and we distinguish their real and complex zeros.

Let ETR denote the language of true existential sentences over the reals, with a quantifier-free Boolean combination of polynomial comparisons and binary integer coefficients and exponents. Its polynomial-time many-one closure is the class $\exists\mathbb{R}$. Write $\mathsf{C}_0\mathsf{P} = \mathsf{P}$, $\mathsf{C}_{k+1}\mathsf{P} = \mathsf{P}^{\mathsf{P}^{\mathsf{C}_k\mathsf{P}}}$, and $\mathsf{CH} = \bigcup_k \mathsf{C}_k\mathsf{P}$. These classes are defined in the ordinary Turing model.

3 Algebraic hardness of the resultant

In this section, we will prove Theorem 1.4. We briefly recall the required notions from algebraic complexity theory. For a more detailed introduction, see [Bür00, Chapter 2].

3.1 Algebraic complexity

We will work over $\mathbb{C}$. An *arithmetic circuit* is a directed acyclic graph that has variables and complex constants as leaves. Each internal node (gate) adds or multiplies two earlier values, and intermediate values may be reused. The *size* of a circuit is the number of gates. A polynomial family $(p_n)_n$ is in VP if its number of variables, degree, and circuit size are polynomially bounded in its index n. It is in VNP if it admits (g_n) in VP and polynomially bounded $q(n)$ such that

$$p_n(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \{0,1\}^{q(n)}} g_n(\boldsymbol{x}, \boldsymbol{y}).$$

A family is in VQP if its number of variables and degree are polynomially bounded, while its circuit size is bounded by $n^{(\log n)^{O(1)}}$ (see [Bür00, Section 2.5]).

A polynomial f is a *projection* of g if substituting variables or field constants for the variables of g gives f. A family (f_n) is a *p-projection* of (g_n) if each f_n is a projection of some $g_{q(n)}$ with $q(n)$ polynomially bounded. An *arithmetic c-reduction* allows polynomial-size circuits to use gates evaluating members of a polynomial family, with polynomially bounded indices (see [Bür00, Section 5.4, Definitions 5.13 and 5.14]). A polynomial family is VNP-hard under a specified reduction if every VNP family has such a reduction to it. Similar to Boolean circuits, these conventions are nonuniform: circuits and their constants may depend on n, without an algorithm constructing them.

In analogy with the coefficients of the characteristic polynomial of matrices, we study the coefficients of z^s in the expansion of the tensor determinant $\mathrm{Det}_n(I + zA)$ as a polynomial in z. The coefficient of z^s will be called $\Phi_{n,s}(A)$ and is in turn a polynomial in the entries of A. It is homogeneous of degree s.

Definition 3.1. Let $n \in \mathbb{N}$, let $I_{ijk} = \delta_{i,j,k}$ be the diagonal unit tensor of dimension n and let A be a symmetric cubic tensor of dimension n. We define

$$\Phi_{n,s}(A) \coloneqq [z^s] \,\mathrm{Det}_n(I + zA). \tag{3}$$

Lemma 2.8 showed that Det_n is invariant under relabeling the indices by a permutation $\pi \in S_n$. Because the invariance must hold in each homogeneous component, it also applies to $\Phi_{n,s}$ for all $s \in \mathbb{N}$. We prove the following theorem in this section.

Theorem 3.2. *The family $\{\Phi_{5m^2,m^2}\}_{m\geq 3}$ on symmetric cubic tensors, with one independent variable for each unordered triple of indices, is* VNP*-hard under p-projections over $\mathbb{C}$. The projections from the permanent use only rational constants.*

3.2 Grid assignments

For $m \geq 3$, consider the toroidal grid on $\mathbb{Z}_m^2$, with one triangle $\{(i, j), (i + 1, j), (i, j + 1)\}$ anchored at each position (i, j). All coordinates are taken modulo m. Each position belongs to three triangles: the one anchored there and those at its two predecessors, which gives m^2 positions and m^2 triangles. The grid is clearly connected.

We endow every position $(i, j) \in \mathbb{Z}_m^2$ with a nonempty index class Ω_{ij}; the classes are pairwise disjoint and of total size n. A symmetric tensor A is *supported on the grid triangles* if its only possibly nonzero entries, up to permutation of their indices, are

$$a_{\alpha\beta\gamma}, \qquad \alpha \in \Omega_{ij}, \quad \beta \in \Omega_{i+1,j}, \quad \gamma \in \Omega_{i,j+1}.$$

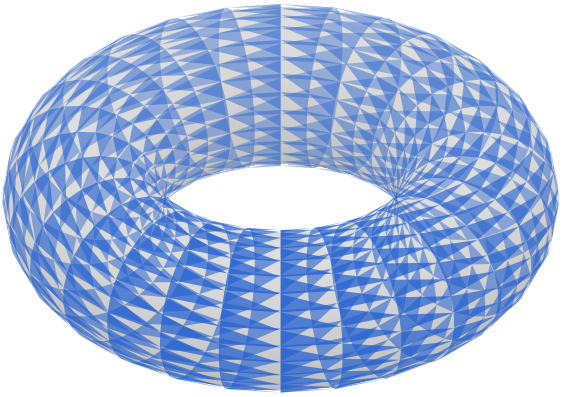


Figure 2: The 28×28 toroidal grid, where $28 = 3^3 + 1$, with 784 positions and 784 triangles. An assignment chooses one tensor index at each position, and each blue triangle contributes one tensor entry to its product. Each position belongs to three triangles.

The corresponding grid assignment sum is

$$H_m(A) := \sum_{\nu_{ij} \in \Omega_{ij}} \prod_{(i,j) \in \mathbb{Z}_m^2} a_{\nu_{ij}, \nu_{i+1,j}, \nu_{i,j+1}}. \tag{4}$$

That is, each summand chooses one index at each position and uses one tensor entry for each triangle. A standard grid encoding and padding argument in Appendix A.1 show the following.

Lemma 3.3 (Permanent as a grid assignment sum)**.** *Given an $n \times n$ matrix $Z = (z_{ij})$, one can construct in polynomial time a symmetric tensor $A_m(Z)$ supported on the grid triangles of side $m = m(n) = 3^r + 1 = O(n)$, with $r \geq 1$, five indices per class, and entries in $\{0, 1\} \cup \{z_{ij} : i, j \in [n]\}$, such that*

$$H_m(A_m(Z)) = \operatorname{per}(Z). \tag{5}$$

The construction preserves the count on $0/1$ inputs.

3.3 Coefficients of grid assignments

Lemma 3.4 (Index occurrences)**.** *In every monomial of $\Phi_{n,s}$, each tensor index occurs a multiple of three times, counting occurrences within each factor with multiplicity. In particular, each monomial uses at most s distinct indices.*

Proof. Take $u_i^3 = 1$ in Equation (2). This fixes I and leaves $\Phi_{n,s}$ invariant under $a_{ij\ell} \mapsto u_i u_j u_\ell a_{ij\ell}$. The exponent of u_i counts the occurrences of i, so this number is divisible by three. There are $3s$ occurrences in total, giving the claimed bound. □

Theorem 3.5 (Nonvanishing of assignment coefficients)**.** *Let $m \geq 3$, put $s = m^2$, and let the index classes defining H_m have total size n. Every monomial of H_m has the same coefficient $\kappa_{n,m} \in \mathbb{Z} \setminus \{0\}$ in $\Phi_{n,s}$. This coefficient depends only on n and m.*

Proof. Distinct assignments in H_m give distinct monomials, since the index classes of each factor identify its triangle. Any two assignment monomials differ only by a permutation of the tensor indices, so by the permutation invariance of $\Phi_{n,s}$ noted before Theorem 3.2, their coefficients in $\Phi_{n,s}$ agree and depend only on n and m. Choose one assignment and relabel its s indices by the grid positions. Its monomial is

$$M = \prod_{(i,j) \in \mathbb{Z}_m^2} a_{(i,j),\,(i+1,j),\,(i,j+1)}.$$

Number these indices by $[s]$. It suffices to show that $[M] \operatorname{Det}_n(I - A) < 0$, since replacing A by $-A$ multiplies every degree-s coefficient by $(-1)^s$.

Introduce independent coefficients $q_i(\boldsymbol{x}) = \sum_{j\le\ell} q_{i;j\ell}x_jx_\ell$ and put $\mathcal{R}(\boldsymbol{q}) = \mathrm{Res}_n^{(2)}(x_i^2 - q_i)_{i=1}^n$. Substitution $x_i \mapsto u_ix_i$ followed by division of equation i by u_i^2 preserves this resultant, by Lemma 2.6(3) and multihomogeneity. Thus, in every occurring coefficient monomial, using a_i factors from equation i forces $2a_i$ occurrences of i among the quadratic indices.

Assigning factors to equations. Symmetrize by setting $q_{i;jj} = a_{ijj}$ and $q_{i;j\ell} = 2a_{ij\ell}$ for $j < \ell$, obtaining $\mathrm{Det}_n(I - A)$. A coefficient monomial then contains $3a_i$ occurrences of index i. Each grid index occurs exactly three times, so a contribution to M uses one coefficient from each grid equation and none from the remaining equations. The possible contributions are therefore

$$Q = \prod_{v=1}^{s} q_{v;j_v\ell_v}, \qquad \prod_{v=1}^{s} a_{vj_v\ell_v} = M.$$

Such an assignment exists explicitly: assign the factor $a_{(i,j),(i+1,j),(i,j+1)}$ to the equation at its anchor (i, j). Every grid equation receives exactly one factor. Since each grid factor has three distinct indices, symmetrization multiplies every contribution by exactly 2^s.

Every contribution is negative. Fix one of these Q and specialize its coefficients to t_v, all others to zero:

$$P_Q(\boldsymbol{t}) = \mathrm{Res}_n^{(2)}\big((x_v^2 - t_vx_{j_v}x_{\ell_v})_{v=1}^s, x_{s+1}^2, \dots, x_n^2\big).$$

The arcs $v \to j_v, \ell_v$ form a connected directed graph with indegree and outdegree two: each grid factor connects its three indices, and each index occurs once as an equation index and twice as a quadratic index. If $P_Q(\boldsymbol{t}) = 0$, take a common nonzero root. Its inactive coordinates are zero. No arc leaves its support, since $x_v \neq 0$ forces $x_{j_v}, x_{\ell_v} \neq 0$. Degree balance gives no entering arcs either, so connectedness forces all grid coordinates to be nonzero. Multiplying their equations gives $\prod_v t_v = 1$.

Put $B(\boldsymbol{t}) = 1 - \prod_v t_v$. We have $V(P_Q) \subseteq V(B)$, so Hilbert's Nullstellensatz implies that P_Q divides a power of B. Since B is irreducible and $P_Q(0) = 1$, there is an integer $c \ge 0$ such that

$$P_Q(\boldsymbol{t}) = \left(1 - \prod_v t_v\right)^c.$$

Here $c \ge 1$: setting all grid coordinates and weights to one, and all inactive coordinates to zero, gives a common nonzero root. Consequently $[Q]\mathcal{R} = [t_1 \cdots t_s]P_Q = -c < 0$. The anchor assignment supplies at least one such contribution, and all have the same negative sign after multiplication by 2^s. □

Lemma 3.6 (Constant grid labelings)**.** *Let $m = 3^r + 1$ with $r \ge 1$. If $b : \mathbb{Z}_m^2 \to \mathbb{F}_3$ satisfies*

$$b_{ij} + b_{i-1,j} + b_{i,j-1} = 0 \qquad \text{for all } (i, j) \in \mathbb{Z}_m^2, \tag{6}$$

then b is constant.

Proof. For $h \ge 1$, write

$$E_h(i, j) := b_{ij} + b_{i-h,j} + b_{i,j-h}.$$

At spacing h, the signed sum of the six triangle equations in Figure 3 is $E_{3h}(i, j) - 3b_{i-h,j-h} = E_{3h}(i, j)$ over $\mathbb{F}_3$. Thus $E_h \equiv 0$ implies $E_{3h} \equiv 0$. Since $E_1 \equiv 0$, induction gives $E_q \equiv 0$ for $q = 3^r = m - 1$.

Fix j and put $x_i = b_{ij}$ and $z_i = b_{i,j-q}$. As $q \equiv -1 \pmod m$, the equation $E_q(i, j) = 0$ gives

$$z_i = -x_i - x_{i-q} = -x_i - x_{i+1}.$$

Moreover, $j - q \equiv j + 1 \pmod m$, so $E_1(i, j + 1) = 0$ gives

$$x_i = -z_i - z_{i-1} = 2x_i + x_{i-1} + x_{i+1}.$$

Thus $x_{i-1} + x_i + x_{i+1} = 0$. Subtracting consecutive equations gives $x_{i+3} = x_i$, and hence $x_i = x_{i+m} = x_{i+1}$, since $m = 3^r + 1$. The row is therefore constant. If its value is a, Equation (6) makes every entry of the row below it $-2a = a$. Thus b is constant. □

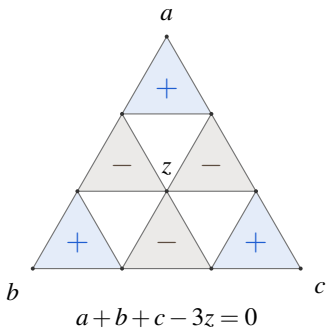


Figure 3: Each marked triangle represents a zero-sum equation. Add the three corner equations and subtract the other three. The intermediate labels cancel except for the center, leaving $a + b + c - 3z = 0$. Over $\mathbb{F}_3$, the outer corners therefore sum to zero.

Lemma 3.7 (Direct extraction of the grid assignment sum)**.** *Let $m = 3^r + 1$ with $r \geq 1$, put $s = m^2$, and let A be a symmetric tensor of dimension n supported on the grid triangles. Then*

$$\Phi_{n,s}(A) = \kappa_{n,m} H_m(A), \tag{7}$$

where $\kappa_{n,m} \neq 0$ is the coefficient from Theorem 3.5.

Proof. Consider a surviving degree-s monomial, and let b_{ij} count its factors on the triangle anchored at (i, j). The total number of occurrences of indices from Ω_{ij} is $b_{ij} + b_{i-1,j} + b_{i,j-1}$. By Lemma 3.4, this number is divisible by three. Lemma 3.6 therefore makes all b_{ij} equal modulo three. Their sum is $s = m^2 \equiv 1 \pmod 3$, so their common residue is one. Each b_{ij} is consequently at least one; there are s of them and they sum to s, so every $b_{ij} = 1$.

Each index class now has exactly three occurrences. Again by Lemma 3.4, all three use the same individual tensor index. The surviving monomials are thus precisely the assignment monomials in $H_m(A)$. By Theorem 3.5, each has coefficient $\kappa_{n,m} \neq 0$, proving the identity. □

3.4 Hardness of the coefficient family

Proof of Theorem 3.2. For an $n \times n$ matrix Z, choose the smallest $m = 3^r + 1 > n + 1$ with $r \geq 1$, as in Lemma 3.3, and put $s = m^2$ and $N = 5m^2$. Write $\kappa = \kappa_{N,m} \neq 0$ for the coefficient from Theorem 3.5. Set $Z' = \operatorname{diag}(Z, \kappa^{-1})$. By Lemmas 3.3 and 3.7,

$$\Phi_{N,s}(A_m(Z')) = \kappa H_m(A_m(Z')) = \kappa \operatorname{per}(Z') = \operatorname{per}(Z).$$

Each entry is a variable or a rational constant, since $\kappa \neq 0$ is integer. Also $m = O(n)$, so dimensions and degrees are polynomially bounded. Valiant's permanent theorem [Val79] proves the claim. □

Through homogenization, we obtain Theorem 1.4.

Proof of Theorem 1.4. By Strassen's homogenization theorem [SY10, Theorem 2.2], a size-$S(N)$ circuit for Det_N gives a circuit of size $O(s^2(S(N)+N))$ for the degree-s homogeneous component of $\operatorname{Det}_N(I+A)$, namely $\Phi_{N,s}(A)$. Taking $N = 5m^2$ and $s = m^2$, polynomial-size circuits for Det_N would therefore put the VNP-hard family of Theorem 3.2 in VP, implying VP = VNP.

The resultant assertion follows because Det_N is a linear specialization of $\operatorname{Res}_N^{(2)}$, obtained from the quadratic forms $(A\boldsymbol{x}^2)_1, \ldots, (A\boldsymbol{x}^2)_N$. The same polynomial overhead gives VNP $\subseteq$ VQP under the quasipolynomial-size hypothesis. □

3.5 Discriminants of cubic forms

The tensor determinant of generic symmetric tensors is closely related to another classical object in algebraic geometry, the *discriminant*.

Definition 3.8 (Cubic Discriminant)**.** Let $f = \sum_{|\boldsymbol{e}|=3} c_{\boldsymbol{e}} \boldsymbol{x}^{\boldsymbol{e}} \in \mathbb{C}[\boldsymbol{x}]$ be a cubic form in the variables $\boldsymbol{x} = (x_1, \ldots, x_n)$. The discriminant of f, denoted $\Delta_n(f)$, is defined by

$$\Delta_n(f) \coloneqq 3^{-a_n} \operatorname{Res}_n^{(2)}\left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right), \qquad \text{where } a_n = \frac{2^n - (-1)^n}{3}.$$

When the coefficients $c_{\boldsymbol{e}}$ are treated as independent indeterminates, the same formula defines the *generic cubic discriminant* $\Delta_n \in \mathbb{C}[c_{\boldsymbol{e}} : |\boldsymbol{e}| = 3]$.

Remark 3.9. Our normalization agrees up to sign with that of Gelfand, Kapranov, and Zelevinsky [GKZ94, Chapter 13, Proposition 1.7]; the preceding definition fixes the sign.

For a generic symmetric cubic tensor A, define its associated cubic form by $f_A(\boldsymbol{x}) = \sum_{i,j,k=1}^{n} a_{ijk} x_i x_j x_k$. Collecting equal monomials, we may write

$$f_A = \sum_{i=1}^{n} a_{iii} x_i^3 + 3 \sum_{1 \le i < j \le n} \left(a_{iij} x_i^2 x_j + a_{ijj} x_i x_j^2\right) + 6 \sum_{1 \le i < j < k \le n} a_{ijk} x_i x_j x_k.$$

Thus, the coefficients of the associated cubic form in the monomial basis are not quite the entries of the symmetric tensor, but are scaled by natural multiplicity factors. The discriminant of this cubic form and the tensor determinant of A differ only by an explicit scalar.

