Introduction to Proof Complexity

Adrian She

December 2020

Abstract

These are notes for an introductory talk on proof complexity given at the University of Toronto Theory Student Seminar Seminar on November 18th, 2020. The study of proof complexity was initiated by Cook and Reckhow in the 1970s as a possible way to resolve the P versus NP problem. The notes will introduce the proof complexity program, the Resolution proof system, and a proof due to Haken that the pigeonhole principle requires exponential length refutations in Resolution.

1 Introduction

The proof complexity program was introduced by Cook and Reckhow in [CR79] as a proposed way to resolve the famous **P** versus **NP** problem.

Definition 1. Given a language $L \subseteq \{0,1\}^*$, a **proof system** for L is a polynomial time algorithm V such that for all $x \in \{0,1\}^*$, $x \in L$ if and only if there is some string $p \in \{0,1\}^*$ such that V accepts the pair (x, p).

Definition 2. A proof system for L is **p-bounded** if it is proof system where for all $x \in L$ the string p certifying its membership can be chosen to have length $|p|$ bounded by some polynomial in $|x|$.

By definition, the languages with p-bounded proof systems are exactly the languages in **NP**. Now recall that **UNSAT**, the set of unsatisfiable formulas, is **coNP**-complete by the Cook-Levin theorem. Putting these together, we get the following fundamental observation.

Theorem 1. *There is a polynomially bounded proof system for* **UNSAT** *if and only if* **NP = coNP.**

Recall that **NP** $\neq$ **coNP** implies that **P** $\neq$ **NP**. Therefore, one of the ideas in [CR79] is that showing that specific proof systems are not polynomially bounded would make progress towards the **P** versus **NP** problem, and hopefully progress towards the general case can be made as stronger and stronger proof systems are investigated.

2 Resolution Proofs

One of the simplest proof systems is the resolution proof system. It is used widely in practice in SAT solving. Before introducing it we will recall some basic definitions.

Definition 3. A Boolean formula F is a **CNF** formula if F is an AND of clauses $F = C_1 \wedge \cdots \wedge C_m$, where each clause C_i is an OR of literals (variables x_i or their negation $\overline{x_i}$.)

A simple example of an unsatisfiable CNF is the formula $F = x_1 \wedge (\overline{x_1} \vee x_2) \wedge (\overline{x_2} \vee x_3) \wedge (\overline{x_3})$

Definition 4. The **resolution rule** takes two clauses $C_1 = A \vee x$ and $C_2 = B \vee \overline{x}$ sharing a common variable, and derives the new clause $C' = A \vee B$ from them.

The resolution rule is sound. We have that if C_1, C_2 are true, then C' must be true as well.

Definition 5. Given a CNF formula $F = C_1 \wedge \cdots \wedge C_m$, a **resolution refutation** is a sequence of clauses $D_1, \ldots, D_s$ where

1. $D_i = C_i$ for all $1 \leq i \leq m$,
2. D_j for $j > m$ is derived from clauses D_k, D_l with $k, l < j$ using the resolution rule, and

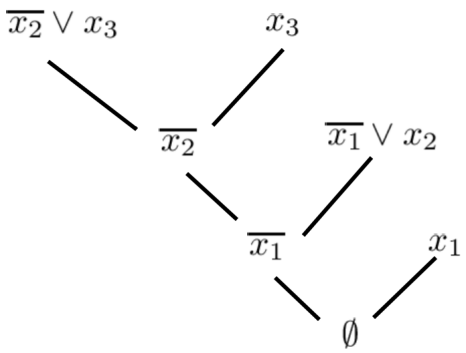

Figure 1: A resolution refutation for $F = x_1 \wedge (\overline{x_1} \vee x_2) \wedge (\overline{x_2} \vee x_3) \wedge (\overline{x_3})$, illustrated graphically

3. The final clause D_s is the empty clause $\emptyset$. We say that s is the size of the refutation.

Every unsatisfiable CNF F has a resolution refutation (although it is possibly of exponential size in the number of variables). One way to see this is that any decision tree that queries the variables of F to find a clause that is falsified by a particular truth assignment to the variables can be turned into a tree-like resolution proof. For details, consult notes available at [Pit19].

3 The Resolution Lower Bound for the Pigeonhole Principle

Now we will prove that the smallest resolution refutation for some unsatisfiable CNFs must have exponential size.

Definition 6. The **pigeonhole principle** PHP^n_{n-1} is the CNF formula over variables x_{ij} with $1 \leq i \leq n, 1 \leq j \leq n-1$ with the following clauses:

- Pigeon axioms: $P_j = \vee_{j=1}^{n-1} x_{ij}$ for $i = 1, \ldots, n$. These clauses encode that the requirement that every pigeon hole belong to some hole.
- Hole axioms: $\overline{x_{i_1 j}} \vee \overline{x_{i_2 j}}$ for $i_1 \neq i_2$ and $j = 1, \ldots, n-1$. These clauses encode the requirement that no two pigeons occupy the same hole.

