Determinant versus permanent

Manindra Agrawal

Abstract. We study the problem of expressing permanents of matrices as determinants of (possibly larger) matrices. This problem has close connections to the complexity of arithmetic computations: the complexities of computing permanent and determinant roughly correspond to arithmetic versions of the classes NP and P respectively. We survey known results about their relative complexity and describe two recently developed approaches that might lead to a proof of the conjecture that the permanent can only be expressed as the determinant of exponential-sized matrices.

Mathematics Subject Classification (2000). Primary 68Q17; Secondary 68W30.

Keywords. Arithmetic computation, complexity classes, determinant, permanent.

1. Introduction

The determinant of square matrices plays a fundamental role in linear algebra. It is a linear function of rows (and columns) of the matrix, and has several nice interpretations. Geometrically it is the volume of the parallelopied defined by rows (or columns) of the matrix, and algebraically it is the product of all eigenvalues, with multiplicity, of the matrix. It also satisfies a number of other interesting properties, e.g., it is multiplicative, invariant under linear combinations of rows (and columns) etc. The *permanent* of a square matrix is a number that is defined in a way similar to the determinant. For a matrix $X = [x_{i,j}]_{1 \le i,j \le n}$,

$$\operatorname{per} X = \sum_{\pi \in S_n} \prod_{i=1}^{n} x_{i,\pi(i)},$$

where S_n is the symmetric group on n elements. The only difference to the determinant is in the signs of terms:

$$\det X = \sum_{\pi \in S_n} \operatorname{sgn}(\pi) \cdot \prod_{i=1}^{n} x_{i,\pi(i)},$$

Proceedings of the International Congress
of Mathematicians, Madrid, Spain, 2006

where $\mathrm{sgn}(\pi) \in \{1, -1\}$ is the sign of the permutation π.[1] Despite the similarity in definition, the permanent has much fewer properties than the determinant. No nice geometric or algebraic interpretation is known for permanent; and it is neither multiplicative nor invariant under linear combinations of rows or columns. Perhaps for this reason, permanents did not get much attention until the late 1970s, and just about everything known about it until then is in the book [10]. In 1979, Leslie Valiant [24] completely changed the view on permanents by showing that the complexity of computing permanent precisely captures the arithmetic version of the class NP, called VNP. Since then, properties of the permanent have been extensively studied by complexity theorists.

One of the most natural questions about permanents is to investigate its relationship with determinants. It is easy to see that the determinant of a matrix X can be expressed as the permanent of a related matrix $\hat{X}$ whose entries are 0, 1, or $x_{i,j}$s and which is of size $O(n)$ (set up entries of $\hat{X}$ such that $\det \hat{X} = \det X$ and the product corresponding to every permutation that has an even cycle is zero). For the converse, the best known bound on the size of a matrix $\hat{X}$ whose entries are constants and $x_{i,j}$s, and for which $\det \hat{X} = \mathrm{per}\, X$ is $2^{\Omega(n)}$. This suggests that the complexity of computing the permanent is much higher than that of the determinant. Although widely believed, this remains a conjecture. This conjecture has a close connection to the conjectured separation of arithmetic NP from arithmetic P (the class of all functions that can be efficiently computed by arithmetic operations, see next section for a precise definition). It is known that the complexity of determinant is close to the complexity of arithmetic P: every function computed by n arithmetic operations can be expressed as determinant of a matrix of size $n^{O(\log n)}$. This lends more importance to the problem of settling the conjecture.

There have been some attempts to answer the conjecture positively [14], [6], [15] [8]. A sequence of arithmetic operations can be modeled as an *arithmetic circuit*, and the size of an arithmetic circuit is the number of arithmetic operations in the sequence. In [8], *monotone* circuits were considered, these are circuits in which no constant is negative. For computability by such restricted circuits, an exponential lower bound was shown for the complexity of permanent. A different restriction on arithmetic circuits is that of depth – the number of layers of operations. These circuits were considered in [14], [19], [6] and lower bounds were shown for the complexity of computing permanent by depth three circuits. Finally, [15] considers yet another restriction. In this restriction, every gate of the circuit is required to compute a multilinear polynomial. A superpolynomial lower bound is shown on *formulas* (circuits with outdegree one) of this kind computing permanent.

