SIGACT News Complexity Theory Column (DRAFT[1])

Meta-Mathematics of Computational Complexity Theory

Igor C. Oliveira[2]

Abstract

We survey results on the formalization and independence of mathematical statements related to major open problems in computational complexity theory. Our primary focus is on recent findings concerning the (un)provability of complexity bounds within theories of bounded arithmetic. This includes the techniques employed and related open problems, such as the (non)existence of a feasible proof that $\mathsf{P} = \mathsf{NP}$.

Contents

[1] **Latest Update: July 10, 2024.** Comments are welcome and would be appreciated.

[2] Department of Computer Science, University of Warwick, UK. Email: **igor.oliveira@warwick.ac.uk**.

1 Introduction

The investigation of the inherent complexity of computational tasks is a central research direction in theoretical computer science. While unconditional results are known in a variety of restricted contexts (i.e., with respect to weak models of computation), despite significant efforts, several central questions of the field remain wide open. Prominent examples include the relation between complexity classes P and NP, understanding the power of non-uniform Boolean circuits, and bounding the length of proofs in propositional proof systems such as Frege and extended Frege.

The investigation of the difficulty of settling these problems has long been an important and influential area of research by itself (e.g., barrier results such as [BGS75, RR97, AW09, CHO+22]). Unfortunately, these results tend to be ad-hoc and do not consider a standard and robust notion of proof. In order to build a general theory, several works have considered provability in the usual sense of mathematical logic. Most importantly, this enables a deeper investigation of complexity theory that considers not only the running time of a program or the size of a circuit but also the feasibility of proving their existence and correctness. In particular, we can explore the fundamental question of what can and cannot be feasibly computed, along with the meta-question of what lower and upper bounds can and cannot be feasibly proven.

A fundamental goal of this research is to

($\star$) identify a suitable logical theory capable of formalizing most, if not all, known results in algorithms and complexity, and determine whether the major open problems mentioned above are provable or unprovable within this theory.[3]

Although we are still far from reaching this goal, progress has been made in understanding the (un)provability of statements concerning the complexity of computations within certain fragments of Peano Arithmetic, collectively known as Bounded Arithmetic. These theories are designed to capture proofs that manipulate and reason with concepts from a specified complexity class. For instance, a proof by induction whose inductive hypothesis can be expressed as an NP predicate is one such example. The earliest theory of this kind was $\mathsf{I}\Delta_0$, introduced by Parikh [Par71], who explored the intuitive concept of feasibility in arithmetic and addressed the infeasibility of exponentiation. The relationship between Parikh's theory and computational complexity was fully recognized and advanced by Paris and Wilkie in a series of influential papers during the 1980s (see [WP87]). Other significant theories include Cook's theory PV_1 [Coo75], which formalizes polynomial-time reasoning; Jeřábek's theory APC_1 [Jeř04, Jeř05, Jeř07], which extends PV_1 by incorporating the dual weak pigeonhole principle for polynomial-time functions and formalizes probabilistic polynomial-time reasoning; and Buss's theories S^i_2 and T^i_2 [Bus86], which include induction principles corresponding to various levels of the polynomial-time hierarchy.

These theories are capable of formalizing advanced results. For instance, it is known that PV_1 can prove the PCP Theorem [Pic15b], while APC_1 can establish several significant circuit lower bounds [MP20], including monotone circuit lower bounds for k-Clique and bounded-depth circuit lower bounds for the Parity function. Further examples include the explicit construction of expander graphs [BKKK20] and the correctness of randomized polynomial-time matching algorithms [LC11], among many others.

Given the expressive power of these theories, even if we are not yet able to establish a breakthrough result of the magnitude of ($\star$), determining the (un)provability of complexity bounds of interest in theories of bounded arithmetic still represents significant progress towards our understanding of the power and limits of feasible computations and proofs. This survey aims to provide an introduction to some of these results,

[3] As we elaborate in Section 5, the unprovability of a statement is equivalent to the consistency of its negation, which can be at least as important.

the underlying techniques, and related open problems. While our primary focus is on recent developments, in order to provide a broader perspective we also cover some classical results. Due to space limitations, the survey is not exhaustive, and several references had to be omitted (although some recent developments are mentioned in Section 6).

2 Preliminaries

2.1 Complexity Theory

We will rely on a few additional standard definitions from complexity theory, such as basic complexity classes, Boolean circuits and formulas, and propositional proof systems. These can be found in textbooks such as [AB09] and [Kra19]. Below we only establish notation and review a classical result that offers a convenient way to talk about polynomial-time computations in some logical theories.

We use $\mathsf{SIZE}[s]$ to denote the set of languages computed by Boolean circuits of size $s(n)$.

