COMS E6998: Proof Complexity and Applications (Spring25) April 9, 2025

Lecture 7: Algebraic Proof Systems

Instructor: *Toniann Pitassi* Scribes: *Hao Cui, Yongyi Wang*

1 Review of Goals

We are mostly interested to develop a super-polynomial lower bounds for $\mathrm{AC}^0[p]$-Frege systems. There are a few remarks on this problem we are trying to tackle:

1. Even though a super-polynomial lower bound for $\mathrm{AC}^0[p]$ have been known for decades, we do not even have conditional results for the Frege system.
2. No apparent lifting results between proof complexity lower bounds and circuit lower bounds have been known.
3. Beigel-Tarui/Yao/Allender-Hertrampf Circuit Normal Form Theorems hold:

   Theorem 1 ([**?**])**.** *Any $AC^0[p]$ Frege Proof π of quasi-polynomial size can be converted into a depth 4 quasi-polynomial size $AC^0[p]$ Frerge Proof, where we have ORs in the first layer, ANDS in the second layer, $\oplus_p$ in the third layer, and small-ANDS at the leaves.*
4. Methods of probabilistic polynomials used to prove superpolynomial lower bounds for $AC^0[p]$ circuits don't seem to work.

These observations directly motivates the study of proofs where lines are low depth $\mathrm{AC}^0[p]$, such those whose lines are $\oplus_p \circ \mathrm{AND}$, or polynomials mod p.

Definition 2. *A polynomial $p(x_1, x_2, \cdots, x_n)$ over $\mathbb{F}$ is of the form*

$$\sum_{(i_1,\cdots,i_n)\in I} a_{(i_1,\cdots,i_n)} x_1^{i_1} \cdots x_n^{i_n}$$

where I is a set of n-tuples of non-negative integers and $a_{(i_1,\cdots,i_n)} \in \mathbb{F}$ for all such tuples. A system of polynomial equations over $\mathbb{F}$ is a collection P of $\mathbb{F}$-polynomials p

Basically, an algebraic proof system certify the unsolvability of P. In other words, P is unsolvable if there does not exist an assignment $\vec{x}$ such that $p(\vec{x}) = 0$ for all $p \in P$.

2 Nullstallensatz Proof System

Theorem 3 (Hilbert's Nullstellensatz, weak form)**.** *$P = \{p_1, \cdots, p_n\}$ is unsolvable over algebraically closed $\mathbb{F}$ iff there exists polynomials $q_1, \cdots, q_n$ such that*

$$\sum_{i=1}^{n} q_i p_i = 1$$

By the above theorem, we can view the polynomials $q_1, \ldots, q_n$ such that $\sum_{i=1}^{n} q_i p_i$ as a *proof* of unsolvability of the system P of polynomially equations over $\mathbb{F}$.

Next we want to apply the above Nullstellensatz theorem to the special case wher the polynomials P correspond to the clauses of an unsatisfiable Boolean CNF formula.

Definition 4 (Translation of Clauses to Polynomials)**.** *Let $\mathcal{C} = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ be a CNF formula over Boolean variables $x_1, \ldots, x_n$. We define the following system of polynomial equations that correspond to $\mathcal{C}$ as follows.*

1. *For each clause C_i, we convert it to a polynomial equation as follows. As an example, suppose that $C_i = (x_1 \vee x_2 \vee \neg x_3)$. Then the corresponding polynomial is: $p_i = (1-x_1)(1-x_2)x_3 = 0$. Note that for any Boolean assignment α to the underlying variables, $C_i(\alpha) = 1$ if and only if $p_i(\alpha) = 0$; thus the assignment satisfies clause C_i if and only if the assignment is a zero of p_i.*

2. *Additionally we add n extra equations, one for each underlying variable x_j in order to force Boolean solutions: for each x_j, we add the add the polynomial equation $p_{x_j} = x_j^2 - x_j = 0$.*

Let P be the resulting system $m + n$ polynomial equations.