Lemma 3.10. *Let A be a* symmetric *cubic tensor of dimension n. Then we have the following polynomial identity in the independent entries of A, where $D_n = n2^{n-1}$ and $a_n = (2^n - (-1)^n)/3$:*

$$\Delta_n(f_A) = 3^{D_n - a_n} \operatorname{Det}_n(A).$$

Proof. Note that since A is symmetric, it is easy to verify that for every $i \in [n]$, we have

$$\frac{\partial f_A}{\partial x_i} = 3(A\boldsymbol{x}^2)_i.$$

Plugging this into Definition 3.8, we get

$$\Delta_n(f_A) = 3^{-a_n} \operatorname{Res}\left(3(A\boldsymbol{x}^2)_1, \ldots, 3(A\boldsymbol{x}^2)_n\right) = 3^{D_n - a_n} \operatorname{Det}_n(A),$$

where the second equality follows from the multihomogeneity of the resultant (Lemma 2.2) and the definition of the tensor determinant (Definition 2.7). □

We immediately get the hardness of the discriminant.

Proof of Corollary 1.6. Let f_A be the cubic form corresponding to a symmetric cubic tensor of dimension n. If the polynomial family $\{\Delta_n(f_A)\}_{n \ge 1}$ has $\mathsf{poly}(n)$ sized circuits, then by Lemma 3.10, $\operatorname{Det}_n(A) = 3^{-(D_n - a_n)} \Delta_n(f_A)$ also has $\mathsf{poly}(n)$ sized circuits. Now use Theorem 1.4. □

4 Modular evaluation of the resultant

In counting complexity, a function is in #P if it counts accepting paths of a polynomial-time nondeterministic Turing machine. A *polynomial-time Turing reduction* is a polynomial-time algorithm that computes one function using an oracle for another. A polynomial-time *counting many-one reduction* from f to g consists of polynomial-time computable functions u, v such that

$$f(x) = v\big(x, g(u(x))\big).$$

It uses one oracle answer and allows polynomial-time post-processing. A function is #P-*hard* under a specified reduction if every #P function admits such a reduction to it; unless specified otherwise, we use Turing reductions. For more on counting complexity, see [AB09, Chapter 17].

Permanent source problems. The permanent of 0/1 matrices is #P-complete under counting many-one reductions [BH93, Theorem 1]. We also use the following threshold consequence.

Lemma 4.1. *Given an integer $n \times n$ matrix Z and an integer threshold K, both encoded in binary, deciding whether* $\operatorname{per}(Z) > K$ *is* PP*-hard under polynomial-time many-one reductions.*

Proof. Ben-Dor and Halevi's construction [BH93, Lemma 2] maps a 3-CNF formula φ with q clauses to an integer matrix Z satisfying $\operatorname{per}(Z) = 12^q \#\mathrm{SAT}(\varphi)$. The factor 12^q is positive and polynomial-time computable, so it also scales the threshold. Threshold counting for 3-CNF formulas is PP-hard by the parsimonious Cook–Levin reduction with the threshold retained. □

The grid construction from Section 3 gives a coefficient that counts perfect matchings up to a nonzero factor. We encode both the count and the factor in one coefficient, then recover it from a single modular determinant evaluation.

4.1 Dyadic determinant evaluations

We consider the problem of computing

$$\operatorname{Det}_n(C) \bmod 2^k \in \{0, \dots, 2^k - 1\},$$

where C is an integer symmetric cubic tensor of dimension n with binary-encoded entries, and $k \geq 1$ is supplied in unary. Equivalently, the modulus 2^k is supplied in ordinary binary notation. The oracle answer has at most k bits. Evaluating at a sufficiently large power of two lets us recover the low-degree coefficients as signed digits. The following bound gives the precision needed for this recovery.

Lemma 4.2 (Low-degree coefficient bound)**.** *Let B be an integer symmetric cubic tensor of dimension M, put $U = \max(1, \max_{i,j,k} |b_{ijk}|)$, and let $\ell \geq 0$ be an integer. For $0 \leq s \leq \ell$,*

$$|\Phi_{M,s}(B)| \leq C(M, \ell, U) := (M^3 2^{M-1} U)^{\ell}.$$

The bound is computable in time polynomial in $M, \ell, \log U$ and has bit length $O(1 + \ell(M + \log U))$.

Proof. By homogeneity and normalization, $\operatorname{Det}_M(\lambda I - B)$ is monic of degree $D = M2^{M-1}$. At any root λ, the resultant criterion gives $\boldsymbol{x} \neq 0$ with $\lambda x_i^2 = (B\boldsymbol{x}^2)_i$ for all i. Choose i with $|x_i| = \max_j |x_j| > 0$. Its equation gives $|\lambda|\,|x_i|^2 \leq M^2 U |x_i|^2$, hence $|\lambda| \leq M^2 U$. Writing the roots with multiplicity as $\lambda_1, \dots, \lambda_D$, homogeneity gives

$$\operatorname{Det}_M(I + tB) = \prod_{j=1}^{D} (1 + t\lambda_j).$$

Thus, for $1 \leq s \leq D$, the coefficient has absolute value at most $\binom{D}{s}(M^2U)^s \leq (M^3 2^{M-1} U)^s$. The constant coefficient is one and coefficients above D vanish, which proves the uniform bound. Its binary logarithm is $\ell(3 \log_2 M + M - 1 + \log_2 U)$, proving the computational claims. □

Corollary 4.3 (Coefficients on 0/1 tensors)**.** *Given a symmetric 0/1 cubic tensor A of dimension n and an integer $1 \leq s \leq n$, computing $\Phi_{n,s}(A)$ is* #P*-hard under polynomial-time Turing reductions.*

Proof. For a 0/1 matrix Z of size $n \times n$, let $m = m(n)$ be as in Lemma 3.3, and put $N = 5m^2$ and $s = m^2$. Query $\Phi_{N,s}$ on $A_m(Z)$ and $A_m(I_n)$. Both tensors are symmetric and 0/1, and Lemma 3.7 gives the answers $\kappa_{N,m} \operatorname{per}(Z)$ and $\kappa_{N,m}$, respectively. Their bit lengths are polynomial in $N + s$ by Lemma 4.2. Exact integer division recovers $\operatorname{per}(Z)$, proving the claim. □

The following lemma will be required in the remaining proof; it is shown in Appendix B.1 by a simple padding argument.

Lemma 4.4 (Affine grid encoding)**.** *Given an integer $n \times n$ matrix Z and integers α, β, all encoded in binary, one can construct in polynomial time a symmetric integer tensor C supported on a grid of side $m = m(n)$ as in Lemma 3.3, with dimension $6m^2$, such that $H_m(C) = \alpha \operatorname{per}(Z) + \beta$.*

Lemma 4.5 (Exact coefficient evaluation)**.** *Given a symmetric integer cubic tensor C of dimension N with binary-encoded entries and an integer $1 \le s \le N$, computing $\Phi_{N,s}(C)$ is $\#\mathsf{P}$-hard under polynomial-time counting many-one reductions.*

Proof. Given an $n \times n$ 0/1 matrix Z, choose $m = m(n) = 3^r + 1 = O(n)$ using Lemma 3.3, and put $s = m^2$ and $N = 6s$. Write $\kappa = \kappa_{N,m} \neq 0$. Consider a 0/1 tensor $C^\star$ of dimension N that allows exactly one prescribed index at each grid position, with entry one on the resulting triangles and zero elsewhere. Then $H_m(C^\star) = 1$, so Lemmas 3.7 and 4.2 give

$$0 < |\kappa| = |\Phi_{N,s}(C^\star)| \le B_0 := C(N, s, 1).$$

Set $L = B_0 + 1$. This bound and L are computable from N, s in polynomial time.

Apply Lemma 4.4 with $(\alpha, \beta) = (L, 1)$. The resulting tensor C has dimension N and entries in $\{0, 1, L\}$. By Lemma 3.7,

$$H_m(C) = L \operatorname{per}(Z) + 1, \qquad V := \Phi_{N,s}(C) = \kappa\big(L \operatorname{per}(Z) + 1\big).$$

Since $\operatorname{per}(Z) \ge 0$ and $0 < |\kappa| < L$, the single answer V can be decoded by

$$W = |V|, \qquad K = W \bmod L = |\kappa|, \qquad \operatorname{per}(Z) = \frac{W/K - 1}{L}. \tag{8}$$

Both divisions are exact, and $K > 0$. The bit length of L is polynomial in N, s. By Lemma 4.2, $|V| \le C(N, s, L)$, so the oracle answer and the decoding also have polynomial size. This gives the required counting many-one reduction from 0/1 permanent. □

The coefficient bound ensures polynomial output length when $s \le N$. The full tensor determinant can have exponentially many bits; the modular problem below keeps the output length polynomial.

Proof of Theorem 1.8 (modular computation). Given a 0/1 matrix Z, construct C, N, s, L as in Lemma 4.5. Put $B_1 = C(N, s, L)$ from Lemma 4.2, choose a power of two $T = 2^b > 4B_1$ with $b \ge 2$, and set $k = b(s + 1)$. One query gives

$$R = \operatorname{Det}_N(I + TC) \bmod 2^k = \operatorname{Det}_N(I + TC) \bmod T^{s+1}.$$

Write $v_j = \Phi_{N,j}(C)$. Since $|v_j| \le B_1 < T/2$ for $0 \le j \le s$, recover $v_0, \ldots, v_s$ in order as signed base-T digits. Starting with $R_0 = R$, take v_j to be the unique representative of $R_j \bmod T$ in $(-T/2, T/2)$ and set $R_{j+1} = (R_j - v_j)/T$. Inductively,

$$R_j \equiv \sum_{i=j}^{s} v_i T^{i-j} \pmod{T^{s+1-j}},$$

so every recovered digit is correct. Terms above degree s disappear modulo T^{s+1}. Apply Equation (8) to $V = v_s$ to recover $\operatorname{per}(Z)$. Since $\log L$ is polynomially bounded, Lemma 4.2 gives $b = O(1 + s(N + \log L))$ and $k = b(s + 1)$, so both b and k are polynomially bounded. The query entries and all recovered integers have polynomial bit length. Thus the unary exponent k, the query construction, and all subsequent arithmetic have polynomial size, establishing the single-query reduction.

The resultant case follows by writing

$$\operatorname{Det}_N(A) = \operatorname{Res}_N^{(2)}((A\boldsymbol{x}^2)_1, \ldots, (A\boldsymbol{x}^2)_N).$$

For symmetric A, the quadratic coefficients are a_{ijj} on x_j^2 and $2a_{ijk}$ on $x_j x_k$ for $j < k$. This specialization is computable in polynomial time, preserves integrality, and uses the same modulus 2^k. □

Theorem 4.6 (Dyadic modular evaluation)**.** *For explicitly encoded integer symmetric cubic tensors A, computing* $\mathrm{Det}_n(A) \bmod 2^k$ *is* $\mathsf{FP}^{\mathsf{PP}}$*-complete under polynomial-time Turing reductions. Here $k \geq 1$ is supplied in unary, or equivalently the modulus 2^k is supplied in ordinary binary notation.*

Proof. For the lower bound, Theorem 1.8 gives #P-hardness under counting many-one reductions. Since $\mathsf{FP}^{\#\mathsf{P}} = \mathsf{FP}^{\mathsf{PP}}$, polynomially many such queries also simulate every $\mathsf{FP}^{\mathsf{PP}}$ computation. For the upper bound, apply Lemma B.2 with $p = 2$ and $d = 2$ to the associated quadratic forms. For symmetric cubic tensors, every mixed-monomial coefficient is even, so the coefficient condition is automatic. □

4.2 Addressed bits

We show that a single addressed output bit can already hold a PP-hard threshold test. For an integer cubic tensor A, define

$$\mathrm{Bit}(A, j) := \left\lfloor \frac{|\mathrm{Det}_n(A)|}{2^j} \right\rfloor \bmod 2, \qquad j \geq 0.$$

Proof of Theorem 1.8 (individual bit). We reduce from the integer permanent threshold problem in Lemma 4.1. Apply Lemma 4.4 to Z with $(\alpha, \beta) = (2, -(2K + 1))$, obtaining C_0 of dimension $N_0 = 6s$, where $s = m^2$ and $m = 3^r + 1$. Then

$$H_m(C_0) = 2\,\mathrm{per}(Z) - 2K - 1 =: h.$$

The integer h is odd, and $h > 0$ exactly when $\mathrm{per}(Z) > K$.

Append one unused index, setting $C = C_0 \oplus 0$ and $N = N_0 + 1$. Include the unused index in any one grid class; its entries are zero, so $H_m(C) = h$. Write

$$P(t) = \mathrm{Det}_N(I + tC) = \sum_i p_i t^i.$$

By Lemma 3.7, $p_s = \kappa_{N,m} h$. Since m and s are even, the proof of Theorem 3.5 gives $\kappa_{N,m} < 0$. Consequently $p_s \neq 0$, and $p_s < 0$ exactly when $\mathrm{per}(Z) > K$.

Put $U = \max(1, \max_{i,j,k} |c_{ijk}|)$ and $B = C(N, s, U)$ from Lemma 4.2. Choose $T = 2^b > 4B$ and set $k = b(s + 1)$ and $j = k - 1$. The padding identity in Lemma 2.6(5) gives

$$W := P(T) = \mathrm{Det}_{N_0}(I + TC_0)^2 > 0.$$

Strict positivity holds because the integer being squared is congruent to one modulo T. Thus the absolute value in the definition of $\mathrm{Bit}(I + TC, j)$ has no effect.

Let $S = \sum_{i=0}^{s} p_i T^i$. Since $|p_i| \leq B < T/4$ for $i \leq s$,

$$\left| \sum_{i=0}^{s-1} p_i T^i \right| \leq B \frac{T^s - 1}{T - 1} < \frac{T^s}{2}, \qquad |S| \leq B \frac{T^{s+1} - 1}{T - 1} < \frac{T^{s+1}}{2}.$$

As p_s is a nonzero integer, S has the sign of p_s. Moreover, $W \equiv S \pmod{T^{s+1}}$, and $T^{s+1} = 2^k$. The bit at position $j = k - 1$ is therefore one exactly when $S < 0$:

$$\mathrm{Bit}(I + TC, j) = 1 \iff p_s < 0 \iff \mathrm{per}(Z) > K.$$

All tensor entries have polynomial bit length, and $\log B = O(1 + s(N + \log U))$. Hence b, k, and the unary address j are polynomially bounded, and the tensor $I + TC$ is constructible in polynomial time. This is a many-one reduction to a single addressed bit. □

5 Hardness of resultant signs

In this section, we show that computing the *sign* of the resultant is $\oplus\mathsf{P}$-hard. The class $\oplus\mathsf{P}$ consists of decision problems for which a polynomial-time verifier accepts an odd number of polynomial-length witnesses exactly on the YES-instances [AB09, Chapter 17]. The canonical complete problem is $\oplus$SAT, where given a Boolean formula, we are asked to decide if it has an odd number of satisfying assignments. To prove our hardness result, we instead reduce from a variant of the subset sum problem, namely counting the parity of subsets summing up to *at most* a given target value T.

Lemma 5.1. *For positive integer weights $\boldsymbol{w} = (w_1, \ldots, w_n)$ and a nonnegative threshold T, let $K_{\boldsymbol{w}}(T)$ count the subsets whose sum is at most T. Deciding whether $K_{\boldsymbol{w}}(T)$ is even is $\oplus\mathsf{P}$-hard under polynomial-time many-one reductions.*

The standard NP-completeness reduction for subset sum with exact target value $= T$ is parsimonious, which directly yields $\oplus\mathsf{P}$-completeness, and the modification to target $\le T$ is elementary. We provide a self-contained proof in Appendix C.

5.1 Encoding parities algebraically

We first describe a more general parity encoding procedure that will also be used in our proof of $\exists\mathbb{R} \subseteq \mathsf{CH}$ (Section 6). The high-level idea is very simple: if each object in a finite collection is encoded as a nonzero real number which is negative exactly when the object satisfies a given condition, then the sign of the product of the object encodings gives the parity of the number of objects that satisfy the condition.

To implement the idea algebraically, we will represent the objects by real points in an *algebraic set* Z, formed by the zeros of a polynomial system. Suppose that we can construct a polynomial Ψ that is negative on the set of real points that we want to count (the feasible set) and positive at all other real points of Z. The lemma below shows how to then compute the parity of the feasible set using the sign of a (coefficient in a) resultant.

Lemma 5.2 (Parity from a resultant sign)**.** *Let $n, d \ge 1$ and $F_1, \ldots, F_n \in \mathbb{R}[\boldsymbol{x}]$ be polynomials in n variables of the form*

$$F_i = \lambda_i x_i^d + \text{terms of total degree less than } d, \qquad \lambda_i > 0.$$

Suppose that their common complex zeros are simple, and we denote this set of zeros by Z. Let $\Psi \in \mathbb{R}[\boldsymbol{x}]$ be nonzero at every real point of Z. For an integer $e \ge \max\{1, \deg \Psi\}$, homogenize Ψ to degree e, and the F_i's to degree d using a new variable z with variable order $(\boldsymbol{x}, z)$. Define

$$\mathcal{R}(t) := \operatorname{Res}\big(F_1^h, \ldots, F_n^h, \Psi^h + tz^e\big),$$

where t is a parameter. Then $\mathcal{R}$ is nonzero, and if ν is the least index for which the coefficient of t^ν is nonzero,

$$\operatorname{sgn}\big([t^\nu]\mathcal{R}\big) = (-1)^{\#\{\alpha \in Z \cap \mathbb{R}^n : \Psi(\alpha) < 0\}}.$$

Proof. Since $F_i^h(\boldsymbol{x}, 0) = \lambda_i x_i^d$, multihomogeneity (Lemma 2.2) and normalization (Definition 2.1) give

$$\operatorname{Res}\big(F_1^h(\boldsymbol{x}, 0), \ldots, F_n^h(\boldsymbol{x}, 0)\big) = \prod_{i=1}^{n} \lambda_i^{d^{n-1}} > 0.$$

By the Poisson formula (Lemma 2.5), Z is finite and, since every common zero is simple,

$$\mathcal{R}(t) = C \prod_{\alpha \in Z} \big(\Psi(\alpha) + t\big), \qquad C = \left(\prod_{i=1}^{n} \lambda_i^{d^{n-1}}\right)^e > 0.$$

The first nonzero coefficient is therefore

$$[t^\nu]\mathcal{R} = C \prod_{\substack{\boldsymbol{\alpha}\in Z\\ \Psi(\boldsymbol{\alpha})\neq 0}} \Psi(\boldsymbol{\alpha}),$$

where ν counts the number of omitted factors due to $\Psi(\boldsymbol{\alpha})$ being zero. No real points of Z are omitted. Since the F_i's and Ψ have real coefficients, the nonreal points in Z do not affect the sign as they occur in conjugate pairs $\boldsymbol{\alpha}, \overline{\boldsymbol{\alpha}}$, and contribute $\Psi(\boldsymbol{\alpha})\Psi(\overline{\boldsymbol{\alpha}}) = |\Psi(\boldsymbol{\alpha})|^2 > 0$ to the product. Since the prefactor C is also positive, the negative factors therefore count exactly the real points with $\Psi(\boldsymbol{\alpha}) < 0$, as needed. □

If Ψ does not vanish even on the nonreal points in Z, we have $\nu = 0$ in the above proof, and we do not need a perturbation. We record this as a useful corollary.

