Notes about Proof Complexity

Xinhao Nie

Aarhus University

April 8, 2025

1 Motivation

Before introducing proof systems and discussing their complexity, we must clarify the underlying motivation. Let's consider two common complexity class NP and coNP. Because all deterministic time complexity classes, i.e. P, are closed under complementation, one possible way to prove P $\neq$ NP is by proving NP is not closed under complementation, which means starting with NP $\neq$ coNP. We already know that TAUT is coNP-complete. Naturally, we have:

Theorem 1.1. NP *is closed under complementation if and only if* TAUT *is in* NP.

The question of whether TAUT is in NP is equivalent to whether there is a propositional proof system in which every tautology has a short proof. In other words, we can view a proof system as a verifier. If the proposition is tautology, there exists a proof such that all steps in it stay consistent with the proof system which means pass the verification. If both proof system and proof are properly defined, we can believe that searching adequate proof can be simulated in NTM and such proof can be verified in polynomial time TM.

Definition 1.1. *If $L \subseteq \Sigma^*$, a proof system for L is a function $f : \Sigma_1^* \to L$ for some alphabet Σ_1 and f such that f is a surjective function. We say that the proof system is polynomially bounded iff there is a polynomial $p(n)$ such that for all $y \in L$ there is $x \in \Sigma_1^*$ such that $y = f(x)$ and $|x| \leq p(|y|)$.*

Because computing function $f(x)$ actually only requires verifying whether each step in the proof x conforms to the rules of reasoning and the size of rules is finite, we can compute $f(x)$ in $O(|x|)$. Therefore, we should focus on $|x|$. [**TBD:** A natural question is why we need different proof system. I think it's hard to proof any kind of problem using only one proof system. Can I take an example? But some proof system may have stronger power than others, does it mean if we have to restrict what system we use the proof complexity cannot perferctly capture the difficulty?] Additionally, a set L is in NP if L has a polynomially bounded proof system where proof system could be any possible one because any bounded proof system can be simulated by a NTM, and the computation steps on NTM to decide a language L is in NP forms a proof! Then we can answer whether TAUT is in NP by giving a polynomially bounded proof system.

Theorem 1.2. *A set L is in* NP *iff $L = \emptyset$ or L has a polynomially bounded proof system.*

[**TBD:** Why there is a case $L = \emptyset$?] [**ToLearn:** Recursive function theory: recursively enumerate set]

2 Some Proof System

2.1 Equivalence Classes On Proof Systems

Definition 2.1. *If $f_1 : \Sigma_1^* \to L$ and $f_2 : \Sigma_2^* \to L$ are proof systems for L, then f_2 p-simulates f_1 provided there is a function $g : \Sigma_1^* \to \Sigma_2^*$ such that $f_2(g(x)) = f_1(x)$ for all x.*

It's easy to show that p-simulation is a transitive reflexive relation, so that its symmetric closure is an equivalence relation. If we require the function g is bounded in length by a polynomial in the length of its argument and a proof system f_2 for L p-simulates a polynomially bounded proof system f_1 for L, then f_2 is also polynomially bounded.

2.2 Nullstellensatz

Definition 2.2. *Fix an algebraic structure $\mathbb{R}$ and a set of polynomials $\mathcal{F} = \{F_i\}$. Let d be a non-negative integer. A d-design for $\mathbb{F}$ is a mapping D from $\mathbb{R}$-polynomials of degree $\leq d$ to $\mathbb{R}$ such that:*

1. $D(1) = 1$
2. *D is linear, which means $D(X + Y) = D(X) + D(Y)$.*
3. *$D(Q \cdot F_i) = 0$ for any polynomial Q and $F_i \in \mathbb{F}$ such that $deg(Q) + deg(F_i) \leq d$.*
4. *$D(x_i^2 Q) = D(x_i Q)$, where x is a variable in $R[x_1, x_2, ..., x_n]$ and $deg(Q) < d - 1$.*

Notice that the mapping D is actually determined by the values it assigns to monomials of degree less than d, since it is a linear mapping and any polynomial can be decomposed into some monomials. Furthermore, due to property 4, it suffices to consider only multilinear monomials when determining the value of D. Alternatively, we can assume that $\mathcal{F}$ includes $x_i(x_i - 1)$ for all i, in which case property 3 implies property 4.

Naturally, for multilinear monomials, we can view them as conjunctions of the atomic statements represented by the propositional variables in the monomial. Furthermore, we found there is a connection between d-designs and Nullstellensatz refutation of degree d. Next, two theorems describe this connection.

