DOI 10.4171/8ECM/03

Propositional proof complexity

Alexander A. Razborov

Abstract. Propositional proof complexity studies efficient provability of those statements that can be expressed in propositional logic, in various proof systems, and under various notions of "efficiency." Proof systems and statements of interest come from a variety of sources that, besides logic and combinatorics, include many other areas like combinatorial optimization and practical SAT solving. This article is an expanded version of the ECM talk in which we will attempt to convey some basic ideas underlying this vibrant area.

1. General overview

Like with many other areas in theoretical computer science, the framework of propositional proof complexity can be easily explained to a mathematically advanced high school student. In fact, its core definitions are so easy to give that we prefer to interlace them with the discussion rather than to separate the two.

Definition 1.1 (preliminaries). We fix a set of *Boolean* (that is, 0-1 valued, where 0 stands for FALSE and 1 stands for TRUE) variables. A *literal* is either a variable x or its negation that will be denoted by $\bar{x}$. The alternate notation $\neg x$ is also used in the literature, and sometimes we will use the uniform notation x^a, $a \in \{0,1\}$, where $x^1 \overset{\text{def}}{=} x$ and $x^0 \overset{\text{def}}{=} \bar{x}$. A *clause* C is a disjunction of literals: $C = x_{i_1}^{a_1} \vee \cdots \vee x_{i_w}^{a_w}$ in which no variable appears twice. A *conjunctive normal form* (CNF in what follows) is a conjunction of clauses $\tau = C_1 \wedge \cdots \wedge C_m$, often identified with the set $\{C_1, \ldots, C_m\}$ of which it is comprised. Whenever n appears as a subscript in τ_n, it always stands for the number of variables.

One very important complexity measure for this article is width. The *width* of a clause is the number of literals w in it. The *width* of a CNF is the maximal width of a clause in it. A k*-CNF* is a CNF of width $\leq k$. An *assignment* (sometimes called *truth assignment*) is a mapping $\alpha : V \to \{0,1\}$. It is naturally extended to literals, clauses, and CNFs. For example, for the assignment α given by $\alpha(x_1) = 1, \alpha(x_2) = 0$,

2020 Mathematics Subject Classification. Primary 03F20; Secondary 68Q17.
Keywords. Proof complexity, lower bounds, resolution.

$\alpha(x_3) = 1$, and $\alpha(x_4) = 0$ we have $\alpha(\bar{x}_2) = 1$, $\alpha(\bar{x}_1 \vee x_2 \vee x_4) = 0$, $\alpha(x_2 \vee x_3) = 1$, and $\alpha((\bar{x}_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_3)) = 0$. A CNF τ is *satisfiable* if there exists at least one truth assignment α such that $\alpha(\tau) = 1$; α itself is called then a *satisfying assignment*. Otherwise, τ is *unsatisfiable*.

The algorithmic problem SATISFIABILITY of determining whether a given CNF τ is satisfiable or not is NP-complete. In fact, it is the most fundamental NP-complete problem, as well as historically the first [4, Chapter 2.4]. It is central to the field of computational complexity.

In proof complexity, accents are slightly shifted. Instead of *deciding* whether τ is satisfiable or not, we want a *proof* of the answer, and we are interested in the resources necessary to *represent* this proof, in most cases abstracting away from the complexity of *finding* it.

If we want to certify the *satisfiability* of τ, then the task becomes trivial: a proof consists of a satisfying assignment α itself. Let us note in passing, however, that this immediately changes once we impose additional restrictions on the verification process. Significantly oversimplifying, any proof can be written in a special "holographic" form such that, once submitted, its validity can be checked by verifying a small number of "lemmas" in it, selected randomly. This leads to one of the most beautiful and difficult topics in the computational complexity theory called *probabilistically checkable proofs* (PCPs). Unfortunately, this topic is way beyond the scope of our article, so we refer the reader to [4, Chapter 11].

The main question of interest in the propositional proof complexity is how to prove efficiently that a CNF τ is *unsatisfiable*.

Remark 1.2. If we view τ itself as representing a mathematical statement, then what we call a "proof" is actually its *refutation*. The reason why this change of direction is very convenient will become clear below. For now, let us just warn the reader that the terminology is unfortunately rather inconsistent. Say, an unsatisfiable CNF may be called in the literature "a contradiction" or even "a tautology." In what follows we also may at times be sloppy about this.

Remark 1.3. We have restricted ourselves to CNFs mostly because this class is sufficiently broad to easily encompass virtually all statements we will be interested in. It will also be a must when we discuss so-called weak proof systems. But sometimes people do consider more complicated Boolean (and not only Boolean in fact) expressions to be proved/refuted.

Once we have determined that our goal is to study efficient provability of (the unsatisfiability of) CNFs, the next task is to define what we mean by a "proof system." In the most abstract form this definition was given in the seminal paper [25] by Cook and Reckhow.

Definition 1.4 (proof systems). Let UNSAT be the set of all unsatisfiable CNFs. A *propositional proof system* is a surjective polynomial-time computable function $P : \{0,1\}^* \twoheadrightarrow \text{UNSAT}$, where $\{0,1\}^*$ is the set of all finite binary strings.

The intuition is that proofs are encoded by binary strings w and the function P first checks whether w is a legitimate proof (and outputs something trivial like $x \wedge \bar{x}$ if it is not). Then $P(w)$ is the theorem that the proof w proves, and the surjectivity of P is the property of a proof system called *completeness*: every unsatisfiable CNF possesses at least one proof (that is, refutation).

In this abstract form, the definition has turned out very useful for general "structural" studies in proof complexity; see e.g. [40, 55]. But the main focus of our article is on *concrete* fixed proof systems that are interesting for some external reasons.

Before branching into specifics, we still can give a few crucial definitions at this level of generality.

Definition 1.5 (size complexity). For a propositional proof system P and $\tau \in \text{UNSAT}$, let $S_P(\tau \vdash 0)$ be the *size complexity* of τ defined as the minimal possible bit length $|w|$ of $w \in \{0,1\}^*$ such that $P(w) = \tau$. The proof system is *p-bounded* if $S_P(\tau \vdash 0)$ is bounded by a polynomial in the bit length $|\tau|$ of τ itself.

Whether p-bounded proof systems P exist is the main motivating question of proof complexity. It is not hard to see, however, that in this generality (that is without any other restrictions on P) this is equivalent to a major question in the computational complexity.

Theorem 1.6 ([25]). *A p-bounded proof system P exists if and only if* $\mathsf{NP} = co - \mathsf{NP}$.

The following will allow us to compare different proof systems according to their strength and arrange them into a hierarchy.

Definition 1.7 (simulation and equivalence). A proof system P *p-simulates* another proof system Q if there is a polynomial-time computable function s such that the following diagram commutes:

$$\begin{tikzcd}
\{0,1\}^* \arrow[rr, "s"] \arrow[dr, "Q"'] & & \{0,1\}^* \arrow[dl, "P"] \\
& \text{UNSAT} &
\end{tikzcd}$$

Informally, any Q-proof w can be efficiently converted into a P-proof $s(w)$ of the same theorem; note that the poly-time computability of s automatically implies that $|s(w)|$ is bounded by a polynomial in $|w|$. Two proof systems are *p-equivalent* if they p-simulate each other.

We now move on to consider concrete proof systems.

2. Strong proof systems

The classification of proof systems into "weak" and "strong" is loosely defined and it is not universally agreed upon. Roughly speaking, a proof system P is considered strong if we cannot rule out that it is p-bounded, and it is sufficiently widely believed that this inability is in a sense inherent. We will see below at least one proof system in the "gray area."

Strong proof systems are usually associated with original motivations for the propositional proof complexity coming from mathematical logic, more exactly from the study of weak theories of bounded arithmetic. On this subject I will be very brief (as I was in my ECM presentation); the reader willing to learn more about these fascinating connections with classical proof theory is referred to the monographs [21, 24, 38, 39]. As before, we precede the discussion with a few definitions.

Definition 2.1 (Frege, informal). Take any textbook in the mathematical logic. It will most likely begin with a description of propositional calculus given as a Hilbert-style proof system. That is, it will contain finitely many *axiom schemes* like $A \Rightarrow (A \vee B)$ or $A \vee \neg A$ and *inference rules* like

$$\frac{A \qquad A \Longrightarrow B}{B} \quad \text{(modus ponens).}$$

Here $A, B, C, \ldots$ are placeholders for which one can substitute an arbitrary Boolean formula. This is a *Frege proof system.*

Remark 2.2. One very important distinction in propositional proof complexity is whether we consider proofs in the tree-like form or allow arbitrary directed acyclic graphs (DAGs). In other words, do we allow intermediate "lemmas" to be used more than once or not? This is of little significance in the classical proof theory since any DAG can be expanded into a tree (if you need to use a lemma more than once, just repeat its inference). But this may result in an exponential increase in the size of the proof and, as a result, for weak proof systems we should strictly distinguish between the two possibilities. It is a non-trivial fact that for the Frege proof system these two versions are actually p-equivalent [37].