All the above lower bounds hold for very restricted settings, and the techniques used do not seem to generalize. Over the last few years, however, two new tech-

[1] Both permanent and determinant are special forms of *immanents* defined as $\mathrm{imm}_\chi\, X = \sum_{\pi \in S_n} \chi(\pi) \cdot \prod_{i=1}^{n} x_{i,\pi(i)}$ where $\chi : S_n \mapsto \mathbb{C}$ is a character of S_n. For the permanent, $\chi = \mathrm{id}$ and for the determinant, χ equals the sign of the permutation.

niques have been proposed that hold some promise. The first of these was proposed by Mulmuley and Sohoni [11]. They transform the problem to algebraic geometry domain where it is reduced to showing that the permanent polynomial does not lie in the closure of a certain *orbit* of the determinant polynomial.

The second approach was proposed by Kabanets and Impagliazzo [9]. They reduced the problem to that of finding a deterministic, subexponential-time algorithm for the *Identity Testing*. The Identity Testing problem is defined as follows: given an arithmetic circuit computing polynomial p in n variables, test if $p = 0$. There exist several randomized polynomial-time algorithms for solving this. Kabanets and Impagliazzo show that *any* deterministic, subexponential-time algorithm for the problem will imply either a superpolynomial lower bound for arithmetic circuits computing permanent, or a boolean lower bound on the class NEXP. This connection was strengthened in [1] to show that if there exist special kinds of deterministic algorithms for testing identities given by superconstant depth arithmetic circuits, then permanent requires superpolynomial sized arithmetic circuits.

In this article, we will describe the known results on lower bounds on permanent as well as the two new approaches outlined above.

2. Definitions

$\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$ are respectively the fields of rational, real, and complex numbers.

An arithmetic circuit over a field F is a directed, acyclic graph with labelled vertices. Vertices of indegree zero are labelled with either a variable x_i or a constant from F. Vertices labelled with variables are called *input gates*. The remaining vertices are labelled with either '+' or '$*$' and are called *addition* or *multiplication gates* respectively. Vertices with outdegree zero are also called *output gates*. We restrict our attention to circuits with exactly one output gate. The *fanin* of a gate equals the number of edges incident to the gate. In this article, gates have unbounded fanin when not mentioned otherwise. The *size* of a circuit C equals the number of gates in it. The *depth* of a circuit C equals the length of the longest path from an input gate to an output gate. The *degree* of C is inductively defined as follows: the degree of an input gate is one or zero depending on whether it is labelled by a variable or constant; the degree of an addition gate is the maximum of the degree of the gates whose edges are incident to the gate; the degree of a multiplication gate is the sum of the degrees of the gates whose edges are incident to the gate; finally, the degree of C is the degree of its output gate.

An arithmetic circuit C computes a polynomial as follows. The polynomial computed at an input gate equals the label of the gate. For any other gate g, let $g_1, \ldots, g_k$ be all the gates that have an edge incident to g and let p_{g_i} be the polynomial computed at gate g_i. Then the polynomial computed at the gate g equals $\sum_{i=1}^{k} p_{g_i}$ if g is an addition gate, and equals $\prod_{i=1}^{k} p_{g_i}$ if g is a multiplication gate. The polynomial computed by the circuit is the polynomial computed at its output gate.

Let $\{p_n\}_{n>0}$ be a family of polynomials with p_n a polynomial of degree $d(n)$ in n variables. A circuit family $\{C_n\}_{n>0}$ is said to compute $\{p_n\}$ if for every n, the polynomial computed by C_n equals p_n. In the following we shall simply write $\{p_n\}$ for the family $\{p_n\}_{n>0}$.

Arithmetic branching programs are a restricted form of arithmetic circuits in which every multiplication gate has fanin exactly two, and in addition at least one of the two gates, from which edges are incident to the multiplication gate, is an input gate. Such circuits are also called *skew* circuits.

The class VP_F, the arithmetic analog of class P, is defined to be the set of polynomial families $\{p_n\}$ over a field F such that (1) each p_n is of degree $n^{O(1)}$, and (2) there exists a circuit family $\{C_n\}$ computing $\{p_n\}$ such that C_n is of size $n^{O(1)}$.[2] The class VNP_F, the arithmetic analog of class NP, is defined to be the set of polynomial families $\{p_n\}$ over a field F such that (1) each p_n is of degree $n^{O(1)}$, and (2) there exists a family of polynomials $\{q_n\} \in \text{VP}_F$ such that for every n,

$$p_n(x_1, x_2, \dots, x_n) = \sum_{y_1=0}^{1} \sum_{y_2=0}^{1} \cdots \sum_{y_m=0}^{1} q_{n+m}(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m)$$

with $m = n^{O(1)}$.[3]