In theoretical computer science, one typically considers functions and predicates that operate over binary strings. This is equivalent to operations on integers, by identifying each non-negative integer with its binary representation. Let $\mathbb{N}$ denote the set of non-negative integers. For $a \in \mathbb{N}$, we let $|a| \triangleq \lceil \log_2(a+1) \rceil$ denote the length of the binary representation of a. For a constant $k \geq 1$, we say that a function $f\colon \mathbb{N}^k \to \mathbb{N}$ is computable in polynomial time if $f(x_1, \ldots, x_k)$ can be computed in time polynomial in $|x_1|, \ldots, |x_k|$. (For convenience, we might write $|\vec{x}| \triangleq |x_1|, \ldots, |x_k|$.) Recall that FP denotes the set of polynomial time functions. While the definition of polynomial time refers to a machine model, FP can also be introduced in a machine independent way as the closure of a set of base functions under *composition* and *limited recursion on notation*. In more detail, we can consider the following class $\mathcal{F}$ of base functions:

$$c(x) \triangleq 0, \quad s(x) \triangleq x+1, \quad a(x) \triangleq \lfloor x/2 \rfloor, \quad d(x) \triangleq 2 \cdot x, \quad \pi^i_\ell(x_1, \ldots, x_\ell) \triangleq x_i, \quad x\#y \triangleq 2^{|x| \cdot |y|},$$

$$x \leq y \triangleq \begin{cases} 1 & \text{if } x \leq y \\ 0 & \text{otherwise,} \end{cases} \qquad \mathsf{Choice}(x,y,z) \triangleq \begin{cases} y & \text{if } x > 0 \\ z & \text{otherwise.} \end{cases}$$

We say that a function $f(\vec{x}, y)$ is defined from functions $g(\vec{x})$, $h(\vec{x}, y, z)$, and $k(\vec{x}, y)$ by *limited recursion on notation* if

$$\begin{aligned} f(\vec{x}, 0) &= g(\vec{x}) \\ f(\vec{x}, y) &= h(\vec{x}, y, f(\vec{x}, \lfloor y/2 \rfloor)) \\ f(\vec{x}, y) &\leq k(\vec{x}, y) \end{aligned}$$

for every sequence $(\vec{x}, y)$ of natural numbers. Cobham [Cob65] proved that FP is the least class of functions that contains $\mathcal{F}$ and is closed under composition and limited recursion on notation.

2.2 Theories of Bounded Arithmetic

Bounded arithmetic has a long and rich history (see [Bus97] for an introduction, and [HP93, Kra95, CN10] for a detailed treatment). The correspondence between the theories and complexity classes manifests in multiple ways. For instance, *witnessing results* show that every provably total function in a given theory $\mathsf{T}_\mathcal{C}$ (i.e., when $\forall x\, \exists! y\, \psi(x,y)$ is provable, for certain formulas ψ) is computable within the corresponding complexity class $\mathcal{C}$ (i.e., the function $y = f(x)$ is in $\mathcal{C}$). There is also a close connection between

theories of bounded arithmetic and propositional proof systems, e.g., *propositional translations* between proofs of certain sentences in PV_1 or S^1_2 and polynomial-size proofs in the extended Frege proof system of the corresponding propositional formulas. We review some related results in Section 3.1 and Section 3.2, respectively. In this section, we provide an overview of some widely investigated theories of bounded arithmetic: PV_1, S^1_2, T^1_2, and APC_1. We assume basic familiarity with first-order logic. Results claimed below without reference can be found in [Kra95].

2.2.1 PV_1

PV_1 [Coo75] (see also [KPT91]) is a first-order theory whose intended model is the set $\mathbb{N}$ of natural numbers, together with the standard interpretation for constants and functions symbols such as $0, +, \times$, etc. The vocabulary (language) of PV_1, denoted $\mathcal{L}_{\mathsf{PV}_1}$, contains a function symbol for each polynomial-time algorithm $f\colon \mathbb{N}^k \to \mathbb{N}$ (where k is any constant). These function symbols, and the axioms defining them, are obtained through Cobham's characterization of polynomial-time functions discussed in Section 2.1.

PV_1 also postulates an induction axiom scheme that simulates binary search, and one can show that it admits induction over quantifier-free formulas (i.e., polynomial-time predicates). We discuss induction axioms in more detail in Section 2.2.2.

We will use later in the text that PV_1 admits a formulation where all axioms are universal formulas (i.e., $\forall \vec{x}\, \phi(\vec{x})$, where ϕ is free of quantifiers). In other words, PV_1 is a *universal theory*.

While the details of the definition of PV_1 are fairly technical (see, e.g., the longer overview in [CLO24b] or the exposition in [Kra95]), such details are often not needed. In particular, PV_1 has an equivalent formalization that does not require Cobham's result [Jeř06].

2.2.2 S^1_2, T^1_2, and Beyond

While PV_1 can be related to polynomial-time computations and feasible proofs, Buss [Bus86] introduced a hierarchy of theories with close ties to the different levels of the polynomial hierarchy. To specify the theories, we will need a few definitions.

