On the Complexity of Hilbert's 17th Problem

Nikhil R. Devanur Richard J. Lipton Nisheeth K. Vishnoi

College of Computing
Georgia Institute of Technology
Atlanta GA 30332
{nikhil,rjl,nkv}@cc.gatech.edu

Abstract

Hilbert posed the following problem as the 17th in the list of 23 problems in his famous 1900 lecture:

> *Given a multivariate polynomial that takes only non-negative values over the reals, can it be represented as a sum of squares of rational functions?*

In 1927, E. Artin gave an affirmative answer to this question. His result guaranteed the existence of such a finite representation and raised the following important question:

> *What is the* **minimum number** *of rational functions needed to represent any non-negative n-variate, degree d polynomial?*

In 1967, Pfister proved that any n-variate non-negative polynomial over the reals can be written as sum of squares of at most 2^n rational functions. In spite of a considerable effort by mathematicians for over 75 years, it is *not* known whether $n+2$ rational functions are sufficient!

In lieu of the lack of progress towards the resolution of this question, we initiate the study of Hilbert's 17th problem from the point of view of Computational Complexity. In this setting, the following question is a natural relaxation:

> *What is the* **descriptive complexity** *of the sum of squares representation (as rational functions) of a non-negative, n-variate, degree d polynomial?*

We consider *arithmetic circuits* as a natural representation of rational functions. We are able to show, assuming a standard conjecture in complexity theory, that it is impossible that every non-negative, n-variate, degree four polynomial can be represented as a sum of squares of a *small* (polynomial in n) number of rational functions, each of which has a *small* size arithmetic circuit (over the rationals) computing it.

Our result points to the direction that it is unlikely that every non-negative, n-variate polynomial over the reals can be written as a sum of squares of a polynomial (in n) number of rational functions. Further, relating to standard (and believed to be hard to prove) complexity-theoretic conjectures sheds some light on why it has been difficult for mathematicians to close the $n+2$ and 2^n gap. We hope that our line of work will play an important role in the resolution of this question.

1 Introduction

Hilbert proposed 23 problems in 1900, in which he tried *to lift the veil behind which the future lies hidden.*[1] His description of the 17th problem is (see [6]):

> A rational integral function or form in any number of variables with real coefficient such that it becomes negative for no real values of these variables, is said to be definite. The system of all definite forms is invariant with respect to the operations of addition and multiplication, but the quotient of two definite forms in case it should be an integral function of the variables is also a definite form. The square of any form is evidently always a definite form. But since, as I have shown [11], not every definite form can be compounded by addition from squares of forms, the question arises which I have answered affirmatively for ternary forms [12] whether every definite form may not be expressed as a quotient of sums of squares of forms. At the same time it is desirable, for certain questions as to the possibility of certain geometrical constructions, to know whether the coefficients of the forms to be used in the expression may always be taken from the realm of rationality given by the coefficients of the form represented.

An affirmative answer to this problem was given by Emil Artin in 1927 [2]:

> For every non-negative polynomial $f \in \mathbb{R}[x_1, \ldots, x_n]$, there exist rational functions $g_1, \ldots, g_s \in \mathbb{R}(x_1, \ldots, x_n)$, such that $f = g_1^2 + \cdots + g_s^2$.

Motzkin's example (see [24]) of $P(x, y, z) = z^6 + x^4z^2 + x^2y^4 - 3x^2y^2z^2$ illustrates that the rational functions in Artin's result cannot, in general, be replaced by polynomials. $P(x, y, z)$ is non-negative everywhere over the reals, and yet, cannot be written as sum of squares of polynomials over the reals. Notice that Artin's result shows that every non-negative polynomial can be written as sum of squares of *finitely* many rational functions. This raised the following important question about the *size* of such a representation:

> What is the smallest number (denoted as $\nu(n, d)$), such that every n-variate, degree d, non-negative polynomial can be written as sum of squares of $\nu(n, d)$ rational functions over the reals?

In 1967, Pfister [19] proved that $\nu(n, d) \le 2^n$. However, this upper bound holds when one is allowed rational functions over a *real closed field* [2]. Remarkably enough, his bound does not depend on the degree of the polynomial. The best lower bound on $\nu(n, 3)$ is $n + 2$. Over 75 years of effort by various mathematicians, these are still the best known bounds in general. We remark that the function $\nu(n, 2)$ is quite well understood from the time of Hilbert (see [11, 13, 14]).

In lieu of the lack of progress towards the determination of $\nu(n, d)$, we initiate the study of Hilbert's 17th problem from the point of view of Computational Complexity. In this setting, the following question is a natural relaxation:

> *What is the* **descriptive complexity** *of the sum of squares representation (as rational functions) of a non-negative, n-variate, degree d polynomial?*

We consider *arithmetic circuits* as a natural representation of rational functions. We are able to show, assuming a standard conjecture in complexity theory, that it is impossible that every non-negative, n-variate, degree four polynomial can be represented as a sum of squares of a *small* (polynomial in n) number of rational functions, each of which has a *small* size arithmetic circuit (over the rationals) computing it.