Given two polynomials $p(x_1, x_2, \dots, x_n)$ and $q(y_1, y_2, \cdots, y_m)$ over a field F, we say that p is a *projection* of q if $p(x_1, x_2, \dots, x_n) = q(\alpha_1, \alpha_2, \cdots, \alpha_m)$ where each $\alpha_i \in F \cup \{x_1, x_2, \dots, x_n\}$. Given two polynomial families $\{p_n\}$ and $\{q_n\}$, we say that $\{p_n\}$ is a *p-projection* of $\{q_n\}$ if for every n there exists an $m = n^{O(1)}$ such that p_n is a projection of q_m.[4]

Let $\text{per}_F = \{\text{per}_{F,n}\}$ and $\det_F = \{\det_{F,n}\}$ denote the families of permanent and determinant polynomials over a field F respectively. Both these families contain polynomials in n^2 variables for each n.

Valiant [24] proved that:

Theorem 2.1 ([24]). *For any F, $\text{per}_F \in \text{VNP}_F$. In addition, for any F, $\text{char}(F) \neq 2$, for any polynomial family $\{p_n\}$ in VNP_F, $\{p_n\}$ is a p-projection of per_F.*

So permanent is as hard to compute as any polynomial family in VNP. In contrast, determinant can be computed efficiently. A nice characterization of determinant was shown in [4], [21], [25]:

Theorem 2.2 ([4], [21], [25]). *For any F, $\det_F$ can be computed by polynomial-sized arithmetic branching programs. In addition, for any F and for any polynomial*

[2] In addition, circuit C_n must be efficiently computable given 1^n. This property does not seem to play any role in obtaining lower bounds.

[3] The class $\#\text{P}_F$ is the 'functional' version of the class VNP_F: a polynomial family $\{p_n\} \in \text{VNP}_F$ belongs to $\#\text{P}_F$ when for each n, p_n is viewed as a map from F^n to F.

[4] Again, given 1^n, the projection specified by $(\alpha_1, \alpha_2, \dots, \alpha_m)$ should be efficiently computable.

family $\{p_n\}$ computed by polynomial-sized arithmetic branching programs, $\{p_n\}$ is a p-projection of $\det_F$.

In fact, all families in VP are *almost* p-projections of determinant.

Theorem 2.3 ([23]). *Let C be a circuit of size s computing a polynomial of degree d. There exists another circuit computing the same polynomial of size $s^{O(1)}$ and depth $O(\log s + \log d)$.*

Corollary 2.4. *Any circuit family in* VP_F *can be computed by circuit families of polynomial size and logarithmic depth.*

Corollary 2.5. *For every circuit family $\{p_n\} \in \mathrm{VP}_F$ and for every n, p_n is a projection of* $\det_{F,m}$ *where $m = n^{O(\log n)}$.*

The above characterizations of complexities of determinant and permanent imply that, in order to separate VP_F from VNP_F, it is enough to show that per_F is not an almost p-projection of $\det_F$ (in the sense above).

3. Known lower bounds on permanent

Lower bounds are known on permanent only for restricted circuits. In this section, we describe important lower bounds of this kind. Three major restrictions have been considered for proving such lower bounds: *monotone* circuits, *constant depth* circuits, and *multilinear* formulas.

3.1. Monotone circuits. A circuit over $\mathbb{Q}$ or $\mathbb{R}$ is *monotone* if all the constants in the circuit are non-negative. This is a very restricted class of circuits since *no* cancellations can occur in it. Jerrum and Snir [8] showed that any monotone circuit family that computes permanent must have exponential size.

3.2. Constant depth circuits. In this restriction, the depth of a circuit family is fixed, i.e., it is independent of n. Permanent (or any polynomial of degree $n^{O(1)}$ for that matter) can be computed by an exponential size depth two circuit family. Conversely, it is easy to see that any depth two circuit family computing permanent must have exponential size.

Depth three circuit families are, however, non-trivial. A depth three circuit can be of the form "sum-of-products-of-sums" or "product-of-sums-of-products." The latter form can easily be seen to require exponential size to compute permanent (the topmost multiplication gate can be shown to be redundant transforming the circuit to a depth two circuit). Circuit families of the first form are powerful: Ben Or observed that they can efficiently compute all symmetric polynomials of degree $n^{O(1)}$ over fields of characteristic zero.