Definition 5 (Nullstallensatz Refutation for CNF)**.** *Let $\mathcal{C} = C_1 \wedge \ldots \wedge C_m$ be a Boolean CNF formula, and let P be the corresponding system of polynomial equations as defined above. A Nullstallensatz (NS) refutation of P (and thus a Nullstellensatz refutation of $\mathcal{C}$) over a field $\mathbb{F}$ is a collection $Q = \{q_1, \cdots, q_{m+n}\}$ of polynomials such that $\sum_{i=1}^{m+n} q_i p_i = 1$. The degree of the refutation is $\max_i \deg(q_i)$, and the monomial size of the refutation is $\sum_i |q_i|$, where $|q_i|$ is the number of nonzero monomials in q_i. We define the Nullstellensatz refutation degree of $\mathcal{C}$, $NS(\mathcal{C})$, to be the min degree of any NS refutation of $\mathcal{C}$.*

Note that the classical weak Nullstellensatz requires that we work in an algebraically closed field. However, in our case we always add the polynomials $x_i^2 - x_i = 0$ which forces all variables to be $\{0, 1\}$ valued in any solution; for this reason, it suffices to work in any field (rather than in the algebraic closure of a field).

Example 6. *Consider the negation of induction $\neg IND_n$, which consists of the following clauses:*

1. *(x_1);*

2. *$(\neg x_n)$;*

3. *and for all $1 \leq i \leq n-1$: $\neg x_i \vee x_{i+1}$).*

By typical induction this is clearly unsatisfiable, and thus we wish to find an NS refutation.

1. *First we convert each clause into a polynomial. Clearly, (x_1) and $(\neg x_n)$ will be converted to $1 - x_1 = 0$ and $x_n = 0$ respectively, and $(\neg x_i \vee x_{i+1})$ will be converted to $x_i(1 - x_{i+1}) = 0$.*

2. *Consider $(\neg x_1 \vee x_2)$ and $(\neg x_2 \vee x_3)$. These two clauses imply $(\neg x_1 \vee x_3)$. Thus, we can think of how to derive the polynomial $x_1(1 - x_3)$ from $x_1(1 - x_2)$ and $x_2(1 - x_3)$. Indeed, we see that*

$$(1 - x_3) \cdot x_1(1 - x_2) + x_1 \cdot x_2(1 - x_3) = x_1(1 - x_3)$$

3. *For any indices i, we can derive the clause $(\neg x_1 \vee x_{i+1})$ from the clauses $(\neg x_1 \vee x_i)$ and $(\neg x_i \vee x_{i+1})$:*

$$(1 - x_{i+1}) \cdot x_1(1 - x_i) + x_1 \cdot x_i(1 - x_{i+1}) = x_1(1 - x_{i+1})$$

4. *Applying the previous step for $i = 2$ to $n - 1$ we eventually derive the polynomial: $x_1(1 - x_n) = 0$. Then using this derived polynomial together with the initial polynomials $1 - x_1 = 0$ and $x_n = 0$, we can derive:*

$$1 - x_1 + x_1(1 - x_n) + x_1 \cdot x_n = 1$$

Hence we have a NS refutation of $\neg IND_n$.

Note that our refutation has degree $\Theta(n)$ since step 3 needs to be repeated $\Theta(n)$ times to get to $x_1 \implies x_n$, and each step we are increasing the degree of the largest polynomial in Q by 1 after multiplying $(1 - x_{i+1})$ (hence the two largest terms in Q is actually just $\prod_{i=3}^{n}(1 - x_i)$ and $x_1 \prod_{i=4}^{n}(1 - x_i)$).

However, this is not the optimal NS refutation. We can infact use a divide and conquor method to get a NS refutation of degree $O(\log n)$: instead of sequentially obtaining $x_1 \implies x_{i+1}$ for every i, we can combine every two consecutive terms at every step.

- For instance, in the first step we could instead obtain $x_1 \implies x_3$ from $x_1 \implies x_2$ and $x_2 \implies x_3$, $x_3 \implies x_5$ from $x_3 \implies x_4$ and $x_4 \implies x_5$, and so on using the same method as above.
- So in the ith step we would have had a list of terms of the form

$$x_j \implies x_{j+2^i}$$

- After $\log(n)$ many steps, we would have obtained $x_1 \implies x_n$, and finish in the same manner as above.

Again, since the maximal degree of polynomials in Q increase by 1 in each step, the claim upperbound follows. In fact, this is tight:

Theorem 7. *[?] Any NS refutation of $\neg IND_n$ has degree $\Omega(\log n)$.*

Automatizability of Nullstellensatz. Next we will show that Nullstellensatz is degree automatizable with respect to Boolean CNF formulas. In particular, we have the following theorem:

Theorem 8 (Nullsatz Degree Automatizability)**.** *There is an algorithm A such that for any unsatisfiable 3CNF formula $\mathcal{C}$, $A(\mathcal{C})$ outputs a NS refutation of $\mathcal{C}$ in time $n^{O(d)}$, where $d = NS(\mathcal{C})$; that is, where d is the NS degree of $\mathcal{C}$.*

Proof. We sketch the main idea behind the above theorem.

