Factorization of Polynomials in the perspective of Algebraic Circuits

Soham Chatterjee

Chennai Mathematical Institute

sohamchatterjee999@gmail.com

July 22, 2023

Presentation Overview

Introduction

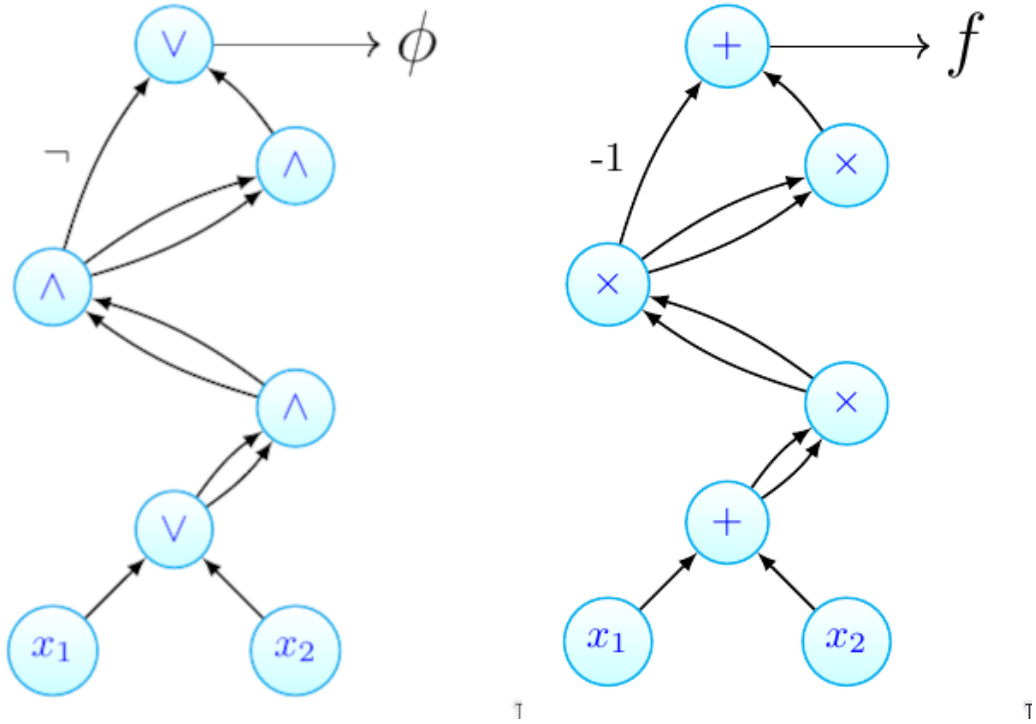
- We mainly study Turing Machines and Different Classes created by considering certain parameters of Turing machines like Time, Space, Randomness, Communication, Nondeterminism etc in Complexity Theory.
- Another model which is often our interest is Circuits. They are basically a kind of DAG (Directed Acyclic Graph)
- First interest of Computer Scientists was Boolean Circuits. Here the nodes are $\wedge$ (AND), $\vee$ (OR) gates. The edges have either $\neg$ (NOT) gate or nothing and the leaves have the variables of the boolean formula.
- To study them better later Computer Scientists Started Algebraic Circuits where instead of $\wedge, \vee$ we have $+, \times$ and the edges have the coefficients of from the ring we are working on and leaves have the variables

Introduction

- To evaluate the circuit we start from the bottom at go up to the root and the root represents the final Boolean formula or circuit and polynomial
- Now the whole reason to study algebraic circuits is not to limit ourselves to $\mathbb{F}_2$ but any field.
- In general, multivariate polynomial factoring has several applications including decoding of Reed-Solomon, Reed-Muller codes [Gur98], [Sud97], integer factoring [Len+90],primary decomposition of polynomial ideals [GTZ88] and algebra isomorphism [KS06],[Iva+08].

In this lecture, we will talk about fields with characteristic 0

Introduction



The left one is a Boolean circuit and the right one is an algebraic circuit or arithmetic circuit.

