Partial Derivatives in Arithmetic Complexity and Beyond

Partial Derivatives in Arithmetic Complexity and Beyond

Xi Chen

Columbia University
USA
xichen@cs.columbia.edu

Neeraj Kayal

Microsoft Research
India
neeraka@microsoft.com

Avi Wigderson

Institute for Advanced Study
USA
avi@ias.edu

the essence of knowledge

Boston – Delft

Foundations and Trends® in Theoretical Computer Science

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
USA
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is X. Chen, N. Kayal and A. Wigderson, Partial Derivatives in Arithmetic Complexity and Beyond, Foundations and Trends® in Theoretical Computer Science, vol 6, nos 1–2, pp 1–138, 2010

ISBN: 978-1-60198-480-7

Foundations and Trends® in Theoretical Computer Science
Volume 6 Issues 1–2, 2010

Editorial Scope

Foundations and Trends® in Theoretical Computer Science will publish survey and tutorial articles in the following topics:

- Algorithmic game theory
- Computational algebra
- Computational aspects of combinatorics and graph theory
- Computational aspects of communication
- Computational biology
- Computational complexity
- Computational geometry
- Computational learning
- Computational Models and Complexity
- Computational Number Theory
- Cryptography and information security
- Data structures
- Database theory
- Design and analysis of algorithms
- Distributed computing
- Information retrieval
- Operations Research
- Parallel algorithms
- Quantum Computation
- Randomness in Computation

Information for Librarians

Foundations and Trends® in Theoretical Computer Science, 2010, Volume 6, 4 issues. ISSN paper version 1551-305X. ISSN online version 1551-3068. Also available as a combined paper and online subscription.

Foundations and Trends® in
Theoretical Computer Science
Vol. 6, Nos. 1–2 (2010) 1–138

DOI: 10.1561/0400000043

Partial Derivatives in Arithmetic Complexity and Beyond

Xi Chen[1], Neeraj Kayal[2] and Avi Wigderson[3]

[1] *Columbia University, New York, NY 10027, USA, xichen@cs.columbia.edu*
[2] *Microsoft Research, Bangalore, 560080, India, neeraka@microsoft.com*
[3] *Institute for Advanced Study, Princeton, NJ 08540, USA, avi@ias.edu*

Abstract

How complex is a given multivariate polynomial? The main point of this survey is that one can learn a great deal about the structure and complexity of polynomials by studying (some of) their partial derivatives. The bulk of the survey shows that partial derivatives provide essential ingredients in proving both upper and lower bounds for computing polynomials by a variety of natural arithmetic models. We will also see applications which go beyond computational complexity, where partial derivatives provide a wealth of structural information about polynomials (including their number of roots, reducibility and internal symmetries), and help us solve various number theoretic, geometric, and combinatorial problems.

Contents

1

Introduction

1.1 Motivation

Polynomials are perhaps the most important family of functions in mathematics. They feature in celebrated results from both antiquity and modern times, like the unsolvability by radicals of polynomials of degree ≥ 5 of Abel and Galois, and Wiles' proof of Fermat's "last theorem." In computer science they feature in, for example, error-correcting codes and probabilistic proofs, among many applications. The manipulation of polynomials is essential in numerous applications of linear algebra and symbolic computation. This survey is devoted mainly to the study of polynomials from a computational perspective. The books [9, 10, 86] and the recent survey [74] provide wide coverage of the area.

Given a polynomial over a field, a natural question to ask is how complex it is? A natural way to compute polynomials is via a sequence of arithmetic operations, for example, by an arithmetic circuit, as shown in Figure 1.1 (formal definitions will be given in Section 1.2). One definition of how complex a polynomial is can be the size of the smallest arithmetic circuit computing it. A weaker model, often employed by mathematicians, is that of a formula (in which the underlying circuit structure must be a tree), and another definition of complexity may be the formula size.

There are many ways to compute a given polynomial. For example,

$$f(x_1,x_2) = x_1 \times (x_1 + x_2) + x_2 \times (x_1 + x_2) = (x_1 + x_2) \times (x_1 + x_2)$$

are two formulae for the same polynomial f, the first requiring 5 operations and the second only 3 operations. Finding the optimal circuit or formula computing a given polynomial is a challenging task, and even estimating that minimum size by giving upper and lower bounds is very difficult. Of course, the same is also true for the study of Boolean functions and their complexity (with respect to Boolean circuits and formulae, or Turing machines), but in the Boolean case we have a better understanding of that difficulty (via results on relativization by Baker et al. [5], natural proofs due to Razborov and Rudich [64], and algebrization due to Aaronson and Wigderson [1]). For the arithmetic setting, which is anyway more structured, there seem to be more hope for progress.

Proving lower bounds for the complexity of polynomials has been one of the most challenging problems in theoretical computer science. Although it has received much attention in the past few decades, the progress of this field is slow. The best lower bound known in the general arithmetic circuit setting is still the classical $\Omega(n \log d)$ result by Baur and Strassen [6] (for some natural degree-d polynomials over n variables). Even for some very restricted models (e.g., constant-depth arithmetic circuits or multilinear formulae), a lot of interesting problems remain widely open. In this survey, we focus on the use of *partial derivatives* in this effort.