1. Suppose $\mathcal{C} = C_1 \wedge \cdots \wedge C_m$ is an unsat 3CNF
2. Let p_i be the degree 3 polynomial corresponding to C_i, and we suppose that the collection $P = \{p_i \mid i \in [m + n]\}$ has a degree $\leq d$ NS refutation Q.
3. We can write a system of linear equations in variables $c_{i,t}$ where $1 \leq i \leq m + n, t \subset [n]$ and $|t| \leq d$, where $c_{i,t}$ represents the coefficient in front of term t in $q_i \in Q$.

4. For each term $t \neq \emptyset$, $|t| \leq d$, there will be an equation that says the coefficients corresponding to term t sum to 0; And for $t = \emptyset$, we have one equation which states that the coefficients corresponding to term t sum to 1.

Since we have $O(m \cdot n^d) = n^{O(d)}$ variables and at most $n^{O(d)}$ equations, we can solve in poly$(n^{O(d)})$ time (since linear programming is in P.) Note that if we do not know if d works, we can just use trial and error to increase d one by one.

□

Recall from previous lectures, that we have proven width-size tradeoffs for Resolution. Similar degree-(monomial)size tradeoffs exist for NS. We have the following theorem (that we won't prove here).

Theorem 9. *For any $3CNF$ $\mathcal{C}$, if $\mathcal{C}$ has NS refutation of monomial size s, then it has a degree $O(\sqrt{n \log s})$ NS refutation*

3 Polynomial Calculus (PC)

Next we define the Polynomial Calculus (PC) refutation system defined by Clegg, Edmonds and Impagliazzo [**?**]. PC is basically a dynamic version of NS, where instead of deriving a contradiction from a family q_i of polynomial such that $\sum_i q_i p_i = 1$, PC is a rule-based system which allows intermediate polynomials to be derived, and in turn this can lead to refutations with smaller degree.

Definition 10. *Let $P = \{p_1 = 0, \ldots, p_q = 0\}$ be a system of polynomial equations. A PC refutation that P is unsolvable over a field F is a derivation of $0 = 1$ from the initial polynomials p_i using the following rules:*

1. *From $f = 0$ or $g = 0$, we can derive $af + bg = 0$ for any $a, b \in \mathbb{F}$;*
2. *From $f = 0$ we can derive $xf = 0$ and $(1 - x)f = 0$.*

Definition 11. *The degree of a PC refutation is the max degree over all polynomials in refutation, and the size is the sum of sizes of all polynomials (total number of occurences of monomials).*

The following theorem shows that PC can simulate NS wrt to degree, since we can construct the polynomials q_i in a NS refutation of a CNF by repeatedly applying the PC rules.

Theorem 12 (PC vs NS)**.** *Let $\mathcal{C}$ be an unsat CNF. For any field F, if there is a degree d NS refutation of $\mathcal{C}$ over F, then there is also a degree d PC refutation of $\mathcal{C}$ over F.*

On the other hand, PC refutations can have much smaller degree. This theorem was proven by [**?**]:

Theorem 13. *NS canot simulate PC with respect to degree: there exists unsatisfiable 3CNFs that have degree $O(1)$ PC refutations but require $\Omega(n)$ degree NS refutations.*

Note that for the negation induction we saw earlier that there exists a PC refutation of size $O(1)$, since every term appearing in the refutation has degree at most 3. And on the other hand any NS refutation requires degree $\Omega(\log n)$; thus the induction principle witnesses a nonconstant degree separation. [**?**] shows that the *strong* induction principle gives a linear separation between the PC versus NS degree.