Note: Some papers also have written algebraic circuits as Straight Line Program (Just a regular day of Kaltofen and Valiant Papers)

Definitions

- **Size:** Size of the whole DAG graph is the size of the circuit
- **Fan-in, Fan-out:** Fan-in $=$ Max in degree, Fan-out $=$ Max out-degree
- **ABP (Arithmetic Branching Program):** These are a special type of circuits. An algebraic branching program (ABP) is a layered graph with a unique source vertex (say s) and a unique sink vertex (say t). All edges are from layer i to $i+1$ and each edge is labeled by a linear polynomial. The polynomial computed by the ABP is defined as

$$f = \sum_{\gamma : s \rightsquigarrow t} wt(\gamma)$$

, where for every path γ from s to t, the weight wt (γ) is defined as the product of the labels over the edges forming γ.
Width is the maximum number of vertices in a layer.

- **Formula:** Formula is an algebraic circuit with fan-out 1. (So you can not store any computation)

Algebraic Circuit Classes

VNP, VP, VBP, VF

From now on n denotes the number of variables

- **VP:** The class VP contains all the algebraic circuits of the polynomials which has degree $poly(n)$ and size $poly(n)$. P analog of algebraic circuits
- **VBP:** The class VBP contains all the ABP's which has degree $poly(n)$ and size $poly(n)$
- **VF:** The class VF contains all the formulas of the polynomials which has degree $poly(n)$ and size $poly(n)$
- **VNP:** A family of polynomials $\{f_n\}_n$ over $\mathbb{F}$ is in VNP if there exist polynomials $t(n), s(n)$ and a family $\{g_n\}_n$ in VP such that for every n,

$$f_n(\bar{x}) = \sum_{w \in \{0,1\}^{t(n)}} g_n\left(\bar{x}, w_1, \ldots, w_{t(n)}\right)$$

Here, witness size is $t(n)$ and versifier circuit g_n has size $s(n)$. Its basically NP analogue of algebraic circuits so VP with non-determinism

Some results on these classes

- $VF \subseteq VBP \subseteq VP \subseteq VNP$
- Any circuit of size s and depth d can be converted to a formula of size s^d
- VP is contained in VNP and it is believed that this containment is strict (Valiant's Hypothesis [Val79]).
- $VP = VNP \implies \#P = FP \implies P = NP$
- DET is $VBP - complete$ [MV97]
- $PERM$ is $VNP - complete$ [Bür98]
- There are other structural results that are well written in [Sap21] and [SY10]

- Since we work on polynomials and power series we assume the function is differentiable

Theorem (Newton Iteration [BCS97, Theorem 2.31], [GG03],[Gat84])

If f is a function on any number of variables, let $(\overline{x}, y)$, has a root at g wrt y i.e. $f(\overline{x}, g(\overline{x})) = 0$ then

$$y_{i+1} = y_i - \left.\frac{f}{\partial_y f}\right|_{y=y_i}$$

where $y_i \equiv g \bmod \langle \overline{x} \rangle^i$ so that

$$y_{i+1} \equiv g \bmod \langle \overline{x} \rangle^{i+1}$$

- This extra variable y comes because we actually shift the variables randomly by adding another variable so that the polynomial does not become zero $\overline{x} \mapsto \overline{x} + \overline{a}y + \overline{b}$ by Schwarz - Zippel Lemma [Sch80]

Lifting [ST21, Section 4.1]]: Let $\mathcal{R}$ be a ring and $\mathcal{I} \subseteq \mathcal{R}$ be an ideal. Let $f, g, h, a, b \in R$ such that $f \equiv gh (\mathrm{mod} \mathcal{I})$ and $ag + bh \equiv 1 (\mathrm{mod} \mathcal{I})$. Then we call $g', h' \in \mathcal{R}$ a lift of g, h, if

1. $f \equiv g'h' \left(\mathrm{mod} \mathcal{I}^2\right)$
2. $g' \equiv g (\mathrm{mod} \mathcal{I})$ and $h' \equiv h (\mathrm{mod} \mathcal{I})$, and
3. $\exists\ a', b' \in \mathcal{R} \quad a'g' + b'h' \equiv 1 \left(\mathrm{mod} \mathcal{I}^2\right)$.