The best known lower bound in fields of characteristic zero is by Shpilka and Wigderson [19] who proved that permanent (and determinant) requires a circuit family of size $\Omega(n^2)$. Their idea is to consider the space spanned by all partial derivatives of the polynomials computed at each gate of a given circuit. They show that for a depth three circuit with small size, the space spanned by the derivatives of the output polynomial would be of small dimension while the space spanned by derivatives of permanent is of large dimension.

Over finite fields, the situation is better. Grigoriev and Razborov [6] showed an exponential lower bound on the size of depth three circuit families computing determinant and permanent. Their approach was to show that polynomial computed by a depth three circuit of small size can be 'approximated' by a low-degree polynomial (approximated in the sense that the two polynomials agree on a large set of points from the field). Then they observed that determinant and permanent cannot be approximated by low-degree polynomials.

3.3. Multilinear formulas. *Multilinear formulas* are circuits such that (1) the out-degree of every gate is at most one, and (2) the polynomial computed at every gate is multilinear. Such circuits have severely limited multiplication gates – the polynomials input to a multiplication gate must be over disjoint sets of variables. Using a combination of partial derivative technique and random restrictions (setting some input variables to random values), Raz [15] proved a lower bound of $n^{\Omega(\log n)}$ on the size of families of multilinear formulas computing permanent and determinant.

4. The algebraic geometry approach

Mulmuley and Sohoni [11] have offered a completely new approach to the problem of proving a lower bound on permanent for unrestricted circuits by transforming the problem to geometric settings. In this section, we give a brief overview of their approach.

Suppose, for $F = \mathbb{Q}$, $\mathrm{per}_{F,n}$ is a projection of $\det_{F,m}$, $m > n$. Define $\widehat{\mathrm{per}}_{F,m} = x_{m^2}^{m-n} \cdot \mathrm{per}_{F,n}$. It follows that $\widehat{\mathrm{per}}_{F,m}$ is also a projection of $\det_{F,m}$ (just multiply all constants of the projection by x_{m^2}). This can be written as

$$\widehat{\mathrm{per}}_{F,m}(x_1, x_2, \ldots, x_{m^2}) = A \cdot \det_{F,m} = \det_{F,m}((x_1, x_2, \ldots, x_{m^2}) \cdot A),$$

where A is an $m^2 \times m^2$ matrix over $\mathbb{Q}$. The matrix A is singular whenever $m > n$ since the variables $x_{n^2+1}, \ldots, x_{m^2-1}$ do not occur in $\widehat{\mathrm{per}}_{F,m}$. Let $A_{\bar{\varepsilon}}$ be a slight 'perturbation' of A obtained by adding $\varepsilon_{i,j}$ to the (i,j)th entry of A. For nearly all values of $\bar{\varepsilon}$ close to zero, $A_{\bar{\varepsilon}}$ is non-singular and the polynomial $A_{\bar{\varepsilon}} \cdot \det_{F,m}$ approximates the polynomial $\widehat{\mathrm{per}}_{F,m}$ very well (all the coefficients of two polynomials are close to each other). Now consider the space $V = \mathbb{C}^M$ with $M = \binom{m^2+m-1}{m}$. Every homogeneous polynomial of degree m in m^2 variables can be viewed as a point in this space (degree m monomials forming the basis). So both $\det_{F,m}$ and $\widehat{\mathrm{per}}_{F,m}$ are points in V (since $F = \mathbb{Q}$ and both polynomials are of degree m in m^2 variables). Let O be the orbit of $\det_{F,m}$ under the action of $\mathrm{GL}_{m^2}(\mathbb{C})$, i.e.,

$$O = \{B \cdot \det_{F,m} \mid B \text{ is an invertible matrix over } \mathbb{C}\}.$$

Set O can be viewed as a set of points in V. The above argument shows the following:

Lemma 4.1 ([11]). *If* $\mathrm{per}_{F,n}$ *is a projection of* $\det_{F,m}$ *then the point corresponding to* $\widehat{\mathrm{per}}_{F,m}$ *in V lies in the closure of the set O in V. Conversely, if* $\widehat{\mathrm{per}}_{F,m}$ *lies in the closure of O then* $\mathrm{per}_{F,n}$ *can be approximated by projections of* $\det_{F,m}$ *to any desired accuracy.*

This (near) characterization is the starting point of their approach. Instead of V, we can work in the projective space $P(V)$ too since both the polynomials are homogeneous. The same near characterization holds in $P(V)$ as well with $\mathrm{GL}_{m^2}(\mathbb{C})$ replaced by $\mathrm{SL}_{m^2}(\mathbb{C})$, the group of all matrices with determinant 1. The advantage of working in $P(V)$ is that the closure of O (under the classical Euclidean topology) coincides with the closure of O under *Zariski topology* [12]. In Zariski topology, there is the well-studied notion of *stability* that captures this problem: $\det_{F,m}$ is $\widehat{\mathrm{per}}_{F,m}$*-stable under* $\mathrm{SL}_{m^2}(\mathbb{C})$ if $\widehat{\mathrm{per}}_{F,m}$ lies in the closure of the orbit O (we abuse notation here by using the same names for polynomials and sets in $P(V)$ as for the corresponding ones in V).