The study of upper bounds — constructing small circuits for computing important polynomials — is of course important for practical applications, and there are many nontrivial examples of such algorithms (e.g., Strassen's matrix multiplication algorithm [76], Berkowitz's algorithm for the determinant [7],[1] and Kaltofen's black-box polynomial factorization algorithm [34]). As we focus here on the uses of partial

[1] The first NC algorithm for the determinant, based on Leverier's method, was given by Csanky in 1976 [18]. However, Csanky's algorithm used divisions and was unsuitable for arbitrary fields. Around 1984, Berkowitz [7] and independently, Chistov [16] came up with

derivatives, we will see relatively few upper bounds, but we are certain that there is room for more, faster algorithms that use the partial derivatives of a polynomial when computing it.

The task of understanding arithmetic circuits and formulae naturally leads to the task of understanding the basic algebraic properties of the polynomials computed by such circuits and formulae. One such question is the following: given an arithmetic circuit, determine whether the polynomial computed by it is the identically zero polynomial or not. It turns out that besides being a natural scientific question, this question is also closely related to proving arithmetic circuit lower bounds, as shown by Impagliazzo and Kabanets [33]. Other natural structural questions relate to the symmetries of polynomials, the algebraic independence of systems of polynomials and more. Again, we will demonstrate the power of partial derivatives to help understand such structural questions.

1.1.1 Organization

The rest of this chapter is devoted to formal definitions of the computational models (arithmetic circuits and formulae, and their complexity measures), and of partial derivatives.

In Part I, we demonstrate how partial derivatives can be used to probe the structure of polynomials, via a list of very different examples. In particular, we will see how to use them to prove that algebraic independence has matroid structure, and to determine the symmetries of a given family of polynomials. Along the way we will see that "most" polynomials have high arithmetic complexity. We will use partial derivatives to derive simple linear algebraic proofs to some important results on the number of solutions of polynomial equations whose initial proofs used algebraic geometry. (These will include Wooley's proof of Bezout's theorem and Stepanov's proof of Weil's theorem). We will also see the power of partial derivatives in resolving a long-standing problem in combinatorial geometry [28, 35].

polylogarithmic depth arithmetic circuits for computing the determinant (and therefore also an NC algorithm for the determinant over arbitrary fields.)

In Part II, we will review some of the most elegant lower bound proofs in the field, which use partial derivatives as a basic tool. Other than the $\Omega(n \log d)$ lower bound by Baur and Strassen for general arithmetic circuits, we will also be looking at some very restricted models of computation. The simplest one is based on the observation that every polynomial of degree d can be expressed as the sum of dth powers of affine linear forms. We will see that partial derivatives allow us to prove pretty sharp lower bounds in this model. We will also use partial derivatives to derive lower bounds for depth-3 arithmetic circuits and multilinear formulae. Another model of computation is based on the observation that every polynomial can be expressed as the determinant of a square matrix whose entries are affine linear forms. We will show how the second-order partial derivatives can be used to prove a quadratic lower bound for the permanent polynomial in this model.

Finally, in Part III we will see how partial derivatives help in deriving upper bounds for various algebraic problems related to arithmetic circuits, such as identity testing, irreducibility testing, and equivalence testing.

Many of the chapters in these three parts can be read independently. For the few which need background from previous chapters, we specify it in the abstract.

1.2 Arithmetic Circuits

In this section, we define arithmetic circuits.

Let $\mathbb{F}$ be a field. Most of the time, it is safe to assume that $\mathbb{F}$ is of characteristic 0 or has a very large characteristic, e.g., $\mathrm{char}(\mathbb{F})$ is much larger than the degree of any relevant polynomial. We will point out explicitly when the results also hold for fields of small characteristic.

The underlying structure of an arithmetic circuit $\mathcal{C}$ is a directed acyclic graph $G = (V, E)$. We use u, v, and w to denote vertices in V, and uv to denote a directed edge in E. The role of a vertex $v \in V$ falls into one of the following cases:

(1) If the in-degree of v is 0, then v is called an input of the arithmetic circuit;

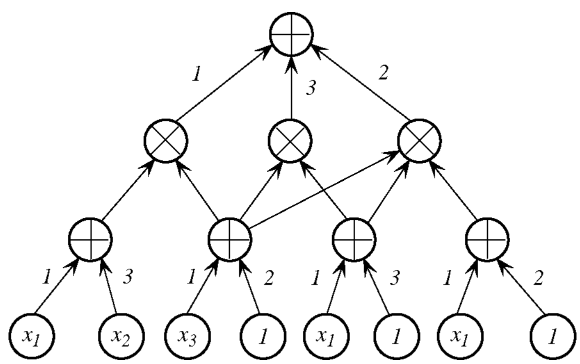

Fig. 1.1 A depth-3 Arithmetic Circuit over $\mathbb{F}[x_1,x_2,x_3]$.

(2) Otherwise, v is called a *gate*. In particular, if the out-degree of v is 0, then v is called an output (gate) of the circuit.