The language $\mathcal{L}_{\mathsf{B}}$ of these theories contains the predicate symbols $=$ and $\leq$, the constant symbols 0 and 1, and function symbols S (successor), $+$, $\cdot$, $\lfloor x/2 \rfloor$, $|x|$ (interpreted as the length of x as in Section 2.1), and $\#$ ("smash"; interpreted as $x\#y = 2^{|x|\cdot|y|}$).

A *bounded quantifier* is a quantifier of the form $Qy \leq t$, where $Q \in \{\exists, \forall\}$ and t is a term not involving y. Similarly, a *sharply bounded quantifier* is one of the form $Qy \leq |t|$. Formally, such quantifiers are simply abbreviations. For instance,

$$\begin{aligned} \forall y \leq t(\vec{x})\ \varphi(\vec{x}, y) &\triangleq \forall y\, (y \leq t(\vec{x}) \to \varphi(\vec{x}, y)), \text{ and} \\ \exists y \leq t(\vec{x})\ \varphi(\vec{x}, y) &\triangleq \exists y\, (y \leq t(\vec{x}) \wedge \varphi(\vec{x}, y))\,. \end{aligned}$$

A formula where each quantifier appears bounded (resp., sharply bounded) is said to be a bounded (resp., sharply bounded) formula. It is not hard to show that every sharply bounded formula defines a polynomial-time predicate over the standard model $\mathbb{N}$ under its usual operations. On the other hand, bounded quantifiers allow us to define predicates in NP, coNP, and beyond.

We can introduce a hierarchy of formulas by counting alternations of bounded quantifiers. The class $\Pi^b_0 = \Sigma^b_0$ contains the sharply bounded formulas. We then recursively define, for each $i \geq 1$, the classes Σ^b_i and Π^b_i according to the quantifier structure of the sentence, ignoring the appearance of sharply bounded quantifiers. For instance, if $\varphi \in \Sigma^b_0$ and $\psi \triangleq \exists y \leq t(\vec{x})\ \varphi(y, \vec{x})$, then $\psi \in \Sigma^b_1$ (see, e.g., [Kra95] for the

technical details in the general case). As alluded to above, it is known that, for each $i \geq 1$, a predicate $P(\vec{x})$ is in Σ_i^p (the i-th level of the polynomial hierarchy) if and only if there is a Σ_i^b-formula that agrees with it over $\mathbb{N}$.

The theories introduced by Buss share a common set BASIC of finitely many axioms postulating the expected arithmetic behavior of the constants, predicates, and function symbols, e.g., $x + y = y + x$ and $|1| = 1$ (see, e.g., [Kra95, Page 68] for the complete list). The only difference among the theories is the kind of induction axiom scheme that each of them postulates.

Theory T_2^1. This is a theory in the language $\mathcal{L}_\mathsf{B}$ extending BASIC by the induction axiom IND

$$\varphi(0) \wedge \forall x\,(\varphi(x) \to \varphi(x+1)) \to\ \forall x\,\varphi(x)$$

for all Σ_1^b-formulas $\varphi(a)$. The formula $\varphi(a)$ may contain other free variables in addition to a.

Informally, we say that T_2^1 admits induction for NP predicates. This definition can be extended to a theory that postulates induction for Σ_i^b-formulas, which gives rise to the theory T_2^i.

Theory S_2^1. This is a theory in the language $\mathcal{L}_\mathsf{B}$ extending BASIC by the polynomial induction axiom PIND

$$\varphi(0) \wedge \forall x\,(\varphi(\lfloor x/2 \rfloor) \to \varphi(x)) \to\ \forall x\,\varphi(x)$$

for all Σ_1^b-formulas $\varphi(a)$. The formula $\varphi(a)$ may contain other free variables in addition to a.

Analogously to T_2^i, we can define the theories S_2^i via polynomial induction for Σ_i^b-formulas. It is known that PV_1 is essentially equivalent to T_2^0 under an appropriate vocabulary and axioms [Jeř06], and that $\mathsf{S}_2^i \subseteq \mathsf{T}_2^i \subseteq \mathsf{S}_2^{i+1}$ for every $i \geq 1$.

When stating and proving results in S_2^1, it is convenient to employ a more expressive vocabulary under which any polynomial-time function can be easily described. Moreover, it is possible to achieve this in a *conservative* way, i.e., without increasing the power of the theory. In more detail, let Γ be a set of $\mathcal{L}_\mathsf{B}$-formulas. We say that a polynomial-time function $f\colon \mathbb{N}^k \to \mathbb{N}$ is Γ-*definable* in S_2^1 if there is a formula $\psi(\vec{x}, y) \in \Gamma$ for which the following conditions hold:

(*i*) For every $a \in \mathbb{N}^k$, $f(\vec{a}) = b$ if and only if $\mathbb{N} \models \varphi(\vec{a}, b)$.