Theorem (Hensel Lifting [ST21, Section 4.1],[GG03],[Gat84])

Let $\mathcal{R}$ be a ring and $\mathcal{I} \subseteq \mathcal{R}$ be an ideal. Let $f, g, h, a, b \in R$ such that $f \equiv h \pmod{\mathcal{I}}$ and $ag + bh \equiv 1 (\mathrm{mod} \mathcal{I})$. Then we have there exists a lift g', h' of g, h and for any other lift g^*, h^* of g, h, $\exists\ u \in \mathcal{I}$ such that

$$g* \equiv g'(1 + u) \pmod{\mathcal{I}^2} \text{ and } h* \equiv h'(1 - u) \pmod{\mathcal{I}^2}$$

Uni-variate Polynomial Factorization I

Field with finite characteristic (prime)

- We are in the field $\mathbb{F}_q$, $q = p^n$ $char(\mathbb{F}_q) = p$. Input f with $\deg f = d$

Preprocess:

- **Square-free** Take $gcd(f, \partial f)$ to get a square free factor. Call this f
- **Distinct-degree** From 1 to $\frac{d}{2}$ we take the $gcd(f, x^{q^i} - x)$ to output the product of all $i-$degree factors of f. $f = \prod_{i \in [k]} f_i$ each f_i irreducible and $\deg f_i = \frac{d}{k}$

Theorem ([GG03, Theorem 14.2])

For any $d \geqslant 1$, $x^{q^d} - x \in \mathbb{F}_q[x]$ is the product of all monic irreducible polynomials in $\mathbb{F}_q[x]$ whose degree divides d

- This also gives an algorithm to check if the polynomial is reducible.
- If f, g are co-prime polynomials in any field $\mathbb{F}$ then $\mathbb{F}[x]/\langle fg \rangle \cong \mathbb{F}[x]/\langle f \rangle \times \mathbb{F}[x]/\langle g \rangle$

Uni-variate Polynomial Factorization II

Field with finite characteristic (prime)

Barlekamp Algorithm:([Ber70],[GG03, Section 14.8],[Knu97, Section 4.6.2],[Sap17])

- Factoring f is factoring $\mathcal{A} := \overset{k}{\underset{i=1}{\times}} \underbrace{\mathbb{F}_q[x]/\langle f_i \rangle}_{=\mathbb{F}_{q'}}$
- Want to find the Isomorphism. We know the RHS
- Any $g \in \mathcal{A}$ is a $k-$tuple $(a_1, a_2, \ldots, a_k)$, $a_i \in \mathbb{F}_q[x]/\langle f_i \rangle \implies g \equiv a_i \pmod{f_i}$. If $a_i \in \mathbb{F}_p$ then $g^p \equiv g$ in $\mathcal{A}$
- Such g gives a factor of f as
  $g^p - g \equiv 0 \bmod f \implies \prod_{\alpha \in \mathbb{F}_p} (g - \alpha) \equiv 0 \bmod f \implies \prod_{i \in [k]} f_i \Big| \prod_{\alpha \in \mathbb{F}_p} (g - \alpha)$
- $\{g \in \mathbb{F}_q[x] \mid g^p \equiv g \pmod{f}\}$ is a vector space over $\mathbb{F}_p$. So we do some system of linear equation solving
- **Time Complexity:** $\tilde{O}(p(dn)^\omega)$ where ω the Matrix Multiplication Exponent

Uni-variate Polynomial Factorization III

Field with finite characteristic (prime)

Barlekamp Algorithm is the oldest polynomial factorization algorithm. Later there are much improvements in the complexity of factorization with the addition of randomization.

Theorem

$\alpha \in \mathbb{F}_p*$ *is a square or quadratic residue* $\iff \alpha^{\frac{p-1}{2}} \equiv 1 \pmod{p}$

- $Pr_{\alpha \in_R \mathbb{F}_p*}[\alpha \text{is a quadratic residue}] = \frac{1}{2}$

Cantor-Zassenhaus Algorithm([GG03, Section 14.3],[Sap17],[Zas69])

- WLOG we assume $p > 2$
- $a \in_R \mathbb{F}_p$. $g = f(x-a)$, the roots if g have different quadratic residuocity.
- Pick $\alpha \in_R \mathbb{F}_p$. If $\alpha + a$ is a zero of $f(x-a)$ and it is a square then it also a root of $gcd(f(x-a), x^{\frac{p-1}{2}} - 1)$
- So we output $gcd(f(x), (x+a)^{\frac{p-1}{2}} - 1)$
- **Time Complexity:** $\tilde{O}(d^\omega \log q)$