Textbooks in the mathematical logic seldom use the same finite sets of axioms and inference rules, and in many cases they use even different sets of Boolean connectives (e.g., we have just seen the implication $\Rightarrow$ that was not in our original de Morgan language $\{\neg, \wedge, \vee\}$). But it turns out that modulo polynomial equivalence all these choices are immaterial.

Theorem 2.3 ([60]). *Any two Frege proof systems, understood as Hilbert-style complete proof systems based on a finite number of axiom schemes and inference rules, are p-equivalent.*

Remark 2.2, along with Theorem 2.3, strongly suggests that the concept of Frege proof system is very robust and hence natural. This system is denoted by F; thus, the function $S_{\mathrm{F}}(\tau \vdash 0)$ is well defined up to a polynomial.

Definition 2.4 (extended Frege, informal). An *extended Frege proof system*, denoted by EF, is the Frege proof system augmented with the following *extension rule*. This rule allows to introduce at any moment a *fresh new* propositional variable x_A as an abbreviation for a formula A. The proof then may proceed using also the *extension axioms* $x_A \equiv A$, and this can happen recursively.

All that has been said about the robustness of Frege proof systems fully applies to EF as well. That is, $S_{\mathrm{EF}}(\tau \vdash 0)$ does not depend on whether it is DAG-like or tree-like or on the choice of the underlying Frege proof system.

Returning to the connections with weak arithmetic, these theories capture various complexity classes in the sense that, roughly speaking, all functions provably total in such a theory T are precisely the functions from that class. Total provability of a function $f(x)$ means that it is representable by a formula $A(x, y)$ such that T proves[1] $(\exists! y \leq t)A(x, y)$ and $A(n, f(n))$ is true for any n. It involves the bounded existential quantifier $(\exists y \leq t)$ in front. It turns out that if we are interested in the provability, in the same theory T, of "almost" quantifier-free formulas (for experts, Δ_0^b formulas), then such formulas can be translated into an increasing sequence $\{\tau_n\}$ of propositional formulas. Then the provability of the original statement in T becomes "essentially equivalent" to the *efficient* provability of its propositional translation in a proof system P_T naturally associated with T. In most cases, it simply means that $S_{P_T}(\tau_n \vdash 0)$ is bounded by a polynomial in n, and F and EF happen to correspond to the most central systems of weak arithmetic. For more details see the monographs [21, 24, 38, 39] already cited above.

Showing that F or EF are not p-bounded is widely believed to be out of reach of the current methods and in general even more difficult than solving notorious open problems in the computational complexity like $\mathsf{NC}^1 \neq \mathsf{P}$ or $\mathsf{P} \neq \mathsf{NP}$. They are paradigmatic strong systems in our informal classification. A good explanation, both philosophical and heuristical, predicates that the most important feature of a proof system P is the *expressive* (in the computational sense of the word) *power of its lines*, that is what computational power is afforded to concepts underlying auxiliary statements appearing in the proof. For a Frege proof system lines are just arbitrary Boolean expressions, and they correspond to the complexity class NC^1. For the extended Frege we get arbitrary Boolean circuits, and those correspond to the class P. It appears to be even more difficult, and usually way more difficult, to analyze what one can *prove* using concepts definable by a complexity class than what we can *compute* within this

[1] The exclamation mark stands for "unique."

class. I am not aware of any good explanation of this fact, this is just what has been happening in the area so far.

The final observation I would like to offer about F and EF strongly differentiates the propositional proof complexity from its sister discipline, circuit complexity. Let me remind the reader that in the latter field we know that almost all Boolean functions are hard; this is the famous *Shannon effect* (see e.g. [36, Chapter 1.4]). Moreover, we strongly believe that a variety of very natural Boolean functions corresponding to NP-complete problems are hard. That is, we have a host of *natural* and *explicit* candidates for hardness; we simply do not know yet how to prove that they are actually hard.

Nothing like that happens in proof complexity, and potential candidates are few and far between. In [16], Bonet, Buss, and Pitassi set off for a slightly modified task to find good tautologies *separating* F and EF, that is hard for F, easy for EF. Their own conclusion, to which I fully concur, was that "no particularly good or convincing examples are known." If we relax the requirement and simply ask for tautologies that would be good candidates to show that the Frege proof system is not p-bounded, I believe there are only two principles that have passed the test of time even by loose standards, and both are equally plausible to be hard for EF.

The first is random k-CNFs. Pick up sufficiently many clauses of width k at random. Then the resulting CNF will be in UNSAT w.h.p. but there does not appear to be even a good starting point for F or EF (or, for that matter, any other conceivable proof system) to certify the unsatisfiability in particular instances.

The second kind of examples is made by CNFs expressing facts like "NP does not have small size circuits." For an extensive discussion of these statements and their relations to other topics in proof and computational complexities I refer the reader to [59, Section 1].

All proof systems in the remainder of this article will be weak ("potentially" weak in one case).

3. Benchmarks

In computer science, a "benchmark" usually stands for a "good" standardized test, or a family of tests, used to run competing pieces of software or hardware to compare these pieces to each other. In the propositional proof complexity, it also turns out that there is a handful of combinatorial principles, expressible as unsatisfiable CNFs, that wander from one framework to another and appear in papers over and over again. This uniformity turns out indispensable for understanding the general picture and trying out new methods for proving both lower and upper bounds that can be then applied to many other tautologies.

For now, let us define two such principles that, arguably, are the most prominent and popular ones (we will see a few more later in the text).

Definition 3.1 (pigeonhole principle). Let $m > n$ be integers; introduce propositional variables x_{ij} ($i \in [m]$, $j \in [n]$). The *pigeonhole principle* (sometimes also called the *Dirichlet principle*, particularly in the Russian literature) is the unsatisfiable CNF PHP_n^m made of the following clauses:

- $x_{i1} \vee \cdots \vee x_{in}$, for all "pigeons" $i \in [m]$ ("every pigeon flies to a hole");
- $\bar{x}_{ij} \vee \bar{x}_{i'j}$, for all pairs of different "pigeons" $i \neq i' \in [m]$ and all "holes" $j \in [n]$ ("no two pigeons fly to the same hole").

This is the so-called "basic" pigeonhole principle. One can also add to it dual axioms, the *functionality axioms* $\bar{x}_{ij} \vee \bar{x}_{ij'}$ or the *surjectivity axioms* $x_{1j} \vee \cdots x_{mj}$. Varying the parameter $m = m(n)$ as well, we obtain a large family of pigeonhole principles and, somewhat surprisingly, they may display very different behavior with respect to the same proof system. I refer the reader to the survey [57] *entirely* devoted to the pigeonhole principle, with the warning that several important results have been obtained since its release.

Our second principle was introduced in [62] that, arguably, was the earliest paper in the propositional proof complexity.

Definition 3.2 (Tseitin tautologies). Let $G = (V, E)$ be a simple graph with odd number of vertices. Introduce propositional variables x_e, one variable per edge $e \in E$. The *Tseitin tautology* $\mathrm{Tseitin}(G)$ is the following system of linear equations over $\mathbb{F}_2$:

$$\bigoplus_{e \ni v} x_e = 1 \quad (v \in V)$$

($\oplus$ is the *parity function*, addition mod 2). This principle says that in any spanning sub-graph of G (determined by the values $(x_e \mid e \in E)$) there exists a vertex of even degree.

Remark 3.3. The attentive reader may have observed that, as stated, $\mathrm{Tseitin}(G)$ is not a CNF. It is usually converted into a CNF by straightforwardly expanding all parities into a family of clauses. For example, $x \oplus y \oplus z$ is the same as $(x \vee y \vee z) \wedge (\bar{x} \vee \bar{y} \vee z) \wedge (\bar{x} \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee \bar{z})$. This expansion incurs an increase in the size of the contradiction by a factor of $2^{\Delta-1}$, where Δ is the maximal vertex degree of G. This is often unacceptable when Δ is large so in most applications Tseitin tautologies are considered only for constant-degree graphs that are also sometimes assumed to be regular (all vertices have the same degree).

It turns out that Tseitin tautologies work best when G is a good expander. There are several standard definitions of graph expansion, very much equivalent in the bounded-degree case. Here we only recall that of *edge* expansion, as the most convenient for our purposes.

Definition 3.4 (edge expansion). For a graph $G = (V, E)$ and $S \subseteq V$, let $E(S, \overline{S})$ be the set of all cross-edges between S and $\overline{S} \stackrel{\text{def}}{=} V \setminus S$. The (*edge*) *expansion* $c(G)$ of G is defined as

$$c(G) \stackrel{\text{def}}{=} \min\left\{\frac{|E(S, \overline{S})|}{|S|} \mid S \subseteq V,\ 1 \le |S| \le |V|/2\right\}.$$

4. Bounded-depth Frege

In this section, we will discuss several restrictions of the Frege proof system to which Theorem 2.3 no longer applies. On the other hand, the remark from Section 2 (that a proof system is largely determined by the expressive power of its lines) applies in full, and a bounded-depth Frege proof system is determined by the bound on depth and the set of propositional connectives (the *basis*) it employs.