(*ii*) $\mathsf{S}_2^1 \vdash \forall \vec{x}\,(\exists y\,(\varphi(\vec{x}, y) \wedge \forall z\,(\varphi(\vec{x}, z) \to y = z))\,.$

Every function $f \in \mathsf{FP}$ is Σ_1^b-definable in S_2^1. By adding all functions in FP to the vocabulary of S_2^1 and by extending the axioms of S_2^1 with their defining equations, we obtain a theory $\mathsf{S}_2^1(\mathcal{L}_\mathsf{PV})$ that can refer to polynomial-time predicates using quantifier-free formulas. $\mathsf{S}_2^1(\mathcal{L}_\mathsf{PV})$ proves the polynomial induction scheme for both Σ_1^b-formulas and Π_1^b-formulas in the extended vocabulary. $\mathsf{S}_2^1(\mathcal{L}_\mathsf{PV})$ is conservative over S_2^1, in the sense that any $\mathcal{L}_\mathsf{B}$-sentence provable in $\mathsf{S}_2^1(\mathcal{L}_\mathsf{PV})$ is also provable in S_2^1.

A $\forall\Sigma_i^b$-sentence is simply a sentence $\psi = \forall \vec{x}\,\varphi(\vec{x})$ where $\varphi \in \Sigma_i^b$. Every $\forall\Sigma_1^b$-sentence provable in $\mathsf{S}_2^1(\mathcal{L}_\mathsf{PV})$ is also provable in PV_1. In other words, $\mathsf{S}_2^1(\mathcal{L}_\mathsf{PV})$ is $\forall\Sigma_1^b$-conservative over PV_1. On the other hand, it is known that if $\mathsf{S}_2^1(\mathcal{L}_\mathsf{PV}) = \mathsf{PV}_1$, then the polynomial-time hierarchy collapses.

2.2.3 APC_1

In order to formalize probabilistic methods and randomized algorithms, Jeřábek [Jeř04, Jeř05, Jeř07] formulated the theory APC_1 (this terminology is from [BKT14]) by extending PV_1 with the *dual Weak Pigeonhole Principle* (dWPHP) for PV_1 functions:[4]

$$\mathsf{APC}_1 \triangleq \mathsf{PV}_1 \cup \{\mathsf{dWPHP}(f) \mid f \in \mathcal{L}_{\mathsf{PV}}\}.$$

Informally, each sentence $\mathsf{dWPHP}(f)$ postulates that, for every length $n = |N|$, there is $y < (1 + 1/n) \cdot 2^n$ such that $f(x) \neq y$ for every $x < 2^n$.

It is known that the dual Weak Pigeonhole Principle for polynomial-time predicates can be proved in T^2_2 [MPW02], and consequently $\mathsf{APC}_1 \subseteq \mathsf{T}^2_2(\mathcal{L}_{\mathsf{PV}})$.

3 Auxiliary Definitions and Results

3.1 Witnessing Theorems

Suppose a sentence ψ of a certain syntactic form admits a proof in a theory T over a vocabulary $\mathcal{L}$. A witnessing theorem allows us to extract computational information from any such proof, by showing that an existential quantifier in ψ can be witnessed by $\mathcal{L}$-terms. The simplest example of such a result is stated next.

Theorem 3.1 (Herbrand's Theorem (see, e.g., [Bus94, McK10]))**.** *Let T be a universal theory over a vocabulary $\mathcal{L}$. Let $\varphi(x, y)$ be a quantifier-free $\mathcal{L}$-formula, and suppose that $T \vdash \forall x\, \exists y\, \varphi(x, y)$. There is a constant $k \geq 1$ and $\mathcal{L}$-terms $t_1(x), \ldots, t_k(x)$ such that*

$$T \vdash \varphi(x, t_1(x)) \vee \varphi(x, t_2(x)) \vee \ldots \vee \varphi(x, t_k(x)).$$

As an immediate consequence, if we apply Theorem 3.1 to $T \triangleq \mathsf{PV}_1$, we obtain $\mathcal{L}_{\mathsf{PV}}$-terms (corresponding to polynomial-time functions over $\mathbb{N}$) such that, given $a \in \mathbb{N}$, at least one of them produces a witness $b \in \mathbb{N}$ such that $\mathbb{N} \models \varphi(a, b)$.

Next, we consider the provability of more complex sentences in a universal theory.

Theorem 3.2 (KPT Theorem [KPT91])**.** *Let T be a universal theory with vocabulary $\mathcal{L}$, ϕ be an open $\mathcal{L}$-formula, and suppose that $T \vdash \forall w\, \exists u\, \forall v\, \phi(w, u, v)$. Then there exist a constant $k \geq 1$ and $\mathcal{L}$-terms $t_1, \ldots, t_k$ such that*

$$T \vdash \phi(w, t_1(w), v_1) \vee \phi(w, t_2(w, v_1), v_2) \vee \ldots \vee \phi(w, t_k(w, v_1, \ldots, v_{k-1}), v_k),$$

where the notation $t_i(w, v_1, \ldots, v_{i-1})$ indicates that these are the only variables occurring in t_i.