For most of the time, we will only discuss arithmetic circuits that compute one polynomial and have a single output gate. In this case, we will denote it by $\mathsf{out}_{\mathcal{C}} \in V$ (or simply $\mathsf{out} \in V$).

Every input vertex in V is labeled with either one of the variables $x_1,\ldots,x_n$ or one of the elements in the field $\mathbb{F}$. Every gate is labeled with either "+" or "×," which are called *plus* gates and *product* gates respectively. Each edge $uv \in E$ entering a plus gate is also labeled with an element c_{uv} in $\mathbb{F}$ (so plus gates perform "weighted addition" or in other words linear combinations of their inputs with field coefficients). See Figure 1.1 for an example.

Given an arithmetic circuit $\mathcal{C}$, we associate with each vertex $v \in V$ a polynomial $\mathcal{C}_v$, as the polynomial computed by $\mathcal{C}$ at v. Let $N^+(v)$ denote the set of successors and $N^-(v)$ denote the set of predecessors of v, then we define $\mathcal{C}_v$ inductively as follows: If v is an input, then $\mathcal{C}_v$ is exactly the label of v. Otherwise (since G is acyclic, when defining $\mathcal{C}_v$, we may assume the $\mathcal{C}_u$'s, $u \in N^-(v)$, have already been defined):

(1) If v is a plus gate, then

$$\mathcal{C}_v = \sum_{u \in N^-(v)} c_{uv} \cdot \mathcal{C}_u,$$

where $c_{uv} \in \mathbb{F}$ is the label of $uv \in E$;

(2) If v is a product gate, then

$$\mathcal{C}_v = \prod_{u \in N^-(v)} \mathcal{C}_u.$$

In particular, the polynomial $\mathcal{C}_{\mathsf{out}}$ associated with the output gate out is the polynomial computed by $\mathcal{C}$. We sometimes use $\mathcal{C}(x_1,\ldots,x_n)$ to denote the polynomial $\mathcal{C}_{\mathsf{out}}$ for short. We also need the notion of the *formal degree* of an arithmetic circuit, which is defined inductively using the following two basic rules:

(1) If $v \in V$ is a plus gate, then the formal degree of v is the maximum of the formal degrees of the vertices $u \in N^-(v)$;
(2) If $v \in V$ is a product gate, then the formal degree of v is the sum of the formal degrees of the vertices $u \in N^-(v)$.

Definition 1.1. The size of an arithmetic circuit, denoted by $\mathsf{S}(\mathcal{C})$, is the number of edges of its underlying graph.

Given a polynomial f, we let $\mathsf{S}(f)$ denote the size of the smallest arithmetic circuit computing f, that is,

$$\mathsf{S}(f) \stackrel{\text{def}}{=} \min_{\mathcal{C}:\, \mathcal{C}_{\mathsf{out}}=f} \mathsf{S}(\mathcal{C}).$$

The second way to define an arithmetic circuit (often referred to as a "straight-line program"), which is more convenient in certain situations, is to view it as a sequence of "+" and "×" operations:

$$\mathcal{C} = \big(g_1,\ldots,g_n,\ldots,g_m\big),$$

in which $g_i = x_i$ for all $i \in [n] = \{1,\ldots,n\}$. For each $k > n$, either

$$g_k = \sum_{i\in S} c_i \cdot g_i + c \quad \text{or} \quad g_k = \prod_{i \in S} g_i,$$

where $c, c_i \in \mathbb{F}$ and S is a subset of $[k-1]$. Similarly, we can define a polynomial $\mathcal{C}_i$ for each g_i and the polynomial computed by $\mathcal{C}$ is $\mathcal{C}_m$.

As a warm up, we take a brief look at the polynomials of the simplest form: univariate polynomials.

Example 1.2. $\mathsf{S}(x^d) = \Theta(\log d)$. This is done via "repeated squaring." Note that in an arithmetic circuit, the out-degree of a gate could be larger than 1 and there could be parallel edges.

Example 1.3. For every polynomial $f \in \mathbb{F}[x]$ of degree d, we have $\mathsf{S}(f) = O(d)$. For example, we can write $f = 3x^4 + 4x^3 + x^2 + 2x + 5$ as $f = x(x(x(3x + 4) + 1) + 2) + 5$.

Although the two bounds above (the lower bound in Example 1.2 and the upper bound in Example 1.3) hold for every univariate polynomial, there is an exponential gap between them. It turns out that even for univariate polynomials, we do not have strong enough techniques for proving general size lower bounds.

Open Problem 1.4. Find an explicit family of polynomials

$$\{f_i\}_{i \in \mathbb{Z}^+} \subset \mathbb{F}[x], \quad \text{where } f_i \text{ has degree } i,$$

such that $\mathsf{S}(f_n) \neq (\log n)^{O(1)}$.

See Section 4 for some more discussion and clarification of what the word "explicit" means in the open problem above. We also provide a possible candidate for this open problem:

Conjecture 1.5. $\mathsf{S}((x + 1)(x + 2)\cdots(x + n)) \neq (\log n)^{O(1)}$.