Let us start with the standard *de Morgan basis* $\{\neg, \vee, \wedge\}$. The first useful observation is that using de Morgan rules $\neg(A \vee B) \equiv (\neg A \wedge \neg B)$, $\neg(A \wedge B) \equiv (\neg A \vee \neg B)$, any formula can be converted into a formula with *tight negations*, that is a formula in which negations occur only at the variables.

Definition 4.1 (bounded-depth Frege). The *logical depth* of a $\{\neg, \vee, \wedge\}$-formula with tight negations is the maximum number of *alternations* $\vee\vee\cdots\wedge\wedge\wedge\cdots\vee$ of $\vee$ and $\wedge$, where the maximum is taken over all paths from the root of the formula to its leaves (i.e., literals). Alternatively, we can allow disjunctions and conjunctions with an arbitrary number of arguments, and then logical depth becomes the ordinary depth (= height) of the tree representing the formula.

The *depth-d* Frege proof system F_d is the fragment of a Frege proof system over $\{\neg, \wedge, \vee\}$ in which all lines are required to have logical depth $\le d$.

As in Definition 2.1, we do not specify axiom schemes and inference rules since all "reasonable" choices lead to p-equivalent systems. For most of this section, we view the depth d as arbitrarily large but fixed constant; this is what we mean by "bounded depth."

The corresponding circuit class, made of sequences of Boolean functions that can be computed by circuits of polynomial size and bounded depth, is well known in circuit complexity. It is denoted by AC^0 and by now it is relatively well understood, beginning with exponential size lower bounds for bounded-depth circuits proved in the celebrated series of papers [1, 34, 63].

While lower bounds for F_d were established with the same general method (so-called *restrictions*), this required to overcome a great deal of additional difficulties as compared to the case of circuits. But before we start discussing concrete results I find it prudent to make the following disclaimer.

This short article is not intended to be a comprehensive survey in the propositional proof complexity or its sub-areas; for more extended account, see e.g. the monograph [39] and historical remarks made therein. Its purpose is limited to giving the first impression about the area to non-specialists, and my choice of illustrating examples is necessarily incomplete and subjective.

That said, the first lower bounds for bounded-depth Frege were proved for the pigeonhole principle.

Theorem 4.2 ([41, 48]). $S_{\mathrm{F}_d}(\mathrm{PHP}_n^{n+1} \vdash 0) \geq \exp(\Omega(n^{1/5^d}))$.

Here, and in what follows, "Ω" is the notation dual to "big-O": $f \geq \Omega(g)$ means that there exists an absolute constant $\varepsilon > 0$ such that $f \geq \varepsilon g$ for all values of the parameters appearing in f, g.

Corollary 4.3. *For any fixed $d > 0$, F_d is not p-bounded.*

To illustrate one point made in Section 3, let us note that once we increase the number of pigeons to $2n$, the situation changes dramatically.

Theorem 4.4 ([5, 42]). $S_{\mathrm{F}_d}(\mathrm{PHP}_n^{2n} \vdash 0) \leq n^{(\log n)^{O(1/d)}}$. *For $d = 2$, this refines as* $S_{\mathrm{F}_2}(\mathrm{PHP}_n^{2n} \vdash 0) \leq n^{O(\log n)}$.

Whether this can be improved to polynomial, perhaps at the expense of using more pigeons, is open despite decades of research.

Problem 4.5. Does there exist a fixed $d > 0$ such that $S_{\mathrm{F}_d}(\mathrm{PHP}_n^{\infty} \vdash 0) \leq n^{O(1)}$?

As we noted above, once a method to analyze a proof system (in particular, to prove lower bounds for it) is established, it usually can be extended to other contradictions as well. As an illustration, the following was proved by a *direct* (albeit, very clever) reduction from Theorem 4.2.

Theorem 4.6 ([13]). *Let $\{G_n\}$ be a sequence of bounded-degree graphs with $c(G_n) \geq \Omega(1)$. Then for any fixed $d > 0$,* $S_{\mathrm{F}_d}(\mathrm{Tseitin}(G_n) \vdash 0) \geq \exp(\Omega(n^{1/5^d}))$.

But sometimes the next improvement/generalization requires a very serious enhancement of known techniques. Let us for example reverse the gears and instead of asking about size lower bounds in any *fixed* depth, ask what is the *largest* depth, as a function of the number of variables n, for which the bound still holds.

The bounds in Theorems 4.2 and 4.6 work up to $d = \varepsilon \log\log n$. It was recently improved to $d = o(\sqrt{\log n})$ in [50]. While this is still the same basic method of restrictions the previous work was based upon, this improvement literally had to take it to a new level of sophistication.

Theorem 4.7 ([50]). *For $\{G_n\}$ as in Theorem 4.6,*

$$S_{\mathrm{F}_d}\big(\mathrm{Tseitin}(G_n) \vdash 0\big) \geq n^{\Omega((\log n)/d^2)}.$$

Note that unlike Theorem 4.6, this bound is only quasi-polynomial. But it is good enough to prove that $\mathrm{F}_{d(n)}$ is not p-bounded when $d(n) = o(\sqrt{\log n})$.

In conclusion of this section, let us briefly discuss one extension.

Definition 4.8 (bounded-depth Frege with modular gates, informal). Let $m > 0$ be a fixed integer and $\mathrm{MOD}_m(x_1, \ldots, x_n)$ the propositional connective with the intended meaning $\mathrm{MOD}_m(x_1, \ldots, x_n) = 1$ iff $m | x_1 + \cdots + x_n$. Let $\mathrm{F}(\mathrm{MOD}_m)$ be a Frege system (p-equivalent to F) in the language $\{\neg, \wedge, \vee, \mathrm{MOD}_m\}$. The proof system $\mathrm{F}_d(\mathrm{MOD}_m)$ is its fragment in which the logical depth of all formulas is restricted to d, where axioms schemes and inference rules are chosen in any reasonable way (in particular, they should describe basic properties of the new connectives).

For some inspiration of what might be expected from this extension, we have to look again into the circuit complexity. The corresponding complexity class is denoted by $\mathsf{ACC}^0[m]$, and it turns out that the story crucially depends on m.

When m is a prime power, exponential lower bounds for this class of circuits have been known since [54,61]. In all other cases (say, when $m = 6$) this is one of the most major and challenging open problems in circuit complexity: for all we know, $\mathsf{ACC}^0[6]$ may contain all of NP or, for that matter, $\mathsf{EXPTIME}$. For details, see e.g. [36, Chapter 12].

Accordingly, when m has at least two different prime divisors, $\mathrm{F}_d(\mathrm{MOD}_m)$ should definitely be classified as "strong." Somewhat embarrassingly, we have not been able to adapt the proofs from [54, 61] (based on the so-called *method of approximations*) to our context so far. The following is one of the main open problems in the area.

Problem 4.9. Prove that for any fixed $d > 0$ and any fixed prime $m > 0$ the system $\mathrm{F}_d(\mathrm{MOD}_m)$ is not p-bounded.

The only known partial results towards this problem pertain to its much weaker subsystems; we will now briefly mention one of them and another will appear in Section 6.1.

Definition 4.10 (counting principles). Let $m \nmid n$, and introduce propositional variables x_e, where $e \in \binom{[n]}{m}$, the family of all m-element subsets of $[n] \overset{\mathrm{def}}{=} \{1, 2, \ldots, n\}$. The *counting principle* Count_m^n is the unsatisfiable CNF consisting of the following clauses:

- $\bar{x}_e \vee \bar{x}_f$, for all $e \neq f$ such that $e \cap f \neq \emptyset$;
- $\bigvee_{e \ni i} x_e$, for all $i \in [n]$.

Intuitively, these clauses state that $(x_e \mid e \in \binom{[n]}{m})$ defines a partition of $[n]$ into sets of size m which may not exist since we assumed $m \nmid n$. The proof system $\mathrm{F}_d + \mathrm{Count}_m$ is obtained from F_d by adding to it all substitutional (de Morgan!) instances of Count_m^n, for arbitrary n, that are of logical depth $\leq d$.

The principle Count_m^n is easily provable in $\mathrm{F}_d(\mathrm{MOD}_m)$, hence $\mathrm{F}_d + \mathrm{Count}_m$ is indeed intermediate between F_d and $\mathrm{F}_d(\mathrm{MOD}_m)$ in the sense of Definition 1.7.

Theorem 4.11 ([11,18]). *Let m, d, ℓ be fixed integers and assume that ℓ has a prime factor which is not a prime factor of m. Then $S_{\mathrm{F}_d+\mathrm{Count}_m}(\mathrm{Count}_\ell^n \vdash 0) \geq \exp(n^{\Omega(1)})$.*

Note, however, that this result holds for all m including, say, $m = 6$. This might be not so good sign for attempts to adapt these methods for solving Problem 4.9.