Uni-variate Polynomial Factorization I

Field with characteristic 0

- We will talk about polynomial factorization in $\mathbb{Q}[x]$ or $\mathbb{Z}[x]$
- Coefficients of f are less than 2^{l-1}
- We will talk about the algorithm discovered by Lenstra, Lenstra, Lovasz in 1982 [LLL82] using shortest vectors.
- $\|f\|_\infty = A$

Theorem (Mignotte's Bound [Mig74])

Any root $\alpha \in \mathbb{C}$ *of a polynomial* $f(x) = \sum_{i=0}^{n} a_i x^i \in \mathbb{Z}[x]$ *satisfies* $\alpha \leqslant n \max\{|a_i\}$

Theorem

Any factor g *of* f *has coefficient of magnitude at most* $2^{(l+\log n-1)n}$

Uni-variate Polynomial Factorization II

Field with characteristic 0

Theorem ([GG03, Lemma 16.20])

Let $f, g \in \mathbb{Z}[x]$ have positive degrees n, k respectively and suppose that $u \in \mathbb{Z}[x]$ is non-constant, monic, and divides both f, g modulo m for some $m \in \mathbb{N}$ with $\|f\|^k \|g\|^n < m$. Then $gcd(f, g) \in \mathbb{Z}[x]$

L^3 Algorithm([LLL82],[GG03, Chapter 16])

- Assume that f is square free. Find the smallest prime p such that $p \nmid a_n$ and $f \pmod p$ is square free
- Using Barlekamp algorithm compute a factorization of $f \equiv f_0 g_0 \pmod p$ where $g_0 \pmod p$ is monic irreducible and co-prime to h_0
- Compute $f \equiv g_k h_k \pmod{p^{2^k}}$ by Hensel Lifting for $k = \lceil \log 2n^3 l \rceil$
- Find $(\tilde{g}, l_k)$ such that $\tilde{g} \equiv g_k l_k \pmod{p^{2^k}}$ with $\deg \tilde{g} < n$. Coefficients of $\tilde{g}$ have bit-size $\leqslant n(l + \log n)$
- Output $gcd(f, \tilde{g})$
- **Time Complexity:** $\tilde{O}(n^{10} + n^8 \log^2 A)$

Uni-variate Polynomial Factorization III

Field with characteristic 0

In 4-th Step this requires a small root of a linear system. Which involves short vector in a lattice system. But Shortest Vector Problem is NP-Hard [Ajt98], even constant approximation of it is also $NP - hard$ [Mic01]. But we need 2^n-approximation so it is doable.

Multi-variate Polynomial Factorization

- Since we can do uni-variate factorization quite easily and in polynomial time in most of the multivariate factorization we first some how make the polynomial uni-variate then factorize there and bring it back to multivariate with some sort of lifting.
- Most of the algorithms mainly first apply a random shift to the variables first.
- In case of multivariate factoring we see if we can say anything about the closure of the algebraic complexity classes.
- There are surveys for polynomial factorization [Kal06],[Kal20],[FS15] to get much more depth

Low Degree Polynomial factorization

- By low degree we mean individual degrees are bounded by some constant

Theorem ([Oli15])

Let $P \in \mathbb{F}[\overline{x}] \backslash \{0\}$ be such that $\deg_{x_i} P \leqslant r$ for $i \in [n]$ and f is a factor of P where $char(\mathbb{F}) = 0$. If there exists a formula (circuit) of size s and depth Δ computing P then there exists a formula (circuit) of depth $\Delta + 5$ and size $poly((nr)^r, s)$ that computes f

- It calculates from 0 to d calculates the Homogeneous part of P
- Then it uses Newton's Iteration to calculate the next degree Homogeneous Part

VP closed under factorization

Theorem ([CKS19],[Kal89])

Let $f \in \mathbb{F}[x_1, x_2, \ldots, x_{n-1}, y]$ be an n-variate degree d polynomial which can be computed by an arithmetic circuit of size at most s. If g and h are polynomials of degree at least 1 such that $f = gh$ and $gcd(g, h) = 1$, then g and h have a circuit of size at most $poly(s, n, d)$.