Corollary 5.3 (The nonvanishing case)**.** *Under the hypotheses of Lemma 5.2, suppose further that Ψ is nonzero at every point of Z. Then $\mathcal{R}(0) = \operatorname{Res}(F_1^h, \dots, F_n^h, \Psi^h)$ is nonzero and*

$$\operatorname{sgn}\mathcal{R}(0) = (-1)^{\#\{\boldsymbol{\alpha}\in Z\cap\mathbb{R}^n : \Psi(\boldsymbol{\alpha})<0\}}.$$

In particular, this applies when all points of Z are real.

We will now construct the polynomials $F_1, \dots, F_n$ and Ψ in the context of the subset sum problem to get our hardness result.

Proof of Theorem 1.11. Let $w_1, \dots, w_n$ be positive integer weights and let $T \geq 0$ be an integer threshold. We call a set $S \subseteq [n]$ *feasible* if $\sum_{i\in S} w_i \leq T$ and write $K_{\boldsymbol{w}}(T)$ for the number of feasible sets. We reduce from deciding whether $K_{\boldsymbol{w}}(T)$ is even, which is $\oplus$P-hard by Lemma 5.1.

Define the affine polynomials

$$F_i(\boldsymbol{x}) = 4w_i x_i(x_i + 1) \quad (i \in [n]), \qquad \Psi(\boldsymbol{x}) = 2\sum_{i=1}^n w_i x_i^2 - (2T + 1).$$

The zero set of the F_i's is exactly $Z = \{0, -1\}^n$, and the zeros are simple real points $\boldsymbol{\alpha}^{(S)} \in \{0, -1\}^n$ indexed by subsets $S \subseteq [n]$, with $\alpha_i^{(S)} = -1$ exactly when $i \in S$. Their leading coefficients $4w_i$ are positive, and

$$\Psi(\boldsymbol{\alpha}^{(S)}) = 2\sum_{i\in S} w_i - (2T + 1). \tag{9}$$

This value is odd, hence nonzero, and is negative exactly when S is feasible. Homogenizing with a fresh variable z, and applying Corollary 5.3 with $d = e = 2$ therefore gives

$$\operatorname{sgn}\operatorname{Res}(F_1^h, \dots, F_n^h, \Psi^h) = (-1)^{K_{\boldsymbol{w}}(T)}.$$

It remains to realize this resultant as a symmetric tensor determinant. Symmetry ties the coefficient of $x_i z$ in the i-th quadratic form to twice the coefficient of x_i^2 in the last form, explaining the choice $4w_i$ above. Set $N = n + 1$ and let A have nonzero entries

$$\begin{aligned} a_{iii} &= 4w_i && i \in [n],\\ a_{iiN} = a_{iNi} = a_{Nii} &= 2w_i && i \in [n],\\ a_{NNN} &= -(2T + 1). \end{aligned}$$

This tensor is symmetric, and its quadratic map is

$$(A(\boldsymbol{x}, z)^2)_i = 4w_i x_i(x_i + z) = F_i^h \quad (i \in [n]), \qquad (A(\boldsymbol{x}, z)^2)_N = 2\sum_{i=1}^n w_i x_i^2 - (2T + 1)z^2 = \Psi^h.$$

Hence $\operatorname{Det}_N(A) = \operatorname{Res}(F_1^h, \dots, F_n^h, \Psi^h)$, and it is positive if and only if $K_{\boldsymbol{w}}(T)$ is even. The tensor A has $4n + 1$ nonzero entries, dimension $N = n + 1$, and its entry bitlengths are polynomial in the input size. The construction is therefore a polynomial-time many-one reduction, proving $\oplus$P-hardness. □

6 $\exists\mathbb{R}$ in the counting hierarchy

In this section, we prove Theorem 1.13. For an overview of the proof, see Section 1.2.2.

6.1 Polynomial arithmetic in CH

In this section, we collect facts about counting-hierarchy and polynomial-arithmetic that are needed for the feasibility proof. We first recall the definition and some closure properties of CH.

Wagner [Wag86] defined the counting hierarchy to classify some combinatorial problems that involved counting. He used the notion of counting quantifiers for his definition, but we state here the equivalent oracle formulation by Torán [Tor91].

Definition 6.1 (Counting hierarchy)**.** Let $\mathsf{C}_0\mathsf{P} = \mathsf{P}$ and $\mathsf{C}_{k+1}\mathsf{P} = \mathsf{P}^{\mathsf{C}_k\mathsf{P}}$. Then, the counting hierarchy is defined as

$$\mathsf{CH} = \bigcup_{k\geq 0} \mathsf{C}_k\mathsf{P}.$$

The counting hierarchy is closed under existential and universal quantifications over strings of polynomial length and polynomially many adaptive queries to an oracle in a fixed level of CH.

Lemma 6.2 (Closure properties of CH [AW93, Section 1])**.** *Fix $k \geq 0$. Let $A \in \mathsf{C}_k\mathsf{P}$, and let p be a polynomial taking nonnegative integer values. Then the languages*

$$L_\exists = \left\{x : \exists y \in \{0,1\}^{\leq p(|x|)} \text{ with } \langle x, y\rangle \in A\right\},$$
$$L_\forall = \left\{x : \forall y \in \{0,1\}^{\leq p(|x|)},\ \langle x, y\rangle \in A\right\}$$

belong to $\mathsf{C}_{k+1}\mathsf{P}$. Moreover,

$$\mathsf{P}^{\mathsf{C}_k\mathsf{P}} \subseteq \mathsf{C}_{k+1}\mathsf{P}.$$

Consequently, a fixed number of such quantifications and polynomial-time oracle computations preserves membership in CH*.*

For more details on the counting hierarchy, we refer the reader to the survey of Allender and Wagner [AW93].

We will now describe the polynomial families that arise in the proof. Unlike in Section 3, these families are indexed by a string w that encodes the original input and any auxiliary choices.

Definition 6.3 (Polynomial family)**.** A *polynomial family* $(f_w)_{w\in\{0,1\}^*}$ satisfies the following conditions:

1. f_w is a polynomial in $\mathbb{Q}[x_1, \ldots, x_{n(w)}]$, where $n(w)$ is polynomially bounded in $|w|$;
2. the degree of f_w is at most $2^{\mathsf{poly}(|w|)}$;
3. the numerator and denominator bit lengths of its coefficients are at most $2^{\mathsf{poly}(|w|)}$.

The bounding polynomials are fixed for each family, so their values can be computed from $|w|$.

A *coefficient representation* for a polynomial family $(f_w)_{w\in\{0,1\}^*}$ is a way of encoding the coefficients of f_w as bit strings.

Definition 6.4 (Coefficient representation)**.** Let $(f_w)_{w\in\{0,1\}^*}$ be a polynomial family. For a binary encoded exponent vector $\boldsymbol{e} = (e_1, \ldots, e_{n(w)})$, let $c_{w,\boldsymbol{e}}$ denote the coefficient of the monomial $\boldsymbol{x}^{\boldsymbol{e}} = x_1^{e_1} \cdots x_{n(w)}^{e_{n(w)}}$ in f_w.

A *representation* of $c_{w,\boldsymbol{e}}$ consists of its sign, and the magnitudes of the numerator and denominator, all satisfying the bit-length bounds of (f_w). The fraction need not be reduced. For a chosen representation, the numerator and denominator are padded to the known bit-length bound and concatenated with the sign bit. The resulting bit string is denoted by $\mathrm{enc}(c_{w,\boldsymbol{e}})$.

Since the polynomials may have singly exponential degree and coefficient bit length, we work with access to individual coefficient bits. This uses Bürgisser's addressed-bit viewpoint of integers definable in CH [Bür09, Section 3].

Definition 6.5 (Compact coefficient access)**.** A polynomial family $(f_w)_w$ has *compact coefficient access* if there exists a coefficient representation for each coefficient $c_{w,\boldsymbol{e}}$ such that a requested bit b of the representation can be computed in CH uniformly. More precisely, for some constant k, we have

$$\{(w, \boldsymbol{e}, b) : \text{bit } b \text{ of } \mathrm{enc}(c_{w,\boldsymbol{e}}) \text{ is } 1\} \in \mathsf{C}_k\mathsf{P}.$$

When $k = 0$, we say that the family has *polynomial-time coefficient-bit access*.

Remark 6.6. A query returns one bit of one coefficient of one polynomial. The representation is chosen once for each coefficient, and all bit queries for that coefficient refer to the same representation. The level k may depend on the family but is independent of $w, \boldsymbol{e}, b$. Every coefficient within the effective degree bound and every bit of its padded representation has an address of length $\mathsf{poly}(|w|)$ by Definition 6.3. The full coefficient array need not be written down.

Whenever a family is indexed by a tuple, we identify the tuple with a single bit string using a fixed encoding computable and decodable in polynomial time, with length polynomial in the total length of its components. Thus, Definitions 6.3 and 6.5 apply without changing the tuple notation. In particular, compact coefficient access always means access uniform in all displayed indices. All auxiliary indices we use will have polynomial length in the original input length, so the combined indices still have polynomial length. We leave this encoding implicit throughout.

Families specified only on valid indices are extended to all bit strings by fixed default polynomials satisfying their stated hypotheses. The validity tests used below lie in a fixed level of CH. Zero coefficients inserted for padding, including coefficients beyond the effective degree bound, use the representation $0/1$ with sign bit zero. Malformed addresses and bit positions outside the padded encoding return zero.

We will now record the polynomial operations used in the proof of Theorem 1.13. Their preservation of compact coefficient access follows from the results of Hesse, Allender, and Barrington [HAB02] and Andrews, Garg, and Schost [AGS26]. We give a proof sketch in Appendix D.1.

Lemma 6.7 (Compact univariate arithmetic)**.** *Let $(F_w(y))_w$ and $(G_{w,j}(y))_{w,\,1\le j\le M(w)}$ be families of univariate polynomials over $\mathbb{Q}$ with compact coefficient access, where $M(w) \ge 1$ is computable in polynomial time in binary. Then, compact coefficient access is preserved by the following operations:*

1. Sums: $S_w(y) = \sum_{j=1}^{M(w)} G_{w,j}(y)$.
2. Products: $P_w(y) = \prod_{j=1}^{M(w)} G_{w,j}(y)$.
3. Derivatives: *the family $(F_w^{(r)})_{w,r\ge 0}$, with r given in binary.*
4. Monic normalization: $p_w = F_w/\mathrm{lc}(F_w)$, *provided every F_w is nonzero.*
5. Squarefree part: $p_w = H_w/\mathrm{lc}(H_w)$, *where $H_w = F_w/\gcd(F_w, F_w')$, provided every F_w is nonconstant. The polynomial p_w is monic and squarefree, with the same roots as F_w.*

The following queries are computable in polynomial time with an oracle in a fixed level of CH*:*

6. Degree: *given w, determine whether F_w is identically zero and, otherwise, compute* $\deg F_w$ *in binary.*
7. Coefficient sign: *given w and e, compute* $\mathrm{sgn}([y^e]F_w) \in \{-1, 0, 1\}$.

All constructions and queries are uniform in the displayed indices, which are encoded in binary. The counting levels may depend on the fixed input families, but not on the indices.

Remark 6.8. The statements about sums, products, derivatives, degrees, and coefficient signs also have multivariate versions under the same effective bounds, using partial derivatives in place of univariate derivatives. We state the univariate versions because these suffice for the arithmetic constructions below.

The next lemma shows that resultants preserve compact coefficient access. The result follows from the constant-depth resultant constructions of [AGS26], together with their arithmetic-to-Boolean simulation and counting-hierarchy evaluation arguments. We provide a proof sketch in Appendix D.1.

Lemma 6.9 (Compact access to resultants)**.** *Suppose the input polynomial families have compact coefficient access and satisfy the bounds of Definition 6.3. Then the following resultant families also satisfy these bounds and have compact coefficient access:*

1. Univariate resultants. *For families of nonzero polynomials $f_w, g_w \in \mathbb{Q}[t][x]$, the family*

$$R_w(t) = \mathrm{Res}_x\big(f_w(x,t), g_w(x,t)\big) \in \mathbb{Q}[t].$$

   The degrees in x are taken over $\mathbb{Q}[t]$, before any specialization of t. This includes ordinary univariate resultants over $\mathbb{Q}$ when the inputs are independent of t.
2. Multivariate resultants. *For families of homogeneous forms*

$$f_{w,1}, \dots, f_{w,n} \in \mathbb{Q}[t][\boldsymbol{x}], \qquad \boldsymbol{x} = (x_1, \dots, x_n),$$

   the family

$$R_w(t) = \mathrm{Res}_{d_1,\dots,d_n}\big(f_{w,1}(\boldsymbol{x},t), \dots, f_{w,n}(\boldsymbol{x},t)\big) \in \mathbb{Q}[t].$$

   Here $n = n(w)$ is polynomially bounded in $|w|$, the positive assigned degrees $d_1, \dots, d_n$ in $\boldsymbol{x}$ are bounded by a fixed constant, and input coefficient access is uniform in (w,i). The number n and the assigned degrees are computable in polynomial time from w. The resultant eliminates $\boldsymbol{x}$ and retains the parameter t.

In both cases, output coefficient access is uniform in w.

6.2 Constructing univariate families preserving feasibility

We start with the problem of testing real feasibility for a homogeneous quadratic system, which is $\exists\mathbb{R}$-complete.

Lemma 6.10 (Quadratic normal form [SCM24, Entry A6], [Sch13, Corollary 3.10])**.** *The following problem is $\exists\mathbb{R}$-complete: given integer homogeneous quadratic forms $q_1, \dots, q_m$ in n variables, decide whether they have a common zero in $\mathbb{R}^n \setminus \{0\}$.*

6.2.1 Constructing a finite set witnessing feasibility

The common zero set of the polynomials q_j may have a positive-dimensional component over $\mathbb{C}$ or even over $\mathbb{R}$. In that case, direct elimination with a linear form may fail to produce a useful univariate polynomial. In what follows, we recall a standard strategy, while explaining why each step is in CH, to obtain a system of polynomials G_i over $\mathbb{R}$ in n variables whose common solution set $Z \subset \mathbb{C}^n$ is finite, and $Z \cap \mathbb{R}^n \neq \emptyset$ if and only if $q_1, \dots, q_m$ have a common nonzero real zero.

Let u denote the (encoding of the) original feasibility instance and $L = |u|$. All quantities below are constructed from u; we suppress this index when discussing a fixed instance. Throughout this section we assume that we are in the situation of Lemma 6.10, i.e., our starting polynomials are quadratic forms $q_1, \dots, q_m$. Since $q_1, \dots, q_m$ are homogeneous, it is enough to test feasibility over the unit sphere $\mathbb{S}^{n-1}$. We also assume that $n \geq 2$, as for $n = 1$, q_j have no common solution on $\mathbb{S}^{n-1}$ unless they are all identically zero. Let

$$S(\boldsymbol{x}) = \left(\sum_{i=1}^{n} x_i^2\right) - 1, \qquad Q(\boldsymbol{x}) = \sum_{j=1}^{m} q_j(\boldsymbol{x})^2 + S(\boldsymbol{x})^2.$$

It is clear that the instance is feasible if and only if Q has a real zero. The reduction and the construction of the explicit quartic Q take polynomial time, so the family $(Q_u)_u$ has polynomial-time coefficient-bit access.

Note that the polynomial Q is *coercive*, i.e., $Q(\boldsymbol{x}) \to \infty$ as $\|\boldsymbol{x}\| \to \infty$. If Q has no real zero, then 0 is not in the Euclidean closure of the image of $Q : \mathbb{R}^n \to \mathbb{R}^+$. In this case there is a $\gamma > 0$ such that $Q(\boldsymbol{x}) \geq \gamma$, for all $\boldsymbol{x} \in \mathbb{R}^n$. The following lemma gives an explicit positive lower bound γ for Q on infeasible instances. The quantity γ will be useful later to perturb Q and choose a nearby level set that detects feasibility.

Lemma 6.11 (Positive-minimum bound)**.** *For a nonnegative coercive integer quartic P in $n \geq 2$ variables of description length L, one can compute in polynomial time a binary integer $B \leq 2^{\mathsf{poly}(L)}$ such that*

$$P \text{ has no real zero} \quad \Longrightarrow \quad P(\boldsymbol{z}) \geq \gamma := 2^{-B} \quad \text{for every } \boldsymbol{z} \in \mathbb{R}^n.$$

The proof of the above lemma follows from the standard bounds in [JPT13, Theorem 2.10]. The full proof along with the computation of bound B is given in Appendix D.2.

Let $\gamma = 2^{-B}$ be the gap from Lemma 6.11 applied to Q. Although γ may have an exponentially long denominator, its power-of-two form gives polynomial-time access to each bit. For an infeasible instance, $Q(\boldsymbol{x}) \geq \gamma$ for every $\boldsymbol{x} \in \mathbb{R}^n$. Set

$$H(\boldsymbol{x}) = Q(\boldsymbol{x}) + \frac{\gamma}{4} \sum_{i=1}^{n} x_i^6.$$

This uses the same pure-even-power deformation idea as [BPR06, Section 12.6], with the least even degree $\bar{d} = 6 > 4$. The new terms of H have degree six, separate from those of Q, so $(H_u)_u$ also has polynomial-time coefficient-bit access.

Consider the square system of n equations in n variables

$$G_1 = H - \gamma/2, \qquad G_i = \partial_i H \ \ (2 \leq i \leq n), \qquad Z = \{\boldsymbol{z} \in \mathbb{C}^n : G_1(\boldsymbol{z}) = \cdots = G_n(\boldsymbol{z}) = 0\}. \tag{10}$$

The degrees are $6, 5, \ldots, 5$. These equations give candidate points for maximizing the coordinate x_1 subject to $G_1 = 0$, including points where $\nabla H = 0$. Differentiation here only multiplies coefficients by integers at most six, and the constant term of G_1 is $1 - \gamma/2$. Thus the family $(G_{u,i})_{u,i}$ still has polynomial-time coefficient-bit access. The coefficient-bit details for Q, H, and the G_i are given in Appendix D.2.