5. Resolution

In our notation, resolution is simply F_1. It obviously does not make much sense to consider *terms* $x_{i_1}^{a_1} \wedge \cdots \wedge x_{i_w}^{a_w}$ as lines in a proof, they can be always split into w lines consisting of single literals. Hence resolution uses clauses only and, given the importance of this proof system (that we will try to explain below), we prefer to break up with our own tradition and formulate its inference rules (there are no default axioms) very explicitly.

Definition 5.1 (resolution). Resoluton is the proof system operating with clauses, denoted by R. It has the inference rules

$$\frac{C}{C \vee D}\ (\text{weakening}) \qquad \frac{C \vee x \qquad D \vee \bar{x}}{C \vee D}\ (\text{resolution rule}).$$

A resolution proof is *regular* if, on any path in this proof, no variable x is resolved more than once. We will denote this subsystem of resolution by RR.

Remark 5.2. Resolution, as well as most systems we will see in the rest of this article, is too weak to speak of CNFs directly. It is therefore paramount (cf. Remark 1.2) that from now on we strictly adopt the "refutational" perspective: all "proofs" will be actually contradictions derived from a set of clauses.

Remark 5.3. The weakening rule is cosmetic and its removal does not change the complexity $S_R(\tau \vdash 0)$. Having this rule, however, is very convenient in many situations.

Resolution, as well as other proof systems that we will see below, is very relevant to various scenarios with practical flavor. The paradigm is somewhat similar in all these cases; let us spell it out for resolution in a few more details. Much more information on the topic, as well as all definitions missing in our description below, can be found e.g. in the very recent survey [19].

There is a large community of practice-oriented researchers working on finding feasible algorithms (which in this context means "actually implemented and delivering concrete results") for solving "interesting" instances of SATISFIABILITY. These programs are called *SAT solvers*. Now, what will happen if we feed a CNF τ to a SAT solver, it runs successfully and produces the correct answer?

When τ is satisfiable, in most cases the solver will be able to justify its answer by producing an actual satisfying assignment. But this case is not very inspiring for our purposes.

More interesting is the case when τ is unsatisfiable because if we understand the code and believe in its correctness, then we also must accept the *transcript of the solver's run* as a *proof* of unsatisfiability of τ. In mathematical terms, any practical scalable algorithm for solving SATISFIABILITY defines a propositional proof system in terms of Definition 1.4.

It turns out that in many scenarios the proof systems automatically associated in this way to algorithms are also mathematically elegant, and it is particularly visible in the case of SAT solvers. Namely, the algorithmic technique that has been dominating in that community for quite a while is called *conflict-driven clause learning* (CDCL). Then, a transcript of a run of a CDCL solver can be *identified* with a resolution proof, modulo a differing terminology. This connection is in fact so strong that it would not be too much of an exaggeration to describe the operation of CDCL solvers in this way: they search, in very ingenuous and specific ways, for resolution refutations of a CNF τ and declare it satisfiable if the search fails.

Thus, any *lower* bounds for the resolution proof system imply *inherent* limitations on CDCL solvers that cannot be overcome by any amount of clever engineering. They can also be used as a rough guidance of what to expect and what to avoid when building CDCL solvers.

An extremely interesting question is whether there is a connection in the opposite direction; that is, what algorithmic applications does the mere *existence* of a short resolution proof entail?

When the word "algorithmic" is understood in its most theoretical sense (that is, poly-time computable), this question is captured by the concept of "automatizability" (or "automation"), and we have recently seen a major progress in this direction [8] followed up in several other papers. Very loosely speaking, if $\mathsf{P} \neq \mathsf{NP}$, then no efficient algorithm will be able to find small resolution refutations in all cases when they exist, ever.

Another meaningful interpretation is to consider only algorithms based on the CDCL-architecture but allow them a limited amount of non-determinism in the choices they make. It turns out that this question is very sensitive to the choice of the model and, in my view, it is far from being answered conclusively. Some partial work in that direction is reported e.g. in [7, 12, 44]; once again, much more information can be found in [19].

Let us now return to mathematics, and we begin with several early prominent results.

Theorem 5.4 ([62]). $S_{\mathrm{RR}}(\mathrm{Tseitin}(\mathrm{Grid}_{n,n}) \vdash 0) \geq \exp(\Omega(n))$, *where* $\mathrm{Grid}_{n,n}$ *is the* $n \times n$ *grid graph.*

Theorem 5.5 ([33]). $S_{\mathrm{R}}(\mathrm{PHP}_n^{n+1} \vdash 0) \geq \exp(\Omega(n))$.

Theorem 5.6 ([22]). *Let* τ_n *be a random* 3*-CNF with* $O(n)$ *clauses. Then with probability* $1 - o(1)$ *we have* $S_{\mathrm{R}}(\tau_n \vdash 0) \geq \exp(\Omega(n))$.

As we already mentioned several times, it is highly desirable to have reasonably general methods for analyzing proof complexity, as opposed to those that are tailored to individual benchmarks. In that respect, the following prominent *width-size relation* clearly stands out.

Given a resolution refutation, its *width* is defined as the maximum width of its clauses, and let $w(\tau_n \vdash 0)$ be the minimum possible width of a resolution refutation of τ_n. In other words, we are trying to refute τ_n using only narrow clauses as our "lemmas," disregarding the question of how many of them we use. Then the width-size relation due to Ben–Sasson and Wigderson has the following neat and general form.

Theorem 5.7 ([15]). *For* any *sequence* $\{\tau_n\}$ *of unsatisfiable CNFs,*

$$w(\tau_n \geq 0) \leq O\left(\sqrt{n \cdot \log S_R(\tau_n \vdash 0)} + w_0\right),$$

where w_0 *is the width of* τ_n *itself.*

Parsing this expression, when w_0 is small (say, a constant) and $w(\tau_n \vdash 0) \geq \Omega(n)$, we get $S_R(\tau_n \vdash 0) \geq \exp(\Omega(n))$. In words, *linear* lower bounds on width imply *exponential* lower bounds on the resolution size.

And it turns out that width lower bounds are often much easier to prove. For example, we have the following (cf. Definition 3.4).

Theorem 5.8 ([15]). *For any sequence of bounded-degree graphs* $\{G_n\}$ *with* $c(G_n) \geq \Omega(1)$, $w(\mathrm{Tseitin}(G_n) \vdash 0) \geq \Omega(n)$ *(and hence by Theorem 5.7,* $S_R(\mathrm{Tseitin}(G_n) \vdash 0) \geq \exp(\Omega(n))$*).*

This recovers a stronger version of Theorem 4.6 for $d = 1$ but, again, the main strength of Theorem 5.7 lies in its generality. Two more important points highlighted by the width-size relation that have turned out very influential in proof complexity (we will see some examples below) are as follows.

(1) Diversity is good. Proof complexity measures more elaborated than the one stipulated by Definition 1.5 are inspiring *even* if one is primarily interested in size.

(2) Expansion is good as well. If a graph property imply hardness in the proof complexity, the odds are that expansion will also do the job.

The width-size relation can be successfully applied to an impressive array of various contradictions τ_n, often after some massaging. But, as is the case with any good method, it has its limitations. One notable principle it completely fails at is the pigeonhole principle with many (say, infinitely many) pigeons, which is the special case of Problem 4.5 for $d = 1$. For that, another technique of *pseudo-width* was developed in [49, 53, 58]. Unfortunately, this concept is a bit too technical to meaningfully address here, so let us simply state the end result for PHP[2].

Theorem 5.9. $S_{\mathrm{R}}(\mathrm{PHP}_n^\infty \vdash 0) \geq \exp(\Omega(n^{1/3}))$.

Remark 5.10. Surprisingly, the best known *upper* bound here is not the trivial $\exp(O(n))$ but $\exp(O(n^{1/2}))$ [20]. That would be nice to close the gap, particularly since most likely this will require developing new methods.

Problem 5.11. Determine the smallest $\alpha \in [1/3, 1/2]$ for which $S_{\mathrm{R}}(\mathrm{PHP}_n^\infty \vdash 0) \leq \exp(n^{\alpha+o(1)})$.

Among other things, Theorem 5.9 implies resolution lower bounds for the statement “NP does not have small size circuits” mentioned at the end of Section 2; see again [59], as well as [52], for more details and the context. The former paper also extends this to the proof system $\mathrm{Res}(O(1))$ operating with $O(1)$-CNFs but the proof is very indirect and complicated. On the other hand, Problem 4.5 remains wide open even for the system (say) $\mathrm{Res}(2)$ intermediate between F_1 and F_2. Moreover, now the upper bound of Theorem 4.4 no longer applies and we can state this conjecture in the stronger form.

Problem 5.12. Prove (or disprove) that $S_{\mathrm{Res}(2)}(\mathrm{PHP}_n^\infty \vdash 0) \geq \exp(n^{\Omega(1)})$.