This conjecture has a surprising connection to the (Boolean!) complexity of factoring integers.

Exercise 1.6. If *Conjecture 1.5* is false, then `Factoring` can be computed by polynomial size Boolean circuits.

As we go from univariate polynomials to multivariate polynomials, we encounter more algebraic structures, and the flavor of problems also changes. As Example 1.3 shows, every univariate polynomial of degree n can always be computed by an arithmetic circuit of size $O(n)$. In contrast, the smallest arithmetic circuit for an n-variate polynomial of degree n can potentially be exponential in n. However, no such explicit family of polynomials is known at present.

Let us say that a family of n-variate polynomials $\{f_n\}_{n\in\mathbb{Z}^+}$ has *low degree* if the degree of f_n is $n^{O(1)}$. A large part of this survey is devoted to understanding families of low-degree polynomials. We will use partial derivatives as a tool to probe the structure of low-degree polynomials, and to prove lower bounds for them.

Open Problem 1.7. Find an explicit family of low-degree polynomials $\{f_n\}_{n\in\mathbb{Z}^+}$, $f_n \in \mathbb{F}[x_1,\ldots,x_n]$, such that $\mathsf{S}(f_n) \neq n^{O(1)}$.

For multivariate polynomials, it even makes sense to study families of constant-degree polynomials. The challenge is the following:

Open Problem 1.8. Find an explicit family of constant-degree polynomials $\{f_n\}_{n\in\mathbb{Z}^+}$, $f_n \in \mathbb{F}[x_1,\ldots,x_n]$, such that $\mathsf{S}(f_n) \neq O(n)$.

In other words, we want to find an explicit family of constant-degree polynomials for which the arithmetic complexity is superlinear, in the number of variables. Below we give a specific family of cubic (degree-3) polynomials for which resolving the above question is of significant practical importance. Let f_n be the following polynomial in $3n^2$ variables $(x_{ij})_{1\leq i,j\leq n}$, $(y_{ij})_{1\leq i,j\leq n}$, and $(z_{ij})_{1\leq i,j\leq n}$:

$$f_n \stackrel{\text{def}}{=} \sum_{i,j\in[n]\times[n]} z_{ij}\left(\sum_{k\in[n]} x_{ik}\cdot y_{kj}\right).$$

Exercise 1.9. For any $\omega \geq 2$, show that the product of two $n\times n$ matrices can be computed by arithmetic circuits of size $O(n^\omega)$ if and only if $\mathsf{S}(f_n) = O(n^\omega)$.

1.3 Formal Derivatives and Their Properties

1.3.1 Univariate Polynomials

Let $\mathbb{F}$ denote a field, e.g., the set of real numbers $\mathbb{R}$. $\mathbb{F}$ could be finite but we normally assume its characteristic is large enough, e.g., much larger than the degree of any relevant polynomial. Let $\mathbb{F}[x]$ denote the set of univariate polynomials in x over $\mathbb{F}$. Every $f \in \mathbb{F}[x]$ can be expressed as

$$f = a_m x^m + a_{m-1} x^{m-1} + \cdots + a_1 x + a_0,$$

where $m \in \mathbb{Z}_{\geq 0}$ and $a_i \in \mathbb{F}$ for all $0 \leq i \leq m$. The *formal derivative of f with respect to x* is defined as

$$\frac{\partial f}{\partial x} \stackrel{\text{def}}{=} (m a_m) x^{m-1} + \big((m-1) a_{m-1}\big) x^{m-2} + \cdots + 2 a_2 x + a_1.$$

It is called the formal derivative of f because it does not depend on the concept of limit.

1.3.2 Multivariate Polynomials

Let $\mathbb{F}[x_1,\ldots,x_n]$, abbreviated as $\mathbb{F}[\mathbf{X}]$, denote the set of n-variate polynomials over $\mathbb{F}$, then every $f \in \mathbb{F}[\mathbf{X}]$ is a finite sum of monomials with coefficients in $\mathbb{F}$. For example,

$$f = x_1^2 x_2^3 x_3 + 2 x_1^4 x_3^2$$

is a polynomial in $\mathbb{F}[x_1,x_2,x_3]$. Similarly we can define the formal partial derivative of f with respect to x_i. To this end, we write f as

$$f(x_1,\ldots,x_n) = g_m x_i^m + g_{m-1} x_i^{m-1} + \cdots + g_1 x_i + g_0,$$

where $g_i \in \mathbb{F}[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]$ for all $0 \leq i \leq m$. Then

$$\frac{\partial f}{\partial x_i} \stackrel{\text{def}}{=} (m g_m) x_i^{m-1} + \big((m-1) g_{m-1}\big) x_i^{m-2} + \cdots + (2 g_2) x_i + g_1.$$

We use $\partial_{x_i}(f)$ as a shorthand for $\frac{\partial f}{\partial x_i}$. When the name of the variables is clear from the context, we shorten this further to simply $\partial_i(f)$.