Lemma 6.12. *The set Z is finite, with cardinality at most $6 \cdot 5^{n-1}$ and total multiplicity exactly $6 \cdot 5^{n-1}$.*

Proof. Consider the projective system obtained by homogenizing $G_1, \ldots, G_n$ with an additional variable x_0. Projective solutions with $x_0 = 0$ are called solutions at infinity. At $x_0 = 0$, the equations become

$$\frac{\gamma}{4} \sum_{i=1}^{n} x_i^6 = 0, \qquad \frac{3\gamma}{2} x_i^5 = 0 \quad (2 \leq i \leq n).$$

The last $n - 1$ equations force $x_2 = \cdots = x_n = 0$, and then the first forces $x_1 = 0$. Thus the projective system has no solutions at infinity. The affine form of Bézout's theorem [CLO05, Chapter 3, Theorem 5.5] therefore gives exactly the product of the degrees, $6 \cdot 5^{n-1}$, complex affine solutions counted with multiplicity. In particular, Z is finite and satisfies both bounds. □

Lemma 6.13. *The original system $q_1(\boldsymbol{x}) = \cdots = q_m(\boldsymbol{x}) = 0$ has a nonzero real solution if and only if $Z \cap \mathbb{R}^n \neq \varnothing$.*

Proof. If $\boldsymbol{z} \in Z$ is real, then $Q(\boldsymbol{z}) \leq H(\boldsymbol{z}) = \gamma/2 < \gamma$, so by the gap bound (Lemma 6.11) Q has a real zero. By the definition of Q, this is a common zero of $q_1, \ldots, q_m$ on the unit sphere.

Conversely, suppose Q has a real zero $\boldsymbol{a} \in \mathbb{R}^n$. Then $\boldsymbol{a} \in \mathbb{S}^{n-1}$ and $H(\boldsymbol{a}) = \frac{\gamma}{4}\sum_i a_i^6 \le \frac{\gamma}{4}(\sum_i a_i^2)^3 = \gamma/4 < \gamma/2$. The set $M = H^{-1}([0,\gamma/2])$ is nonempty, since it contains $\boldsymbol{a}$. It is closed by continuity of H. It is also bounded, since for every $\boldsymbol{x} \in M$ and every i we have

$$\frac{\gamma}{4}x_i^6 \le H(\boldsymbol{x}) \le \frac{\gamma}{2}, \qquad \text{and hence} \qquad |x_i| \le 2^{1/6}.$$

By the extreme value theorem [Boo13, Theorem 17.6], the first coordinate attains a maximum on the nonempty closed and bounded set M. Let $\boldsymbol{\alpha} \in M$ be a maximizing point. Then $H(\boldsymbol{\alpha}) = \gamma/2$: otherwise a small ball around $\boldsymbol{\alpha}$ would lie in M. Indeed, if $H(\boldsymbol{\alpha}) < \gamma/2$, put $\varepsilon = \gamma/2 - H(\boldsymbol{\alpha}) > 0$. By the ε–δ definition of continuity of H at $\boldsymbol{\alpha}$, there is a $\delta > 0$ such that $\|\boldsymbol{x} - \boldsymbol{\alpha}\| < \delta$ implies $|H(\boldsymbol{x}) - H(\boldsymbol{\alpha})| < \varepsilon$. Since $H \ge 0$, every such $\boldsymbol{x}$ lies in M. This ball contains points with larger first coordinate, a contradiction. If $\partial_i H(\boldsymbol{\alpha}) \ne 0$ for some $i \ge 2$, choose $c \in \mathbb{R}$ such that

$$\partial_1 H(\boldsymbol{\alpha}) - c\partial_i H(\boldsymbol{\alpha}) < 0.$$

Since H is a polynomial, for some polynomial $P(t)$ we have

$$H(\boldsymbol{\alpha} + t(\boldsymbol{e}_1 - c\boldsymbol{e}_i)) - H(\boldsymbol{\alpha}) = t\big(\partial_1 H(\boldsymbol{\alpha}) - c\partial_i H(\boldsymbol{\alpha}) + tP(t)\big).$$

The expression in parentheses is negative for sufficiently small $t > 0$, so $H(\boldsymbol{\alpha} + t(\boldsymbol{e}_1 - c\boldsymbol{e}_i)) < \gamma/2$. As $H \ge 0$, this point belongs to M; its first coordinate is $\alpha_1 + t$, contradicting the maximality of $\boldsymbol{\alpha}$. Hence $\partial_i H(\boldsymbol{\alpha}) = 0$ for every $i \ge 2$, and $\boldsymbol{\alpha} \in Z$. □

We do not construct the set Z, nor do we enumerate it. We only need the following properties: if the instance is feasible, then Z contains a real point; if the instance is infeasible, then Z is finite and contains no real point. In the next section, we use properties of Z to construct a family of univariate polynomials which can be used to test the existence of a real point in Z. The bound $6 \cdot 5^{n-1}$ on the size of Z has polynomial bit-length. All coefficient-bit predicates constructed so far lie in P.

6.2.2 Constructing a family of univariate polynomials preserving feasibility

Recall the finite set Z from the previous section. Consider the linear forms $\ell_a(\boldsymbol{x}) \coloneqq \sum_{i=1}^n a^i x_i$ for integers $a \ge 0$. They map real points to real numbers. We will show that if Z contains no real point, then some ℓ_a in an explicitly bounded family satisfies $\ell_a(Z) \cap \mathbb{R} = \varnothing$. For a fixed input u, retain the notation n, G_i, Z for the quantities constructed above. Consider the resultant

$$R_{u,a}(t) = \operatorname{Res}(G_1^h, \dots, G_n^h, tx_0 - \ell_a(\boldsymbol{x})), \tag{11}$$

where G_1^h is the homogenization of G_1 to degree 6, and G_i^h is the homogenization of G_i to degree 5 for $i \ge 2$, using the additional variable x_0. The two indices of $R_{u,a}$ are the input description u and the integer parameter a. For each fixed pair (u,a), we regard $R_{u,a}(t)$ as a univariate polynomial in t. At $x_0 = 0$, the equations $G_i^h = 0$ for $i \ge 2$ force $x_i = 0$, and then $G_1^h = 0$ forces $x_1 = 0$. Thus the G_i^h have no common projective zero at infinity, and the Poisson formula [AGS26, Theorem 4.3] gives

$$R_{u,a}(t) = C\prod_{\boldsymbol{\alpha}\in Z}(t - \ell_a(\boldsymbol{\alpha}))^{m_{\boldsymbol{\alpha}}}, \qquad C \ne 0, \tag{12}$$

where $m_{\boldsymbol{\alpha}}$ are the local multiplicities and C is independent of a and t, but may depend on u. The formula shows that every $R_{u,a}$ has root set $\ell_a(Z)$ and, by Lemma 6.12, degree $6 \cdot 5^{n-1}$.

Lemma 6.14 (Universal projections)**.** *Fix an input u and put $K = K(u) = 6n \cdot 5^{n-1} + 1$. Then $Z \cap \mathbb{R}^n = \varnothing$ if and only if some $R_{u,a}$ with $0 \le a < K$ has no real root.*

Proof. Suppose Z contains no real point. For $\boldsymbol{\alpha} \in Z$ the condition $\ell_a(\boldsymbol{\alpha}) \in \mathbb{R}$ reads $\sum_{i=1}^{n} a^i \operatorname{Im} \alpha_i = 0$, a nonzero polynomial in a of degree at most n, so at most n indices give this point a real image. The union over the at most $6 \cdot 5^{n-1}$ points of Z, by Lemma 6.12, excludes at most $6n \cdot 5^{n-1} = K - 1$ indices. Hence some $a < K$ has $\ell_a(Z) \cap \mathbb{R} = \varnothing$, and $R_{u,a}$ has no real root.

Conversely, if some $R_{u,a}$ has no real root, then Z cannot contain a real point: its image under ℓ_a would be a real root of $R_{u,a}$. □

The following lemma shows that the projection family $(R_{u,a})_{u,a}$ has compact coefficient access. It lets us apply the real-root test to these polynomials without writing out their coefficients. This is the first use of the CH closure machinery in the construction.

Lemma 6.15 (Compact access to the projection family)**.** *The polynomial family $(R_{u,a})_{u,\, 0 \le a < K(u)}$, with a written in binary, has compact coefficient access.*

To see this, apply Lemma 6.9(2) to the homogeneous family $(G^h_{u,1}, \dots, G^h_{u,n}, tx_0 - \ell_a(\boldsymbol{x}))$ with index (u, a). Its coefficients have polynomial-time bit access and its assigned degrees are constant, so the output family $(R_{u,a})_{u,a}$ has compact coefficient access. The full proof of Lemma 6.15, including the size bounds, is in Appendix D.3.

6.3 Univariate real-root existence testing

We know by Lemmas 6.13 and 6.14 that the original instance u is feasible if and only if every $R_{u,a}$ with $0 \le a < K(u)$ has a real root. By Lemma 6.15, we can construct the family $(R_{u,a})_{u,a}$ of univariate polynomials with compact coefficient access. We will now show that, given the index of a polynomial in any such univariate family, we can decide in CH whether it has a real root. The following proposition is the main technical result of this section.

Proposition 6.16. *For every fixed family $(f_w(x))_w$ of univariate polynomials in $\mathbb{Q}[x]$ with compact coefficient access,*

$$\{w : \textit{there exists } \alpha \in \mathbb{R} \textit{ with } f_w(\alpha) = 0\} \in \mathsf{CH}.$$

In this section, we write $L = |w|$ for the length of the family index. The coefficient tests in Lemma 6.7(6) can detect zero and constant polynomials: the zero polynomial has a real root, whereas a nonzero constant does not. For the nonconstant members of $(f_w)_w$, applying Lemma 6.7(5) gives the monic squarefree family $(p_w)_w$ with the same roots as before, but with multiplicity one. Further, using Lemma 6.7(6) we can compute $D_w = \deg p_w$ in CH. The construction also gives effective upper bounds on degree and coefficient bit lengths.

6.3.1 Short description of roots

For a fixed w, we let $p = p_w$ and $D = D_w$. We will first show that if p has a real root, then there is a short description for it. To do this, we will need the classical notion of Thom encoding [CR88].

Lemma 6.17 (Thom encoding [BPR06, Proposition 2.28])**.** *Let p be a univariate polynomial of degree D. The Thom encoding of a real root α of p is given by the vector*

$$\boldsymbol{c}(\alpha) = (\operatorname{sgn} p^{(j)}(\alpha))_{j=1}^{D} \in \{-1, 0, 1\}^D.$$

For distinct real roots α, α' of p, we have $\boldsymbol{c}(\alpha) \neq \boldsymbol{c}(\alpha')$.

Our resultant sign test (Lemma 5.2) can detect whether an odd number of real roots satisfy a given set of sign conditions. The Thom encoding uniquely identifies a root using derivative-sign tests, and we could guess it. To use it for real-root existence, we will need to construct a polynomial Ψ such that for every real root, it is negative exactly if it matches the guess. But the entire Thom encoding is of length D, which could be $2^{\mathsf{poly}(L)}$. The closure of CH under existential quantification (Lemma 6.2) only allows

a guess of polynomial length. Therefore, we will instead guess a short partial list of the encoding. We call a list

$$\boldsymbol{\eta} = [(j_1, \sigma_1), \ldots, (j_t, \sigma_t)], \qquad j_i \in [D] \text{ and } \sigma_i \in \{-1, 0, 1\}$$

a *fingerprint*. A real root α of p matches the fingerprint $\boldsymbol{\eta}$ if $\operatorname{sgn} p^{(j_i)}(\alpha) = \sigma_i$ for every $i \in [t]$. The fingerprint is *isolating* if exactly one real root matches it. The following lemma shows that, if p has a real root, some short fingerprint isolates one of its roots.

Lemma 6.18 (Isolating fingerprint)**.** *Let p be a univariate polynomial of degree D as above. If p has a real root, there is an isolating fingerprint with at most $\lceil \log_2 D \rceil$ entries for some real root of p.*

Proof. The Thom labels $\boldsymbol{c}(\alpha)$ are distinct at distinct real roots (Lemma 6.17), so the real roots contribute at most D distinct strings. Start with all of them. Whenever at least two remain, distinctness gives a coordinate taking at least two values, hence at least two nonempty sign classes. Retain a smallest nonempty class, containing at most half the strings, and record its coordinate value. After at most $\lceil \log_2 D \rceil$ halvings one string remains, and it matches every recorded test while each dropped string fails at least one. □

Figure 1 illustrates how two derivative signs isolate a root of a sextic. An isolating fingerprint for p uses $O(\log^2(D+1))$ bits, which we can guess. We now want to construct a polynomial Ψ such that, at every real root α of p, $\Psi(\alpha) < 0$ exactly when α matches the guessed fingerprint. The sign function is not a polynomial, so we use rational functions to approximate it. In order to do this, we need uniform bounds on the nonzero derivative values of p at its roots separating them from zero. The lemma below gives such a bound.

Lemma 6.19 (Separation of derivative evaluations)**.** *Let $p = p_w$ and $D = D_w$ be as above. There exists an integer $1 \le U \le 2^{\mathsf{poly}(L)}$ computable in binary in polynomial time from w such that at every complex root α of p, the following holds:*

$$p^{(j)}(\alpha) \neq 0 \quad \Longrightarrow \quad 2^{-U} \le |p^{(j)}(\alpha)| \le 2^{U} \qquad (0 \le j \le D).$$

We give a simple proof for the above lemma in Appendix D.4. All the derivatives have compact coefficient access uniformly in their binary orders by Lemma 6.7(3). We will handle the case when derivatives evaluate to zero using the sign approximations that we construct.

6.3.2 Rational approximation and combined sign tests

Approximating the sign function by rational functions is a standard technique in complexity theory developed by Beigel, Reingold and Spielman [BRS95] (also see [For97, Section 4.4]) to study the closure properties of PP. We will need the following version, for which a self-contained proof is given in Appendix D.4.

Lemma 6.20 (Rational approximation of the sign function)**.** *Let $U \ge 1$ be an integer and $0 < \varepsilon \le 1/2$. There exist polynomials $\mathcal{N}, \mathcal{D} \in \mathbb{Q}[x]$ such that $\mathcal{D}(\lambda) > 0$ for every $\lambda \in \mathbb{R}$ and the rational function $\mathcal{S} = \mathcal{N}/\mathcal{D}$ satisfies*

$$\mathcal{S}(0) = 0, \qquad |\mathcal{S}(\lambda)| \le 1 \quad (\lambda \in \mathbb{R}),$$

and

$$|\mathcal{S}(\lambda) - \operatorname{sgn}(\lambda)| \le \varepsilon \qquad \text{whenever } 2^{-U} \le |\lambda| \le 2^{U}.$$

Both $\mathcal{N}$ and $\mathcal{D}$ have degree $O(U \log(1/\varepsilon))$, and their coefficients admit representations with numerator and positive denominator bit lengths $O(U^2 \log(1/\varepsilon))$.

We will now see how to replace a conjunction of sign tests by the sign of one polynomial.

Lemma 6.21 (A polynomial for a sign conjunction)**.** *For $s \geq 1$, let $h_1, \dots, h_s \in \mathbb{Q}[x]$, be polynomials with requested sign vector $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_s) \in \{-1, 0, 1\}^s$. Let $U \geq 1$ be an integer, and suppose a real point α satisfies either $h_i(\alpha) = 0$ or $2^{-U} \leq |h_i(\alpha)| \leq 2^U$ for every $i \in [s]$.*

Then there exists a polynomial $\Psi_{\wedge,\boldsymbol{\sigma}} \in \mathbb{Q}[x]$, depending only on the σ_i, the h_i, and U (suppressed in the notation), and not on α, of degree $O(sU \log(s+1) \max\{1, \max_i \deg h_i\})$, such that $\Psi_{\wedge,\boldsymbol{\sigma}}(\alpha) \neq 0$ and

$$\Psi_{\wedge,\boldsymbol{\sigma}}(\alpha) < 0 \quad \Longleftrightarrow \quad \operatorname{sgn} h_i(\alpha) = \sigma_i \text{ for all } i \in [s].$$

Proof. Choose $\varepsilon = 1/(4s)$. Now Lemma 6.20 gives a rational sign approximation $\mathcal{S} = \mathcal{N}/\mathcal{D}$ with $\mathcal{D} > 0$ on $\mathbb{R}$, $\mathcal{S}(0) = 0$, and error at most ε on $2^{-U} \leq |\lambda| \leq 2^U$. Its numerator and denominator have degree $O(U \log(s+1))$. Let

$$E(x) = \sum_{i=1}^{s} \big(\mathcal{S}(h_i(x)) - \sigma_i\big)^2.$$

If every sign matches at a permitted point α, then $E(\alpha) \leq s\varepsilon^2 \leq 1/16$. If a sign fails, its distance from the requested sign is at least one, so that summand is at least $(1-\varepsilon)^2 \geq 9/16$. All summands are nonnegative, hence $E(\alpha) \geq 9/16$ in this case.

Write $E = A/B$ by setting

$$B = \prod_i \mathcal{D}(h_i)^2, \qquad A = \sum_i \big(\mathcal{N}(h_i) - \sigma_i \mathcal{D}(h_i)\big)^2 \prod_{\ell \neq i} \mathcal{D}(h_\ell)^2,$$

and define

$$\Psi_{\wedge,\boldsymbol{\sigma}} = 2A - B. \tag{13}$$

Since $B > 0$ on $\mathbb{R}$, this polynomial has the sign of $2E - 1$, which is at most $-7/8$ when all tests pass and at least $1/8$ otherwise. The products defining A and B give the asserted degree bound. □

In our case of interest, the h_i's are the derivatives selected by an isolating fingerprint. The lemma below shows that the resulting sign-conjunction polynomial has compact coefficient access. We adjoin the always true test $\operatorname{sgn}(1) = 1$, so the conjunction is nonempty even when the fingerprint is empty.

Lemma 6.22 (Compact access to a sign conjunction)**.** *Let $(p_w)_w$ be a family of nonconstant monic squarefree polynomials in $\mathbb{Q}[x]$ with compact coefficient access. Given a fingerprint $\boldsymbol{\eta} = [(j_1, \sigma_1), \dots, (j_t, \sigma_t)]$, let*

$$h_i = p_w^{(j_i)} \ (1 \leq i \leq t), \qquad h_{t+1} = 1, \qquad \boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_t, 1).$$

Choose $U \geq 1$ from the effective degree and coefficient bounds so that the separation condition (Lemma 6.19) holds at every real root of p_w. Then the polynomials $\Psi_{\wedge,\boldsymbol{\sigma}}$ constructed in Lemma 6.21 form a family with compact coefficient access indexed by $(w, \boldsymbol{\eta})$.

The above lemma follows easily from the previous results in the section and CH closure results. We defer the proof to Appendix D.4.

We can now use the sign-conjunction polynomial and our method of encoding parities using resultant signs from Section 5 to test for the existence of a real root matching a given fingerprint.