Our result points to the direction that it is unlikely that every non-negative, n-variate polynomial over the reals can be written as a sum of squares of a polynomial (in n) number of rational functions. Further, relating to standard (and believed to be hard to prove) complexity-theoretic conjectures sheds some light on why it has been difficult for mathematicians to close the $n + 2$ and 2^n gap.

[1] A quote taken from [26].

[2] See [4, 21] for a definition.

1.1 Related work

Like all of Hilbert's problems, the 17th has received a lot of attention from the mathematical community and beyond. For an extensive survey of the development and impact of Hilbert's 17th problem on Mathematics, the reader is referred to excellent surveys by [9, 22, 24, 25]. The books [4, 21] also provide good accounts of this and related problems.

Apart from what can be found in the references above, we are aware of some recent work on various quantitative aspects of Hilbert's 17th problem. For instance, in [3], it has been proved that if the degree is fixed and the number of variables are allowed to increase, then there are significantly many more non-negative polynomials than those that can be written as sum of squares of polynomials. Further, in [23], it is shown that in general, one cannot obtain a sum of squares representation in which each rational function has the *same* denominator.

To the best of our knowledge the problem raised by this work, about the representational complexity of non-negative polynomials in the computational setting, is new.

2 Overview of our result

Notations

For $k = \mathbb{R}, \mathbb{Q}$ or $\mathbb{Z}$, $k[x_1, \ldots, x_n]$ denotes the ring of polynomials over k and $k(x_1, \ldots, x_n)$ denotes the corresponding field of fractions. The following notation about polynomials is used throughout this paper: A polynomial is written as $f = \sum_\alpha c_\alpha \mathbf{x}^\alpha$. Here $\mathbf{x}^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. $\deg(f)$ denotes the maximum *total degree* of f. $H(f) := \max_\alpha |c_\alpha|$.

Arithmetic circuits

An *arithmetic circuit* C over k [3] is a directed acyclic graph. Each vertex has indegree 0 or 2 and is labeled either by addition, multiplication, one of the input variables: $\{x_1, \ldots, x_n\}$, or scalars from k. If the vertex is labeled by a scalar or an input variable, then its indegree must be 0. If the vertex has indegree 2, then it must be labeled either by $+$ or by $\times$. There is exactly one vertex with no outgoing edge, which naturally corresponds to the polynomial (over $k[x_1, \ldots, x_n]$) computed by C. The size of C is the number of gates along with description size of all the constants used. As observed, C computes a polynomial $f(x_1, \ldots, x_n) \in k[x_1, \ldots, x_n]$. The size of the smallest arithmetic circuit that computes $f \in k[x_1, \ldots, x_n]$ is denoted by $\mathcal{L}_k(f)$. We will drop the subscript wherever k is clear from the context.

Unsatisfiability

Consider a boolean function $\phi : \{0,1\}^n \mapsto \{0,1\}$ in the conjunctive normal form (3-CNF), that is $\phi(x_1, \ldots, x_n) = \bigwedge_{i=1}^m C_i$, where each C_i is a boolean OR of at most 3 literals from $\{x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\}$. ϕ is said to be *satisfiable* if there is a *satisfying assignment* $a_1, \ldots, a_n \in \{0,1\}$, such that $\phi(a_1, \ldots, a_n) = 1$. The set of such boolean functions, in 3-CNF form, that have a satisfying assignment is denoted 3SAT. It is well known that 3SAT is NP-complete. The corresponding co-NP problem is UN3SAT, i.e. the set of boolean functions in 3-CNF that have no satisfying assignment. It follows that UN3SAT is complete for co-NP.

Now we give the key definition and the main result of this paper. (Readers not familiar with standard notations and definitions in computational complexity theory, should refer to the appendix.)

[3] In general k could be a commutative ring, but here k will be either the fields $\mathbb{R}$ and $\mathbb{Q}$, or the ring of integers $\mathbb{Z}$.

Definition 2.1.

$$\mathsf{H}^{\mathbb{Z}}(n,d,h) := \{f \in \mathbb{Z}[x_1,\ldots,x_n] : \deg(f) \le d, H(f) = O(h), \forall (x_1,\ldots,x_n) \in \mathbb{R}^n \ f \ge 0\}.$$

Further, let $\mathsf{H}^{\mathbb{Z}}(d,h) := \cup_{n\ge 0}\mathsf{H}^{\mathbb{Z}}(n,d,h)$.

Remark 2.2. *Note that we are implicitly viewing* $\mathsf{H}^{\mathbb{Z}}(d,h)$ *as a language. Fixing a unique representation of polynomials (say the smallest arithmetic circuit over* $\mathbb{Q}$*), we can view polynomials in this set as binary strings, thus, justifying our viewpoint. Hence, the length of the input is related to the description of the polynomial and* **not** n*. But we concern ourselves only with the case when the smallest arithmetic circuit computing an* n*-variate polynomial* f *is of size at most a fixed polynomial in* n*, say* n^6*.* [4]

2.1 Main theorems

Theorem 2.3. *Assuming* $\mathsf{PH}{\neq}\Sigma_2$*, for all* $n \ge 1$*, there exists a polynomial* $f \in \mathsf{H}^{\mathbb{Z}}(n,6,1)$ *such that no representation of* f *as sum of squares of rational functions over* $\mathbb{Q}$*,* $f = \sum_{i=1}^{s} g_i^2$*,* $g_i \in \mathbb{Q}(x_1,\ldots,x_n)$*, satisfies both of the following:*

1. $s = \mathrm{poly}\,(\mathcal{L}(f))$.
2. *For all* $i = 1,2,\ldots,s$*,* $\mathcal{L}(g_i) = \mathrm{poly}\,(\mathcal{L}(f))$.