Points in the orbit O have a useful property. For any point $p \in P(V)$, let

$$G_p = \{A \in \mathrm{SL}_{m^2}(\mathbb{C}) \mid A \cdot p = p\}.$$

The group G_p is called the *stabilizer* of p.

Lemma 4.2. *For any point $p \in O$, G_p is a conjugate of $G_{\det_{F,m}}$.*

Proof. Let $p = B \cdot \det_{F,m} \in O$. Then $G_p = B \cdot G_{\det_{F,m}} \cdot B^{-1}$. □

Suppose the orbit of the polynomial $\widehat{\mathrm{per}}_{F,m}$ under $\mathrm{SL}_{m^2}(\mathbb{C})$ is a closed set (such polynomials are called *stable*). Let Q be the orbit of $\widehat{\mathrm{per}}_{F,m}$ under $\mathrm{SL}_{m^2}(\mathbb{C})$. By *Luna's slice theorem*, there is a neighborhood N of Q such that for any point $p \in N$, G_p is a conjugate of a subgroup of $G_{\widehat{\mathrm{per}}_{F,m}}$. Since the closure of O contains $\widehat{\mathrm{per}}_{F,m}$, there is a point in N, say q, such that $q = B \cdot \det_{F,m}$. This means G_q is a conjugate of $G_{\det_{F,m}}$. Therefore, $G_{\det_{F,m}}$ is a conjugate of a subgroup of $G_{\widehat{\mathrm{per}}_{F,m}}$. On the other hand, it is well known that $G_{\det_{F,m}}$ is 'larger' than $G_{\widehat{\mathrm{per}}_{F,m}}$: $G_{\det_{F,m}}$ is characterized by the transformations of the kind $X \mapsto A \cdot X \cdot B^{-1}$ where $A, B \in \mathrm{GL}_m(\mathbb{C})$ while $G_{\widehat{\mathrm{per}}_{F,m}}$ is characterized by the transformations of the kind $X \mapsto A \cdot X \cdot B^{-1}$ where $A, B \in \mathrm{GL}_m(\mathbb{C})$ and both A and B are either diagonal or permutation matrices. Therefore, $G_{\det_{F,m}}$ cannot be a conjugate of a subgroup of $G_{\widehat{\mathrm{per}}_{F,m}}$. (This is a rough argument; to make it precise, more work is needed.)

Unfortunately, $\widehat{\text{per}}_{F,m}$ is *not* stable (interestingly, $\text{per}_{F,n}$ is stable in the smaller dimensional space defined by degree n homogeneous polynomials in n^2 variables; the translation to higher dimensional space ruins the stability). Mulmuley and Sohoni define the notion of *partial stability* and show that $\widehat{\text{per}}_{F,m}$ is partially stable. Now their aim is to make the above argument work even for partially stable points. A more detailed explanation of their approach is in [16].

5. The derandomization approach

Kabanets and Impagliazzo [9] have discovered another new approach for proving lower bounds on permanent. Unlike the previous one, this approach is based on arithmetic circuits. In this section we outline their approach and its variation in [1].

The *Identity Testing* problem is defined as follows: given an arithmetic circuit C over a field F as input, decide if the polynomial computed by the circuit is the zero polynomial. This is a classical problem in computational algebra and there exist several randomized polynomial-time algorithms for it. Perhaps the simplest one is by Schwartz and Zippel [17], [26]: randomly choose values for variables of C from a set in F of size $2d$, here d is the degree of C (if $|F| < 2d$ then extend F slightly); output ZERO if C evaluates to zero, otherwise NON-ZERO. An easy argument shows that this test is correct with probability at least $\frac{1}{2}$ when C computes a non-zero polynomial and always correct when C computes a zero polynomial.

Kabanets and Impagliazzo show that if there exists a deterministic subexponential ($= 2^{n^{o(1)}}$) time algorithm for solving Identity Testing problem then at least one of the following two lower bounds hold:

1. NEXP requires superpolynomial sized boolean circuits.
2. Permanent requires superpolynomial sized arithmetic circuits.