Theorem 2.1. *If $\mathcal{F}$ exists a d-design, then there cannot be a Nullstellensatz refutation of degree $\leq d$ to refutate $\mathcal{F}$.*

Proof. If there exists a Nullstellensatz refutation of degree $\leq d$, then

$$1 = \sum_i P_i \cdot F_i + \sum_j Q_j \cdot (x_j^2 - x_j)$$

. Using the d-design for $\mathcal{F}$,

$$D(1) = 1 \neq 0 = \sum_i D(P_i \cdot F_i) + \sum_j D(Q_j x_j^2) - D(Q_j x_j)$$

. □

A converse of the above theorem, to some extent, also holds:

Theorem 2.2. *Suppose the algebraic structure $\mathbb{R}$ is a field. If $\mathcal{F}$ does not have a Nullsltellensatz refutation of degree d, then there is a d-design for $\mathcal{F}$.*

Proof. Suppose there are n variables used in our proposition. Let δ be the number of monomials of degree $\leq d$. Then we can calculate $\delta = \sum_{i=0}^{d} \binom{n+i-1}{i}$. Recall that $\binom{n+i-1}{i} = \binom{n+i-1}{n-1}$. It means δ is not infinite. Then for any polynomial H of degree $\leq d$, it can be viewed as a vector $\mathbf{v_H}$ where each element a_Q is the coefficient of a monomial Q. Therefore, $\mathbf{v_H}$ has dimension δ.

Let $\mathcal{G}$ be a set of polynomials in the form $Q \cdot F$, where Q is a monomial and $F \in \mathcal{F}$, and $\deg(Q \cdot F) \leq d$. For simplicity, we consider $\mathcal{F}$ to include all polynomials of the form $x_i(x_i - 1)$. Each $g_i \in \mathcal{G}$ can be viewed as a vector $\mathbf{v_{g_i}}$, analogous to what was done for H.

Define a vector $\mathbf{v_1}$ with only one nonzero element on the constant term. A Nullstellensatz refutation of degree at most d exists if and only if there is an R-linear combination of the vectors $\{\mathbf{v_{g_i}}\}$. Let $\mathbf{M}$ be the $\delta \times |\mathcal{G}|$ matrix with columns $\mathbf{v_{g_i}}$,, where $\delta = \sum_{i=0}^{d} \binom{n+i-1}{i}$. This is equivalent to finding a solution $\mathbf{w}$ to the linear equation $\mathbf{Mw} = \mathbf{v_1}$.

It's equivalent to checking whether the rank of coefficient matrix $\mathbf{M}$ equals the rank of the augmented matrix $[\mathbf{M}, \mathbf{v_1}]$. Since there is only one nonzero element in $\mathbf{v_1}$, that element cannot be a linear combination of zeros. It's therefore equivalent to checking whether the last row of $\mathbf{M}$ is a linear combination of other rows in $\mathbf{M}$.

Thus, there is a Nullstellensatz refutation of degree $\leq d$ if and only if the last row of $\mathbf{M}$ is linearly independent of the rest of the row vectors. If the last row is a linear combination, we can indeed find a possible d-design. Denote each row as $\mathbf{u_Q}$ where Q corresponds to that row. Suppose the combination is

$$\sum_{\deg(Q)\leq d} \alpha_Q \mathbf{u_Q} = 0. \tag{1}$$

Then we can actually use α_Q as $D(Q)$ for all monomials Q, and $D(1) = 1$.

The rest is to check if D is valid. Since it's sufficient that all properties hold for monomials, we only need to consider the case for monomials. For any monomial Q and $F \in \mathcal{F}$, $Q \cdot F$ must be in $\mathcal{G}$. Because this is a column in matrix $\mathbf{M}$, equation (1) holds for the column $Q \cdot F$. Then $D(Q \cdot F) = D(\sum_{\deg(Q)\leq d} \alpha_Q u_Q^{Q\cdot F}) = 0$. It's not difficult to check that D is linear. □

2.3 Sherali-Adams Proof System

Originally, it was designed to handle mixed linear programming. Amazing!

2.4 Mixed-integer Zero-one Programming

2.4.1 Relaxation and Linearization

Consider a linear mixed-integer zero-one programming problem whose feasible region is given as follows:

$$\begin{aligned} X = \{(x, y) : \sum_{j=1}^{n} \alpha_{rj} x_j + \sum_{k=1}^{m} \gamma_{rk} y_k \geq \beta_r, \forall r \in [R], \\ x_j \in \{0, 1\}, \forall j \in [n], \\ 0 \leq y_k \leq 1, \forall k \in [m]\} \end{aligned} \tag{2}$$

One classical approach involves relaxing the integrality constraint by simply treating the variables as continuous real values rather than integers, thereby allowing for an optimal but potentially infeasible solution. The Sherali-Adams reformulation-linearization is another approach.