More applications of the pseudo-width method can be found in the recent paper [28].

Are there prominent unsatisfiable CNFs that (in terms of their resolution complexity) resist analysis by both the width-size and pseudo-width methods? Let me conclude this section with my favorite example, the *small clique problem*.

[2] The last paper in the series [58] generalized the method to a much wider class of general *perfect matching principles* including, among others, the counting principles from Definition 4.10.

Definition 5.13. Let G be a *k-partite graph*, that is its vertices can be partitioned into k blocks, $V(G) = V_1 \dot{\cup} \cdots \dot{\cup} V_k$ (let us also assume that $|V_1| = \cdots = |V_k|$) such that there is no edge within each block. The CNF $\mathrm{Clique}_{\mathrm{Block}}(G, k)$ is defined as the following set of clauses in the variables $(x_v \mid v \in V(G))$:

- $\bigvee_{v \in V_i} x_v$ $(1 \leq i \leq k)$;
- $\bar{x}_v \wedge \bar{x}_w$ $((v, w)$ is *not* an edge of $G)$.

This CNF says that $(x_v \mid v \in V(G))$ encodes a k-clique in G and when the clique number $\omega(G)$ is at most $(k-1)$, this is a contradiction.

The obvious brute-force resolution refutation has size at most n^k, and the question is whether we can do any better. Motivated by the framework of parameterized (computational) complexity [29] and some research in circuit complexity, it is natural to ask about the existence of resolution refutations of size $f(k) \cdot n^{O(1)}$, where $f(k)$ is *any* function. Assuming that k is a fixed constant, the first term disappears and the question is whether $S_{\mathrm{R}}(\mathrm{Clique}_{\mathrm{Block}}(G, k) \vdash 0) \leq n^{O(1)}$, where the degree of the polynomial in the right-hand side must not depend on k.

The small clique problem is usually considered when G is the *Erdös–Renyi random graph*, that is when every potential edge between $v \in V_i$ and $w \in V_j$ is included i.i.d. with probability $p_{kn} > 0$, $n \overset{\mathrm{def}}{=} |V_i|$. Let us fix for definiteness

$$p_{kn} \overset{\mathrm{def}}{=} n^{-C/(k-1)}, \tag{5.1}$$

where $C > 2$ is an arbitrary constant, and let $\mathbf{G_{k,n}}$ be the corresponding Erdös–Renyi graph. The value (5.1) is a (weak) *threshold value*; it guarantees that the probability of the event $\omega(\mathbf{G_{k,n}}) = k$ is bounded away from both 0 and 1.

Theorem 5.14 ([6]). *For $k \leq n^{1/4} - \Omega(1)$, with probability $1 - o(1)$ we have*

$$S_{\mathrm{RR}}\big(\mathrm{Clique}_{\mathrm{Block}}(\mathbf{G_{k,n}}) \vdash 0\big) \geq n^{\Omega(k)}.$$

Problem 5.15. Prove that for any fixed $k > 0$, $S_{\mathrm{R}}(\mathrm{Clique}_{\mathrm{Block}}(\mathbf{G_{k,n}}) \vdash 0) \geq n^2$.

6. Algebraic and semi-algebraic proof systems

When you say 0 and 1, it is only mathematical logicians and computer scientists whose first association would be FALSE and TRUE. For anyone else, these are distinguished elements of a ring with particular algebraic (or semi-algebraic if the ring is ordered) properties. In this section, we will review, very briefly, a prominent family of proof systems heavily adopting this latter point of view and entirely abstracting from the logical interpretation of the statements they are proving. Besides [39, Chapter 16], the foundational material for this section, as well as a taxonomy of these proof systems, can be found in the early paper [32].

The first thing to decide is how exactly we are going to translate logic to algebra/geometry, and we should start with encoding clauses. There are essentially two different ways of doing it, and this choice largely determines what kind of proof systems we are aiming at.

The first possibility is to encode clauses by polynomial *equations* over a ground field $\mathbb{F}$. This is done in a very straightforward way; for example, the clause $C = x_1 \vee \bar{x}_2 \vee x_3$ is encoded as the equation $(1 - x_1)x_2(1 - x_3) = 0$.

For the second option we must assume that our ground field $\mathbb{F}$ is ordered, say $\mathbb{F} = \mathbb{Q}$ or $\mathbb{F} = \mathbb{R}$. In that case, we can encode clauses by linear *inequalities*. For example, $C = x_1 \vee \bar{x}_2 \vee x_3$ will be translated as $x_1 + (1 - x_2) + x_3 \geq 1$ that can be further simplified to $x_1 + x_3 \geq x_2$, if desired.

In either case, the original CNF is unsatisfiable if and only if the algebraic/semi-algebraic set defined by the corresponding system of polynomial equations/inequalities over $\mathbb{F}$ does not have 0-1 solutions. This reformulation allows us to employ tools from algebra/geometry, and we now treat the two cases separately.

6.1. Algebraic models

If we are allowed to use non-linear polynomials, the assumption that we are interested only in 0-1 solutions can be hardwired into the framework by introducing the default axioms $x_i^2 - x_i = 0$. It turns out to be very handy, albeit not strictly necessary, to factor out these relations at once and work in the $\mathbb{F}$-algebra

$$\Lambda_n \stackrel{\text{def}}{=} \mathbb{F}[x_1, \ldots, x_n]/(x_i^2 - x_i \mid 1 \leq i \leq n).$$

This algebra was introduced to complexity theory (apparently) in [54, 61]; it consists of all *multi-linear* polynomials and hence has linear dimension 2^n. On the other hand, it is isomorphic to the algebra of all functions $\{0,1\}^n \to \mathbb{F}$; $\mathrm{Hom}(\Lambda_n, \mathbb{F})$ is the set of all Boolean assignments to the variables $x_1, \ldots, x_n$ etc.

Hilbert's Nullstellensatz tells us that a polynomial system $f_1(x_1, \ldots, x_n) = \cdots = f_m(x_1, \ldots, x_n) = 0$ ($f_i \in \Lambda_n$) does not have 0-1 solutions if and only if there exist $Q_1, Q_2, \ldots, Q_m \in \Lambda_n$ such that

$$f_1 Q_1 + f_2 Q_2 + \cdots + f_m Q_m = 1. \tag{6.1}$$

Every such system of polynomials $(Q_1, \ldots, Q_m) \in \Lambda_n$ can thus be considered as a *proof* of the statement that the algebraic set $(f_1 = 0, \ldots, f_m = 0)$ does not contain 0-1 points. This proof system is called the *Nullstellensatz* proof system (over the field $\mathbb{F}$).

Remark 6.1. The question whether this system formally fits Definition 1.4 is slightly non-trivial. It may depend on the way the polynomials are represented, on their coefficients etc. We prefer not to dwell into these details as it has become much more customary (and it is way more clean mathematically, too) to measure the complexity of the proof $(Q_1, \dots, Q_m)$ by its *degree* defined as $\max_{1 \leq i \leq m}(\deg(Q_i) + \deg(f_i))$.

By now, the Nullstellensatz proof system is fairly well understood. But since most results proved for it have been eventually generalized (and sometimes strengthened) to a stronger system that we will consider next, let us confine ourselves to just one prominent example.

Theorem 6.2 ([10]). *Every Nullstellensatz refutation of* PHP_n^∞ *must have degree* $\Omega(\sqrt{n})$.

Remark 6.3. Both for this result and those below, Definition 2.1 should be slightly adjusted. Namely, to avoid polynomials of prohibitively high degree, the pigeon axioms $x_{i1} \vee \dots \vee x_{in}$ should be translated as $x_{i1} + \dots + x_{in} - 1 = 0$ (note that this also implies that the funcionality axioms $\bar{x}_{ij_1} \vee \bar{x}_{ij_2}$ $(j_1 \neq j_2 \in [n])$ are also implicitly included).

The polynomial calculus (PC) is a *dynamic* version of this system in which we attempt to prove that 1 is in the ideal $(f_1, \dots, f_m) \subseteq \Lambda_n$ by generating its elements one by one instead of writing down a single expression like (6.1).

Definition 6.4. *Polynomial calculus* (over a ground field $\mathbb{F}$) is the algebraic proof system whose lines are elements of Λ_n. It has the following inference rules:

$$\frac{f = 0 \qquad g = 0}{\alpha f + \beta g = 0};\ \alpha, \beta \in \mathbb{F} \text{ (addition rule)}, \qquad \frac{f = 0}{fg = 0} \text{ (multiplication rule)}.$$

The *degree* of a PC proof is the maximum degree of its lines.

Remark 6.5. The main source of non-triviality of this system stems from the fact that at every step we completely expand the result as a sum of terms. When doing this, cancelations may (and typically do) lead to a substantial decrease in degree. On the other hand, there is a degree-size relation for the PC perfectly analogous to Theorem 5.7 (and actually proved earlier in [23]).