Thus, unless the polynomial hierarchy collapses to the second level, not every non-negative polynomial has a *succinct* sum of squares representation. It is a standard hypothesis in complexity theory that $\mathsf{PH}{\neq}\Sigma_2$.[5] In fact this theorem says that even if the polynomial has degree 6 and all coefficients are integers and bounded by a constant, there is no such representation. As remarked earlier, the degree 2 case is well understood. We strengthen the previous result by bringing the degree down to 4, at the cost of blowing up the size of the coefficients. It is an interesting open problem if such a statement can be obtained for degree 3.

Theorem 2.4. *Assuming* $\mathsf{PH}{\neq}\Sigma_2$*, for all* $n \ge 1$*, there exists a polynomial* $f \in \mathsf{H}^{\mathbb{Z}}(n,4,\mathrm{poly}(n))$ *such that no representation of* f *as sum of squares of rational functions over the rationals,* $f = \sum_{i=1}^{s} g_i^2$*,* $g_i \in \mathbb{Q}(x_1,\ldots,x_n)$*, satisfies both of the following:*

1. $s = \mathrm{poly}\,(\mathcal{L}(f))$.
2. *For all* $i = 1,2,\ldots,s$*,* $\mathcal{L}(g_i) = \mathrm{poly}\,(\mathcal{L}(f))$.

A remark about the representation field

Although we state our theorems for $\mathbb{Q}$, one can replace it by a finite real algebraic extension of $\mathbb{Q}$. The details are easy and we omit the details for the ease of presentation. It is important to note though, that Artin's result does not, in general, imply existence of a sum of squares representation, where each rational function is over $\mathbb{Q}$. The *hard to represent* polynomials guaranteed by our results have a further property that these have small arithmetic circuits over the integers. It is conceivable that for such polynomials, a succinct representation (in our sense) exists if and only if a succinct representation exists over the reals. This is an interesting question for which we do not know an answer.

[4] For a non-negative, n-variate polynomial with arithmetic circuit complexity not bounded by any polynomial in n, one cannot hope to write an efficient (polynomial in n) sum of square representation by rational functions. Hence it makes sense only to consider polynomials which are efficiently computable by small circuits.

[5] Refer to the appendix for a substantiation of this belief.

Outline of the proofs

As the first step in the proof of Theorems 2.3 and 2.4, we reduce an instance ϕ of $\mathsf{UN3SAT}$ to a polynomial F_ϕ which is non-negative if and only if ϕ is unsatisfiable. This is a variant of an often used trick, which allows one to use algebraic considerations to study a boolean formula. We give two such reductions, corresponding to the two theorems: for Theorem 2.3 we give a reduction such that $F_\phi \in \mathsf{H}^{\mathbb{Z}}(6,1)$ and for Theorem 2.4, $F_\phi \in \mathsf{H}^{\mathbb{Z}}(4, \mathrm{poly}(\cdot))$. These results establish the $\mathsf{co\text{-}NP}$ hardness of the classes $\mathsf{H}^{\mathbb{Z}}(6,1)$ and $\mathsf{H}^{\mathbb{Z}}(4, \mathrm{poly}(\cdot))$. Artin's Theorem guarantees a sum of squares representation of F_ϕ over the reals. If there is some such representation which is *succinct* (describable by a polynomial number of polynomial size arithmetic circuits), in NP we can guess it and in $\mathsf{co\text{-}RP}$, check if the guessed representation is the same as F_ϕ. (This last step is done by invoking polynomial identity testing.) Formally we prove the following theorem:

Theorem 2.5. *For all $n, d, h \geq 1$, if for all $f \in \mathsf{H}^{\mathbb{Z}}(n,d,h)$, there exist $g_1, g_2, \ldots, g_s \in \mathbb{Q}(x_1, \ldots, x_n)$, such that $f = \sum_{i=1}^{s} g_i^2$, $s = \mathrm{poly}(\mathcal{L}(f))$, and for all $i = 1, 2, \ldots, s$, $\mathcal{L}(g_i) = \mathrm{poly}(\mathcal{L}(f))$, then $\mathsf{H}^{\mathbb{Z}}(d,h) \in \mathsf{NP}^{\mathsf{co-RP}}$.*

To derive the desired contradiction, in the end we invoke a result of Boppana, Hastad and Zachos [5], which states that $\mathsf{co\text{-}NP} \not\subseteq \mathsf{NP}^{\mathsf{co-RP}}$, unless $\mathsf{PH} = \Sigma_2$.

Organization

Section 3 contains the arithmetizations of SAT needed to prove Theorems 2.3 and 2.4. The main results, viz proofs of Theorems 2.3, 2.4, 2.5, are proved in Section 4.

3 Arithmetization of SAT

In this section we give two different arithmetizations of instances of $\mathsf{UN3SAT}$, each of which will be used in proving one of Theorems 2.3, 2.4.