Theorem 3.2 has a natural interpretation as an interactive game with finitely many rounds, which we revisit in Section 5.1.1 in the context of the provability of circuit upper bounds.

A similar form of Theorem 3.2 holds under the provability of a $\forall\exists\forall\exists$-sentence (see, e.g., [CKK+24] for a concrete application in the context of circuit lower bounds). In contrast, there is no straightforward analogue of the KPT Theorem for a larger number of quantifier alternations. In this case, more general formulations are needed, such as the ones considered in [Pud06, BKT14, LO23].

[4] The dWPHP axiom scheme is also referred to as the surjective Weak Pigeonhole Principle in some references.

It is also possible to establish witnessing theorems for theories that are not universal. This can be done either by first transforming the theory into a universal theory through the inclusion of new function symbols and quantifier elimination, or via direct approaches (see, e.g., [Kra95, Section 7.3]). Another example is Buss's Theorem for S^1_2, which can be used to show that every $\forall\Sigma^b_1$-sentence provable in $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$ is also provable in PV_1. This has two implications. First, we can combine this result with Theorem 3.1, which yields polynomial-time algorithms from proofs of $\forall\Sigma^b_1$-sentences in $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$. Second, this means that in some situations we can establish the provability of a sentence in PV_1 using the more convenient theory $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$ (see Section 4.2 for an example).

3.2 Bounded Arithmetic and Propositional Proofs

In this section, we explain a connection between PV_1 and the extended Frege proof system discovered by [Coo75]. In short, it says that if a universal $\mathcal{L}_{\mathsf{PV}}$-sentence $\phi(x)$ is provable in PV_1, then there is a translation of $\phi(x)$ into a sequence $\{G_n\}_{n\geq 1}$ of propositional formulas $G_n(p_1,\ldots,p_n)$ such that each G_n has an extended Frege proof π_n of size polynomial in n.[5]

First, we review some concepts and fix notation, deferring the details to a standard textbook (e.g., [Kra19]). Recall that a propositional formula $G(p_1,\ldots,p_n)$ is formed using variables $p_1,\ldots,p_n$, constants 0 and 1, and logical connectives $\wedge$, $\vee$, and $\neg$. A *Frege* ($\mathcal{F}$) proof system is a "textbook" style proof system for propositional logic. It can be formulated as a finite set of axiom schemes together with the modus ponens rule. $\mathcal{F}$ is known to be sound and complete. We measure the *size* of a Frege proof in terms of the number of symbols occurring in the proof. In the *extended Frege* ($e\mathcal{F}$) proof system, we also allow repeated subformulas appearing in a proof to be abbreviated via new symbols.

Cook's Translation [Coo75]. Let φ be a universal $\mathcal{L}_{\mathsf{PV}}$-sentence of the form $\varphi \triangleq \forall x\, \psi(x)$, where $\psi(x)$ is free of quantifiers. Cook [Coo75] established that if φ is provable in PV_1, then there is a sequence $\{G_n\}_{n\geq 1}$ of propositional tautologies such that

- Each $G_n(p_1,\ldots,p_n)$ is a polynomial-size formula.
- G_n encodes that $\psi(x)$ is true whenever $|x| \leq n$, i.e., over all integers encoded as n-bit strings.
- G_n admits polynomial-size $e\mathcal{F}$-proofs.
- Moreover, the existence of polynomial-size $e\mathcal{F}$-proofs for each G_n is provable in PV_1. (We will need this additional property of the translation in Section 5.2.2.)

For a formula $\psi(x)$ as above, we write $||\psi||_n$ to denote the corresponding propositional formula over inputs of length n.

For more information about the relation between proofs in bounded arithmetic and propositional proofs, including additional examples of propositional translations, we refer to [Bey09, Kra19].

[5] Conceptually, this is analogous to the translation of a polynomial-time Turing machine M into a sequence $\{C_n\}_{n\geq 1}$ of polynomial-size Boolean circuits, one for each input length n.

3.3 Cuts of Models of Bounded Arithmetic

Many fundamental results in bounded arithmetic are established using model-theoretic techniques (see, e.g., the exposition of Parikh's Theorem in [Kra95]). We will provide an example in Section 5.2.2. In this section, we include the required background for the result. We assume basic familiarity with model theory.

While the definitions and results presented below can be adapted to other theories of bounded arithmetic, we focus on the theory S^1_2 for concreteness.

Definition 3.3 (Cut in a Model of Arithmetic)**.** A *cut* in a model M of S^1_2 is a nonempty set $I \subseteq M$ such that:

1. For every $a, b \in M$, if $b \in I$ and $a < b$ then $a \in I$.
2. For every $a \in M$, if $a \in I$ then $a + 1 \in I$.