Remark 6.6. It is not very hard to see that every PC proof over $\mathbb{F}_p$ can be p-simulated by $F_2(\mathrm{MOD}_p)$. Thus, polynomial calculus over a finite field can be reasonably viewed as an "algebraic" component of $F_2(\mathrm{MOD}_p)$ while F_2 is its logical part. The main reason why Problem 4.9 appears to be so difficult is that the existing methods for understanding these two parts seem to be totally disjoint from each other.

There has been a fair amount of work attempting to build actual SAT solvers based upon algebraic principles, primarily the Gröbner basis algorithm. These solvers relate to the PC in precisely the same way CDCL-based solvers are related to resolution, cf. our discussion in Section 5. It would be fair to say that so far they have not been competitive with CDCL solvers but there does not seem to exist any good theoretical explanation of this fact. So perhaps the true potential of algebraic SAT solvers is yet to be revealed; we refer the reader to [19, Section 7.5.7] for more details.

As usual, we conclude with a few sample results. Historically the first lower bound for the PC generalized and strengthened Theorem 6.2.

Theorem 6.7 ([56]). *Every polynomial calculus refutation of* PHP_n^∞ *must have degree* $\Omega(n)$.

This also implies PC degree lower bounds for the statement "NP does not have small size circuits" we already mentioned several times before.

The proof method of Theorem 6.7 is rather ad hoc, it is based on the so-called "pigeon dance" specifically designed for the purpose. The next paper [17] introduced a very nice and remarkably simple method of analyzing PC refutations from *binomial*[3] axioms. Here is one concrete application that strengthens Theorem 5.8.

Theorem 6.8 ([17]). *For any sequence of bounded-degree graphs* $\{G_n\}$ *with* $c(G_n) \geq \Omega(1)$, *every polynomial calculus refutation of* $\mathrm{Tseitin}(G_n)$ *over any field of* odd or zero *characteristic must have degree* $\Omega(n)$.

The extension to random 3-CNFs, with the same restriction on the ground field $\mathbb{F}$, is not very difficult [14]. But the binomial method completely breaks down for $\mathbb{F} = \mathbb{F}_2$ which is one of the most interesting cases. Another method for proving PC degree lower bounds over an arbitrary field based on a general hardness criterion was proposed in [2]; see also [43] and the literature cited therein for more recent developments.

Theorem 6.9 ([2, 14]). *Let* τ_n *be a random* 3*-CNF with* $O(n)$ *clauses. Then any polynomial calculus refutation of* τ_n *over an* arbitrary *field* $\mathbb{F}$ *must have degree* $\Omega(n)$.

Let us finally note that the degree-size relation mentioned above immediately implies exponential *size* lower bounds for PC refutations in Theorems 6.8 and 6.9.

6.2. Semi-algebraic case

There are many prominent semi-algebraic proof systems: Sum-of-Squares, Cutting Planes, Lovász–Schrijver, Sherali–Adams to name a few. We will only touch, very briefly, on the first two; for a nicely organized exposition see [32]. Throughout this section we assume that $\mathbb{F} = \mathbb{Q}$ or $\mathbb{F} = \mathbb{R}$.

The Sum-of-Squares is also known under the name Positivestellensatz and is closely related to the so-called Lassierre hierarchy. There are several slight variations in its definition, we only present here (as many other authors do) the simplest version in which the original axioms are given as polynomial equations, like in Section 6.1.

[3] In the Rademacher $\{\pm 1\}$ framework.

Definition 6.10. An *SOS* (or *Positivestellensatz*) refutation of a polynomial system $(f_1 = \cdots = f_m = 0)$ $(f_i \in \Lambda_n)$ is a family of polynomials $(Q_1, \ldots, Q_m, g_1, \ldots, g_t)$ in Λ_n such that

$$f_1 Q_1 + \cdots + f_m Q_m + \sum_{j=1}^{t} g_j^2 = -1. \tag{6.2}$$

Its *degree* is defined as $\max(\max_{1 \le i \le m} \deg(f_i) + \deg(Q_i),\ 2 \max_{1 \le j \le t} \deg(g_j))$.

The corresponding algorithmic technique has in recent years become extremely important in combinatorial optimization and approximation algorithms, largely due to the fact that it has turned out unexpectedly powerful. We refer the reader to the expository paper [9] although a great deal of important work has been done since that. The relation between combinatorial optimization and proof complexity follows the familiar pattern, and in fact in this case it is even more transparent. But one important difference is that unlike SAT solvers, algorithms in combinatorial optimization seldom output the exact answer but only an optimistic approximation to it which in most cases means relaxing the integrality constraints $x_i \in \{0, 1\}$ to $x_i \in [0, 1]$. In any case, the computation implies that one cannot beat the value of the goal function delivered by this relaxation, and then after a straightforward application of the PSD duality, it becomes an SOS proof in the sense of Definition 6.10. See again [9] for more details.

As for degree lower bounds, SOS is also relatively well understood although some important problems still remain open. The first lower bound had been proven by Grigoriev [31] and largely forgotten until the realization of the algorithmic significance of the SOS method came. This is the same binomial method we saw in Section 6.1, wisely put to a different use.

Theorem 6.11 ([31]). *Every SOS refutation of* $\mathrm{Tseitin}(G_n)$*, where* $\{G_n\}$ *is a sequence of bounded-degree graphs with* $c(G_n) \ge \Omega(1)$*, must have degree* $\Omega(n)$.

More modern methods of handling SOS proofs are based upon the concept of a *pseudoexpectation* which is essentially an object dual to the expression (6.2) (therefore, it exists if and only if the system (6.2) is *not* solvable in Q_i, g_j of given degree).

The last system we discuss is Cutting Planes.

Definition 6.12. *Cutting Planes* is the proof system operating with affine inequalities, denoted by PC. It has default axioms $x \ge 0$ and $x \le 1$ for all variables x, as well as the following inference rules:

$$\frac{f \ge 0 \qquad g \ge 0}{\alpha f + \beta g \ge 0}; \quad \alpha, \beta \ge 0 \text{ (convex closure)},$$

$$\frac{f \ge a}{f \ge \lceil a \rceil}; \quad f \in \mathbb{Z}[x_1, \ldots, x_n] \text{ (cut rule)}.$$

To explain this terminology, it is convenient to adapt the dual, more geometric point of view. Namely, if we allow to apply all possible convex closure rules at once, then the set of constraints inferrable in this way will form a (convex) polyhedra. Its dual will be a polytope P that is actually a sub-polytope of $[0,1]^n$ (due to default axioms). The task is to show that $P \cap \{0,1\}^n = \emptyset$, that is that P does not contain any *integer* points. In this language, applying the cut rule means cutting off a small piece from this polytope (whence the name) guaranteed to not contain integer points.

From the algorithmic perspective, cutting planes correspond to the "geometric" part of very prominent method in combinatorial optimization called *Branch and Cut*. The full power of this method is captured by the proof system that, in the geometric language above, operates with finite unions of polytopes. This system is currently out of reach of the current methods although I would hesitate to classify it as "strong." A major development has been very recently reported on its subsystem $\mathrm{Res}(\mathrm{lin}_{\mathbb{R}})$ in which all polytopes are confined to the form $H \cap [0,1]^n$, H a hyperplane [47].

One proof complexity measure for cutting planes that has been extensively considered in the literature is their (Chvátal) *rank* (or *depth*). It is defined as follows: we allow to apply in parallel not only all possible convex closure rules but cut rules as well. Then the rank is simply the number of rounds that are necessary to arrive at the empty polytope. This complexity measure is rather well understood due to a very powerful technique called "protection lemmas," see [36, Chapter 19] for an excellent exposition.

As far as the *size* of cutting planes refutation is concerned, the situation is way more intriguing and dynamic. The first lower bounds were proved by Pudlák [51] using a prominent *feasible interpolation* method (or rather property). In the next theorem, Clique-Coloring(n,k) is the principle that says that a graph on n vertices may not simultaneously have a clique on k vertices and be k-colorable.

Theorem 6.13 ([51]). $S_{\mathrm{CP}}(\text{Clique-Coloring}(n, \sqrt{n})) \geq \exp(n^{\Omega(1)})$.

Remarkably, the method of feasible interpolation is not combinatorial or direct, instead it reduces a difficult problem in proof complexity to a difficult problem in circuit complexity (lower bounds for monotone circuits) that we fortunately know how to solve. As a by-side remark, let me mention that this kind of reductions is very important and welcome for the proof complexity. Still, it is also natural to wonder whether there are any "direct" methods (all other results in this article certainly qualify) to handle cutting planes. On this frontier we have seen recent exciting developments that defy several pieces of "common wisdom."

Firstly, it somehow makes sense to assume that random $O(\log n)$-CNFs and $O(1)$-CNFs should be "morally similar." Nonetheless, the proof method of the following theorem (a very clever use of feasible interpolation) seems to completely break apart for $O(1)$-CNFs.