To see this, suppose that permanent has polynomial sized arithmetic circuits for some field F of characteristic different from two. Consider a non-deterministic machine that, on input 1^n, guesses the circuit that computes $\text{per}_{F,n}$ and verifies it to be correct. It does this by inductively verifying that the circuit, under appropriate settings of its inputs, computes $\text{per}_{F,n-1}$ correctly and then verifying the equation for $\text{per}_{F,n}$ that expresses it in terms of $\text{per}_{F,n-1}$. Verifying the equation is an instance of Identity Testing problem and so can be done in subexponential time by assumption. Therefore, given any matrix $A \in F^{n^2}$, per A can be computed in non-deterministic subexponential time. Now assume that NEXP has polynomial sized boolean circuits. By [3], [22], it follows that $\text{NEXP} \subseteq \text{P}^{\#\text{P}}$. Since the complexity of #P is exactly the complexity of computing permanent, it follows that NEXP is in non-deterministic subexponential time contradicting the non-deterministic time hierarchy theorem [18].

This result falls short of pointing a way for proving lower bounds on permanent – besides finding a deterministic algorithm for Identity Testing, one needs to assume

NEXP has polynomial sized boolean circuits which is very unlikely to be true. However, it *does* point to a connection between Identity Testing problem and permanent lower bounds. This connection was strengthened in [1] by defining *pseudo-random generators* for arithmetic circuits. Pseudo-random generators in the boolean settings have been studied intensively (see, e.g., [5], [13], [7], [20]). It is known that constructing pseudo-random generators is equivalent to proving lower bounds in the boolean settings. In [1], pseudo-random generators are defined in arithmetic settings and a similar equivalence is observed.

Let $\mathcal{A}C_F$ be the class of all arithmetic circuits over F and $\mathcal{A}_F \subseteq \mathcal{A}C_F$.

Definition 5.1. A function $f: \mathbb{N} \to (F[y])^*$ is called an $(\ell(n), n)$-*pseudo-random generator* against $\mathcal{A}_F$ if the following holds:

- $f(n) \in (F[y])^{n+1}$ for every $n > 0$.
- Let $f(n) = (f_1(y), \ldots, f_n(y), g(y))$. Then each $f_i(y)$ as well as $g(y)$ is a polynomial of degree at most $2^{\ell(n)}$.
- For any circuit $C \in \mathcal{A}_F$ of size n with $m \leq n$ inputs:

$$C(x_1, x_2, \ldots, x_m) = 0 \text{ iff } C(f_1(y), f_2(y), \ldots, f_m(y)) = 0 \pmod{g(y)}.$$

A direct application of Schwartz–Zippel lemma [17], [26] shows that there always exist $(O(\log n), n)$-pseudo-random generators against $\mathcal{A}C_F$. Call such generators *optimal* pseudo-random generators. Pseudo-random generators that can be efficiently computed are of special interest.

Definition 5.2. An $(\ell(n), n)$-pseudo-random generator f against $\mathcal{A}_F$ is *efficiently computable* if $f(n)$ is computable in time $2^{O(\ell(n))}$.

An easy argument shows that if there exists an efficiently computable $(\ell(n), n)$-pseudo-random generator against $\mathcal{A}C_F$ then the Identity Testing problem can be solved deterministically in time $2^{O(\ell(n))}$: evaluate the given circuit C of size n modulo $g(y)$ after substituting for the i^{th} input variable the polynomial $f_i(y)$ where $f(n) = (f_1(y), \ldots, f_n(y), g(y))$. In particular, if there exists an efficiently computable optimal pseudo-random generator against $\mathcal{A}C_F$ then Identity Testing can be solved in polynomial time.

An efficiently computable pseudo-random generator also results in a lower bound.

Theorem 5.3 ([1]). *Let f be an efficiently computable $(\ell(n), n)$-pseudo-random generator against $\mathcal{A}_F$. Then there is a multilinear polynomial in $2\ell(n)$ variables, computable in time $2^{O(\ell(n))}$, that cannot be computed by any circuit in $\mathcal{A}_F$ of size n.*

Proof. For any $m = \ell(n)$, define the polynomial $q_f(x_1, x_2, \ldots, x_{2m})$ by

$$q_f(x_1, x_2, \ldots, x_{2m}) = \sum_{S \subseteq [1,2m]} c_S \cdot \prod_{i \in S} x_i.$$

The coefficients c_S satisfy the condition

$$\sum_{S \subseteq [1,2m]} c_S \cdot \prod_{i \in S} f_i(y) = 0$$

where $f(n) = (f_1(y), f_2(y), \ldots, f_n(y), g(y))$. Such a q_f always exists as the following argument shows.