For any $d \in [N]$, let us define the polynomial factors of degree d as

$$F_d(J_1, J_2) = (\prod_{j\in J_1} x_j)(\prod_{j\in J_2}(1 - x_j))$$

for each $J_1, J_2 \subseteq [N]$ such that $J_1 \cap J_2 = \phi$, and $|J_1 \cup J_2| = d$. Sepcially, $F_0(\phi, \phi) = 1$.

Sherali and Adams use each polynomial factor of differing degree to construct a hierarchy of relaxations. In the description of relaxation algorithm, we just fix a d. We multiply $F_d(J_1, J_2)$ on both sides and remove the integerality constraints, but, to some extent, hidenly add constraints $x_i(x_i - 1) = 0$. Then the programming problem (2) would be updated to

$$(\sum_{j\in J_1} \alpha_{rj})F_d(J_1, J_2) - \beta_r F_d(J_1, J_2) + \sum_{j\in[N]\setminus(J_1\cup J_2)} \alpha_{rj} F_{d+1}(J_1 + j, J_2) + \sum_{k=1}^{m} \gamma_{rk} y_k F_d(J_1, J_2) \geq 0, \forall r \in [R], \forall F_d. \tag{3}$$

Because we remove integerality constraints, it's suffice to restrict $0 \leq x, y \leq 1$. Thus, we have another two constraints

$$\begin{aligned} F_D(J_1, J_2) \geq 0, D = min(d + 1, N), \forall F_D \\ F_d(J_1, J_2) \geq y_k F_d(J_1, J_2) \geq 0, \forall F_d, \forall k \in [m]. \end{aligned} \tag{4}$$

Viewing the constraints above in expanded form as a sum of monomials, linearize them by substituting the following variables for the corresponding nonlinear terms $\forall J \subseteq [N], \forall k \in [m], w_J = \prod_{j \in J} x_j, v_{Jk} = y_k \prod_{j \in J} x_j$. Specially, $w_\phi = 1$. Furthermore, denoting by $f_d(J_1, J_2)$ and $f_d^k(J_1, J_2)$ the respective linearized forms of the polynomial expressions $F_d(J_1, J_2)$ and $y_k F_d(J_1, J_2)$.

The original problem of finding the optimal (X, Y) is transformed into a new problem of finding the optimal (W, V). Actually, W includes $w_j = x_j$ and V includes $v_{\phi k} = y_k$. For simplicity, we can denote the solution as (X, Y, W, V). We then need to demonstrate the validity of this relaxation.

2.4.2 Validity

Theorem 2.3. $Conv(X) \subseteq X_{P_n} \subseteq X_{P_{n-1}} \subseteq ... \subseteq X_{P_1} \subseteq X_{P_0} = X_0$ *where* X_{P_d} *is the corresponding* $(\mathbf{x}, \mathbf{y})$ *to* $X_d = \{(\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{v})\}$. *In particular,* $X_{P_d} \cap \{(x, y) : x \text{ } is \text{ } binary\} = X$ *for all* $d \in [N]$.

Proof. First, we show $X_{P_d} \subseteq X_{P_{d-1}}$.

$$F_d(J_1 + j, J_2) \geq 0, F_d(J_1, J_2 + j) \geq 0 \Rightarrow F_{d-1}(J_1, J_2) \geq 0$$

Then constraints (4) hold. For constraints (3), all terms are analogous except $\sum_{j \in N \setminus J_1 \cup J_2} \alpha_{r_j} F_{d+1}(J_1 + j, J_2)$. We choose pairs $(J_1 + t, J_2)$ and $(J_1, J_2 + t)$ where $t \notin J_1 \cup J_2$. Then $\sum_{j \in N \setminus J_1 \cup J_2 \cup t}(F_{d+2}(J_1 + t + j, J_2)) + \sum_{j \in N \setminus J_1 \cup J_2 \cup t}(F_{d+2}(J_1 + j, J_2 + t)) + F_{d+1}(J_1 + t, J_2) = \sum_{j \in N \setminus J_1 \cup J_2} \alpha_{r_j} F_{d+1}(J_1 + j, J_2)$. Then we can conclude $X_{P_d} \subseteq X_{P_{d-1}}$.

Second, let us show that $Conv(X) \subseteq X_{P_n}$. $X \subseteq X_{P_n}$ is trivial. And X_{P_n} is convex. Thus $Conv(X) \subseteq X_{P_n}$. In particular, $X = Conv(X) \cap \{(x, y) : x \text{ } is \text{ } binary\} = X_0 \cap \{(x, y) : x \text{ } is \text{ } binary\}$ because X_0 just removes integrality restrictions. Finally, it's suffice that $X_{P_d} \cap \{(x, y) : x \text{ } is \text{ } binary\} = X$ for all $d \in [N]$. □

2.4.3 Convexity

Theorem 2.4. $X_{P_n} = Conv(X)$

2.5 Application on Proof System