Given an instance $\phi = \wedge_{i=1}^{m} C_i$ of a $\mathsf{UN3SAT}$ problem: Call a literal $z \in \{z_1, \overline{z}_1, \ldots, z_n, \overline{z}_n\}$ *positive*, if $z \in \{z_1, \ldots, z_n\}$. Else, call it *negative*. For a clause $C = C_+ \vee C_-$ (C_+ consists of positive literals while C_- consists of negative literals), define

$$\mathcal{A}(C) := \left(\prod_{z \in C_+} (1 - z) \right) \cdot \left(\prod_{z \in C_-} z \right).$$

For instance, if $C = x_1 \vee \overline{x}_2 \vee x_3$, then $\mathcal{A}(C) = (1 - x_1) x_2 (1 - x_3)$. Further for $a_1, a_2, a_3 \in \{0,1\}$, $\mathcal{A}(C)(a_1, a_2, a_3) = 0$ if and only if $C(a_1, a_2, a_3) = 1$, (or C is satisfiable). Now define

$$F_\phi(z_1, \ldots, z_n) := 300 \left(\sum_{i=1}^{n} z_i^2 (1 - z_i)^2 + \sum_{j=1}^{m} (\mathcal{A}(C_j))^2 \right) - 1. \tag{1}$$

Thus for all ϕ, $F_\phi \in \mathbb{Z}[z_1, \ldots, z_n]$. It is convenient to let $f_\phi := F_\phi / 300$. The problem remains the same though, as the sign of f_ϕ is the same as that of F_ϕ. Let $\epsilon = \frac{1}{300}$.

Lemma 3.1. *ϕ is not satisfiable if and only if $f_\phi \geq 0$ over the reals.*

Proof. If ϕ is satisfiable, let $a = (a_1, \ldots, a_n) \in \{0,1\}^n \subset \mathbb{R}^n$ be a satisfying assignment. Then by definition $f_\phi(a) = -\epsilon < 0$. To prove the converse, consider the case when ϕ is unsatisfiable. We need to show that $f_\phi \geq 0$ over the reals. Let $\delta = 1/4$. We consider two cases:

1. Case 1: Let $(s_1, \ldots, s_n) \in \mathbb{R}^n$ be a point such that there is an $1 \leq i \leq n$ such that s_i does not lie in either of the two intervals: $[-\delta, \delta], [1-\delta, 1+\delta]$. In this case $s_i^2(1-s_i)^2 > \delta^4$. Since $\epsilon \leq \delta^4$, $f_\phi(s_1, \ldots, s_n) > 0$.

2. Case 2: Hence, we may assume that for a point $(s_1, \ldots, s_n)$, all s_i are in one of the intervals: $[-\delta, \delta], [1-\delta, 1+\delta]$. From this we construct a point $a = (a_1, \ldots, a_n) \in \{0,1\}^n$ as follows:

   - If $s_i \in [-\delta, \delta]$ then let $a_i = 0$.
   - If $s_i \in [1-\delta, 1+\delta]$ then let $a_i = 1$.

   Since ϕ is unsatisfiable, there is a clause, say C, which is not satisfied by a. Let $\mathcal{A}(C) = \left(\prod_{z \in C_+}(1-z)\right) \cdot \left(\prod_{z \in C_-} z\right)$. If $z_i \in C_+$, since C is not satisfied by a, it must be that $a_i = 0$, and hence $s_i \in [-\delta, \delta]$, or equivalently $(1-s_i) \in [1-\delta, 1+\delta]$. Similarly, if $\overline{z}_i \in C_-$, $a_i = 1$, and hence $s_i \in [1-\delta, 1+\delta]$. This implies that at the point $(s_1, \ldots, s_n)$, $f_\phi \geq \mathcal{A}^2(C) \geq (1-\delta)^6 > \epsilon$.

Thus, if ϕ is unsatisfiable, $f_\phi > 0$ over the reals. This completes the proof. □

The above arithmetization reduces UN3SAT to $\mathsf{H}^{\mathbb{Z}}(6,1)$. Thus, the following proposition follows from Lemma 3.1 and co-NP hardness of UN3SAT.

Proposition 3.2. $\mathsf{H}^{\mathbb{Z}}(6,1)$ *is* co-NP *hard.*

Next we show how to obtain a quantitatively better result, if we allow the coefficients to grow with the input size. First, we need a new reduction. As before, let ϕ be a boolean function given in 3-CNF form on n variables and m clauses.

$$f'_\phi(z_1, \ldots, z_n) := \sum_{i=1}^{n} \frac{(3^3+1)m}{\delta(m)^4} z_i^2(1-z_i)^2 + \sum_{j=1}^{m}(\mathcal{A}(C_j)) - \epsilon(m). \tag{2}$$

Here δ and ϵ are positive functions (but less than 1) of m such that $\epsilon < (1-\delta)^3 - m\delta(1+\delta)^2$. Note that one can choose such a δ and an ϵ since $(1-\delta)^3 \to 1$ and $m\delta(1+\delta)^2 \to 0$ as $\delta \to 0$. As in the previous case, we can always multiply f'_ϕ suitably to obtain a polynomial F'_ϕ over the integers.