Furthermore, we can take higher-order derivatives of f. Let $x_{i_1}, x_{i_2}, \ldots, x_{i_t}$ be a sequence of t variables. Then we can take the tth order derivative of f:

$$\frac{\partial}{\partial x_{i_t}}\left(\cdots\left(\frac{\partial}{\partial x_{i_1}}\left(f\right)\right)\right) \in \mathbb{F}[\mathbf{X}],$$

which we write compactly as $\partial_{i_t}\ldots\partial_{i_1}(f)$. Just like in calculus, it can be shown that the tth order derivatives do not depend on the sequence but only depend on the multiset of variables $\{x_{i_1},\ldots,x_{i_t}\}$.

Let $\mathbf{f} = (f_1,\ldots,f_k)$ be a sequence of k polynomials, where $f_1,\ldots,f_k \in \mathbb{F}[\mathbf{X}]$. We define the *Jacobian matrix* of $\mathbf{f}$ as follows. For $f \in \mathbb{F}[\mathbf{X}]$ we use $\partial(f)$ to denote the n-dimensional vector:

$$\partial(f) \stackrel{\text{def}}{=} \begin{pmatrix} \partial_{x_1}(f) \\ \vdots \\ \partial_{x_n}(f) \end{pmatrix}.$$

Then the Jacobian matrix $\mathbf{J}(\mathbf{f})$ of $\mathbf{f}$ is the following $n \times k$ matrix:

$$\mathbf{J}(\mathbf{f}) \stackrel{\text{def}}{=} \Big(\partial_{x_i}(f_j)\Big)_{i\in[n],\,j\in[k]} = \Big(\partial(f_1)\ \partial(f_2)\ \cdots\ \partial(f_k)\Big).$$

Exercise 1.10. Show that given an arithmetic circuit $\mathcal{C}$ of size s, one can efficiently compute another arithmetic circuit of size $O(s\cdot n)$ with n outputs, the outputs being the polynomials $\partial_{x_i}(\mathcal{C}(\mathbf{X}))$ for $i \in [n]$.

In [6], Baur and Strassen showed that these first-order partial derivatives of $\mathcal{C}(\mathbf{X})$ can actually be computed by an arithmetic circuit of size $O(s)$. We will see a proof in Section 9.

1.3.3 Substitution Maps

Consider now a univariate polynomial

$$f = a_m x^m + a_{m-1}x^{m-1} + \cdots + a_1 x + a_0$$

and its derivative

$$\frac{\partial f}{\partial x} = (m a_m)x^{m-1} + \big((m-1)a_{m-1}\big)x^{m-2} + \cdots + 2a_2 x + a_1.$$

Knowing $\partial_x(f)$ alone is not enough to determine f itself, but observe that knowing $\partial_x(f)$ and the value $f(\alpha)$ of f at *any* point $\alpha \in \mathbb{F}$, we can recover the polynomial f. More generally, for an n-variate polynomial f, we can determine f completely if we know all its first-order partial derivatives and the value $f(\alpha)$ for any point $\alpha \in \mathbb{F}^n$. This means that knowing the partial derivatives of f and a substitution of f is sufficient to determine all the properties of f, including its complexity. In some of the results presented in the survey, we will combine the use of partial derivatives with carefully chosen substitutions in order to enhance our understanding of a given polynomial f.

The substitution that is most natural and occurs frequently is the one where we substitute some of the variables to zero. For a polynomial $f \in \mathbb{F}[\mathbf{X}]$, we denote by $\sigma_i(f)$ the polynomial obtained by setting x_i to zero. For example, for $f = x_1^2 x_2^3 x_3 + 2x_1^4 x_3^2$, we have that $\sigma_1(f) = 0$ and $\sigma_2(f) = 2x_1^4 x_3^2$.

Exercise 1.11. Let $f \in \mathbb{F}[x]$ be a univariate polynomial of degree at most d. Show that f is the identically zero polynomial if and only if $\sigma(\partial^i(f)) = 0$ for all $0 \leq i \leq d$.

1.3.4 Properties

The following properties of derivatives and substitution maps are easy to verify.

Property 1.12. For any $f, g \in \mathbb{F}[\mathbf{X}]$, $\alpha, \beta \in \mathbb{F}$, and $i \in [n]$:

- *Linearity of derivatives*: $\partial_i(\alpha f + \beta g) = \alpha \cdot \partial_i(f) + \beta \cdot \partial_i(g)$.
- *Derivative of product*: $\partial_i(f \cdot g) = \partial_i(f) \cdot g + f \cdot \partial_i(g)$.
- *Linearity of substitution*: $\sigma_i(\alpha f + \beta g) = \alpha \cdot \sigma_i(f) + \beta \cdot \sigma_i(g)$.
- *Substitution preserves multiplication*: $\sigma_i(f \cdot g) = \sigma_i(f) \cdot \sigma_i(g)$.

We also need the counterpart of the *chain rule* in calculus.