Corollary 6.23 (Parity of a fingerprint)**.** *Let $(p_w)_w$ and $\boldsymbol{\eta}$ be as in Lemma 6.22, and denote its sign-conjunction polynomial by $\Psi_{w,\boldsymbol{\eta}}$. Define*

$$\mathcal{R}_{w,\boldsymbol{\eta}}(t) = \operatorname{Res}_x\big(p_w(x), \Psi_{w,\boldsymbol{\eta}}(x) + t\big).$$

This polynomial is nonzero. If ν is its least nonzero coefficient index, then

$$\operatorname{sgn}\big([t^\nu]\mathcal{R}_{w,\boldsymbol{\eta}}\big) = (-1)^{\#\{\alpha \in \mathbb{R} : p_w(\alpha) = 0,\, \alpha \text{ matches } \boldsymbol{\eta}\}}. \tag{14}$$

Deciding whether $[t^\nu]\mathcal{R}_{w,\boldsymbol{\eta}} < 0$ belongs to CH*, uniformly in $(w, \boldsymbol{\eta})$.*

Proof of the parity formula. By Lemmas 6.21 and 6.22, $\Psi_{w,\boldsymbol{\eta}}$ is nonzero at every real root of p_w and is negative exactly at the roots matching $\boldsymbol{\eta}$. Since p_w is monic and squarefree, we can apply Lemma 5.2 with $n = 1$ and $F_1 = p_w$. Homogenizing the second polynomial to any degree at least $\max\{1, \deg \Psi_{w,\boldsymbol{\eta}}\}$ gives the same resultant $\mathcal{R}_{w,\boldsymbol{\eta}}$, by the univariate product formula and monicity of p_w. The lemma therefore gives nonvanishing and the claimed sign. We defer the proof of compact coefficient access and CH containment to Appendix D.4. □

We are now ready to show that univariate real-root existence testing lies in CH.

Proof of Proposition 6.16. Accept the zero polynomial and reject nonzero constants using Lemma 6.7(6). Otherwise, Lemma 6.7(5) gives the monic squarefree part p_w with the same roots; compute $D = \deg p_w$ by Lemma 6.7(6). Guess a fingerprint $\boldsymbol{\eta}$ with at most $\lceil \log_2 D \rceil$ entries, rejecting malformed guesses and indices outside $[D]$. Use Corollary 6.23 to accept exactly when an odd number of real roots match $\boldsymbol{\eta}$.

Acceptance implies that p_w has a real root. Conversely, if a real root exists, Lemma 6.18 supplies an isolating fingerprint, which is accepted. The fingerprint has polynomial length in $|w|$, and the preprocessing and parity test lie in fixed levels of CH. Existential quantification over the fingerprint preserves membership in CH by Lemma 6.2. □

6.4 Completing the proof

We can now complete the proof of Theorem 1.13 by combining the results of the previous sections.

Proof of Theorem 1.13. Fix an input u. The instance is feasible if and only if Z has a real point, by Lemma 6.13, if and only if every $R_{u,a}$, $0 \le a < K(u)$, has a real root, by Lemma 6.14. By Lemma 6.15, the family $(R_{u,a})_{u,a}$ has compact coefficient access. By Proposition 6.16, its real-root tests therefore lie in one fixed level of CH. The index a has polynomial length, and the bound $K(u) = 6n \cdot 5^{n-1} + 1$ is computable in polynomial time in binary. Thus the bounded universal quantifier over $0 \le a < K(u)$ adds at most one counting level by Lemma 6.2.

The construction uses a fixed number of arithmetic and quantifier stages, each adding a fixed number of levels. All quantified strings and coefficient addresses have polynomial length. Thus the whole test lies in $\mathsf{C}_k\mathsf{P}$ for a fixed k, and hence in CH. □

References

[AHPS26] Sanyam Agarwal, Archit Hari, Anurag Pandey, and Himanshu Shukla. "Approximate Polynomial Satisfiability and Integral Closure Membership in the Counting Hierarchy". In: (2026). arXiv: 2610.42138 [cs.CC].

[ABKM09] Eric Allender, Peter Bürgisser, Johan Kjeldgaard-Pedersen, and Peter Bro Miltersen. "On the Complexity of Numerical Analysis". In: *SIAM Journal on Computing* 38.5 (2009), pp. 1987–2006. DOI: 10.1137/070706015.

[AW93] Eric Allender and Klaus W. Wagner. "Counting Hierarchies: Polynomial Time and Constant Depth Circuits". In: *Current Trends in Theoretical Computer Science - Essays and Tutorials*. Vol. 40. World Scientific Series in Computer Science. World Scientific, 1993, pp. 469–483. DOI: 10.1142/9789812794499_0035.

[AGS26] Robert Andrews, Abhibhav Garg, and Éric Schost. *Hilbert's Nullstellensatz Is in the Counting Hierarchy*. 2026. arXiv: 2602.17904v1 [cs.CC]. To appear in the 67th IEEE Symposium on Foundations of Computer Science (FOCS 2026).

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity*. Cambridge University Press, Cambridge, 2009. ISBN: 978-0-521-42426-4. DOI: 10.1017/CBO9780511804090. URL: https://doi.org/10.1017/CBO9780511804090.

[BSTW26] Nikhil Balaji, Mahsa Shirmohammadi, Sébastien Tavenas, and James Worrell. *Approximate Polynomial Satisfiability is in the Counting Hierarchy*. 2026. arXiv: 2610.00644 [cs.CC].

[Bar95] Alexander I. Barvinok. "New Algorithms for Linear k-Matroid Intersection and Matroid k-Parity Problems". In: *Mathematical Programming* 69 (1995), pp. 449–470. DOI: 10.1007/BF01585571.

[Bas17] Saugata Basu. "Algorithms in Real Algebraic Geometry: A Survey". In: *Real Algebraic Geometry*. Vol. 51. Panor. Synthèses. Soc. Math. France, Paris, 2017, pp. 107–153. ISBN: 978-2-85629-857-2.

[BPR06] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry*. Second. Vol. 10. Algorithms and Computation in Mathematics. Springer-Verlag, Berlin, 2006. ISBN: 978-3-540-33098-1.

[BRS95] Richard Beigel, Nick Reingold, and Daniel Spielman. "PP Is Closed under Intersection". In: *Journal of Computer and System Sciences* 50 (1995), pp. 191–202. ISSN: 0022-0000,1090-2724. DOI: 10.1006/jcss.1995.1017.

[BH93] Amir Ben-Dor and Shai Halevi. "Zero-One Permanent Is #P-Complete, a Simpler Proof". In: *Proceedings of the Second Israel Symposium on Theory of Computing and Systems*. IEEE Computer Society, 1993, pp. 108–117. DOI: 10.1109/ISTCS.1993.253457.

[BSS89] Lenore Blum, Mike Shub, and Steve Smale. "On a Theory of Computation and Complexity over the Real Numbers: NP-Completeness, Recursive Functions and Universal Machines". In: *Bulletin of the American Mathematical Society* 21.1 (1989), pp. 1–46. DOI: 10.1090/S0273-0979-1989-15750-9.

[Boo13] Carl de Boor. *Linear Algebra*. Draft of January 25, 2013. 2013. URL: https://pages.cs.wisc.edu/~deboor/443/book.pdf.

[BCKLOSW26] Cornelius Brand, Radu Curticapean, Petteri Kaski, Baitian Li, Ian Orzel, Tim Seppelt, and Jiaheng Wang. "Beyond Bilinear Complexity: What Works and What Breaks with Many Modes?" In: *41st Computational Complexity Conference (CCC 2026)*. Vol. 383. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026, 11:1–11:23. DOI: 10.4230/LIPIcs.CCC.2026.11.

[BHO14] Murray Bremner, Jiaxiong Hu, and Luke Oeding. "The $3 \times 3 \times 3$ Hyperdeterminant as a Polynomial in the Fundamental Invariants for $\mathrm{SL}_3(\mathbb{C})^3$". In: *Mathematics in Computer Science* 8.2 (2014), pp. 147–156. DOI: 10.1007/s11786-014-0186-9.

[Bür00] Peter Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*. Vol. 7. Algorithms and Computation in Mathematics. Berlin, Heidelberg: Springer, 2000. DOI: 10.1007/978-3-662-04179-6.

[Bür09] Peter Bürgisser. "On Defining Integers and Proving Arithmetic Circuit Lower Bounds". In: *Computational Complexity* 18.1 (2009), pp. 81–103. ISSN: 1016-3328,1420-8954. DOI: 10.1007/s00037-009-0260-x.

[Can88] John F. Canny. "Some Algebraic and Geometric Computations in PSPACE". In: *STOC*. 1988, pp. 460–467. DOI: 10.1145/62212.62257. URL: https://doi.org/10.1145/62212.62257.

[CD05] Eduardo Cattani and Alicia Dickenstein. "Introduction to Residues and Resultants". In: *Solving Polynomial Equations: Foundations, Algorithms, and Applications*. Ed. by Alicia Dickenstein and Ioannis Z. Emiris. Vol. 14. Algorithms and Computation in Mathematics. Springer, Berlin, Heidelberg, 2005, pp. 1–61. DOI: 10.1007/3-540-27357-3_1.

[CR88] M. Coste and M.-F. Roy. “Thom’s lemma, the coding of real algebraic numbers and the computation of the topology of semi-algebraic sets”. In: *Journal of Symbolic Computation* 5.1-2 (1988), pp. 121–129. ISSN: 0747-7171,1095-855X. DOI: 10.1016/S0747-7171(88)80008-7.

[CLO05] David A. Cox, John Little, and Donal O’Shea. *Using Algebraic Geometry*. 2nd ed. Vol. 185. Graduate Texts in Mathematics. Springer, New York, 2005. ISBN: 978-0-387-20706-3.

[CT26] Dean Cureton and Jesse Thorner. *A quasi-Riemann hypothesis for Dedekind zeta functions*. Hexagon preprint 2610.00200v2. 2026. URL: https://hexagonmath.org/2610.00200v2.

[Cur24] Radu Curticapean. “Count on CFI Graphs for #P-Hardness”. In: *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Society for Industrial and Applied Mathematics, 2024, pp. 1854–1871. DOI: 10.1137/1.9781611977912.74.

[CM14] Radu Curticapean and Dániel Marx. “Complexity of Counting Subgraphs: Only the Boundedness of the Vertex-Cover Number Counts”. In: *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, 2014, pp. 130–139. DOI: 10.1109/FOCS.2014.22. arXiv: 1407.2929 [cs.CC].

[CW26] Radu Curticapean and Jiaheng Wang. *Planar Perfect Matching Counting is as Hard as Determinants*. 2026. DOI: 10.48550/arXiv.2606.03975. arXiv: 2606.03975 [cs.CC].

[DD01] Carlos D’Andrea and Alicia Dickenstein. “Explicit formulas for the multivariate resultant”. In: *Journal of Pure and Applied Algebra* 164.1–2 (2001), pp. 59–86. DOI: 10.1016/S0022-4049(00)00145-6.

[DJS23] Carlos D’Andrea, Gabriela Jeronimo, and Martín Sombra. “The Canny–Emiris conjecture for the sparse resultant”. In: *Foundations of Computational Mathematics* 23 (2023), pp. 741–801. DOI: 10.1007/s10208-021-09547-3.

[Dav73] Martin Davis. “Hilbert’s Tenth Problem Is Unsolvable”. In: *American Mathematical Monthly* 80.3 (1973), pp. 233–269. DOI: 10.1080/00029890.1973.11993265.

[Dem12] Michel Demazure. “Résultant, discriminant”. In: *L’Enseignement Mathématique* 58.3-4 (2012), pp. 333–373. ISSN: 0013-8584. DOI: 10.4171/LEM/58-3-5. URL: https://doi.org/10.4171/LEM/58-3-5.

[DPS26] Prateek Dwivedi, Benedikt Pago, and Tim Seppelt. “Lower Bounds in Algebraic Complexity via Symmetry and Homomorphism Polynomials”. In: *Proceedings of the 58th Annual ACM Symposium on Theory of Computing (STOC 2026)*. Full version contains the detailed statements cited here. Association for Computing Machinery, 2026, pp. 631–640. DOI: 10.1145/3798129.3800779. arXiv: 2601.09343v1 [cs.CC].

[For09] Lance Fortnow. “A simple proof of Toda’s theorem”. In: *Theory of Computing. An Open Access Journal* 5 (2009), pp. 135–140. ISSN: 1557-2862. DOI: 10.4086/toc.2009.v005a007.

[For97] Lance Fortnow. “Counting Complexity”. In: *Complexity Theory Retrospective, II*. Springer, New York, 1997, pp. 81–107. ISBN: 978-0-387-94973-4. URL: https://lance.fortnow.com/papers/files/counting.pdf.

[FPV15] Hervé Fournier, Sylvain Perifel, and Rémi de Verclos. “On fixed-polynomial size circuit lower bounds for uniform polynomials in the sense of Valiant”. In: *Information and Computation* 240 (2015), pp. 31–41. DOI: 10.1016/j.ic.2014.09.006. arXiv: 1304.5910 [cs.CC].

[FSW26] Christopher Frei, Efthymios Sofos, and Martin Widmer. *Some applications of the quasi-Riemann hypothesis to class groups and small generators of number fields*. 2026. arXiv: 2610.18491. URL: https://arxiv.org/abs/2610.18491.

[GKZ94] I. M. Gelfand, M. M. Kapranov, and A. V. Zelevinsky. *Discriminants, Resultants, and Multidimensional Determinants*. Mathematics: Theory & Applications. Birkhäuser Boston, Inc., Boston, MA, 1994. ISBN: 978-0-8176-3660-9. DOI: 10.1007/978-0-8176-4771-1.

[GS86] Shafi Goldwasser and Michael Sipser. "Private coins versus public coins in interactive proof systems". In: *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*. 1986, pp. 59–68. DOI: 10.1145/12130.12137.

[GKP13] Bruno Grenet, Pascal Koiran, and Natacha Portier. "On the Complexity of the Multivariate Resultant". In: *Journal of Complexity* 29.2 (2013), pp. 142–157. DOI: 10.1016/j.jco.2012.10.001.

[GSS19] Zeyu Guo, Nitin Saxena, and Amit Sinhababu. "Algebraic Dependencies and PSPACE Algorithms in Approximative Complexity over Any Field". In: *Theory of Computing. An Open Access Journal* 15 (2019), Paper No. 16, 30. ISSN: 1557-2862. DOI: 10.4086/toc.2019.v015a016.

[HAB02] William Hesse, Eric Allender, and David A. Mix Barrington. "Uniform Constant-Depth Threshold Circuits for Division and Iterated Multiplication". In: *Journal of Computer and System Sciences* 65 (2002), pp. 695–716. ISSN: 0022-0000,1090-2724. DOI: 10.1016/S0022-0000(02)00025-9.

[HL13] Christopher J. Hillar and Lek-Heng Lim. "Most Tensor Problems Are NP-Hard". In: *Journal of the ACM* 60.6 (2013), 45:1–45:39. DOI: 10.1145/2512329.

[HHLQ13] Shenglong Hu, Zheng-Hai Huang, Chen Ling, and Liqun Qi. "On Determinants and Eigenvalue Theory of Tensors". In: *Journal of Symbolic Computation* 50 (2013), pp. 508–531. ISSN: 0747-7171,1095-855X. DOI: 10.1016/j.jsc.2012.10.001.

[JPT13] Gabriela Jeronimo, Daniel Perrucci, and Elias Tsigaridas. "On the Minimum of a Polynomial Function on a Basic Closed Semialgebraic Set and Applications". In: *SIAM Journal on Optimization* 23.1 (2013), pp. 241–255. DOI: 10.1137/110857751.

[JN25] Antoine Joux and Anand Kumar Narayanan. "A High Dimensional Cramer's Rule Connecting Homogeneous Multilinear Equations to Hyperdeterminants". In: *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*. Vol. 325. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025, 62:1–62:16. DOI: 10.4230/LIPIcs.ITCS.2025.62.

[Koi96] Pascal Koiran. "Hilbert's Nullstellensatz Is in the Polynomial Hierarchy". In: *Journal of Complexity* 12.4 (1996). The theorem numbering in the text refers to this journal version, pp. 273–286. DOI: 10.1006/jcom.1996.0019.

[Mat70] Yuri V. Matiyasevich. "Enumerable Sets Are Diophantine". In: *Doklady Akademii Nauk SSSR* 191.2 (1970). English translation in Soviet Mathematics Doklady 11 (1970), 354–358, pp. 279–282.

[Mei26] Alex Meiburg. $\exists\mathbb{R} \subseteq \mathsf{CH}$. 2026. DOI: 10.48550/arXiv.2610.10514. arXiv: 2610.10514 [cs.CC].

[Nar25] Anand Kumar Narayanan. *Hyperdeterminants Are Hard in Four Dimensions*. Tech. rep. TR25-131. Electronic Colloquium on Computational Complexity, 2025. URL: https://eccc.weizmann.ac.il/report/2025/131/.

[Ope26a] OpenAI. *Existential–universal real sentences in the counting hierarchy*. OpenAI Math Release preprint. Released 6 October 2026. Oct. 4, 2026. URL: https://github.com/openai/math/blob/main/preprints/Existential-universal-real-sentences-in-the-counting-hierarchy-October-4-2026/etr-counting-hierarchy.pdf (visited on 10/07/2026).

[Ope26b] OpenAI. *Hilbert's tenth problem over the rational numbers*. Preprint. 2026. URL: https://github.com/openai/math/tree/main/preprints/Hilberts-tenth-problem-over-the-rational-numbers-October-6-2026.

[Ope26c] OpenAI. *The Quasi-Riemann Hypothesis: A Zero-Free Half-Plane* $\Re s > 7/8$. Preprint. 2026. URL: https://github.com/openai/math/tree/main/preprints/The-Quasi-Riemann-Hypothesis-September-30-2026.

[Poo08] Bjorn Poonen. "Undecidability in Number Theory". In: *Notices of the American Mathematical Society* 55.3 (2008), pp. 344–350.

[Qi05] Liqun Qi. "Eigenvalues of a Real Supersymmetric Tensor". In: *Journal of Symbolic Computation* 40.6 (2005), pp. 1302–1324. ISSN: 0747-7171,1095-855X. DOI: 10.1016/j.jsc.2005.05.007.

[QL17] Liqun Qi and Ziyan Luo. *Tensor Analysis*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017. ISBN: 978-1-61197-474-4. DOI: 10.1137/1.9781611974751.

[Sch13] Marcus Schaefer. "Realizability of Graphs and Linkages". In: *Thirty Essays on Geometric Graph Theory*. Springer, 2013, pp. 461–482. URL: https://ovid.cs.depaul.edu/documents/realizability.pdf.

[SCM24] Marcus Schaefer, Jean Cardinal, and Tillmann Miltzow. *The Existential Theory of the Reals as a Complexity Class: A Compendium*. July 2024. DOI: 10.48550/arXiv.2407.18006. arXiv: 2407.18006 [cs.CC].