In this case, we write $I \subseteq_e M$.

Note that a cut is not necessarily closed under operations such as addition and multiplication.

Claim 3.4. *Let M be a model of S^1_2, and let $I \subseteq_e M$. Moreover, assume that I is closed under $+$, $\cdot$, and $\#$ operations. Let $\varphi(a, \vec{b})$ be a bounded formula with all free variables displayed. Let $\vec{v}$ be elements of I. Then for every $u \in I$,*

$$I \models \varphi(u, \vec{v}) \quad \iff \quad M \models \varphi(u, \vec{v}).$$

Claim 3.4 can be proved by induction on the complexity of φ. Using the claim, one can establish the following lemma.

Lemma 3.5. *Let M be a model of S^1_2, and let $I \subseteq_e M$. Moreover, assume that I is closed under $+$, $\cdot$, and $\#$ operations. Then I is a model of S^1_2.*

Since it is not hard to check that a cut I as above satisfies the BASIC axioms of S^1_2, the proof of Lemma 3.5 essentially amounts to verifying that I satisfies the corresponding induction principle (see, e.g., [Kra95, Lemma 5.1.3] for a similar argument).

For a model M, we say that $n \in M$ is a *length* if there is $N \in M$ such that $n = |N|$.

Lemma 3.6. *Let M_0 be a nonstandard countable model of S^1_2. Then there is a (countable) cut M of M_0 that is a model of S^1_2 and a length $n \in M$ for which the following holds. For every $b \in M$ there is a standard number k such that $M \models |b| \leq n^k$.*

Proof. Let $e \in M_0$ be nonstandard, and let $n \triangleq |e|$. Consider the set

$$I_e \triangleq \{a \in M \mid a \leq t(e) \text{ for some } \mathcal{L}_{\mathsf{B}}\text{-term } t\}.$$

Note that I_e is a cut of M_0. Moreover, it is not hard to check that it is closed under addition, multiplication, and smash operations. By Lemma 3.5, I_e is a model of S^1_2. Finally, by construction, for every $b \in I_e$ we have $b \leq t(e)$ for some $\mathcal{L}_{\mathsf{B}}$-term t. A simple induction on the structure of t shows the existence of a standard number k such that $|b| \leq n^k$ in M. □

Finally, we will need the following definition.

Definition 3.7 (Cofinal extension)**.** An extension M' of a model M is *cofinal* if for every $a \in M'$ there is $b \in M$ such that $a \leq b$ in M'. If this is the case, we write $M' \supseteq_{\mathsf{cf}} M$.

4 The Strength of Bounded Arithmetic

In connection with the fundamental research goal mentioned in Section 1, research on the provability of complexity bounds has achieved significant progress on two complementary fronts: the *formalization* of several established results from algorithms and complexity within theories of bounded arithmetic, and the *unprovability* of complexity bounds in the same theories, often conditional on a computational assumption.

In Section 4.1, we explore what it means to formalize results from algorithms and complexity theory within the framework of bounded arithmetic, highlighting some of the nuances involved. In Section 4.2, we present some concrete details of the formalization of a formula lower bound in PV_1.

4.1 Formalization of Results from Algorithms and Complexity

Several central theorems from mathematics and computer science can be proved in bounded arithmetic. They include results from number theory [Woo81, PWW88], graph theory and extremal combinatorics [Oja04], randomized algorithms and probabilistic arguments [Jeř05, LC11, Lê14], probabilistic checkable proofs [Pic15b], circuit lower bounds [MP20], expander graphs [BKKK20], linear algebra [TC21], Zhuk's CSP algorithm [Gay23, Gay24], etc. The reader can find numerous other examples in [CN10, Kra19, MP20] and references therein.

In some cases, the formalization of an existing result in bounded arithmetic is straightforward, specially once an appropriate framework has been developed (e.g., the approximate counting framework of [Jeř07], which enables the use of tools from probability theory in APC_1). However, sometimes one needs to discover a new proof whose concepts can be defined in the theory and their associated properties established using the available inductive axioms (e.g., Razborov's formalization of the Switching Lemma [Raz95a]).

We provide two instructive examples below. The first is a consequence of the formalization of the PCP Theorem in PV_1, while the second concerns different ways of formulating a circuit lower bound statement in bounded arithmetic.