> The number of coefficients of q_f are exactly 2^{2m}. These need to satisfy a polynomial equation of degree at most $2m \cdot 2^m$. So the equation gives rise to at most $2m \cdot 2^m + 1$ homogeneous constraints on the coefficients. Since $(2m \cdot 2^m + 1) < 2^{2m}$ for $m \geq 3$, there is always a non-trivial polynomial q_f satisfying all the conditions.

The polynomial q_f can be computed by solving a system of $2^{O(m)}$ linear equations in $2^{O(m)}$ variables over the field F. Each of these equations can be computed in time $2^{O(m)}$ using computability of f. Therefore, q_f can be computed in time $2^{O(m)}$. Now suppose q_f can be computed by a circuit $C \in \mathcal{A}_F$ of size n. By the definition of the polynomial q_f, it follows that $C(f_1(y), f_2(y), \ldots, f_{2m}(y)) = 0$. The size of circuit C is n and it computes a non-zero polynomial. This contradicts the pseudo-randomness of f. □

A partial converse of this theorem can also be shown: if there exists a polynomial family computable in time $2^{O(\ell(n))}$ that cannot be computed by any size n circuit family in $\mathcal{A}_F$ then there exists an efficiently computable $(\ell^2(n), n)$-pseudo-random generator against $\mathcal{A}_F$, when the degree of every size n circuit in $\mathcal{A}_F$ is bounded by $n^{O(1)}$.

An efficient optimal pseudo-random generator against $\mathcal{AC}_F$ yields a polynomial that requires exponential (in the number of variables) sized circuits. However, it is not clear whether the polynomial q_f can be computed as permanent of a matrix of size $m^{O(1)}$. To get this, one needs to show that all the coefficients c_S of q_f are themselves efficiently computable. If this is done, then using the VNP characterization of permanent, it follows that q_f equals the permanent of a matrix of size $m^{O(1)}$. This results in an exponential lower bound on permanent.

For a superpolynomial lower bound one needs either an $(n^{o(1)}, n)$-pseudo random generator against $\mathcal{AC}_F$ or an optimal pseudo-random generators against a much smaller class of circuits.

Theorem 5.4 ([1]). *Let f be an efficiently computable optimal pseudo-random generator against the class of circuits of depth $\omega(1)$ such that the associated polynomial q_f is in* VNP. *Then permanent cannot computed by any polynomial sized circuit.*

Proof. From the previous theorem, it follows that the polynomial q_f cannot be computed by exponential sized circuits of depth $\omega(1)$. A size n^d, depth $d \log n$ arithmetic circuit with fanin two multiplication gates can be translated to a subexponential sized depth d circuit by "cutting" the circuit into $\log n$ layers of depth d each, and then "flattening" each layer to a subexponential sized circuit of depth two. Since every polynomial sized circuit computing permanent can be transformed to a depth $O(\log n)$, size $n^{O(1)}$ circuit with fanin two multiplication gates [23], the theorem follows. □

It is not clear at the moment how to construct optimal pseudo-random generators against constant depth circuits. In [1] a generator is conjectured. Unconditionally, we only know generators against depth two, polynomial sized circuits (the proof is easy, see [1]). We know an optimal generator against the following very special class of circuits too:

$$\mathcal{A} = \{C_n(x) \mid C_n(x) = (1+x)^n - 1 - x^n \text{ over the ring } \mathbb{Z}_n\}.$$

Notice that the circuits in the class $\mathcal{A}$ are not over a fixed field (or ring), and the size of the circuit C_n is $O(\log n)$ and the degree is n. In [2], the following optimal generator was constructed against $\mathcal{A}$:

$$f(m) = \Big(x, 0, \dots, 0, x^{16m^5} \cdot \prod_{r=1}^{16m^5} \prod_{a=1}^{4m^4} ((x-a)^r - 1)\Big).$$

6. Concluding remarks

The problem of proving that the permanent of a size n matrix cannot be expressed as determinant of size $n^{O(\log n)}$ matrix is of great importance in complexity theory. While the existing approaches have failed to shed light on this, one hopes that at least one of the two new approaches will eventually lead to a solution of the problem.

Acknowledgements. I wish to thank Somenath Biswas for enjoyable discussions and help in preparing this article.