It is clear that PHP^n_{n-1} is not satisfiable, since there cannot be any injective function $[n] \to [n-1]$.

Theorem 2 ([Hak85])**.** *There is a constant $c > 1$ such that any resolution refutation of PHP^n_{n-1} requires size c^n.*

We will present a simplification of Haken's original argument due to Beame and Pitassi in [BP96], also presented in [Juk12]. The argument has two parts. First we will show that any resolution proof of PHP^n_{n-1} must contain a wide clause (a clause with many variables). Next, we will show that if the size of the proof is too small, then it is possible to "kill" all of the wide clauses with a suitable restriction. Thus, the width lower bound implies that the size must be large.

3.1 The Width Lower Bound

Firstly, we will need the notion of a critical assignment.

Definition 7. A truth assignment α is i-critical if it falsifies only the pigeon axiom P_i in PHP^n_{n-1}. A truth assignment is **critical** if it is i-critical for some $1 \leq i \leq n$.

In other words, a critical truth assignment encodes a matching between the pigeons and the holes where exactly one pigeon has been left out. For example, encoding the truth assignment as an $n \times n-1$ matrix, we have that $\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$ is a 2-critical assignment for PHP^3_2. We will refer to truth assignments as matrices whenever it is convenient by arranging the variables appropriately. Given a truth assignment α and clause C, we will use $C(\alpha)$ to denote the truth value of C evaluated at α.

Lemma 1. *Let C be any clause and C^+ be the clause C with any negated variable $\overline{x_{ij}}$ in C replaced by the subclause $X_{ij} = \bigvee_{i' \neq i} x_{i'j}$. For any critical truth assignment α, $C^+(\alpha) = C(\alpha)$.*

Proof. If $x_{ij}(\alpha) = 1$, then $\overline{x_{ij}}(\alpha) = X_{ij}(\alpha) = 0$ since no other pigeon has been matched to hole j in α. Otherwise, if $x_{ij}(\alpha) = 0$, then $\overline{x_{ij}}(\alpha) = X_{ij}(\alpha) = 1$ since some other pigeon must be matched to hole j. □

Lemma 2. *Every resolution refutation of PHP^n_{n-1} must contain a clause C where the width of C^+ satisfies $w(C^+) \geq \frac{2n^2}{9}$.*

Proof. Given a clause C let $Pigeon(C) \subseteq [n]$ be the set of pigeons where there is some i-critical assignment α for which $C(\alpha) = 0$. Let $\mu(C) = |Pigeon(C)|$. We can observe that

- $\mu(P_i) = 1$ for each pigeon axiom P_i.
- $\mu(\emptyset) = n$ since every assignment falsifies $\emptyset$.
- If clause C was derived from clauses A, B using the resolution rule then $\mu(C) \leq \mu(A) + \mu(B)$. This is because every assignment falsifying C must have falsified either A or B.

Therefore, we can conclude from the conditions above that if R was a resolution refutation of PHP^n_{n-1} there is some clause C in R where $\frac{n}{3} < \mu(C) \leq \frac{2n}{3}$. Now we want to argue that for that clause C, we have $w(C^+) \geq \frac{2n^2}{9}$.

Fix $i \in Pigeon(C)$ and some $j \notin Pigeon(C)$ so that α is an i-critical assignment that falsifies C. Let α' be the truth assignment created from α by exchanging row i and j. Now α' is j-critical by definition. Furthermore, $C(\alpha) = C^+(\alpha) = 0$ and $C(\alpha') = C^+(\alpha') = 1$ by construction and the previous lemma. Therefore, since C^+ contains positive literals only and there was exactly one variable x_{ik} whose truth value was switched from 0 to 1 from α to α', we conclude that C^+ must contain x_{ik}.

Repeating this argument for all pairs (i, j) for $i \in Pigeon(C)$ and $j \notin Pigeon(C)$ since all pairs must yield a distinct variable in C^+, we can conclude that the width of C^+ is at least $s(n - s)$ where $s = |Pigeon(C)|$. By assumption that $\frac{n}{3} < |Pigeon(C)| \leq \frac{2n}{3}$, we get the width lower bound that $w(C^+) \geq s(n - s) \geq \frac{2n^2}{9}$. □

3.2 The Restriction Step

Now we will use the width lower bound to get the size lower bound. We let R be a resolution refutation of PHP^n_{n-1} and let R^+ be the positive version where each clause $C \in R$ is replaced by its positive version C^+. Let $\epsilon > 0$ be a constant whose exact value we will choose later.