[SQH15] Jia-Yu Shao, Liqun Qi, and Shenglong Hu. "Some new trace formulas of tensors with applications in spectral hypergraph theory". In: *Linear and Multilinear Algebra* 63.5 (2015), pp. 971–992. DOI: 10.1080/03081087.2014.910208. arXiv: 1307.5690 [math.SP].

[SY10] Amir Shpilka and Amir Yehudayoff. "Arithmetic Circuits: A Survey of Recent Results and Open Questions". In: *Foundations and Trends in Theoretical Computer Science* 5.3–4 (2010), pp. 207–388. DOI: 10.1561/0400000039.

[Sta18] Giovanni Staglianò. "A Package for Computations with Classical Resultants". In: *Journal of Software for Algebra and Geometry* 8.1 (2018), pp. 21–30. DOI: 10.2140/jsag.2018.8.21.

[Tor91] Jacobo Torán. "Complexity Classes Defined by Counting Quantifiers". In: *Journal of the ACM* 38.3 (1991), pp. 753–774. DOI: 10.1145/116825.116858.

[Val79] Leslie G. Valiant. "Completeness Classes in Algebra". In: *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*. Association for Computing Machinery, 1979, pp. 249–261. DOI: 10.1145/800135.804419.

[Wag86] Klaus W. Wagner. "The Complexity of Combinatorial Problems with Succinct Input Representation". In: *Acta Informatica* 23.3 (1986), pp. 325–356. DOI: 10.1007/BF00289117.

A Proofs omitted from Section 3

A.1 Grid assignments

Proof of Lemma 3.3. We first give the encoding on an $m \times m$ grid for an arbitrary $m \times m$ matrix Z, with $m \geq 3$. At position (i, j) use the five states

$$S = \{(0, p, q) : p, q \in \{0, 1\}\} \cup \{(1, 0, 0)\}.$$

For $\boldsymbol{a} = (x, p, q)$, the bit x selects the matrix entry z_{ij}; p records the number of selections earlier in column j, and q the number earlier in row i. The state set ensures that both outgoing counts $p + x$ and $q + x$ are at most one. Use a separate copy of S for each grid position.

For $i, j \in [m]$, write $i^+ = i + 1$ if $i < m$ and $i^+ = 1$ if $i = m$, and define j^+ analogously. For the factor anchored at (i, j), with states $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ at $(i, j), (i^+, j), (i, j^+)$ respectively, set its tensor entry to $z_{ij}^{x(\boldsymbol{a})}$ if all the following conditions hold, and to zero otherwise:

$$\begin{aligned} i = 1 &\implies p(\boldsymbol{a}) = 0, & \qquad i = m &\implies p(\boldsymbol{a}) + x(\boldsymbol{a}) = 1, \\ j = 1 &\implies q(\boldsymbol{a}) = 0, & \qquad j = m &\implies q(\boldsymbol{a}) + x(\boldsymbol{a}) = 1, \\ i < m &\implies p(\boldsymbol{b}) = p(\boldsymbol{a}) + x(\boldsymbol{a}), & \qquad j < m &\implies q(\boldsymbol{c}) = q(\boldsymbol{a}) + x(\boldsymbol{a}). \end{aligned}$$

Here z_{ij}^0 means the constant 1, including when $z_{ij} = 0$. There is no compatibility test across the periodic boundary: the boundary conditions supply the initial and final counts instead. Extend these entries symmetrically and set all other entries to zero. The three grid positions uniquely determine the factor and its anchor, so this symmetric tensor is well defined. Each entry is 0, 1, or z_{ij}, and there are only $5m^2$ tensor indices.

For any state assignment satisfying the tests, the column recurrences give $p(\boldsymbol{a}_{ij}) = \sum_{h<i} x(\boldsymbol{a}_{hj})$, and the final boundary condition gives $\sum_i x(\boldsymbol{a}_{ij}) = 1$. Similarly, each row has exactly one selection. Thus the selected positions form a permutation matrix. Conversely, a permutation matrix uniquely determines all the states by these prefix sums and satisfies every test. Its weight is $\prod_i z_{i,\sigma(i)}$, since the factor anchored at (i, j) contributes exactly the weight of that position. Thus its assignment sum is

$$\sum_{\sigma \in S_m} \prod_{i=1}^{m} z_{i,\sigma(i)} = \operatorname{per}(Z).$$

This proves the polynomial identity. On 0/1 inputs, nonzero-weight assignments correspond bijectively to perfect matchings of the matrix's bipartite graph. For an $n \times n$ input Z, choose the smallest $m = m(n) = 3^r + 1 > n$ with $r \geq 1$, and define $A_m(Z)$ by applying this encoding to

$$Z' = \operatorname{diag}(Z, I_{m-n}).$$

Since $\operatorname{per}(Z') = \operatorname{per}(Z)$, the padding preserves the permanent and 0/1 entries. The smallest such m is $O(n)$, so the padding and encoding have polynomial size. □

A.2 Coefficient expansions and VNP membership

A common coefficient expansion. Let $h_1, \ldots, h_n$ be homogeneous forms of positive assigned degrees $\boldsymbol{d} = (d_1, \ldots, d_n)$, with coefficients in a polynomial ring over $\mathbb{Z}$; zero forms are allowed. Write

$$Q(t) = \operatorname{Res}_{\boldsymbol{d}}(x_1^{d_1} + t h_1, \ldots, x_n^{d_n} + t h_n), \qquad a_s = [t^s] Q(t).$$

Set $\delta = 1 + \sum_i (d_i - 1)$ and $\mathcal{M} = \{\boldsymbol{e} \in \mathbb{Z}_{\geq 0}^n : \sum_i e_i = \delta\}$. Every corresponding monomial is divisible by some $x_i^{d_i}$, so put $i(\boldsymbol{e}) = \min\{i : e_i \geq d_i\}$ and define

$$C_{\boldsymbol{e},\boldsymbol{\beta}} = [\boldsymbol{x}^{\boldsymbol{\beta} - \boldsymbol{e} + d_{i(\boldsymbol{e})} \boldsymbol{e}_{i(\boldsymbol{e})}}] h_{i(\boldsymbol{e})}, \tag{15}$$

where $\boldsymbol{e}_i$ is the ith standard unit vector and a coefficient with a negative exponent is zero. Using the same ordering for rows and columns, row $\boldsymbol{e}$ of $I + tC$ is the coefficient vector of $\boldsymbol{x}^{\boldsymbol{e}-d_{i(\boldsymbol{e})}\boldsymbol{e}_{i(\boldsymbol{e})}}(x_{i(\boldsymbol{e})}^{d_{i(\boldsymbol{e})}} + th_{i(\boldsymbol{e})})$. Let $\mathcal{M}'$ consist of the monomials divisible by at least two distinct pure powers $x_i^{d_i}$, and set $C' = C[\mathcal{M}']$. Macaulay's formula [Sta18, Theorem 1.8 and its preceding construction] gives

$$Q(t) = \frac{P(t)}{R(t)}, \qquad P(t) = \det(I + tC), \qquad R(t) = \det(I + tC'). \tag{16}$$

The constant terms are $P(0) = R(0) = 1$, so the denominator is invertible as a formal power series.

Write $P(t) = \sum_j p_j t^j$ and $R(t) = \sum_j r_j t^j$. The principal-minor expansion gives

$$p_j = \sum_{\substack{S \subseteq \mathcal{M} \\ |S| = j}} \sum_{\pi \in S_j} \operatorname{sgn}(\pi) \prod_{v=1}^{j} C_{\boldsymbol{e}^{(v)}, \boldsymbol{e}^{(\pi(v))}}, \tag{17}$$

where $\boldsymbol{e}^{(1)} < \cdots < \boldsymbol{e}^{(j)}$ lists S; the formula for r_j restricts S to $\mathcal{M}'$. Expanding $1/R(t)$ geometrically yields

$$a_s = \sum_{\substack{j \geq 0,\ r \geq 0,\ k_1, \ldots, k_r \geq 1 \\ j + k_1 + \cdots + k_r = s}} (-1)^r p_j \prod_{u=1}^{r} r_{k_u}. \tag{18}$$

The empty product contributes p_s. After expanding each minor, a term is specified by its sorted row lists and permutations. Their total number of rows is s, and each row index uses $O(n \log(\delta + 1))$ bits. Thus, for unary s, terms have polynomial-length descriptions even though the matrices may be exponentially large. Each term is a signed product of s entries of C.

Symbolic coefficients.

Lemma A.1 (Membership of coefficient families)**.** *For arbitrary polynomially bounded dimensions n and coefficient indices $s \geq 0$, the family $\{\Phi_{n,s}\}$ on symmetric cubic tensors belongs to* VNP *over* $\mathbb{C}$.

Proof. Apply Equation (18) with $d_i = 2$ and $h_i = \sum_{v,w=1}^{n} a_{ivw} x_v x_w$. Then $a_s = \Phi_{n,s}$, $\delta = n + 1$, and each entry of C is zero, a tensor variable, or twice a tensor variable, determined by Equation (15).

A function is in GapP if it is the difference of two #P functions [For97]. By Valiant's coefficient criterion [FPV15, Proposition 1], it suffices to show that

$$(1^n, 1^s, \boldsymbol{e}) \longmapsto [\mathbf{a}^{\boldsymbol{e}}]\Phi_{n,s}(\mathbf{a})$$

is in GapP, where $\mathbf{a}$ lists the independent symmetric tensor entries. The degree is at most s, and there are at most n^3 variables.

Use the polynomial-length term descriptions above, treating each entry $2a$ as two copies of a. Given $(1^n, 1^s, \boldsymbol{e})$, one can check a description, its sign, and whether its product has exponent vector $\boldsymbol{e}$ in polynomial time. Use canonical padding for the lists so that each description has a unique fixed-length encoding. The coefficient is the number of positive descriptions minus the number of negative ones, hence is a GapP function. For arbitrary polynomially bounded choices of n and s, hardwire these parameters in the circuits, as permitted by our nonuniform convention, to obtain the stated VNP membership. □

Together with Theorem 3.2, this proves the VNP-completeness assertion of Theorem 1.7.

B Proofs omitted from Section 4

B.1 Dyadic determinant evaluations

Proof of Lemma 4.4. Start with $A_m(Z)$ and add one dummy index per grid position. On one distinguished triangle, multiply the original entries by α and give the all-dummy entry weight β. On every other

triangle, keep the original entries and give the all-dummy entry weight one. Set mixed original–dummy entries and unsupported entries to zero, and extend by symmetry. Connectivity forces every contributing assignment to use original indices throughout or dummy indices throughout. These sectors have total weights $\alpha \operatorname{per}(Z)$ and β, respectively. The construction has polynomial size and entry bit length. □

Low-degree coefficients. Let $h_1, \dots, h_n$ be explicitly listed integer homogeneous forms of positive assigned degrees $\boldsymbol{d} = (d_1, \dots, d_n)$; zero forms are allowed. Use $Q(t)$, a_s, and the Macaulay matrices from Appendix A.2. For quadratic forms coming from a tensor A, the coefficients a_s are $\Phi_{n,s}(A)$.

Lemma B.1 (Low-degree coefficients)**.** *Given these forms and a unary integer s, the integer a_s is a* GapP *function, has polynomial bit length, and can be computed in* $\mathsf{FP}^{\mathsf{PP}}$.

Proof. Use the term descriptions in Equation (18). Given row and column indices, Equation (15) computes the matrix entry by a coefficient lookup in the explicitly listed h_i. Validity of the indices and membership in $\mathcal{M}'$ are also polynomial-time tests. This remains true for binary-encoded degrees and exponents, since $\log \delta$ is polynomially bounded in the input length.

Each term uses s row indices and has a signed integer weight equal to a product of s input coefficients. Both its description length and its weight's bit length are polynomially bounded. Use fixed-length encodings with canonical padding; invalid encodings have weight zero. Thus a_s is a signed weighted count, hence a GapP function. The same bounds give polynomial bit length for a_s, and such counts can be computed in $\mathsf{FP}^{\mathsf{PP}}$. □

Prime-adic precision.

Lemma B.2 (Prime-adic pure-power systems)**.** *Let p be a supplied prime, and let $f_1, \dots, f_n$ be explicitly listed integer homogeneous forms of a common positive degree d. Suppose every mixed-monomial coefficient is divisible by p, so*

$$f_i \equiv \sum_{j=1}^{n} M_{ij} x_j^d \pmod{p}.$$

Here $M_{ij} = [x_j^d] f_i$ is the integer pure-power coefficient matrix. Given the modulus p^k explicitly in binary, computing $\operatorname{Res}(f_1, \dots, f_n) \bmod p^k$ *is in* $\mathsf{FP}^{\mathsf{PP}}$.

Proof. Write $\boldsymbol{f} = (f_1, \dots, f_n)$ and put $E = d^{n-1}$, so the total coefficient degree is nE. The pure-power coefficient matrix M and the integer E have polynomial-size binary descriptions. Also k is at most the bit length of the explicitly supplied modulus.

Suppose first that $M \bmod p$ is singular. A nonzero row dependence can be normalized to give coefficient one on one selected row. Add the prescribed multiples of the other rows to that row. This is an integer equation transformation of determinant one, after which every coefficient of the selected form is divisible by p. The resultant is unchanged by this row operation and is homogeneous of degree E in that form's coefficients. Hence

$$p^E \mid \operatorname{Res}(f_1, \dots, f_n). \tag{19}$$

If $k \le E$, the requested residue is zero. If $k > E$, then $nE < nk$ is polynomially bounded. Lemma B.1 computes the entire resultant as the coefficient of degree nE in $\operatorname{Res}(x_1^d + z f_1, \dots, x_n^d + z f_n)$; reduce it modulo p^k.

Suppose instead that $M \bmod p$ is invertible. Compute an integer lift U of $M^{-1} \bmod p^k$ by modular Gaussian elimination. Its determinant is a unit modulo p^k. Equal-degree resultants satisfy the equation-change identity

$$\operatorname{Res}(U\boldsymbol{f}) = \det(U)^E \operatorname{Res}(\boldsymbol{f}).$$

It follows from invariance under elementary row additions and homogeneity under row scaling, and hence holds as an integer polynomial identity. Reduce the coefficients of $U\boldsymbol{f}$ modulo p^k. The pure-power

coefficients become those of $(x_1^d, \dots, x_n^d)$; the mixed coefficients remain divisible by p. Therefore the reduced system has the form $(x_i^d + ph_i)_{i=1}^n$, with integer forms h_i of polynomial description size.

Let $a_r = [z^r] \operatorname{Res}(x_1^d + zh_1, \dots, x_n^d + zh_n)$. Integrality of the resultant gives

$$\operatorname{Res}(\boldsymbol{f}) \equiv \det(U)^{-E} \sum_{r=0}^{k-1} a_r p^r \pmod{p^k}. \tag{20}$$

There are only k coefficients to compute, each covered by Lemma B.1. The inverse determinant and its binary exponentiation are polynomial-time modular operations. Thus the whole computation is in $\mathsf{FP}^{\mathsf{PP}}$. □

C Proofs omitted from Section 5

Lemma C.1. *Given positive integers $a_1, \dots, a_r$ and a positive target T in binary, counting subsets with sum exactly T is #P-complete under parsimonious reductions. Deciding whether this count is odd is $\oplus$P-hard under polynomial-time many-one reductions.*

Proof. Given a 3-CNF formula, construct an exact subset-sum instance as follows. Use one base-ten digit position per variable and per clause. For each variable make two items, corresponding to its two truth values. Each has digit one at that variable and, at each clause, the number of literal occurrences made true by that value. For each clause also make two slack items with only that clause digit nonzero, respectively one and two. The target has digit one at each variable and four at each clause.

There are no carries: even selecting all items gives at most two in a variable column and six in a clause column. The target therefore forces exactly one truth item for each variable. A clause with j true occurrences needs slack $4 - j$; for $j = 0$ this is impossible, and for $j = 1, 2, 3$ it has exactly one representation by a subset of the two slack items. Thus subsets of the target sum correspond bijectively to satisfying assignments. All items and the target are positive and have polynomial bit length. One may ensure a nonempty variable set by adjoining a variable forced to be true.

The counting and parity hardness of 3-SAT follow directly from their witness definitions: a polynomial-time Boolean predicate on witness bits has a polynomial-size Boolean circuit; introduce a variable for each gate, impose its truth table by clauses of width at most three, and require the output to be true. Each accepted witness has exactly one extension to the gate variables, so this transformation preserves the count.

The exact subset count is in #P by guessing the subset. □

Proof of Lemma 5.1. For the exact subset count $N_{\boldsymbol{a}}(T)$ from Lemma C.1, append an item of weight T to $\boldsymbol{a}$ to obtain $\boldsymbol{b}$, and then an item of weight one to obtain $\boldsymbol{w}$. Since all original weights are positive, the new item of weight T contributes exactly one additional subset of sum T: $N_{\boldsymbol{b}}(T) = N_{\boldsymbol{a}}(T) + 1$. Therefore

$$K_{\boldsymbol{w}}(T) = K_{\boldsymbol{b}}(T) + K_{\boldsymbol{b}}(T-1) = N_{\boldsymbol{a}}(T) + 1 + 2K_{\boldsymbol{b}}(T-1).$$

Thus $K_{\boldsymbol{w}}(T)$ is even exactly when $N_{\boldsymbol{a}}(T)$ is odd. The added weights have polynomial bit length, proving the claim. □

D Proofs omitted from Section 6

D.1 Polynomial arithmetic in CH

We use a common transfer principle for both proofs. Put $L = |w|$. For polylogtime-uniform, constant-depth threshold circuits of size $2^{\mathsf{poly}(L)}$, gate addresses have polynomial length and the wiring is computable in polynomial time. If their input bits have fixed-level CH access, so do their output bits: apply the gate-depth induction of [AGS26, Lemma 6.3], using the input predicate at the leaves. For circuits

of polynomial size in the expanded input length, this is precisely the composition principle of [AGS26, Lemmas 6.5–6.6]. A fixed number of such stages, together with polynomial-time fixed-level oracle computations, stays in CH by Lemma 6.2. Polynomial-length circuit parameters and output addresses may themselves be computed with a fixed-level CH oracle and then supplied as auxiliary inputs to the uniform evaluator. If their computation uses a $\mathsf{C}_j\mathsf{P}$ oracle and the evaluator's bit predicate lies in $\mathsf{C}_k\mathsf{P}$, the composition lies in

$$\mathsf{P}^{\mathsf{C}_{\max\{j,k\}}\mathsf{P}} \subseteq \mathsf{C}_{\max\{j,k\}+1}\mathsf{P}$$

by Lemma 6.2.

For weakly division-free, constant-free arithmetic circuits of constant depth, [AGS26, Lemma 6.1] supplies the threshold simulation and bounds the numerator and denominator bit lengths by a polynomial in the input bit-length bound and circuit size. These bounds remain singly exponential here. For polynomial-size threshold circuits, their output lengths give the corresponding bounds directly. We always allow unreduced fractions and make their denominators positive. Thus it suffices below to identify the uniform circuits and check the degree and size bounds.