The PCP Theorem in PV_1. Pich [Pic15b] proved the PCP Theorem in PV_1 by formalizing Dinur's proof [Din07]. Exploiting the standard connection between PCPs and hardness of approximation, Pich's result can be used to show that PV_1 establishes the NP-hardness of approximating the value of a k-SAT instance. This means in particular that, for a suitable $\mathcal{L}_{\mathsf{PV}}$-function symbol f obtained from Dinur's argument, PV_1 proves that f is a gap-inducing reduction from the Boolean Formula Satisfiability Problem to k-SAT (for a sufficiently large k):

$$\mathsf{PV}_1 \vdash \forall\varphi\left(\mathsf{Valid\text{-}Fla}(\varphi) \wedge \exists y\, \mathsf{Sat}(\varphi, y) \rightarrow \mathsf{Valid}\text{-}k\text{-}\mathsf{CNF}(f(\varphi)) \wedge \exists z\, \mathsf{Sat}(f(\varphi), z)\right)$$

$$\mathsf{PV}_1 \vdash \forall\varphi\left(\mathsf{Valid\text{-}Fla}(\varphi) \wedge \forall y\, \neg\mathsf{Sat}(\varphi, y) \rightarrow \mathsf{Valid}\text{-}k\text{-}\mathsf{CNF}(f(\varphi)) \wedge \forall z\, \mathsf{Value}_{\leq 1-\delta}(f(\varphi), z)\right)$$

where all the expressions are quantifier-free $\mathcal{L}_{\mathsf{PV}}$-formulas: $\mathsf{Valid\text{-}Fla}(x)$ checks if x is a valid Boolean formula, $\mathsf{Valid}\text{-}k\text{-}\mathsf{CNF}(x)$ checks if x is a valid k-CNF, $\mathsf{Sat}(u, v)$ checks if v is a satisfying assignment for u, and $\mathsf{Value}_{\leq 1-\delta}(u, v)$ holds if v satisfies at most a $(1-\delta)$-fraction of the clauses in u (with $\delta > 0$ being a universal constant from the formalized Dinur's proof).

In the formalization the key point is that PV_1 proves that the function symbol f behaves as expected. In practice, in order to achieve this, a typical formalization is presented in a semi-formal way, and might claim on a few occasions that some algorithm f_1 constructed in a particular way from another algorithm f_2 can be defined in PV_1. This means that PV_1 proves that f_1 behaves as described in the definition.

This is possible thanks to Cobham's characterization of FP and the axioms of PV_1, which ensure that the theory "understands" how different algorithms are constructed from one another. In many cases, the verification that PV_1 proves the desired properties is straightforward but tedious, requiring some initial setup of basic capabilities of PV_1 (often referred to as "bootstrapping") which is part of the standard background in bounded arithmetic.

Circuit Lower Bound Statements. We discuss two ways of formalizing a complexity lower bound. In this example, for a given size bound $s(n)$ (e.g., $s(n) = n^2$), we consider an $\mathcal{L}_{\mathsf{PV}}$-sentence FLB_s stating that Boolean formulas for the parity function on n bits require at least $s(n)$ leaves:

$$\forall N\, \forall n\, \forall F\, (n = |N| \wedge n \geq 1 \wedge \mathsf{Valid\text{-}Fla}(F) \wedge \mathsf{Size}(F) < s(n) \rightarrow \exists x\, (|x| \leq n \wedge \mathsf{Eval}(F, x) \neq \oplus(x))\,,$$

where we identify n-bit strings with natural numbers of length at most n, and employ a well-behaved $\mathcal{L}_{\mathsf{PV}}$-function symbol $\oplus$ such that PV_1 proves the basic properties of the parity function, e.g., $\mathsf{PV}_1 \vdash \oplus(x1) = 1 - \oplus(x)$.[6]

Note that FLB_s is a $\forall\Sigma^b_1$-sentence. Consequently, if $\mathsf{PV}_1 \vdash \mathsf{FLB}_s$, we obtain via Herbrand's Theorem (Theorem 3.1) a polynomial-time algorithm A that, when given N of length n and the description of an n-bit formula F of size $< s(n)$, $A(N, F)$ outputs a string $x \in \{0,1\}^n$ such that $F(x) \neq \oplus(x)$. In other words, circuit lower bounds provable in PV_1 are constructive in the sense that they also provide an efficient refuter witnessing that F does not compute parity (see [CJSW21] for more on this topic).

The aforementioned formalization is informally referred to as a "Log" formalization of circuit lower bounds. This is because the main parameter n is the length of a variable N and all objects quantified over are of length polynomial in n. It is also possible to consider a formalization where $n = ||N||$ (n is the length of the length of N), which is known as a "LogLog" formalization. This allows us to quantify over exponentially larger objects, e.g., under such a formalization the entire truth-table of a formula F has length polynomial in the length of N.

Obtaining a Log formalization (e.g., [MP20]) is a stronger result than obtaining a LogLog formalization (e.g., [Raz95a]). In particular, in contrast to the discussion above, a witnessing theorem applied to a LogLog formalization provides a refuter with access to N and thus running in time $\mathsf{poly}(N) = \mathsf{poly}(2^n)$. Conversely, the unprovability of a LogLog circuit lower bound statement (e.g., [PS21, LO23]) is a stronger result than the unprovability of a Log statement. We refer to the introduction of [MP20] for a more extensive discussion on this matter.