Let $g \in \mathbb{F}[z_1, \ldots, z_k] = \mathbb{F}[\mathbf{Z}]$, and $\mathbf{f} = (f_1, \ldots, f_k)$ be a tuple where each f_i is a polynomial in $\mathbb{F}[\mathbf{X}]$. The composition $g \circ \mathbf{f}$ of g and $\mathbf{f}$ is a polynomial in $\mathbb{F}[\mathbf{X}]$ where

$$g \circ \mathbf{f}(\mathbf{X}) = g\big(f_1(\mathbf{X}), f_2(\mathbf{X}), \ldots, f_k(\mathbf{X})\big).$$

Property 1.13 (The Chain Rule). For every $i \in [n]$, we have

$$\partial_{x_i}\big(g \circ \mathbf{f}\big) = \sum_{j=1}^{k} \partial_{f_j}(g) \cdot \partial_{x_i}(f_j),$$

where we use $\partial_{f_j}(g)$ to denote $\partial_{z_j}(g) \circ \mathbf{f} \in \mathbb{F}[\mathbf{X}]$ for all $j \in [k]$.

In the rest of this survey, unless mentioned otherwise, we will assume the underlying field $\mathbb{F}$ to be $\mathbb{C}$, the field of complex numbers. A notable exception is Section 8, where we will work with finite fields. This is all that we need for now. We will introduce some shorthand notation later as needed.

References

[1] S. Aaronson and A. Wigderson, "Algebrization: A new barrier in complexity theory," *ACM Transactions on Computation Theory*, vol. 1, pp. 1–54, 2009.

[2] M. Agrawal, "Proving lower bounds via pseudo-random generators," in *Proceedings of the 25th Conference on Foundations of Software Technology and Theoretical Computer Science*, pp. 92–105, 2005.

[3] M. Agrawal and S. Biswas, "Primality and identity testing via Chinese remaindering," *Journal of the ACM*, vol. 50, 2003.

[4] M. Agrawal and N. Saxena, "Equivalence of F-algebras and cubic forms," in *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science*, pp. 115–126, 2006.

[5] T. Baker, J. Gill, and R. Solovay, "Relativizations of the P =? NP question," *SIAM Journal on Computing*, vol. 4, pp. 431–442, 1975.

[6] W. Baur and V. Strassen, "The complexity of partial derivatives," *Theoretical Computer Science*, vol. 22, pp. 317–330, 1983.

[7] S. J. Berkowitz, "On computing the determinant in small parallel time using a small number of processors," *Information Processing Letters*, vol. 18, pp. 147–150, 1984.

[8] M. Bläser, M. Hardt, R. J. Lipton, and N. K. Vishnoi, "Deterministically testing sparse polynomial identities of unbounded degree," *Information Processing Letters*, vol. 109, pp. 187–192, 2009.

[9] A. Borodin and I. Munro, *The Computational Complexity of Algebraic and Numeric Problems*. American Elsevier, 1st ed., 1975.

[10] P. Bügisser, M. Clausen, and A. Shokrollahi, *Algebraic Complexity Theory*. Springer, 1997.

[11] P. Bürgisser, J. M. Landsberg, L. Manivel, and J. Weyman, "An overview of mathematical issues arising in the geometric complexity theory approach to VP v.s. VNP," in *CoRR*, Vol. abs/0907.2850, 2009.

[12] J.-Y. Cai, "A note on the determinant and permanent problem," *Information and Computation*, vol. 84, pp. 119–127, 1990.

[13] J.-Y. Cai, X. Chen, and D. Li, "A quadratic lower bound for the permanent and determinant problem over any characteristic $\neq 2$," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pp. 491–498, 2008.

[14] E. Carlini, "Reducing the number of variables of a polynomial," in *Algebraic Geometry and Geometric Modelling*, pp. 237–247, Springer, 2006.

[15] Z.-Z. Chen and M.-Y. Kao, "Reducing randomness via irrational numbers," *SIAM Journal of Computing*, vol. 29, pp. 1247–1256, 2000.

[16] A. L. Chistov, "Fast parallel calculation of the rank of matrices over a field of arbitary characteristic," in *Proceedings of the International Conference Foundations of Computation Theory*, pp. 63–69, 1985.

[17] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *Journal of Symbolic Computation*, vol. 9, pp. 251–280, 1990.

[18] L. Csanky, "Fast parallel inversion algorithm," *SIAM Journal on Computing*, vol. 5, pp. 618–623, 1976.

[19] Z. Dvir, "From randomness extraction to rotating needles," *SIGACT News*, vol. 40, pp. 46–61, 2009.

[20] Z. Dvir, "On the size of Kakeya sets in finite fields," *Journal of the American Mathematical Society*, vol. 22, pp. 1093–1097, 2009.

[21] Z. Dvir, A. Gabizon, and A. Wigderson, "Extractors and rank extractors for polynomial sources," *Computational Complexity*, vol. 18, pp. 1–58, 2009.

[22] Z. Dvir and A. Shpilka, "Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pp. 592–601, 2005.

[23] R. Ehrenborg and G.-C. Rota, "Apolarity and canonical forms for homogeneous polynomials," *European Journal of Combinatorics*, vol. 14, pp. 157–181, 1993.

[24] W. J. Ellison, "A 'waring's problem' for homogeneous forms," *Proceedings of the Cambridge Philosophical Society*, vol. 65, pp. 663–672, 1969.

[25] I. Fischer, "Sums of like powers of multivariate linear forms," *Mathematics Magazine*, vol. 67, pp. 59–61, 1994.