Automatizability of PC. Similar to NS, we also have a degree-automatizability result of PC,

Theorem 14 ([?]). *There is an algorithm A such that for any unsat 3CNF $\mathcal{C}$, $A(\mathcal{C})$ outputs a PC refutation in time $n^{O(d)}$ where d is the minimum degree of any PC refutation of $\mathcal{C}$.*

The proof is more complicated than the analogous theorem for NS, and uses a modified version of the Grobner basis algorithm which generates a generating set for any ideal I of the polynomial ring $\mathbb{F}[X_1, \cdots, X_n]$

Finally, a similar size-degree tradeoff also holds for PC:

Definition 15. *The monomial size of a PC refutation $\{\{P_1, \cdots, P_r, 1 = 0\}$ is the sum of all the monomial sizes of P_i. The monomial size of an CNF F is the min monomial size over all PC refutation of it.*

Theorem 16. *For any 3CNF $\mathcal{C}$, if $\mathcal{C}$ has a PC refutation of monomial-size S, then $\mathcal{C}$ has a degree $O(\sqrt{n \log s}$ PC refutation.*

There are also other $\Omega(n)$lower bound results for PC which are tight: such as PHP and the mod-q counting principle (this one requires the PC refutation to be over a field of characteristic not q (see [?]).

4 Ideal Proof System (IPS)

4.1 Motivation

The Ideal Proof System (IPS) was defined by Grochow and Pitassi [?]; this proof system greatly generalizes NS and PC by working directly with polynomials represented as algebraic circuits. This leads to refutations that can be much shorter than PC/NS refutations. However a important property of IPS refutations is that while they can be verified in *randomized* polynomial time, they are not known to be verifiable in deterministic polynomial time. Thus, they are not known to be a Cook-Reckhow proof system. IPS refutations are very powerful; in fact, Grochow and Pitassi prove a nontrivial connection between IPS lower bounds and algebraic circuit lower bound, that we will explain.

Definition 17. *An IPS refutation of $P = \{p_1, \ldots, p_m\}$ is an algebraic circuit $C(x_1, \ldots, x_n, y_1, \ldots, y_m)$ that satisfies the following properties:*

1. $C(x_1, \ldots, x_n, p_1(x), \ldots, p_m(x)) = 1$;
2. $C(x_1, \ldots, x_n, 0, \ldots, 0) = 0$.

The size of the proof is the circuit size of C, and the IPS size of P is the minimum circuit size over all circuits C that are IPS refutations of P.

We note that IPS refutations can be verified in randomized polynomial time (RP). To see this, we first use the fact that PIT (polynomial identity testing) is in RP.

Theorem 18 (Swartz-Zippel). *Given an algebraic circuit C over a field F, there is a randomized algorithm A that runs in time polynomial in the size of C, and that has one-sided error at most $1/3$ such that: If C computes the identically 1 polynomial, then $A(C)$ outputs 1 with probability 1, and if C does not compute the identically 1 polynomial, then $A(C)$ outputs 0 with probability at least $2/3$.*

Using the above theorem, it is easy to see that IPS refutations can be verified in randomized polynomial time, since each of the two properties in the definition of IPS refutation are polynomial-identity testing properties: For the first item, we simply run the PIT algorithm on the algebraic circuit C with the y-variables replaced by the polynomials $p_1, \ldots, p_m$; and for the second property, it suffices to show that the circuit $1 - C(x_1, \ldots, x_n, 0, \ldots, 0)$ compute the identically-1 polynomial.

4.2 Properties of IPS Refutations

Grochow and Pitassi [**?**] prove that IPS simulates NS and PC. Furthermore than prove that IPS can even simulate the very powerful Extended Frege (EF) proof system.

They also prove that superpolynomial lower bounds for IPS for an unsatisfiable family of CNFs implies a longstanding algebraic circuit lower bound, namely that $VNP \neq VP$. In simple terms, superpolynomial lower bounds implies that the permanent polynomial cannot be computed by polynomial-size algebraic circuits, a longstanding open problem in algebraic circuit complexity that is viewed as the algebraic analog of the P versus NP question.

References

[1] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on hilbert's nullstellensatz and propositional proofs. *Proceedings of the London Mathematical Society*, 73(1):1–26, 1994.

[2] Samuel R Buss and Toniann Pitassi. Good degree bounds on nullstellensatz refutations of the induction principle. *computational complexity*, 7:162–178, 1996.

[3] Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. Using the groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 174–183, 1996.

[4] Joshua A. Grochow and Toniann Pitassi. Circuit complexity, proof complexity, and polynomial identity testing: The ideal proof system. *J. ACM*, 65(6):37:1–37:59, 2018.

[5] A. Kolodziejczyk S. Buss and K. Zdanowski. Collapsing modular counting in bounded arithmetic and constant depth propositional proofs. *Transactions of the American Math Society*, 367:7517–7563, 2015.