Lemma 3.3. *ϕ is not satisfiable if and only if $f'_\phi \geq 0$ over the reals.*

Proof. If ϕ is satisfiable, let $a = (a_1, \ldots, a_n) \in \{0,1\}^n \subset \mathbb{R}^n$ be a satisfying assignment. Then by definition $f'_\phi(a) = -\epsilon < 0$. To prove the converse, consider the case when ϕ is unsatisfiable. We need to show that $f'_\phi \geq 0$ over the reals. We consider two cases:

Case 1: Suppose that for a point $s := (s_1, \ldots, s_n)$, all s_i are in one of the intervals: $[-\delta, \delta], [1-\delta, 1+\delta]$. From this we construct a point $a = (a_1, \ldots, a_n) \in \{0,1\}^n$ as follows:

- If $s_i \in [-\delta, \delta]$ then let $a_i = 0$.
- If $s_i \in [1-\delta, 1+\delta]$ then let $a_i = 1$.

Since ϕ is unsatisfiable, there is a clause, say C, which is not satisfied by a. Let $\mathcal{A}(C) = \left(\prod_{z \in C_+}(1-z)\right) \cdot \left(\prod_{z \in C_-} z\right)$. If $z_i \in C_+$, since C is not satisfied by a, $s_i \in [-\delta, \delta]$, or equivalently $(1-s_i) \in [1-\delta, 1+\delta]$. Similarly, if $\overline{z}_i \in C_-$, $a_i = 1$ and hence $s_i \in [1-\delta, 1+\delta]$. This means that at the point s, $\mathcal{A}(C) \geq (1-\delta)^3$.

Now consider a clause C' satisfied by a. Writing $C' = C'_+ \bigvee C'_-$, we see that either some variable in C'_+ is set to 1, or some variable in C'_- is set to 0 in the assignment a. Without loss of generality, assume that $z_i \in C'_+$ is set to 1 ($a_i = 1$). Thus, $s_i \in [1-\delta, 1+\delta]$, or $(1-s_i) \in [-\delta, \delta]$. Thus

$$\mathcal{A}(C') = \left(\prod_{z \in C'_+} (1-z) \right) \cdot \left(\prod_{z \in C'_-} z \right) \geq -\delta(1+\delta)^2.$$

Adding the inequalities for unsatisfied and satisfied clauses, one gets that

$$\sum_{j=1}^{m} \mathcal{A}(C_j) \geq (1-\delta)^3 - m\delta(1+\delta)^2.$$

By the choice of ϵ and δ, we have $\epsilon < (1-\delta)^3 - m\delta(1+\delta)^2$, and therefore $f'_\phi(s) > 0$.

Case 2: Now consider a point $s = (s_1, \ldots, s_n)$ such that, there is an $1 \leq i \leq n$, such that s_i does not lie in either of the two intervals: $[-\delta, \delta], [1-\delta, 1+\delta]$. For a clause C, define

$$\Delta_C := \max \left\{ \{|1-s_i| : z_i \in C_+\} \cup \{|s_j| : \overline{z}_j \in C_-\} \right\}.$$

It follows that $\mathcal{A}(C)(s) \geq -\Delta_C^3$. Now consider the following 2 cases:

Case 2a $\Delta_C > 3$

Let s_{j^*} be such that either $|s_{j^*}|$ or $|1-s_{j^*}|$ is equal to Δ_C. Then $(s_{j^*})^2(1-s_{j^*})^2 \geq \Delta_C^2(\Delta_C - 1)^2 > \Delta_C^3 + 1$. This implies that $\mathcal{A}(C)(s) + \frac{3^3+1}{\delta^4}(s_{j^*})^2(1-s_{j^*})^2 > -\Delta_C^3 + \frac{3^3+1}{\delta^4}(\Delta_C^3 + 1) > 1$. The last inequality follows by noticing that $\delta < 1$.

Case 2b $\Delta_C \leq 3$

From the definition of case 2, $\exists s_{j^*}$ such that $(s_{j^*})^2(1-s_{j^*})^2 \geq \delta^4$. Hence, $\frac{3^3+1}{\delta^4}(s_{j^*})^2(1 - s_{j^*})^2 > 3^3 + 1$. By definition of Δ_C, $\mathcal{A}(C)(s) \geq -3^3$. Combining these inequalities, we get $\mathcal{A}(C)(s) + \frac{3^3+1}{\delta^4}(s_{j^*})^2(1-s_{j^*})^2 > 1$.

Now summing over all clauses, we get, $\sum_{j=1}^m \mathcal{A}(C_j)(s) + \sum_{i=1}^n \frac{(3^3+1)m}{\delta^4} s_i^2(1-s_i)^2 > m$. This is exactly what we set out to prove: $f_\phi(s) > 0$.

Thus, if ϕ is unsatisfiable, $f'_\phi > 0$ over the reals. This completes the proof. □

This leads to the following:

Proposition 3.4. $\mathsf{H}^{\mathbb{Z}}(4, \mathrm{poly}(\cdot))$ *is* co-NP *hard.*

Amplifying positivity

Using the PCP Theorem of [1], one can transform the given formula so that, if it is unsatisfiable, then a large fraction (say $c, 0 < c < 1$) of clauses are unsatisfiable. This gives rise to an arithmetization such that $f_\phi > cm - 1$ if and only if ϕ is unsatisfiable. This shows that even if one is given that whenever $f > 0$, $f > cm - 1$, it is still co-NP hard to decide the positivity of f.