[26] S. Gao, "Factoring multivariate polynomials via partial differential equations," *Mathematics of computation*, vol. 72, pp. 801–822, 2003.

[27] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Transactions on Information Theory*, vol. 45, pp. 1757–1767, 1999.

[28] L. Guth and N. H. Katz, *Algebraic Methods in Discrete Analogs of the Kakeya Problem.* arXiv:0812.1043, 2008.

[29] R. Hartshorne, "Algebraic geometry," in *Graduate Texts in Mathematics*, No. 52. Springer-Verlag, New York, 1977.

[30] D. R. Heath-Brown, "An estimate for Heilbronn's exponential sum," in *Analytic Number Theory: Proceedings of a Conference in Honor of Heini Halberstam*, pp. 451–463, 1996.

[31] D. Hilbert, *Theory of Algebraic Invariants.* Cambridge University Press, 1993.

[32] P. Hrubes and A. Yehudayoff, "Arithmetic complexity in algebraic extensions," in *Manuscript*, 2009.

[33] V. Kabanets and R. Impagliazzo, "Derandomizing polynomial identity tests means proving circuit lower bounds," *Computational Complexity*, vol. 13, pp. 1–46, 2004.

[34] E. Kaltofen, "Factorization of polynomials given by straight-line programs," *Randomness and Computation*, vol. 5, pp. 375–412, 1989.

[35] H. Kaplan, M. Sharir, and E. Shustin, "On lines and joints," *Discrete and Computational Geometry*, to appear, 2010.

[36] Z. S. Karnin and A. Shpilka, "Black box polynomial identity testing of depth-3 arithmetic circuits with bounded top fan-in," *Electronic Colloquium on Computational Complexity*, vol. 14, 2007.

[37] N. Kayal, "The complexity of the annihilating polynomial," in *Proceedings of the 24th Annual IEEE Conference on Computational Complexity*, pp. 184–193, 2009.

[38] N. Kayal, "Algorithms for arithmetic circuits," Technical report, Electronic Colloquium on Computational Complexity, 2010.

[39] N. Kayal and S. Saraf, "Blackbox polynomial identity testing for depth 3 circuits," *Electronic Colloquium on Computational Complexity*, vol. 16, 2009.

[40] N. Kayal and N. Saxena, "Polynomial identity testing for depth 3 circuits," in *Proceedings of the Twenty-first Annual IEEE Conference on Computational Complexity*, 2006.

[41] O. Keller, "Ganze cremona-transformationen," *Monatshefte Mathematics and Physics*, vol. 47, pp. 299–306, 1939.

[42] A. Klivans and D. A. Spielman, "Randomness efficient identity testing of multivariate polynomials," in *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pp. 216–223, 2001.

[43] S. Kopparty, S. Saraf, and S. Yekhanin, "High-rate codes with sublinear-time decoding," in *Proceedings of ACM Symposium on Theory of Computing*, 2011.

[44] L. D. Kudryavtsev, "Implicit function," *Encyclopaedia of Mathematics*, 2001.

[45] E. Kushilevitz and N. Nisan, *Communication Complexity.* Cambridge University Press Cambridge, 1997.

[46] D. Lewin and S. P. Vadhan, "Checking polynomial identities over any field: Towards a derandomization?," in *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 438–447, 1998.

[47] S. Lovett, "Computing polynomials with few multiplications," Technical Report 094, Electronic Colloquium on Computational Complexity, 2011.

[48] R. Meshulam, "On two extremal matrix problems," *Linear Algebra and its Applications*, vol. 114, 115, pp. 261–271, 1989.

[49] T. Mignon and N. Ressayre, "A quadratic bound for the determinant and permanent problem," *International Mathematics Research Notices*, pp. 4241–4253, 2004.

[50] D. A. Mit'kin, "Stepanov method of the estimation of the number of roots of some equations," *Matematicheskie Zametki*, vol. 51, pp. 52–58, 1992. (Translated as Mathematical Notes, 51, 565–570, 1992).

[51] J. Morgenstern, "How to compute fast a function and all its derivatives: A variation on the theorem of Baur-Strassen," *SIGACT News*, vol. 16, pp. 60–62, 1985.

[52] K. Mulmuley, "Lower bounds in a parallel model without bit operations," *SIAM Journal on Computing*, vol. 28, pp. 1460–1509, 1999.

[53] K. Mulmuley, "On P versus NP, geometric complexity theory and the Riemann hypothesis," Technical report, The University of Chicago, August 2009.

[54] K. Mulmuley, "On P versus NP, geometric complexity theory, explicit proofs and the complexity barrier," Technical report, The University of Chicago, August 2009.

[55] K. Mulmuley and M. A. Sohoni, "Geometric complexity theory I: An approach to the P vs. NP and related problems," *SIAM Journal on Computing*, vol. 31, pp. 496–526, 2001.

[56] K. Mulmuley and M. A. Sohoni, "Geometric complexity theory II: Towards explicit obstructions for embeddings among class varieties," *SIAM Journal on Computing*, vol. 38, pp. 1175–1206, 2008.