Theorem ([CKS19],[Kal87])

Let $f \in \mathbb{F}[x_1, x_2, \ldots, x_{n-1}, y]$ be an n-variate degree d polynomial which can be computed by an arithmetic circuit of size at most s. If there is a polynomial g and an integer e such that $f = g^e$, then g has a circuit of size at most $poly(s, n, d)$.

VBP closed under factorization

Theorem ([ST21])

Let p be a polynomial over a field F with characteristic 0. For all factors q of p, we have

$$size_{ABP}(q) \leqslant poly(size_{ABP}(p))$$

- We preprocess by random shift and introduce new variable z to track total degree for $\overline{x}$ and bring multiplicity of one irreducible factor to 1 to get $\hat{p}(\overline{x}, y, z)$
- Factor $\hat{p}(\overline{x}, y, 0)$ to have $\hat{q}(\overline{x}, y)$ then Hensel Lift
- At the end we do some solving linear equations to get the exact q

More Factorization Results I

- These are true for exponential degree circuits also.

Theorem ([DSS17, Theorm 1])

If $f = u_0 u_1$ is a nonzero product in the polynomial ring $\mathbb{F}[\overline{X}]$ with $size(f) + size(u_0) \leqslant s$ then every factor of u_1 has a circuit of size $poly(s + \deg(sqfree(u_1)))$

Theorem ([DSS17, Theorem 3])

The classes $VF\left(n^{\log n}\right), VBP\left(n^{\log n}\right), VNP\left(n^{\log n}\right)$ are all closed under factoring.
Moreover, there exists a randomized poly $\left(n^{\log n}\right)$-time algorithm that: for a given $n^{O(\log n)}$ sized formula (resp. ABP) f of poly(n)-degree, outputs $n^{O(\log n)}$ sized formula (resp. ABP) of a nontrivial factor of f (if one exists).

More Factorization Results II

- In both theorems we random shift and get $\hat{f}(\overline{x}, y)$ and think as if we are in the ring $\mathbb{F}[[\overline{x}]][y]$
- $\hat{f}$ splits in this ring and then we approximate till $\deg \hat{f}$
- Now we factor $\hat{f}(\overline{0}, y) = \hat{f} \pmod{\langle \overline{x} \rangle}$. Then Newton's Iteration to approximate
- In the end we again do some solving of the system of linear equations to get the final factor of f
- Then we multiply some factors to get a factor of u_1 where we use a trick of Kaltofen

Theorem ([DSS17, Theorem 14])

The approximative classes $\overline{VF}\left(n^{\log n}\right), \overline{VBP}\left(n^{\log n}\right), \overline{VNP}\left(n^{\log n}\right)$ are all closed under factoring.

Black-Box Multi-variate Polynomial Factorization I

Theorem (Hilbert Irreducibility Theorem [Kal85],[Sap17],[KT90])

Let $f(x, y_1, \ldots, y_n)$ *be a degree* d *polynomial which is monic in* x*. Suppose* $S \subseteq \mathbb{F}$ *and* $Pr_{\bar{a},\bar{b}\in S^n}[f(x, a_1t + b_1, \ldots, a_nt + b_n)$ *is reducible* $] \geqslant \frac{O(d^5)}{|S|}$ *then* $f(x, y_1, \ldots, y_n)$ *is reducible.*

- Hilbert's irreducibility theorem, which in one formulation says that restricting an irreducible polynomial to a random two-dimensional subspace keeps the polynomial irreducible with high probability.
- They restrict a polynomial f to a random two-dimensional subspace does not change the number of irreducible factors of f .
- Given this, multivariate factoring algorithms proceed as follows. First, restrict the polynomial to a randomly chosen two-dimensional space.
- Then, perform bi-variate factorization of the restricted polynomial. Finally, lift each factor to the whole space.