Circuit complexity of the arithmetized polynomials

It is important to note that for any 3CNF formula ϕ, there is an arithmetic circuit over $\mathbb{Z}$ which computes F_ϕ and F'_ϕ, whose sizes are at most n^6. [6] In fact, the explicit arithmetizations written down earlier can be converted into such circuits.

[6] Since ϕ is in 3CNF, $m \leq (2n)^3$.

4 Main results

Testing identities The *Identity Testing* problem for arithmetic circuits is to decide if two given arithmetic circuits evaluate the same polynomial. More formally, given two arithmetic circuits C_1, C_2 over $\mathbb{Z}$, let $f, g \in \mathbb{Z}[x_1, \ldots, x_n]$ be the polynomials computed by them respectively. The problem is to decide efficiently if $f - g$ is identically zero over the integers. Here, efficiency is measured in terms of the input size, which in this case, is the sum of the sizes of C_1 and C_2. The following result by Ibarra and Moran [15] establishes that, in the presence of randomness, there is an efficient solution to this problem. Formally, there is an efficient randomized algorithm which takes as input two circuits and decides if they compute the same polynomial. The algorithm is always correct when it says NO, but there is a small chance that is is wrong when it says YES. This simple but important result will play a crucial role in the proof of the main results which we describe next.

Lemma 4.1. *([15]) The Identity Testing problem for arithmetic circuits over* $\mathbb{Z}$ *is in* $\mathsf{co\text{-}RP}$*.*

The fact that non-negative polynomials can be represented as sum of squares suggests the following algorithm for checking if $f \in \mathsf{H}^{\mathbb{Z}}(n, d, \cdot)$. Suppose it is true that $f = g_1^2 + \cdots + g_s^2$, where each $g_i = \frac{\alpha_i}{\beta_i}$, α_i and β_i are polynomials over the integers, and $\beta_i \neq 0$. Further assume that this representation is *succinct*, that is $s = \text{poly}(n)$ and for all $1 \leq i \leq s$, $\mathcal{L}(\alpha_i), \mathcal{L}(\beta_i) \leq \text{poly}(n)$. Then in NP, we can *guess* these polynomials α_i, β_i, as the total bits one has to guess is a polynomial in n. Once we have guessed the representation, one checks the following identity:

$$f \prod_{j=1}^{s} \beta_j^2 - \sum_{i=1}^{s} \left(\alpha_i \prod_{j \neq i} \beta_j \right)^2 \equiv 0 \tag{3}$$

Since f itself has an arithmetic circuit over the integers of size at most n^6, the polynomial on the LHS of the above identity has a polynomial size circuit. Hence using the identity testing algorithm for arithmetic circuits over the integers, one can verify the above identity in $\mathsf{co\text{-}RP}$. Thus checking the validity of the guessed representation.
This is formalized in the following proof:

Proof of Theorem 2.5. Using NP, guess each $g_i = \frac{\alpha_i}{\beta_i}$ where α_i and β_i are polynomials over $\mathbb{Q}$. By hypothesis, we know that $f \prod_{j=1}^{s} \beta_j^2$ and $\sum_{i=1}^{s} \left(\alpha_i \prod_{j \neq i} \beta_j \right)^2$ are arithmetic circuits with length a polynomial in n. Hence by Lemma 4.1, checking whether they are equal is in $\mathsf{co\text{-}RP}$. The time required to evaluate the g_i's is also a polynomial in n. Hence we get $\mathsf{H}^{\mathbb{Z}}(d, h) \in \mathsf{NP}^{\mathsf{co-RP}}$.

□

Finally, we need the following result of Bopanna, *et al* [5].

Theorem 4.2. *[5]* $\mathsf{co\text{-}NP} \subseteq \mathsf{NP}^{\mathsf{co-RP}} \Rightarrow \mathsf{PH} = \Sigma_2$.

Now we are ready to prove Theorem 2.3.

Proof of Theorem 2.3. Assume on the contrary. From Theorem 2.5, $\mathsf{H}^{\mathbb{Z}}(6, 1) \in \mathsf{NP}^{\mathsf{co-RP}}$. But $\mathsf{H}^{\mathbb{Z}}(6, 1)$ is $\mathsf{co\text{-}NP}$-Hard by Proposition 3.2. Now by Theorem 4.2, $\mathsf{PH} = \Sigma_2$, a contradiction is achieved. □

Using Proposition 3.4 instead of Proposition 3.2 in the above proof, one obtains a proof of Theorem 2.4.

Acknowledgments

We are grateful to Marie-Françoise Roy for valuable comments on an earlier draft of this paper. We also thank Bruce Reznick for pointing us to references [3, 7]. We would also like to thank Peter Bürgisser for his encouragement.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1) (1998), 70–122.

[2] E. Artin. Über die Zerlegung definiter Funktionen in Quadrate. *Abh. Math. Sem. Univ. Hamburg*, 5 (1927), 100–115.