Theorem 6.14 ([30, 35]). *With probability* $1 - o(1)$, *for a random* $\Theta(\log n)$*-CNF* τ_n *with* $n^{O(1)}$ *clauses we have* $S_{\mathrm{CP}}(\tau_n \vdash 0) \geq \exp(n^{\Omega(1)})$.

Even more striking and unexpected is the following recent result. In all our previous scenarios, random $O(1)$-CNFs and Tseitin tautologies for expanders went hand in hand, and it was a general feeling that morally they should be sort of the same (well, unless the characteristics of the field is 2). Given this feeling, the following *upper* bound, very surprising in itself, also does not seem to generalize to random $O(1)$-CNFs.

Theorem 6.15 ([26]). *For any sequence* $\{G_n\}$ *of bounded-degree graphs,*

$$S_{\mathrm{CP}}\big(\mathrm{Tseitin}(G_n) \vdash 0\big) \leq n^{O(\log n)}.$$

All these developments make the following problem particularly exciting.

Problem 6.16. Is it true that for random $O(1)$-CNFs τ_n with $O(n)$ clauses,

$$S_{\mathrm{CP}}(\tau_n \vdash 0) \geq \exp(n^{\Omega(1)}) \text{ w.h.p.?}$$

It is expected that solving this in the affirmative would require development of long-sought *direct* techniques, combinatorial or geometric, for analyzing the size complexity of cutting planes. But then it also had been expected from the principles featuring in the last two theorems.

7. In lieu of conclusion

There are several important topics in the modern proof complexity that, due to time and space constraints, we have either skipped entirely or given them much less attention than they deserve. Let me conclude with a list of such topics, saying (literally) a few words about each of them and providing some pointers to the literature.

Space complexity. Size complexity measures roughly correspond to the framework in which a complete proof is written as a single piece made ready for submission or verification. Space complexity deals with more dynamic, “classroom” scenario when the proof is presented on a blackboard and lemmas that are no longer needed can be erased to save space. See [45] for a nice exposition.

Feasible interpolation and automatizability. These were already mentioned in Sections 5 and 6.1. The book [39] treats the subject extensively in Chapters 17 and 18.

Relations between various proof systems and complexity measures. We have already seen some of those but, with the exception for Theorem 5.7, they were somewhat straightforward. There are, however, many other realtions, particularly involving space complexity measures, that are rather intricate and unexpected. The paper [46] aims at providing a general picture using an appropriate notion of reduction.

Pseudo-random generators in proof complexity. This is an ongoing effort to adjust to the needs of proof complexity the concept that is omnipresent in computational complexity. It is largely motivated by studying (efficient) provability of the principle "NP does not possess small circuits" we already mentioned several times. See (again) [59, Section 1] or [39, Chapter 19.4] for more details.

Lifting techniques. This is a very recent general approach to lower bounds in circuit complexity, communication complexity, and proof complexity remarkably uniting the three themes. I am not aware of an expository source (this is very much work in progress!) so let me instead refer to one of the latest papers in this direction [27].

Ideal proof system. This is an intriguing and bold attempt to stretch the Cook–Reckhow framework (Definition 1.4) and bring it closer to the concept of PCPs discussed earlier in Section 1. The paper [3] is one of the latest texts on the subject.

References

[1] M. Ajtai, Σ_1^1-formulae on finite structures. *Ann. Pure Appl. Logic* **24** (1983), no. 1, 1–48 Zbl 0519.03021 MR 706289

[2] M. V. Alekhnovich and A. A. Razborov, Lower bounds for polynomial calculus: the non-binomial ideal case. *Tr. Mat. Inst. Steklova* **242** (2003), 23–43 Zbl 1079.03047 MR 2054483

[3] Y. Alekseev, D. Grigoriev, E. A. Hirsch, and I. Tzameret, Semi-algebraic proofs, IPS lower bounds, and the τ-conjecture: can a natural number be negative? In *STOC '20—Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 54–67, ACM, New York, 2020 Zbl 07298230 MR 4141742

[4] S. Arora and B. Barak, *Computational Complexity. A Modern Approach*. Cambridge University Press, Cambridge, 2009 Zbl 1193.68112 MR 2500087

[5] A. Atserias, Improved bounds on the weak pigeonhole principle and infinitely many primes from weaker axioms. In *Mathematical Foundations of Computer Science, 2001 (Mariánské Lázně)*, pp. 148–158, Lecture Notes in Comput. Sci. 2136, Springer, Berlin, 2001 Zbl 0999.03052 MR 1907008

[6] A. Atserias, I. Bonacina, S. F. De Rezende, M. Lauria, J. Nordström, and A. A. Razborov, Clique is hard on average for regular resolution. *J. ACM* **68** (2021), no. 4, Art. 23 Zbl 1427.68102 MR 4313110

[7] A. Atserias, J. K. Fichte, and M. Thurley, Clause-learning algorithms with many restarts and bounded-width resolution. *J. Artificial Intelligence Res.* **40** (2011), 353–373 Zbl 1214.68340 MR 2805245

[8] A. Atserias and M. Müller, Automating resolution is NP-hard. *J. ACM* **67** (2020), no. 5, Art. 31 Zbl 07273098 MR 4152445

[9] B. Barak and D. Steurer, Sum-of-squares proofs and the quest toward optimal algorithms. In *Proceedings of the International Congress of Mathematicians—Seoul 2014. Vol. IV*, pp. 509–533, Kyung Moon Sa, Seoul, 2014 Zbl 1373.68253 MR 3727623

[10] P. Beame, S. Cook, J. Edmonds, R. Impagliazzo, and T. Pitassi, The relative complexity of NP search problems. In *Proceedings of the 27th Annual ACM Symposium on the Theory of Computing (STOC'95)*, pp. 303–314, ACM, New York, 1995 Zbl 0978.68526

[11] P. Beame, R. Impagliazzo, J. Krajíček, T. Pitassi, and P. Pudlák, Lower bounds on Hilbert's Nullstellensatz and propositional proofs. *Proc. London Math. Soc. (3)* **73** (1996), no. 1, 1–26 Zbl 0853.03017 MR 1387081

[12] P. Beame, H. Kautz, and A. Sabharwal, Towards understanding and harnessing the potential of clause learning. *J. Artificial Intelligence Res.* **22** (2004), 319–351 Zbl 1080.68651 MR 2129471

[13] E. Ben-Sasson, Hard examples for bounded depth Frege. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pp. 563–572, ACM, New York, 2002 Zbl 1192.03041 MR 2121182

[14] E. Ben-Sasson and R. Impagliazzo, Random CNF's are hard for the polynomial calculus. *Comput. Complexity* **19** (2010), no. 4, 501–519 Zbl 1216.03064 MR 2746277

[15] E. Ben-Sasson and A. Wigderson, Short proofs are narrow—resolution made simple. *J. ACM* **48** (2001), no. 2, 149–169 Zbl 1089.03507 MR 1868713

[16] M. L. Bonet, S. R. Buss, and T. Pitassi, Are there hard examples for Frege systems? In *Feasible Mathematics, II (Ithaca, NY, 1992)*, pp. 30–56, Progr. Comput. Sci. Appl. Logic 13, Birkhäuser, Boston, MA, 1995 Zbl 0834.03021 MR 1322273

[17] S. Buss, D. Grigoriev, R. Impagliazzo, and T. Pitassi, Linear gaps between degrees for the polynomial calculus modulo distinct primes. *J. Comput. System Sci.* **62** (2001), no. 2, 267–289 Zbl 1007.03052 MR 1820593

[18] S. Buss, R. Impagliazzo, J. Krajíček, P. Pudlák, A. A. Razborov, and J. Sgall, Proof complexity in algebraic systems and bounded depth Frege systems with modular counting. *Comput. Complexity* **6** (1996/97), no. 3, 256–298 Zbl 0890.03030 MR 1486929

[19] S. Buss and J. Nordström, Proof complexity and SAT solving. In *Handbook of Satisfiability*, chap. 7, pp. 233–350, Frontiers in Artificial Intelligence and Applications 336, IOS Press, Amsterdam, 2021 Zbl 1456.68001

[20] S. Buss and T. Pitassi, Resolution and the weak pigeonhole principle. In *Computer Science Logic (Aarhus, 1997)*, pp. 149–156, Lecture Notes in Comput. Sci. 1414, Springer, Berlin, 1998 Zbl 0910.03036 MR 1727809

[21] S. R. Buss, *Bounded Arithmetic*. Studies in Proof Theory. Lecture Notes 3, Bibliopolis, Naples, 1986 Zbl 0649.03042 MR 880863

[22] V. Chvátal and E. Szemerédi, Many hard examples for resolution. *J. Assoc. Comput. Mach.* **35** (1988), no. 4, 759–768 Zbl 0712.03008 MR 1072398

[23] M. Clegg, J. Edmonds, and R. Impagliazzo, Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996)*, pp. 174–183, ACM, New York, 1996 Zbl 0938.68825 MR 1427512