Proof of Lemma 6.7. Fix effective bounds $D, H, M \leq 2^{\mathsf{poly}(L)}$ on input degrees, coefficient numerator and denominator bit lengths, and the number of summands or factors. The padded coefficient arrays have singly exponential length and compact bit access, uniformly in all indices.

For Lemmas 6.7(1) and 6.7(2), clear positive common denominators by iterated integer multiplication and division, then use integer addition and iterated polynomial multiplication. These operations have uniform TC^0 circuits by [HAB02, Corollaries 6.3 and 6.5]; see also [AGS26, Theorem 2.12 and Section 2.3] for the rational representations. Their sizes and output lengths are polynomial in the expanded input length. The degrees are at most D for sums and MD for products, so the transfer principle applies.

For Lemma 6.7(3), use

$$[y^e]F_w^{(r)} = [y^{e+r}]F_w \prod_{a=1}^{r}(e+a).$$

Return zero if $e + r > D$; otherwise the product has at most D factors, each of polynomial bit length, and is covered by iterated integer multiplication. The empty product is 1.

For Lemmas 6.7(6) and 6.7(7), a coefficient is zero exactly when all its numerator bits vanish. Quantification over polynomial-length coefficient and bit addresses therefore tests whether the polynomial is zero or has a nonzero coefficient above a given index. Binary search finds its degree using polynomially many fixed-level oracle queries; a nonzero coefficient's sign is its stored sign. Monic normalization in Lemma 6.7(4) then divides each coefficient by the leading coefficient, using integer multiplication on their rational representations.

For Lemma 6.7(5), clear denominators in F_w, F_w' and apply the uniform GCD circuits of [AGS26, Lemma 6.13]. After clearing denominators in the resulting gcd, compute its actual degree using Lemma 6.7(6). Use this as the divisor-degree parameter for integer pseudodivision [AGS26, Theorem 2.12(4)], so the designated leading divisor coefficient is nonzero. The remainder is zero, and the pseudoquotient is a nonzero scalar multiple of $F_w/\gcd(F_w, F_w')$. Monic normalization removes this scalar and yields the required squarefree part. These operations have polynomial-size circuits in their expanded inputs and do not increase polynomial degrees. Only a fixed number of stages is used, so all output families satisfy the effective bounds of Definition 6.3 and have compact coefficient access. □

Proof of Lemma 6.9. Fix effective bounds $D, H \leq 2^{\mathsf{poly}(L)}$ on input degrees and coefficient numerator and denominator bit lengths. For Lemma 6.9(1), determine the degrees in x over $\mathbb{Q}[t]$ by the zero tests and binary search used above, now quantifying also over coefficient indices in t. With these degrees, [AGS26, Lemma 6.10] supplies polylogtime-uniform TC^0 circuits over $\mathbb{Q}[t]$, of polynomial size in the expanded input length. The transfer principle gives compact access and singly exponential coefficient bit lengths. The Sylvester matrix gives $\deg_t R_w \leq 2D^2$. If one input has degree zero in x, the resultant is that input raised to the degree of the other; if both do, it is 1. These cases follow from Lemma 6.7.

For Lemma 6.9(2), put

$$P = \sum_{i=1}^{n} \prod_{j \neq i} d_j \leq 2^{\mathsf{poly}(L)}.$$

Multihomogeneity of the resultant gives $\deg_t R_w \leq PD$. Since the d_i are bounded by a fixed constant, each form has polynomially many monomials in $\boldsymbol{x}$; including the coefficients in t, its expanded representation has singly exponential length. For $n \geq 2$, [AGS26, Corollary 5.13] supplies polylogtime-uniform, weakly division-free, constant-free arithmetic circuits of constant depth for the resultant coefficients. Their size is at most $(P(D+1))^{\mathsf{poly}(n)} = 2^{\mathsf{poly}(L)}$. The arithmetic simulation and transfer principle above give compact access and effective singly exponential numerator and denominator bit-length bounds. For $n = 1$, the resultant of $a(t)x_1^{d_1}$ is simply $a(t)$. Thus both output families satisfy Definition 6.3, with access uniform in w. □

D.2 Constructing a finite set witnessing feasibility

A function $P\colon \mathbb{R}^n \to \mathbb{R}$ is *coercive* if $P(\boldsymbol{z}) \to +\infty$ as $\|\boldsymbol{z}\| \to \infty$.

We state the minimum bound of Jeronimo, Perrucci, and Tsigaridas [JPT13] in their formulation and derive Lemma 6.11 from it.

Theorem D.1 ([JPT13, Theorem 2.10])**.** *Let $n \geq 2$ and let d be a positive even integer. Let $T = \{\boldsymbol{x} \in \mathbb{R}^n \mid f_1(\boldsymbol{x}) = \cdots = f_l(\boldsymbol{x}) = 0, f_{l+1}(\boldsymbol{x}) \geq 0, \ldots, f_m(\boldsymbol{x}) \geq 0\}$ be defined by polynomials $f_1, \ldots, f_m \in \mathbb{Z}[x_1, \ldots, x_n]$ with degrees bounded by d and coefficients of absolute value at most H, and let C be a connected component of T. Let $g \in \mathbb{Z}[x_1, \ldots, x_n]$ be a polynomial of degree $d_0 \leq d$ and coefficients of absolute value bounded by $H_0 \leq H$, and let $g_{\min,C}$ be the minimum value that g takes over C. Assume that the set $C_{\min} = \{\boldsymbol{z} \in C \mid g(\boldsymbol{z}) = g_{\min,C}\}$ is compact. Then $g_{\min,C}$ is an algebraic number of degree at most $2^{n-1}d^n$ and, if it is not zero, its absolute value is greater than or equal to $(2^{4-\frac{n}{2}}\tilde{H}d^n)^{-n2^n d^n}$, where $\tilde{H} = \max\{H, 2n + 2m\}$.*

Proof of Lemma 6.11. Since P is continuous, each sublevel set $\{\boldsymbol{x} \in \mathbb{R}^n : P(\boldsymbol{x}) \leq c\}$ is closed, and by coercivity there is R with $P(\boldsymbol{x}) > c$ for all $\|\boldsymbol{x}\| > R$, so it is contained in the closed ball of radius R, hence bounded and compact. Let $m = \inf_{\mathbb{R}^n} P$. Then m is finite: $m \geq 0$ since $P \geq 0$, and $m \leq P(0) < \infty$ since P is real-valued. By definition of the infimum, for each $k \geq 1$ there is $\boldsymbol{x}_k$ with $P(\boldsymbol{x}_k) < m + 1/k$; this $(\boldsymbol{x}_k)$ is a minimizing sequence, with $P(\boldsymbol{x}_k) \to m$. It eventually lies in the compact set $\{P \leq m + 1\}$, so some subsequence converges to a point $\boldsymbol{x}^\star$, and continuity gives $P(\boldsymbol{x}^\star) = m$. Thus the minimum is attained and the minimizer set $P^{-1}(\{m\})$ is nonempty; it is closed and contained in $\{P \leq m + 1\}$, hence compact. If P has no real zero then, P being nonnegative, $\min_{\mathbb{R}^n} P > 0$. Let 2^h with $h \geq 1$ bound the absolute values of the integer coefficients. Apply Theorem D.1 with $T = C = \mathbb{R}^n$ (using the tautological inequality $1 \geq 0$), $n \geq 2$ by the global assumption of Section 6, $g = P$ (of degree at most 4), $d = 4$, and $H = 2^h$. With $\widetilde{H} = \max(2^h, 2n + 2)$ this bounds a nonzero minimum from below by $(2^{4-n/2}\widetilde{H}\,4^n)^{-n8^n}$, so a sufficient choice is

$$B = n8^n\big(h + \lceil \log_2(2n + 2)\rceil + 2n + 6\big). \tag{21}$$

This integer is computable in polynomial time and has polynomial binary length, so $B \leq 2^{\mathsf{poly}(L)}$. □

Polynomial-time access to the initial families. Recall the polynomials Q_u, H_u, and $G_{u,i}$ constructed in Section 6. As in that section, write $L = |u|$ and suppress u when discussing a fixed instance.

The reduction in Lemma 6.10 is polynomial-time, and the $n = 1$ case is decided in polynomial time. Forming the quartic Q from the explicit quadratic forms also takes polynomial time. These initial coefficient bits are therefore accessible in $\mathsf{P} = \mathsf{C}_0\mathsf{P}$.

The explicit formula in Equation (21) computes B in polynomial time in binary. The dyadic $\gamma = 2^{-B}$ has P-decidable addressed bits: its numerator is 1, and its denominator has its only nonzero bit at position B. The constants $\gamma/4$ and $\gamma/2$ are obtained by shifting this position. The family $(H_u)_u$ still has coefficient-bit access in P: its coefficients of degree at most four are those of Q_u, and its only additional

nonzero coefficients are $2^{-(B+2)}$ at the monomials x_i^6. These degree ranges are disjoint, and the new denominator has just one nonzero bit. Its degree is six and its coefficient bit lengths are $O(B + \mathsf{poly}(L))$; individual addressed bits are computable in polynomial time even though the full denominators may be exponentially long.

The whole family $(G_{u,i})_{u,i}$ also has coefficient-bit access in P. Since $Q(0) = 1$, the only coefficient changed in passing from H to G_1 is the constant term,

$$1 - \gamma/2 = \frac{2^{B+1} - 1}{2^{B+1}},$$

whose numerator has exactly $B + 1$ consecutive 1-bits. For $i \geq 2$,

$$G_i = \partial_i Q + \frac{3}{2^{B+1}} x_i^5.$$

The coefficients of $\partial_i Q$ are polynomial-length integers computable in polynomial time and occur only in degrees at most three. The additional degree-five coefficient has numerator 3 and a power-of-two denominator. Thus each addressed coefficient bit is computable in polynomial time, uniformly in (u, i), with the same singly exponential bit-length bounds.

D.3 Constructing the univariate polynomial preserving feasibility

Proof of Lemma 6.15. For valid indices, $a < K(u) = 6n \cdot 5^{n-1} + 1$ has polynomial bit length in $|u|$, as do the powers a^i for $1 \leq i \leq n$. The coefficients of the G_i have polynomial-time bit access, and homogenization only reindexes those coefficients. The coefficients of $tx_0 - \ell_a(\boldsymbol{x})$ are also computable in polynomial time, uniformly in (u, a). Each $R_{u,a}$ is one parameterized resultant with polynomially many variables, constant assigned degrees $6, 5, \ldots, 5, 1$, and the single parameter t. The degree $6 \cdot 5^{n-1}$ and the resultant coefficient-bit bounds from Lemma 6.9 are singly exponential. Apply Lemma 6.9(2) with index (u, a). The input is the homogeneous family $(G^h_{u,1}, \ldots, G^h_{u,n}, tx_0 - \ell_a(\boldsymbol{x}))$, viewed as forms in $x_0, \ldots, x_n$ with the parameter t; the output is the univariate family $(R_{u,a}(t))_{u,a}$. The lemma supplies each addressed coefficient bit of this output family in one fixed level of CH, uniformly in (u, a). □

D.4 Univariate real-root testing

Proof of Lemma 6.19. Suppose every coefficient of p has a numerator and positive denominator of bit lengths at most H'. Then every derivative $p^{(j)}$, $0 \leq j \leq D$, admits coefficient representations with numerator and positive denominator bit lengths at most $H = H' + D\lceil \log_2(D+1) \rceil + 2$. Let $g \in \mathbb{Q}[x]$ have degree at most d. Suppose every coefficient numerator and positive denominator of both polynomials has magnitude at most 2^H, where $H \geq 1$. We will show that the integer

$$U = 8(D + d + 2)^2 \big(H + \lceil \log_2(D + d + 2) \rceil + 1\big)$$

satisfies $2^{-U} \leq |g(\alpha)| \leq 2^U$ at every complex root α of p with $g(\alpha) \neq 0$.

Clear denominators separately, writing $P = ap$ and $G = bg$ with positive integers a, b that are products of the individual denominators. The nonzero integer polynomial $\mathrm{Res}_x(P, by - G)$ has every $g(\alpha)$ as a root. Its Sylvester matrix has size at most $D + d$; the entries are affine polynomials in y. Each entry has coefficient absolute sum bounded by $2^{(D+d+3)H+1}$. The determinant expansion therefore bounds every resultant coefficient by $(D + d)!\, 2^{(D+d)((D+d+3)H+1)}$, which is at most 2^{U-2}. Remove any power of y from this resultant. Its constant and leading coefficients are now nonzero integers. The elementary Cauchy bound gives $|\rho| \leq 1 + 2^{U-2}$ for its roots; applying the same bound to the reciprocal polynomial gives $|\rho| \geq (1 + 2^{U-2})^{-1}$. These imply the asserted bounds. The constant-polynomial case for g obeys the same estimate.

Applying this bound with $g = p^{(j)}$ and replacing the degrees and coefficient bit lengths by the family's fixed effective upper bounds gives a common $U \leq 2^{\mathsf{poly}(L)}$ computable in binary in polynomial time from w. □

Proof of Lemma 6.20. Set $k = \lceil \log_9(2/\varepsilon) \rceil$ and define

$$P_+(x) = \prod_{j=-U}^{U} (x + 2^j)^{2k}, \qquad P_-(x) = \prod_{j=-U}^{U} (x - 2^j)^{2k}.$$

Let $\mathcal{N} = P_+ - P_-$, $\mathcal{D} = P_+ + P_-$, and $\mathcal{S} = \mathcal{N}/\mathcal{D}$. These polynomials have rational coefficients and degree at most $2k(2U+1) = O(U \log(1/\varepsilon))$. For real λ, both $P_\pm(\lambda)$ are nonnegative and cannot vanish simultaneously. Thus $\mathcal{D}(\lambda) > 0$ and $|\mathcal{S}(\lambda)| \le 1$. Moreover $P_+(0) = P_-(0)$, so $\mathcal{S}(0) = 0$. For $2^{-U} \le \lambda \le 2^U$, choose a dyadic scale $a = 2^j \in [\lambda/2, \lambda]$ with $-U \le j \le U$. Then $|(\lambda - a)/(\lambda + a)| \le 1/3$, while every other factor ratio is at most one. Consequently

$$|\mathcal{S}(\lambda) - 1| = \frac{2P_-(\lambda)}{P_+(\lambda) + P_-(\lambda)} \le 2 \prod_{j=-U}^{U} \left| \frac{\lambda - 2^j}{\lambda + 2^j} \right|^{2k} \le 2 \cdot 9^{-k} \le \varepsilon.$$

The negative case follows from $P_+(-\lambda) = P_-(\lambda)$ and $\mathcal{S}(-\lambda) = -\mathcal{S}(\lambda)$.

To bound the coefficient bit lengths, multiply $P_\pm$ by the common positive denominator $C = 2^{kU(U+1)}$. This gives the integer polynomials

$$CP_\pm(\lambda) = \prod_{r=1}^{U} (2^r \lambda \pm 1)^{2k} \prod_{j=0}^{U} (\lambda \pm 2^j)^{2k}.$$

The sum of the absolute values of the coefficients of a product is at most the product of the corresponding sums for its factors. Thus every coefficient of $CP_\pm$ has absolute value at most

$$\prod_{r=1}^{U} (2^r + 1)^{2k} \prod_{j=0}^{U} (1 + 2^j)^{2k} \le 2^{O(k(U+1)^2)}.$$

The same bound holds for $C\mathcal{N}$ and $C\mathcal{D}$, and C itself has $O(k(U+1)^2)$ bits. Since $k = O(\log(1/\varepsilon))$ and $U \ge 1$, these representations give the claimed coefficient bit-length bounds. □

Proof of Lemma 6.22. Write $v = (w, \boldsymbol{\eta})$ for the combined index and put $s = t + 1$. By Lemma 6.7(3), the selected derivatives $h_i = p_w^{(j_i)}$ have compact coefficient access uniformly in v and i; the last test is $h_{t+1} = 1$. The requested signs are read from $\boldsymbol{\eta}$, followed by 1. Validity of the derivative indices can be checked using Lemma 6.7(6); invalid indices may be assigned a fixed constant polynomial.

Choose U from the family's effective bounds as in the proof of Lemma 6.19. It is computable in polynomial time in binary and is singly exponential in $|v|$. For $\varepsilon = 1/(4s)$, the construction in Lemma 6.20 uses

$$P_\pm(h_i(x)) = \prod_{j=-U}^{U} (h_i(x) \pm 2^j)^{2k}, \qquad k = \lceil \log_9(8s) \rceil = O(\log(s+1)).$$

The dyadic constants $\pm 2^j$ have polynomial-time coefficient-bit access. By Lemmas 6.7(1) and 6.7(2), these products therefore have compact coefficient access: first form the factors $h_i(x) \pm 2^j$, then take one iterated product, repeating each factor $2k$ times. All factor indices have polynomial length in $|v|$. Taking the difference and sum gives $\mathcal{N}(h_i)$ and $\mathcal{D}(h_i)$. The formulas for A, B, and $\Psi_{\wedge,\sigma} = 2A - B$ in Lemma 6.21 use only a fixed number of further sum and product stages.

For completeness, let D_0 and H_0 be effective upper bounds on the degrees and coefficient numerator and denominator bit lengths of the h_i. Each dyadic product has degree at most $2k(2U+1)D_0$. Clearing the denominators of its factors and bounding the sum of the absolute values of the coefficients shows that its coefficient bit lengths are polynomial in k, U, D_0, H_0. The squares, products, and sum in A and B give final degree and bit-length bounds polynomial in s, k, U, D_0, H_0, hence singly exponential in $|v|$. Thus

all intermediate polynomials satisfy the required size bounds. Since the number of arithmetic stages is fixed, the resulting family has compact coefficient access in a fixed level of CH, uniformly in $(w, \boldsymbol{\eta})$.

The same argument applies to any polynomially many test polynomials with compact coefficient access, provided the requested signs and binary U are computable in polynomial time from the input index. □

Proof of the complexity claim in Corollary 6.23. By Lemma 6.22, the family $(\Psi_{w,\boldsymbol{\eta}})_{w,\boldsymbol{\eta}}$ has compact coefficient access. Applying Lemma 6.9(1) to $p_w(x)$ and $\Psi_{w,\boldsymbol{\eta}}(x) + t$ gives compact coefficient access to $\mathcal{R}_{w,\boldsymbol{\eta}}(t)$, uniformly in $(w, \boldsymbol{\eta})$.