Definition 8. A clause in R^+ is **ϵ-wide** if its width satisfies $w(C^+) \geq \epsilon n^2$.

Let S be the number of wide clauses in R^+. We can conclude that there is some variable x_{ij} appearing in at least ϵS wide clauses in R^+ by counting the number of variables in the wide clauses. Therefore, we can define a restriction of the variables by matching pigeon i to hole j, that is we set $x_{ij} = 1, x_{i'j} = 0, x_{ij'} = 0$ for all $i' \neq i, j' \neq j$.

Notice that once we apply this restriction to R (and hence R^+), R is now a resolution refutation of PHP^{n-1}_{n-2} and there are now at most $(1 - \epsilon)S$ wide clauses in R^+. Therefore, we can inductively apply k restrictions for $k = \frac{\ln S}{\epsilon}$ so that we get a proof of PHP^{n-k}_{n-1-k} where the positive version R^+ contains no wide clauses since there are at most $S(1 - \epsilon)^k$ wide clauses and $S(1 - \epsilon)^k < Se^{-k\epsilon} \leq 1$ by the choice of ϵ and k.

However, from the previous lemma, we know that in any proof of PHP^{n-k}_{n-k-1} there is some clause C where $w(C^+) \geq \frac{2(n-k)^2}{9}$ but on the other hand, we have produced a proof of PHP^{n-k}_{n-k-1} with no wide clauses by restricting the proof of PHP^n_{n-1}. Therefore, the inequality

$$\frac{2(n-k)^2}{9} = \frac{2(n - \frac{\ln S}{\epsilon})^2}{9} \leq w(C^+) < \epsilon n^2$$

must be satisfied. After some algebra , we can conclude that $\ln S \geq \epsilon n - \epsilon\sqrt{\frac{9\epsilon}{2}}n$, so picking $\epsilon = \frac{8}{81}$ to maximize the bound means that the resolution proof of PHP^n_{n-1} must contain at least

$$S \geq \exp(\frac{8n}{243}) \geq 1.033^n$$

wide clauses, which completes the proof.

4 Other Proof Systems and Problems

We have now proved an exponential lower bound on the pigeonhole principle in the resolution proof system. A natural question to ask is what other lower bounds are known for the pigeonhole principle, which we will summarize below.

- Exponential lower bounds are known in AC^0-Frege systems using a switching lemma argument. [BIK+92]
- For algebraic proof systems, Razborov proved $\Omega(n)$ degree lower bounds for the polynomial calculus [Raz98].
- For semi-algebraic proof systems, an $\Omega(n)$ degree lower bound is known for the Sherali-Adams proof system based on linear programming [GM08].

In contrast, efficient (polynomial size) proofs are known for PHP^n_{n-1} are known for the following proof systems: Cutting Planes [Juk12], Sum of Squares [FKP+19], and Frege [Bus87].

For a recent overview of other interesting problems in proof complexity, see Prof. Paul Beame's talk titled "Proof Complexity 2020" available at [Bea20].

References

[Bea20] Paul Beame. Proof complexity 2020, Jan 2020. `https://www.birs.ca/events/2020/5-day-workshops/20w5144/videos/watch/202001200905-Beame.html`.

[BIK+92] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, Pavel Pudlák, and Alan Woods. Exponential lower bounds for the pigeonhole principle. In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '92, page 200–220, New York, NY, USA, 1992. Association for Computing Machinery.

[BP96] Paul Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, FOCS '96, page 274, USA, 1996. IEEE Computer Society.

[Bus87] Samuel R Buss. Polynomial size proofs of the propositional pigeonhole principle. *Journal of Symbolic Logic*, pages 916–927, 1987.

[CR79] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979.

[FKP+19] Noah Fleming, Pravesh Kothari, Toniann Pitassi, et al. Semialgebraic proofs and efficient algorithm design., 2019. `https://eccc.weizmann.ac.il/report/2019/106/`.

[GM08] Konstantinos Georgiou and Avner Magen. Limitations of the Sherali-Adams lift and project system: Compromising local and global arguments. Technical report, Technical Report CSRG-587, University of Toronto, 2008.

[Hak85] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297 – 308, 1985. Third Conference on Foundations of Software Technology and Theoretical Computer Science.

[Juk12] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*. Algorithms and Combinatorics. Springer Berlin Heidelberg, 2012.

[Pit19] Toniann Pitassi. Resolution completeness notes, 2019. Notes for CSC 438, `http://www.cs.toronto.edu/~toni/Courses/438/Mynotes/res.pdf`.

[Raz98] Alexander A Razborov. Lower bounds for the polynomial calculus. *computational complexity*, 7(4):291–324, 1998.