Black-Box Multi-variate Polynomial Factorization II

Theorem ([KT90])

The Black Box Polynomial Factorization algorithm can output a factor in polynomially many arithmetic steps as a function of n *and* $\deg(f) = d$ *with the right answer probability at least* $\frac{O(d^6)}{|S|}$

Open Questions

1. Randomized uni-variate algorithm in $\tilde{O}(d \log^2 q)$
2. VF or $\overline{VF}$ is closed under factors
3. Factor closure is also not known for other models like ROABP (Read-once Oblivious ABP), Constant depth circuits
4. We also don't know any algorithm to find the poly size ABP factor. We have existential proof in [ST21]
5. If we have the circuit of $f = g^p$ with size s where g is irreducible in $\mathbb{F}_p[\overline{x}]$ there is a circuit of g in $poly(s, n)$
6. If we have a circuit of f with size s and $\deg f = d$, $f = gh$ where $\deg g = \delta$ then there is a randomized algorithm to find a circuit of g with size $poly(s, n, \delta)$ [Factor Conjecture]

References I

[Ajt98] Miklós Ajtai. "The Shortest Vector Problem in L2 is NP-Hard for Randomized Reductions (Extended Abstract)". In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. STOC '98. Dallas, Texas, USA: Association for Computing Machinery, 1998, pp. 10–19. ISBN: 0897919629. DOI: 10.1145/276698.276705. URL: https://doi.org/10.1145/276698.276705.

[BCS97] Peter Bürgisser, Michael Clausen, and Amin Shokrollahi. *Algebraic Complexity Theory*. Vol. 315. Jan. 1997. ISBN: 978-3-642-08228-3. DOI: 10.1007/978-3-662-03338-8.

[Ber70] E. R. Berlekamp. "Factoring Polynomials Over Large Finite Fields". In: *Mathematics of Computation* 24.111 (1970), pp. 713–735. ISSN: 00255718, 10886842. URL: http://www.jstor.org/stable/2004849 (visited on 06/09/2023).

[Bür98] Peter Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*. Vol. 7. Aug. 1998. DOI: 10.1007/978-3-662-04179-6_1.

References II

[CKS19] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. *Closure of VP under taking factors: a short and simple proof*. 2019. arXiv: 1903.02366 [cs.CC].

[DSS17] Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. *Discovering the roots: Uniform closure results for algebraic classes under factoring*. 2017. arXiv: 1710.03214 [cs.CC].

[FS15] Michael A. Forbes and Amir Shpilka. "Complexity Theory Column 88: Challenges in Polynomial Factorization1". In: *SIGACT News* 46.4 (2015), pp. 32–49. DOI: 10.1145/2852040.2852051. URL: https://doi.org/10.1145/2852040.2852051.

[Gat84] Joachim von zur Gathen. "Hensel and Newton methods in valuation rings". In: *Mathematics of Computation* 42 (1984), pp. 637–661.

[GG03] Joachim Gathen and Jürgen Gerhard. *Modern computer algebra (2. ed.)*. Jan. 2003. ISBN: 978-0-521-82646-4.

References III

[GTZ88] Patrizia Gianni, Barry Trager, and Gail Zacharias. "GröBner Bases and Primary Decomposition of Polynomial Ideals". In: *J. Symb. Comput.* 6.2–3 (Dec. 1988), pp. 149–167. ISSN: 0747-7171. DOI: 10.1016/S0747-7171(88)80040-3. URL: https://doi.org/10.1016/S0747-7171(88)80040-3.

[Gur98] Venkatesan Guruswami. "Improved decoding of Reed-Solomon and algebraic-geometric codes". In: vol. 45. Dec. 1998, pp. 28–37. ISBN: 0-8186-9172-7. DOI: 10.1109/SFCS.1998.743426.

[Iva+08] Gábor Ivanyos et al. "Trading GRH for algebra: algorithms for factoring polynomials and related structures". In: *CoRR* abs/0811.3165 (2008). arXiv: 0811.3165. URL: http://arxiv.org/abs/0811.3165.

[Kal06] Erich Kaltofen. "Polynomial factorization 1987–1991". In: Apr. 2006, pp. 294–313. ISBN: 3-540-55284-7. DOI: 10.1007/BFb0023837.

[Kal20] Erich L. Kaltofen. "Polynomial Factorization 1982–1986". In: *Computers in Mathematics* (2020).

References IV

[Kal85] Erich Kaltofen. "Effective Hilbert irreducibility". In: *Information and Control* 66.3 (1985), pp. 123–137. ISSN: 0019-9958. DOI: https://doi.org/10.1016/S0019-9958(85)80056-5. URL: https://www.sciencedirect.com/science/article/pii/S0019995885800565.

[Kal87] E. Kaltofen. "Single-Factor Hensel Lifting and Its Application to the Straight-Line Complexity of Certain Polynomials". In: *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*. STOC '87. New York, New York, USA: Association for Computing Machinery, 1987, pp. 443–452. ISBN: 0897912217. DOI: 10.1145/28395.28443. URL: https://doi.org/10.1145/28395.28443.