The univariate product formula shows that $\mathcal{R}_{w,\boldsymbol{\eta}}$ is monic of degree $\deg p_w$, so a nonzero coefficient exists. By Lemmas 6.2 and 6.7(7), testing whether any coefficient at or below a given index is nonzero belongs to CH. Binary search over the effective degree bound therefore finds the least nonzero coefficient index ν in polynomial time with an oracle in a fixed level of CH. Finally, Lemma 6.7(7) determines whether $[t^\nu]\mathcal{R}_{w,\boldsymbol{\eta}} < 0$. These computations use only a fixed number of counting levels, proving the claim. □

E Parity hardness of univariate resultant signs

The same sign approximation and resultant-sign principle also give a lower bound for two univariate polynomials represented by arithmetic circuits.

An *integer constant-free arithmetic circuit* has inputs x and 1 and gates for addition, subtraction, and multiplication. It computes a polynomial in $\mathbb{Z}[x]$; constants such as 0, 2, and 3 are constructed from 1. We also use the straight-line-program (SLP) terminology. The negative and positive resultant-sign problems ask whether $\operatorname{Res}_x(p, g) < 0$ or $\operatorname{Res}_x(p, g) > 0$, respectively, for nonzero polynomials of positive degree given by such circuits.

We reduce from Parity Circuit SAT, which asks whether $\#\mathrm{SAT}(C) = \sum_{\boldsymbol{b}\in\{0,1\}^m} C(\boldsymbol{b})$ is odd for a Boolean circuit C with m inputs and s gates. We use fan-in-two AND and OR gates, unary NOT gates, and Boolean constants. We may assume $m, s \geq 1$: a dummy input constrained to zero preserves the number of satisfying assignments, and a dummy gate can be inserted if needed. Parity Circuit SAT is $\oplus$P-complete: accepting computation paths are encoded bijectively by satisfying assignments of a polynomial-size Boolean circuit, with unused nondeterministic choices forced to zero.

Theorem E.1. *The negative and positive resultant-sign problems for univariate integer polynomials given by constant-free arithmetic circuits are $\oplus$P-hard under deterministic polynomial-time many-one reductions, even with the promise that the resultant is nonzero. More precisely, from a Boolean circuit C on m inputs with s gates one can construct, in time polynomial in $m + s$, constant-free circuits for $p_m, \Phi_C \in \mathbb{Z}[x]$ such that*

$$\begin{aligned} p_m &= T_{2^m}, & \operatorname{size}(p_m) &= O(m), \\ \operatorname{Res}_x(p_m, \Phi_C) &\neq 0, & \operatorname{size}(\Phi_C) &= O\big(m(m+s)+s\big), \\ \operatorname{sgn}\operatorname{Res}_x(p_m, \Phi_C) &= (-1)^{\#\mathrm{SAT}(C)}. \end{aligned}$$

Both polynomials have positive degree. The 2^m roots of p_m are real and simple, and p_m depends only on m.

E.1 Chebyshev roots and Boolean assignments

Define

$$R_0(x) = x, \qquad R_{i+1}(x) = 2R_i(x)^2 - 1. \tag{22}$$

Thus $R_i = T_{2^i}$, and the shared circuit for $R_0, \ldots, R_m$ has size $O(m)$. Induction gives $\deg R_m = 2^m$ and $\operatorname{lc}(R_m) = 2^{2^m-1} > 0$.

Lemma E.2. *Put $p_m = R_m$ and $Z = \{\alpha \in \mathbb{R} : p_m(\alpha) = 0\}$. The polynomial p_m has 2^m simple roots, all in $(-1, 1)$, and*

$$\alpha \longmapsto \boldsymbol{b}(\alpha), \qquad b_i(\alpha) = \mathbf{1}[R_i(\alpha) < 0] \quad (0 \le i < m),$$

is a bijection from Z to $\{0,1\}^m$. At every such root,

$$2^{-(m-i)} \le |R_i(\alpha)| \le 1 \qquad (0 \le i < m). \tag{23}$$

Proof. Write $q(u) = 2u^2 - 1$. Given $\boldsymbol{b} \in \{0,1\}^m$, set $y_m = 0$ and, working backwards, put

$$y_i = (-1)^{b_i}\sqrt{\frac{1 + y_{i+1}}{2}} \qquad (i = m-1, \dots, 0).$$

Every y_i with $i < m$ lies in $(-1, 1) \setminus \{0\}$, and $q(y_i) = y_{i+1}$. Hence $\alpha = y_0$ satisfies $R_i(\alpha) = y_i$ and $p_m(\alpha) = 0$, with precisely the prescribed signs. Different sign vectors give different roots. Their number equals $\deg p_m = 2^m$, so these are all the roots and they are simple.

For the separation bound, let $u = R_i(\alpha)$ and $r = m - i \ge 1$. The map q is 4-Lipschitz on $[-1, 1]$, maps that interval into itself, and $|q^{\circ r}(0)| = 1$. Since $q^{\circ r}(u) = 0$,

$$1 = |q^{\circ r}(u) - q^{\circ r}(0)| \le 4^{r-1}|q(u) - q(0)| = 2 \cdot 4^{r-1}|u|^2.$$

Thus $|u| \ge 2^{1/2-r} \ge 2^{-r}$, proving the claim. □

E.2 A polynomial for the Boolean predicate

For a real root α of p_m, the sign tests $R_i(\alpha) < 0$ encode its Boolean assignment. We approximate these indicators and arithmetize C, then clear positive denominators to obtain a polynomial negative exactly when $C(\boldsymbol{b}(\alpha)) = 1$. This uses the dyadic approximation from Appendix D.4, with $U = m$ and accuracy $2^{-(s+5)}$.

Put $k = s + 5$ and define integer polynomials

$$P_{\pm}(t) = \left(\prod_{j=-m}^{m} (2^m t \pm 2^{m+j})\right)^{2k}, \qquad A = P_-, \qquad D = P_+ + P_-, \qquad r = \frac{A}{D}. \tag{24}$$

Scaling each dyadic factor by 2^m clears its denominator and leaves r unchanged. The even powers give $P_{\pm}(t) \ge 0$ for real t; their zero sets are disjoint, so $D(t) > 0$ and $0 \le r(t) \le 1$. Moreover,

$$|r(t) - \mathbf{1}[t < 0]| \le 9^{-k} \le 2^{-(s+5)} \qquad \text{if } 2^{-m} \le |t| \le 2^m. \tag{25}$$

This is the approximation estimate proved in that appendix.

Set $B_i(x) = r(R_i(x))$. By Lemma E.2, at every $\alpha \in Z$,

$$0 \le B_i(\alpha) \le 1, \qquad |B_i(\alpha) - b_i(\alpha)| \le \varepsilon := 2^{-(s+5)}.$$

Replace the inputs of C by the B_i and its gates by

$$\mathsf{NOT}(u) \mapsto 1 - u, \qquad \mathsf{AND}(u, v) \mapsto uv, \qquad \mathsf{OR}(u, v) \mapsto u + v - uv,$$

obtaining a rational function H_C. These operations preserve $[0, 1]$. On that interval, each gate's error is at most the sum of its incoming errors: for AND use $uv - u'v' = (u - u')v + u'(v - v')$, and for OR use $\mathsf{OR}(u, v) = 1 - (1 - u)(1 - v)$. A topological induction through the s gates therefore gives

$$|H_C(\alpha) - C(\boldsymbol{b}(\alpha))| \le 2^s\varepsilon = \frac{1}{32}, \qquad 0 \le H_C(\alpha) \le 1. \tag{26}$$

This bound allows gates to reuse earlier outputs.

We represent each rational function by a pair (N, D) of integer polynomials with $D(x) > 0$ for all real x. The input B_i has pair $(A(R_i), D(R_i))$, and the Boolean constants have pairs $(0, 1)$ and $(1, 1)$. For $u = N_u/D_u$ and $v = N_v/D_v$, use the exact updates

operation	numerator–denominator pair
$1 - u$	$(D_u - N_u,\ D_u)$
uv	$(N_u N_v,\ D_u D_v)$
$u + v - uv$	$(N_u D_v + N_v D_u - N_u N_v,\ D_u D_v)$.

Every denominator remains positive on $\mathbb{R}$; it need not be constant. The resulting pair (N_C, D_C) satisfies $H_C = N_C/D_C$. Define

$$\Phi_C(x) = (1 + x^2)\big(D_C(x) - 2N_C(x)\big). \tag{27}$$

For a root $\alpha \in Z$, both $1 + \alpha^2$ and $D_C(\alpha)$ are positive, so $\Phi_C(\alpha)$ has the same sign as $1 - 2H_C(\alpha)$. By Equation (26), if the assignment $\boldsymbol{b}(\alpha)$ satisfies C, then $H_C(\alpha) \geq 31/32$ and $\Phi_C(\alpha) < 0$. If it does not satisfy C, then $H_C(\alpha) \leq 1/32$ and $\Phi_C(\alpha) > 0$. Thus Φ_C is negative exactly at the roots corresponding to satisfying assignments. In particular, $|1 - 2H_C(\alpha)| \geq 15/16$, and

$$\operatorname{sgn} \Phi_C(\alpha) = (-1)^{C(\boldsymbol{b}(\alpha))}, \qquad \Phi_C(\alpha) \neq 0 \quad (\alpha \in Z). \tag{28}$$

In particular $D_C - 2N_C$ is not the zero polynomial. The positive factor $1 + x^2$ ensures $\deg \Phi_C \geq 2$, including when C is constant.

The construction uses only constant-free arithmetic gates. Build the powers of two in Equation (24) by successive doubling, and form both products at each R_i before raising them to $2k$ by repeated squaring. This uses $O(m^2 + m \log(s + 1))$ gates in total. The pair updates add $O(s)$ gates, so the size is $O(m(m + s) + s)$ as claimed. All gates and wires can be listed in polynomial time; intermediate polynomials and integer constants are stored as circuits and are never expanded.

E.3 The resultant sign

Proof of Theorem E.1. The preceding construction gives the circuit bounds and positive degrees. All roots of p_m are real and simple, and $\operatorname{lc}(p_m) > 0$. Apply Corollary 5.3 with $n = 1$, $F_1 = p_m$, and $\Psi = \Phi_C$. The homogeneous resultant there is the ordinary univariate resultant when the actual degrees are used for homogenization. By Equation (28) the resultant is nonzero, and the sign bijection in Lemma E.2 gives

$$\operatorname{sgn} \operatorname{Res}_x(p_m, \Phi_C) = \prod_{\alpha \in Z} (-1)^{C(\boldsymbol{b}(\alpha))} = (-1)^{\sum_{\boldsymbol{b} \in \{0,1\}^m} C(\boldsymbol{b})} = (-1)^{\#\mathrm{SAT}(C)}.$$

Thus odd satisfying-assignment count is equivalent to a negative resultant.

For strict positivity, let E recognize the all-zero assignment and set $C^\star = C \oplus E$. This adds $O(m)$ Boolean gates and changes the output at exactly one assignment. Hence $\#\mathrm{SAT}(C^\star) \equiv \#\mathrm{SAT}(C) + 1 \pmod 2$. Applying the construction to $C^\star$ reverses the resultant sign, so odd satisfying-assignment count is equivalent to a positive resultant. The circuit bounds and nonvanishing promise are preserved. □

The result concerns polynomials represented by constant-free arithmetic circuits. It does not require or assert efficient access to individual bits of their expanded coefficients, nor membership of these sign problems in $\oplus\mathsf{P}$.

F Hilbert Nullstellensatz in AM

Proposition F.1. *Let $\boldsymbol{f} = (f_1, \ldots, f_r)$ be a system of polynomials in $\mathbb{Z}[x_1, \ldots, x_n]$, given in sparse binary encoding. Deciding whether $\boldsymbol{f}$ has a solution over $\mathbb{C}$ can be done in* AM*, assuming the QRH bound in [CT26, Theorem 1.1].*

The proof follows from the methods of Koiran [Koi96] and a recent estimate [FSW26] that depends on [CT26, Theorem 1.1]. In order to prove the above proposition, we first recall these results.

Theorem F.2 ([FSW26, Theorem 1.1])**.** *Let K be a number field with degree $d_K = [K : \mathbb{Q}]$, absolute discriminant $D_K = |\operatorname{disc}(K)|$, and ring of integers $\mathcal{O}_K$. Let $\pi_K(x)$ be the number of nonzero prime ideals of $\mathcal{O}_K$ of norm at most x. Assuming [CT26, Theorem 1.1], for all $x > 2$ we have*

$$|\pi_K(x) - \operatorname{Li}(x)| \le Cx^{31/32}\big(\log D_K + d_K \log x\big),$$

where $C > 0$ is an absolute effectively computable constant and $\operatorname{Li}(x)$ is the logarithmic integral $\int_2^x \frac{dt}{\ln(t)}$.

We now briefly recall the methods used by Koiran in [Koi96]. For $x \ge 2$, let

$$S_{\boldsymbol{f}}(x) \coloneqq \{p \le x : \ p \text{ is prime and } \boldsymbol{f} \text{ has a common zero over } \mathbb{F}_p\}.$$

The following theorem is the combination of results in [Koi96, Theorems 4, 5, and 7 and Lemma 3].

Theorem F.3. *Let $\boldsymbol{f} = (f_1, \dots, f_r) \in \mathbb{Z}[x_1, \dots, x_n]^r$ have sparse binary input length N, and let $S_{\boldsymbol{f}}(x)$ be as above. There is a polynomial q, independent of $\boldsymbol{f}$, such that, with $B = 2^{q(N)}$, the following hold. If $\boldsymbol{f}$ has no common zero in $\mathbb{C}^n$, then for each $x \ge 2$,*

$$|S_{\boldsymbol{f}}(x)| \le B.$$

If $\boldsymbol{f}$ has a solution over $\mathbb{C}$, then there is a number field K and a common zero $\boldsymbol{a} = (a_1, \dots, a_n) \in K^n$ of $\boldsymbol{f}$ such that

$$d_K \le B, \qquad \log D_K \le B,$$

and there is a positive integer c such that $ca_i \in \mathcal{O}_K$ for every i. The set E of rational primes dividing c has size at most B.

We now prove Proposition F.1.

Proof. If the system $\boldsymbol{f}$ has no solution, then recall that by Theorem F.3, $|S_{\boldsymbol{f}}(x)| \le B$. We will first show that if $\boldsymbol{f}$ has a solution, then $|S_{\boldsymbol{f}}(x)| \ge 4B$ for large enough x. Then we show how checking this dichotomy of $|S_{\boldsymbol{f}}(x)|$ is in AM. Let N be the sparse binary input length of $\boldsymbol{f}$. In the satisfiable case, choose K, the common zero $\boldsymbol{a}$, the positive denominator c, and the finite set E as in Theorem F.3. Let $\mathcal{P}_K(x)$ be the set of non-zero prime ideals in $\mathcal{O}_K$ of norm at most x. Let $\mathfrak{p} \in \mathcal{P}_K(x)$ be a prime ideal above a rational prime $p \notin E$, i.e., $\mathfrak{p} \cap \mathbb{Z} = (p)$. Then the inertia degree $f_{\mathfrak{p}}$ is defined as the residue-field extension degree

$$f_{\mathfrak{p}} \coloneqq \left[\frac{\mathcal{O}_K}{\mathfrak{p}} : \mathbb{F}_p\right].$$

Since $p \notin E$, the image of c in $\mathcal{O}_K/\mathfrak{p}$ is nonzero and hence invertible. Reducing the coordinates as $\overline{ca_i}/\overline{c}$ therefore gives a common zero of $\boldsymbol{f}$ over the finite field $\mathcal{O}_K/\mathfrak{p}$. In particular, if $f_{\mathfrak{p}} = 1$, then $\boldsymbol{f}$ has a solution over $\mathbb{F}_p$. So every such prime ideal $\mathfrak{p}$ contributes to a prime p in $S_{\boldsymbol{f}}(x)$, and there are at most d_K prime ideals above each p. Write

$$\mathcal{P}^E_{K,1}(x) = \{\mathfrak{p} \in \mathcal{P}_K(x) : f_{\mathfrak{p}} = 1, \ \mathfrak{p} \cap \mathbb{Z} = (p) \text{ for some } p \notin E\}.$$

Therefore $|\mathcal{P}^E_{K,1}(x)|/d_K \le |S_{\boldsymbol{f}}(x)|$.

Moreover, a prime ideal $\mathfrak{p}$ with $f_{\mathfrak{p}} \ge 2$ has norm at least p^2. So there are at most $d_K\sqrt{x}$ prime ideals in $\mathcal{P}_K(x)$ of inertia degree at least two. Hence

$$|\mathcal{P}^E_{K,1}(x)| \ge \pi_K(x) - d_K\sqrt{x} - d_K B,$$

where the $d_K B$ comes from the bound $d_K|E| \le d_K B$ on the number of prime ideals above the primes in E. Therefore,

$$|S_{\boldsymbol{f}}(x)| \ge \frac{\pi_K(x)}{d_K} - \sqrt{x} - B.$$

Now take $x = B^{128}$. Since $d_K \le B$ and $\log D_K \le B$, the error term in the estimate for $\pi_K(x)$ in Theorem F.2 is $O(B^{125} \log B)$, and $\mathrm{Li}(x) \ge x/(2\log x)$ for all sufficiently large x. Thus, for all sufficiently large B,

$$\frac{\pi_K(x)}{d_K} - \sqrt{x} - B \ge \frac{B^{127}}{256 \log B} - O(B^{125} \log B) - B^{64} - B \ge 4B. \tag{29}$$

Enlarging the constant term of q if necessary makes B large enough for this bound to hold for every N. Hence

$$\boldsymbol{f} \text{ unsatisfiable} \implies |S_{\boldsymbol{f}}(x)| \le B, \qquad \boldsymbol{f} \text{ satisfiable} \implies |S_{\boldsymbol{f}}(x)| \ge 4B. \tag{30}$$

Let $\ell = \lceil \log_2(x+1) \rceil$. Since the required x for Equation (29) to hold is $2^{\mathsf{poly}(N)}$, $\ell = \mathsf{poly}(N)$. Represent each integer from 0 to x by its binary expansion padded to length ℓ, and view $S_{\boldsymbol{f}}(x)$ as a subset of $\{0,1\}^\ell$. Membership in this set is in NP. Given a string representing an integer p, a witness consists of a primality certificate for p and a common zero of $\boldsymbol{f}$ over $\mathbb{F}_p$. The verifier also checks $2 \le p \le x$. Since $\ell = \mathrm{poly}(N)$, the witnesses have polynomial length. Evaluating the sparse input polynomials modulo p takes polynomial time, using repeated squaring for the binary exponents.

The Goldwasser–Sipser set-size lower-bound protocol [GS86] distinguishes in AM the two cases of the dichotomy for $|S_{\boldsymbol{f}}(x)|$ in (30); see also [AB09]. Apply it with threshold $2B$ and universe $\{0,1\}^\ell$. The protocol uses only B and x, both computed from N, and does not need to construct K. Hence $\mathrm{HN}_{\mathbb{C}} \in \mathsf{AM}$. □