[57] N. Nisan and A. Wigderson, "Lower bounds on arithmetic circuits via partial derivatives," *Computational Complexity*, vol. 6, pp. 217–234, 1997.

[58] P. Olver, *Classical Invariant Theory*. London Mathematical Society, 1999.

[59] J. G. Oxley, "Matroid theory," in *Oxford Graduate Texts in Mathematics*, Oxford University Press, 2006.

[60] G. Pólya, "Aufgabe 424," *Archives of Mathematics and Physics*, vol. 20, p. 271, 1913.

[61] R. Raz, "Multi-linear formulas for permanent and determinant are of super-polynomial size," *Journal of the Association for Computing Machinery*, vol. 56, 2009.

[62] R. Raz and A. Shpilka, "Deterministic polynomial identity testing in non-commutative models," in *IEEE Conference on Computational Complexity*, pp. 215–222, 2004.

[63] R. Raz and A. Yehudayoff, "Lower bounds and separations for constant depth mutilinear circuits," in *Proceedings of the 23rd IEEE Annual Conference on Computational Complexity*, pp. 128–139, 2008.

[64] A. A. Razborov and S. Rudich, "Natural proofs," *Journal of Computer and System Sciences*, vol. 55, pp. 204–213, 1994.

[65] W. M. Ruppert, "Reducibility of polynomials $f(x,y)$ modulo p," *Journal of Number Theory*, vol. 77, pp. 62–70, 1999.

[66] H. J. Ryser, *Combinatorial Mathematics*. Mathematical Association of America, 1963.

[67] N. Saxena, "Diagonal circuit identity testing and lower bounds," in *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, pp. 60–71, 2008.

[68] N. Saxena and C. Seshadhri, "Proceedings of the 24th annual ieee conference on computational complexity," in *IEEE Conference on Computational Complexity*, IEEE Computer Society, 2009.

[69] W. M. Schmidt, *Equations over Finite Fields*. Kendrick Press, 2004.

[70] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *Journal of the Association for Computing Machinery*, vol. 27, pp. 701–717, 1980.

[71] A. Shpilka, "Affine projections of symmetric polynomials," *Journal of Computer and System Sciences*, vol. 65, pp. 639–659, 2002.

[72] A. Shpilka and I. Volkovich, "Read-once polynomial identity testing," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pp. 507–516, 2008.

[73] A. Shpilka and A. Wigderson, "Depth-3 arithmetic circuits over fields of characteristic zero," *Computational Complexity*, vol. 10, pp. 1–27, 2001.

[74] A. Shpilka and A. Yehudayoff, "Arithmetic circuits: A survey of recent results and open questions," *Foundations and Trends in Theoretical Computer Science*, vol. 5, pp. 207–388, 2010.

[75] M. Soltys, "Berkowitz's algorithm and clow sequences," *Electronic Journal of Linear Algebra*, vol. 9, pp. 42–54, 2002.

[76] V. Strassen, "Gaussian elimination is not optimal," *Numerische Mathematik*, vol. 13, pp. 354–356, 1969.

[77] V. Strassen, "Die berechnungskomplexität von elementarsymmetrischen funktionen und von interpolationskoeffizienten," *Numerische Mathematik*, vol. 20, pp. 238–251, 1973.

[78] V. Strassen, "Vermeidung von divisionen," *Journal of Reine Angewanatte Mathematics*, vol. 264, pp. 184–202, 1973.

[79] M. Sudan, "Decoding of Reed Solomon codes beyond the error-correction bound," *Journal of Complex*, vol. 13, pp. 180–193, 1997.

[80] G. Szegö, "Zu aufgabe 424," *Archives of Mathematics and Physics*, vol. 21, pp. 291–292, 1913.

[81] L. G. Valiant, "Completeness classes in algebra," in *Proceedings of the 11th Annual ACM Symposium on Theory of Computing*, pp. 249–261, 1979.

[82] L. G. Valiant, "The complexity of computing the permanent," *Theoretical Computer Science*, vol. 8, pp. 189–201, 1979.

[83] L. G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff, "Fast parallel computation of polynomials using few processors," *SIAM Journal on Computing*, vol. 12, pp. 641–644, 1983.

[84] B. L. van der Waerden, *Moderne Algebra*. Berlin, Springer, 2nd ed., 1937.

[85] J. von zur Gathen, "Permanent and determinant," *Linear Algebra and its Applications*, vol. 96, pp. 87–100, 1987.

[86] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*. Cambridge University Press, 1999.

[87] T. D. Wooley, "A note on simultaneous congruences," *Journal of Number Theory*, vol. 58, pp. 288–297, 1996.

[88] D. Wright, "On the Jacobian conjecture," *Illinois Journal of Mathematics*, vol. 15, pp. 423–440, 1981.

[89] J. T. Yu, "On the Jacobian conjecture: Reduction of coefficients," *Journal of Algebra*, vol. 171, pp. 515–523, 1995.

[90] R. Zippel, "Interpolating polynomials from their values," *Journal of Symbolic Computation*, vol. 9, pp. 375–403, 1990.