References

[1] Agrawal, M., Proving lower bounds via pseudo-random generators. In *Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Comput. Sci. 3821, Springer-Verlag, Berlin 2005, 92–105.

[2] Agrawal, M., On derandomizing tests for certain polynomial identities. In *Proceedings of 18th Annual IEEE Conference on Computational Complexity*, IEEE Computer Society, Los Alamitos, CA, 2003, 355–362.

[3] Babai, L., Fortnow, L., Nisan, N., and Wigderson, A., BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Comput. Complexity* **3** (4) (1963), 307–318.

[4] Damm, C., DET=L$^{\#L}$. Technical Report Informatik, Preprint 8, Fachbereich Informatik der Humboldt Universität zu Berlin, 1991.

[5] Goldreich, O., *Foundation of Cryptography I: Basic Tools*. Cambridge University Press, Cambridge 2001.

[6] Grigoriev, D., and Razborov, A:, Exponential lower bounds for depth 3 arithmetic circuits in algebras of functions over finite fields. *Appl. Algebra Engrg. Comm. Comput.* **10** (6) (2000), 467–487, 2000.

[7] Impagliazzo, R., and Wigderson, A., P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, ACM Press, New York 1997, 220–229.

[8] Jerrum, M., and Snir, M., Some exact complexity results for straight-line computations over semirings. *J. ACM* **29** (3) (1982), 874–897.

[9] Kabanets, Valentine, and Impagliazzo, Russell, Derandomizing polyonmial identity tests means proving circuit lower bounds. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, ACM Press, New York 2003, 355–364.

[10] Minc, H., *Permanents*. Addision-Wesley, 1978.

[11] Mulmulay, K., and Sohoni, M., Geometric complexity theory, P vs. NP, and explicit obstructions. *SIAM J. Comput.* **31** (2) (2002), 496–526.

[12] D. Mumford, D., *Algebraic Geometry I: Complex Projective Varieties*. Grundlehren Math. Wiss. 221, Springer-Verlag, Berlin 1976.

[13] Nisan, N., and Wigderson, A., Hardness vs. randomness. *J. Comput. System Sci.* **49** (2) (1994), 149–167.

[14] Nisan, N., and Wigderson, A., Lower bounds on arithmetic circuits via partial derivatives. *Comput. Complexity* **6** (3) (1996/97), 217–234.

[15] Raz, Ran, Multi-linear formulas for permanent and determinant and of super-polynomial size. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, ACM Press, New York 2004, 633–641.

[16] Regan, K., Understanding the Mulmuley-Sohoni approach to P vs. NP. *Bulletin of the European Association for Theoretical Computer Science* **78** (2002), 86–97. Lance Fortnow's Computational Complexity Column.

[17] Schwartz, J. T., Fast probabilistic algorithms for verification of polynomial identities. *J. ACM* **27** (4) (1980), 701–717.

[18] Seiferas, J., Fischer, M., and Meyer, A., Separating nondeterministic time complexity classes. *J. ACM* **25** (1) (1987), 146–167.

[19] Shpilka, A., and Wigderson, A., Depth-3 arithmetic circuits over fields of characteristic zero. *Comput. Complexity* **10** (1) (2001), 1–27.

[20] Sudan, M., Trevisan, L., and Vadhan, S., Pseudorandom generators without the XOR lemma. In *Proceedings of the 31th Annual ACM Symposium on Theory of Computing*, ACM Press, New York 1999, 537–546.

[21] Toda, S., Counting problems computationally equivalent to the determinant. Manuscript, 1991.

[22] Toda, S., PP is as hard as the polyonmial-time hierarchy. *SIAM J. Comput.* **20** (1991), 865–877.

[23] Valiant, L., Skyum, S., Berkowitz, S., and Rackoff, C., Fast parallel computation of polynnomials using few processors. *SIAM J. Comput.* **12** (1983), 641–644.

[24] Valiant, L., Completeness classes in algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing*, ACM Press, New York 1979, 249–261.

[25] Vinay, V., Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits. In *Proceedings of the Structure in Complexity Theory Conference*, Lecture Notes in Comput. Sci. 223, Springer-Verlag, Berlin 1991, 270–284.

[26] Zippel, R. E., Probabilistic algorithms for sparse polynomials. In *EUROSCAM'79*, Lecture Notes in Comput. Sci. 72, Springer-Verlag, Berlin 1979, 216–226.

Department of Computer Science and Engineering, Indian Institute of Technology,
Kanpur 208016, India
E-mail: manindra@iitk.ac.in