[3] G. Blekherman. There are significantly more non-negative polynomials than sums of squares. *Preprint.*

[4] H. Bochnak, M. Coste, M.-F. Roy. **Real algebraic geometry**. Springer, 1998.

[5] R. Boppana, J. Hastad, and S. Zachos. Does Co-NP Have Short Interactive Proofs? *Information Processing Letters*, 25 (1987), 127–132.

[6] Felix Browder (ed.) Mathematical developments arising from Hilbert's Problems. *Proc. Symp. Pure Math.*, 28 (1976), Amer. Math. Soc.

[7] M. D. Choi, Z. D. Dai, T. Y. Lam, B. Reznick: The pythagoras number of some affine algebras and local algebras, *J. Reine Angew. Math.*, 336 (1982), 45-82.

[8] S.A. Cook. The complexity of theorem-proving procedures. *Proceedings of the Third ACM Symposium on the Theory of Computing*, 151–158, 1971.

[9] D. W. Dubois. Note on of Hilbert's 17th problem. *Bull. Amer. Math. Soc.*, 73 (1967), 540–541.

[10] M. R. Garey, D. S. Johnson. **Computers and Intractability: A Guide to the Theory of NP-Completeness**. Freeman, 1979.

[11] D. Hilbert. Über die Darstellung definiter Formen als Summen von Formenquadraten. *Math. Ann.*, 32 (1888), 342–350.

[12] D. Hilbert. Über ternäre definite Formen. *Acta Math.*, 17 (1893), 169–198.

[13] D. Hilbert. **Grundlagen der Geometrie**. Leipzig, Chap. 7, 1899.

[14] D. Hilbert. Darstellung definiter Formen durch Quadrate. *Akad. Wiss. Göttingen* (1900), 284–285.

[15] Oscar H. Ibarra, Shlomo Moran. Probabilistic Algorithms for Deciding Equivalence of Straight-Line Programs. *JACM* , 30(1) (1983), 217–228.

[16] R.M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, (R.E. Miller and J.M. Thatcher, eds.), 85–103, Plenum Press, 1972.

[17] Leonid A. Levin. Universal'nye perebornye zadachi (Universal search problems : in Russian). *Problemy Peredachi Informatsii*, 9(3) (1973), 265–266.

[18] C. Papadimitriou. **Computational Complexity**. Addison-Wesley, 1994.

[19] A. Pfister. Zur Darstellung definiter Funktionen als Summe von Quadraten. *Invent. Math.* 4 (1967), 229–237.

[20] V. Powers, B. Reznick. A new bound for Po'lya's Theorem with applications to polynomials positive on polyhedra, J. Pure Appl. Alg. 164 (2001) 221–229.

[21] A. Prestel, C.N. Delzell. **Positive Polynomials: From Hilbert's 17th Problem to Real Algebra**. Springer Monographs in Mathematics, 2001.

[22] B. Reznick. Some concrete aspects of Hilbert's 17th Problem. *Publ. Math. Univ. Paris VII*, No. 56, Jan. 1996.

[23] B. Reznick. On the absence of uniform denominators in Hilbert's Seventeenth Problem. *Preprint.*

[24] Marie-Francoise Roy. The role of Hilbert's problems in real algebraic geometry. *Proceedings of the ninth EWM Meeting*, Loccum, Germany 1999.

[25] G. Stengle. A Nullstellensatz and a Positivstellensatz in semialgebraic geometry. *Math. Ann.*, 207 (1974), 87–97.

[26] R. Thiele. Hilbert's Twenty-Fourth Problem. *American Math. Monthly*, 110(1) (2003), 1–23.

Computational Complexity preliminaries

The aim of this section is to present the definitions and notions in Computational Complexity Theory. [7] The reader is refered to the book by Papadimitriou [18] for a comprehensive treatment of this subject.