[24] S. Cook and P. Nguyen, *Logical Foundations of Proof Complexity*. Perspect. Log., Cambridge University Press, Cambridge, 2014 Zbl 1284.03001 MR 3241240

[25] S. A. Cook and R. A. Reckhow, The relative efficiency of propositional proof systems. *J. Symbolic Logic* **44** (1979), no. 1, 36–50 Zbl 0408.03044 MR 523487

[26] D. Dadush and S. Tiwari, On the complexity of branching proofs. In *35th Computational Complexity Conference*, p. Art. No. 34, LIPIcs. Leibniz Int. Proc. Inform. 169, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2020 MR 4129296

[27] S. de Rezende, O. Meir, J. Nordström, T. Pitassi, R. Robere, and M. Vinyals, Lifting with simple gadgets and applications to circuit and proof complexity. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science*, pp. 24–30, IEEE Computer Soc., Los Alamitos, CA, 2020 MR 4232019

[28] S. F. de Rezende, J. Nordström, K. Risse, and D. Sokolov, Exponential resolution lower bounds for weak pigeonhole principle and perfect matching formulas over sparse graphs. In *35th Computational Complexity Conference*, p. Art. No. 28, LIPIcs. Leibniz Int. Proc. Inform. 169, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2020 MR 4129290

[29] R. G. Downey and M. R. Fellows, *Parameterized Complexity*. Monographs in Computer Science, Springer, New York, 1999 MR 1656112

[30] N. Fleming, D. Pankratov, T. Pitassi, and R. Robere, Random $\Theta(\log n)$-CNFs are hard for cutting planes. In *58th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2017*, pp. 109–120, IEEE Computer Soc., Los Alamitos, CA, 2017 MR 3734222

[31] D. Grigoriev, Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity. *Theoret. Comput. Sci.* **259** (2001), no. 1-2, 613–622 Zbl 0974.68192 MR 1832812

[32] D. Grigoriev, E. A. Hirsch, and D. V. Pasechnik, Complexity of semialgebraic proofs. *Mosc. Math. J.* **2** (2002), no. 4, 647–679 Zbl 1027.03044 MR 1986085

[33] A. Haken, The intractability of resolution. *Theoret. Comput. Sci.* **39** (1985), no. 2-3, 297–308 Zbl 0586.03010 MR 821207

[34] J. Håstad, *Computational limitations on Small Depth Circuits*. Ph.D. thesis, Massachusetts Institute of Technology, 1986

[35] P. Hrubeš and P. Pudlák, Random formulas, monotone circuits, and interpolation. In *58th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2017*, pp. 121–131, IEEE Computer Soc., Los Alamitos, CA, 2017 MR 3734223

[36] S. Jukna, *Boolean Function Complexity. Advances and Frontiers*. Algorithms Combin. 27, Springer, Heidelberg, 2012 Zbl 1235.94005 MR 2895965

[37] J. Krajíček, Lower bounds to the size of constant-depth propositional proofs. *J. Symbolic Logic* **59** (1994), no. 1, 73–86 Zbl 0798.03056 MR 1264964

[38] J. Krajíček, *Bounded Arithmetic, Propositional Logic, and Complexity Theory*. Encyclopedia Math. Appl. 60, Cambridge University Press, Cambridge, 1995 Zbl 0835.03025 MR 1366417

[39] J. Krajíček, *Proof Complexity*. Encyclopedia Math. Appl. 170, Cambridge University Press, Cambridge, 2019 Zbl 07044161 MR 3929744

[40] J. Krajíček and P. Pudlák, Propositional proof systems, the consistency of first order theories and the complexity of computations. *J. Symbolic Logic* **54** (1989), no. 3, 1063–1079 Zbl 0696.03029 MR 1011192

[41] J. Krajíček, P. Pudlák, and A. Woods, An exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle. *Random Structures Algorithms* **7** (1995), no. 1, 15–39 Zbl 0843.03032 MR 1346282

[42] A. Maciel, T. Pitassi, and A. R. Woods, A new proof of the weak pigeonhole principle. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pp. 368–377, ACM, New York, 2000 Zbl 1296.03033 MR 2114552

[43] M. Mikša and J. Nordström, A generalized method for proving polynomial calculus degree lower bounds. In *30th Conference on Computational Complexity*, pp. 467–487, LIPIcs. Leibniz Int. Proc. Inform. 33, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2015 Zbl 1434.03134 MR 3441818

[44] N. Mull, S. Pang, and A. A. Razborov, On CDCL-based proof systems with the ordered decision strategy. In *Theory and Applications of Satisfiability Testing—SAT 2020*, pp. 149–165, Lecture Notes in Comput. Sci. 12178, Springer, Cham, 2020 Zbl 07331019 MR 4140309

[45] J. Nordström, Pebble games, proof complexity and time-space trade-offs. *Log. Methods Comput. Sci.* **9** (2013), no. 3, 3:15, 63 Zbl 1285.03070 MR 3109599

[46] T. Papamakarios and A. A. Razborov, Space characterizations of complexity measures and size-space trade-offs in propositional proof systems. Tech. Rep. TR21-074, Electronic Colloquium on Computational Complexity, 2021

[47] F. Part and I. Tzameret, Resolution with counting: dag-like lower bounds and different moduli. *Comput. Complexity* **30** (2021), no. 1, Paper No. 2 Zbl 07355181 MR 4199854

[48] T. Pitassi, P. Beame, and R. Impagliazzo, Exponential lower bounds for the pigeonhole principle. *Comput. Complexity* **3** (1993), no. 2, 97–140 Zbl 0784.03034 MR 1233662

[49] T. Pitassi and R. Raz, Regular resolution lower bounds for the weak pigeonhole principle. *Combinatorica* **24** (2004), no. 3, 503–524 Zbl 1063.03044 MR 2085370

[50] T. Pitassi, B. Rossman, R. A. Servedio, and L.-Y. Tan, Poly-logarithmic Frege depth lower bounds via an expander switching lemma. In *STOC'16—Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 644–657, ACM, New York, 2016 Zbl 1373.03125 MR 3536603

[51] P. Pudlák, Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symbolic Logic* **62** (1997), no. 3, 981–998 Zbl 0945.03086 MR 1472134

[52] R. Raz, P $\neq$ NP, propositional proof complexity, and resolution lower bounds for the weak pigeonhole principle. In *Proceedings of the International Congress of Mathematicians, Vol. III (Beijing, 2002)*, pp. 685–693, Higher Ed. Press, Beijing, 2002 Zbl 1012.68081 MR 1957570

[53] R. Raz, Resolution lower bounds for the weak pigeonhole principle. *J. ACM* **51** (2004), no. 2, 115–138 Zbl 1317.03036 MR 2145651

[54] A. A. Razborov, Lower bounds on the size of bounded-depth networks over a complete basis with logical addition. *Mat. Zametki* **41** (1987), no. 4, 598–607; English translation *Math. Notes* **41** (1987), 333–338 MR 897705

[55] A. A. Razborov, On provably disjoint **NP**-pairs. Tech. Rep. RS-94-36, Basic Research in Computer Science Center, Aarhus, Denmark, 1994, https://www.brics.dk/RS/94/36/BRICS-RS-94-36.pdf

[56] A. A. Razborov, Lower bounds for the polynomial calculus. *Comput. Complexity* **7** (1998), no. 4, 291–324 Zbl 1026.03043 MR 1691494

[57] A. A. Razborov, Proof complexity of pigeonhole principles. In *Developments in Language Theory (Vienna, 2001)*, pp. 110–116, Lecture Notes in Comput. Sci. 2295, Springer, Berlin, 2002 Zbl 1073.03540 MR 1964164

[58] A. A. Razborov, Resolution lower bounds for perfect matching principles. *J. Comput. System Sci.* **69** (2004), no. 1, 3–27 Zbl 1106.03049 MR 2070797

[59] A. A. Razborov, Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. *Ann. of Math. (2)* **181** (2015), no. 2, 415–472 Zbl 1376.03055 MR 3275844

[60] R. A. Reckhow, On the lengths of proofs in the propositional calculus. Tech. Rep. 87, University of Toronto, 1976

[61] R. Smolensky, Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the 19th ACM Symposium on Theory of Computing*, pp. 77–82, 1987

[62] G. S. Tseitin, On the complexity of derivations in propositional calculus. In *Studies in Constructive Mathematics and Mathematical Logic, Part II*, Consultants Bureau, New York, 1968

[63] A. Yao, Separating the polynomial-time hierarchy by oracles. In *Proceedings of the 26th IEEE FOCS*, pp. 1–10, 1985

Alexander A. Razborov
Departments of Mathematics and Computer Science, University of Chicago, Eckhart Hall,
5734 S University Ave, Chicago, IL 60637, USA; and Steklov Mathematical Institute of
the Russian Academy of Sciences, 8 Gubkina St., Moscow 119991, Russia;
razborov@uchicago.edu, razborov@mi-ras.ru