[Kal89] Erich L. Kaltofen. "Factorization of Polynomials Given by Straight-Line Programs". In: *Adv. Comput. Res.* 5 (1989), pp. 375–412.

[Knu97] Donald E. Knuth. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. USA: Addison-Wesley Longman Publishing Co., Inc., 1997. ISBN: 0201896842.

References V

[KS06] Neeraj Kayal and Nitin Saxena. "Complexity of Ring Morphism Problems". In: *computational complexity* 15 (2006), pp. 342–390.

[KT90] Erich Kaltofen and Barry M. Trager. "Computing with polynomials given byblack boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators". In: *Journal of Symbolic Computation* 9.3 (1990). Computational algebraic complexity editorial, pp. 301–320. ISSN: 0747-7171. DOI: https://doi.org/10.1016/S0747-7171(08)80015-6. URL: https://www.sciencedirect.com/science/article/pii/S0747717108800156.

[Len+90] Arjen Lenstra et al. "The Number Field Sieve". In: Jan. 1990, pp. 564–572. DOI: 10.1145/100216.100295.

[LLL82] Arjen Lenstra, H. Lenstra, and Lovász László. "Factoring Polynomials with Rational Coefficients". In: *Mathematische Annalen* 261 (Dec. 1982). DOI: 10.1007/BF01457454.

References VI

[Mic01] D. Micciancio. "The shortest vector problem is NP-hard to approximate to within some constant". In: *Siam Journal on Computing - SIAMCOMP* 30 (Jan. 2001). DOI: 10.1137/S009753970037303 9.

[Mig74] Maurice Mignotte. "An inequality about factors of polynomials". In: *Mathematics of Computation* 28 (1974), pp. 1153–1157.

[MV97] Meena Mahajan and V. Vinay. "A combinatorial algorithm for the determinant". In: *ACM-SIAM Symposium on Discrete Algorithms*. 1997.

[Oli15] Rafael Oliveira. "Factors of Low Individual Degree Polynomials". In: *Proceedings of the 30th Conference on Computational Complexity*. CCC '15. Portland, Oregon: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015, pp. 198–216. ISBN: 9783939897811.

[Sap17] Ramprasad Saptharishi. "Algebra and Computation Lecture Notes, 2017". In: (2017). URL: https://www.tifr.res.in/~ramprasad.saptharishi/assets/courses/2017-algComp/algComp_2017.pdf.

References VII

[Sap21] Ramprasad Saptharishi. "A survey of lower bounds in arithmetic circuit complexity". In: (2021). URL: https://github.com/dasarpmar/lowerbounds-survey.

[Sch80] J. T. Schwartz. "Fast Probabilistic Algorithms for Verification of Polynomial Identities". In: *J. ACM* 27.4 (1980), pp. 701–717. ISSN: 0004-5411. DOI: 10.1145/322217.322225. URL: https://doi.org/10.1145/322217.322225.

[ST21] Amit Sinhababu and Thomas Thierauf. "Factorization of Polynomials Given by Arithmetic Branching Programs". In: *computational complexity* 30 (Dec. 2021). DOI: 10.1007/s00037-021-00215-0.

[Sud97] Madhu Sudan. "Decoding of Reed Solomon Codes beyond the Error-Correction Bound". In: *J. Complex.* 13.1 (Mar. 1997), pp. 180–193. ISSN: 0885-064X. DOI: 10.1006/jcom.1997.0439. URL: https://doi.org/10.1006/jcom.1997.0439.

References VIII

[SY10] Amir Shpilka and Amir Yehudayoff. "Arithmetic Circuits: A survey of recent results and open questions". In: *Foundations and Trends in Theoretical Computer Science* 5 (Jan. 2010), pp. 207–388. DOI: 10.1561/0400000039.

[Val79] Leslie G. Valiant. "Completeness classes in algebra". In: *Proceedings of the eleventh annual ACM symposium on Theory of computing* (1979).

[Zas69] Hans Zassenhaus. "On Hensel factorization, I". In: *Journal of Number Theory* 1 (1969), pp. 291–311.

The End

Questions? Comments? Suggestions?