Some complexity classes

A *language* is a subset of $\{0,1\}^*$. For a language L, $\bar{L} := \{0,1\}^* \backslash L$. A p-ary *relation* is a language over the following p-ary product: $\{0,1\}^* \times \cdots \times \{0,1\}^*$.[8] The complexity class $\mathsf{DTIME}(f(n))$ is the set of all languages for which membership can be tested in time $f(n)$, by a deterministic Turing machine, in time $f(n)$. $\mathsf{P} := \cup_{t \geq 0} \mathsf{DTIME}(n^t)$. NP is the collection of all languages L, such that there is a 2-ary relation $R_L \in \mathsf{P}$ (called a *polynomially decidable relation*) and a polynomial $p(\cdot)$, such that $x \in L$ if and only if there is a $y \in \{0,1\}^*$, with $|y| = O(p(|x|))$, and $(x,y) \in R_L$. The class $\mathsf{co\text{-}NP}$ is defined as $\cup_{L \in \mathsf{NP}} \bar{L}$. It follows that a language L is in $\mathsf{co\text{-}NP}$ if and only if there is a polynomially decidable 2-ary relation R_L and a polynomial $p(\cdot)$, such that $x \in L$ if and only if and for all $y \in \{0,1\}^*$, with $|y| = O(p(|x|))$, $(x,y) \in R_L$. It is natural to define complexity classes based on compositions of these *existential* and *universal* quantifiers. Starting with $\mathsf{\Sigma}_1 = \mathsf{NP}$ and $\mathsf{\Pi}_1 = \mathsf{co\text{-}NP}$, one can define $\mathsf{\Sigma}_i$ and $\mathsf{\Pi}_i$ as follows. For $i \geq 2$, $\mathsf{\Sigma}_i$ is the collection of all languages L such that there is a i-ary relation $R_L \in \mathsf{\Pi}_{i-1}$, and a polynomial $p(\cdot)$, such that $x \in L$ if and only if there exists a $y \in \{0,1\}^*$, with $|y| = O(p(|x|))$, $(x,y) \in R_L$. $\mathsf{\Pi}_i$ is defined similarly as $\mathsf{co\text{-}\Sigma}_i$. Further, define $\mathsf{\Delta}_i := \mathsf{\Sigma}_i \cap \mathsf{\Pi}_i$. One often thinks of $\mathsf{\Delta}_0 = \mathsf{\Sigma}_0 = \mathsf{\Pi}_0 = \mathsf{P}$ and $\mathsf{\Delta}_1 = \mathsf{NP} \cap \mathsf{co\text{-}NP}$. *Polynomial Hierarchy* (PH) is defined to be the collection of classes $\mathsf{\Delta}_i, \mathsf{\Sigma}_i$ and $\mathsf{\Pi}_i$, for all $i \geq 0$. It follows from definitions that if NP=$\mathsf{co\text{-}NP}$ then $\mathsf{\Sigma}_i = \mathsf{\Delta}_i$ for all $i \geq 1$.

Completeness

A language L is said to be *hard* for a a complexity class $\mathcal{C}$, for all $L' \in \mathcal{C}$, there is a polynomial $p(\cdot)$ and a Turing machine $M_{L,L'} : \{0,1\}^* \to \{0,1\}^*$, such that $x \in L$ if and only if $M_{L,L'}(x) \in L'$. Moreover, for the complexity classes we will be interested in, we assume that $M_{L,L'}$ runs in time $O(p(|x|))$. If $L \in \mathcal{C}$ and L is

[7] The reason we do so is it to broaden the scope of this paper to mathematicians who may not be familiar with these notions, but are interested in understanding our results on Hilbert's 17th problem.

[8] A 1-ary relation is just a language.

hard for $\mathcal{C}$, then L is said to be *complete* for $\mathcal{C}$. Complete problems for a complexity class can be thought of as the hardest problems in their class and can be thought of as characterizing the complexity class.

Next we define a problem which is known to be NP-complete. Consider a boolean function $\phi : \{0,1\}^n \mapsto \{0,1\}$ in the conjunctive normal form (3-CNF), that is $\phi(x_1, \ldots, x_n) = \bigwedge_{i=1}^{m} C_i$, where each C_i is a boolean OR of at most 3 literals from $\{x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\}$. ϕ is said to be *satisfiable* if there is a *satisfying assignment* $a_1, \ldots, a_n \in \{0,1\}$, such that $\phi(a_1, \ldots, a_n) = 1$. The set of such boolean functions, in 3-CNF form, that have a satisfying assignment is denoted $\mathsf{3SAT}$. One of the earliest and most important results in complexity Theory (see [8, 16, 17]) was establishing that $\mathsf{3SAT}$ is NP-complete. The corresponding $\mathsf{co\text{-}NP}$ problem is $\mathsf{UN3SAT}$, i.e. the set of boolean functions in 3-CNF that have no satisfying assignment. It follows that $\mathsf{UN3SAT}$ is complete for $\mathsf{co\text{-}NP}$. Generalizing these results, it is known that there is a complete problem for Σ_i (and hence for each Π_i), for all $i \geq 1$. This is precisely the reason why it is widely believed that for all $i \geq 1$, $\Sigma_i \neq \Pi_i$. This implies that $\mathsf{PH} \neq \Sigma_2$, a conjecture on which our result will be based on.

Probabilistic complexity classes

Randomized complexity classes are defined with respect to Turing machines which have access to an additional tape which contains an infinite number of uniform and independent random bits. For this paper, we are just concerned with *probabilistic polynomial time* Turing machines which always halt (independently of the random tape) after a polynomial number of steps (in the length of the input). Naturally, for an input x to such a randomized machine M, one associates probabilities to the computation $M(x)$. The class RP is the class of all languages L, such that there is a probabilistic polynomial time Turing machine M_L, such that for all $x \in L$, $\Pr[M_L(x) \text{ accepts}] = 1$ and for all $x \notin L$, $\Pr[M_L(x) \text{ accepts}] \leq 1/2$. The probabilistic complexity classes important for this paper will be RP and $\mathsf{co\text{-}RP}$. Finally, we define the class $\mathsf{NP}^{\mathsf{co-RP}}$ as the collections of languages L, for which there is a probabilistic polynomial time machine M_L, and a polynomial $p(\cdot)$, such that if $x \in L$ there is a $y \in \{0,1\}^*$, with $|y| = O(p(|x|))$, $\Pr[M_L(x,y) \text{ accepts}] \leq 1/2$, and if $x \notin L$, then for all $y \in \{0,1\}^*$, with $|y| = O(p(|x|))$, $\Pr[M_L(x,y) \text{ accepts}] = 1$.