4.2 Concrete Example: Subbotovskaya's Formula Lower Bound in PV_1

In this section, we explore some details of a formalization in PV_1 that the parity function $\oplus$ on n bits requires Boolean formulas of size $\geq n^{3/2}$ [Sub61]. We follow the notation introduced in Section 4.1.

Theorem 4.1 ([CKK+24])**.** *Let* $s(n) \triangleq n^{3/2}$. *Then* $\mathsf{PV}_1 \vdash \mathsf{FLB}_s$.

The formalization is an adaptation of the argument presented in [Juk12, Section 6.3], which proceeds as follows:

1. [Juk12, Lemma 6.8]: For any formula F on n-bit inputs, it is possible to fix one of its variables so that the resulting formula F_1 satisfies $\mathsf{Size}(F_1) \leq (1 - 1/n)^{3/2} \cdot \mathsf{Size}(F)$.

[6] We often abuse notation and treat x as a string in semi-formal discussions.

2. [Juk12, Theorem 6.10]: If we apply this result $\ell \triangleq n-k$ times, we obtain a formula F_ℓ on k-bit inputs such that

$$\mathsf{Size}(F_\ell) \leq \mathsf{Size}(F) \cdot (1-1/n)^{3/2} \cdot (1-1/(n-1))^{3/2} \dots (1-1/(k+1))^{3/2} = \mathsf{Size}(F) \cdot (k/n)^{3/2}.$$

3. [Juk12, Example 6.11]: Finally, if the initial formula F computes the parity function, by setting $\ell = n-1$ we get $1 \leq \mathsf{Size}(F_\ell) \leq (1/n)^{3/2} \cdot \mathsf{Size}(F)$, and consequently $\mathsf{Size}(F) \geq n^{3/2}$.

We present the argument in a more constructive way when formalizing the result in PV_1. In more detail, given a small formula F, we recursively construct (and establish correctness by induction) an n-bit input y witnessing that F does not compute the parity function.[7]

Proof. We follow closely the presentation from [CKK+24]. For brevity, we only discuss the formalization of the main inductive argument. More details can be found in [CKK+24]. Given $b \in \{0,1\}$, we introduce the function $\oplus^b(x) \triangleq \oplus(x) + b \pmod 2$. In order to prove FLB_s in PV_1, we explicitly consider a polynomial-time function $R(1^n, F, b)$ with the following property:[8]

If $\mathsf{Size}(F) < s(n)$ then $R(1^n, F, b)$ outputs an n-bit string y_n^b such that $\mathsf{Eval}(F, y_n^b) \neq \oplus^b(y_n^b)$.

In other words, $R(1^n, F, b)$ witnesses that the formula F does not compute the function $\oplus^b$ over n-bit strings. Note that the correctness of R is captured by a sentence $\mathsf{Ref}_{R,s}$ described as follows:

$$\forall 1^n \, \forall F \left(\mathsf{Valid\text{-}Fla}(F) \wedge \mathsf{Size}(F) < s(n) \rightarrow |y_n^0|_\ell = |y_n^1|_\ell = n \wedge F(y_n^0) \neq \oplus^0(y_n^0) \wedge F(y_n^1) \neq \oplus^1(y_n^1)\right),$$

where we employ the abbreviations $y_n^0 \triangleq R(1^n, F, 0)$ and $y_n^1 \triangleq R(1^n, F, 1)$, and for convenience use $|z|_\ell$ to denote the bitlength of z. Our plan is to define R and show that $\mathsf{PV}_1 \vdash \mathsf{Ref}_{R,s}$. Note that this implies FLB_s in PV_1 by standard first-order logic reasoning.

The correctness of $R(1^n, F, b)$ will be established by polynomial induction on N (equivalently, induction on $n = |N|$). Since $\mathsf{Ref}_{R,s}$ is a universal sentence and $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$ is $\forall\Sigma^b_1$-*conservative* over PV_1 (i.e., provability of such a sentence in $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$ implies its provability in PV_1), it is sufficient to describe a formalization in the more convenient theory $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$. For this reason, polynomial induction for NP and coNP predicates (admissible in $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$; see, e.g., [Kra95, Section 5.2]) is available during the formalization. More details follow.

The procedure $R(1^n, F, b)$ makes use of a few polynomial-time sub-routines (briefly discussed in the comments in the pseudocode below) and is defined in the following way:

[7] Actuallly, for technical reasons related to the induction step, we will simultaneously construct an n-bit input y_n^0 witnessing that F does not compute the parity function and an n-bit input y_n^1 witnessing that F does not compute the negation of the parity function.

[8] For convenience, we often write 1^n instead of explicitly considering parameters N and $n = |N|$. We might also write just $F(x)$ instead of $\mathsf{Eval}(F, x)$.

Input: 1^n for some $n \geq 1$, formula F over n-bit inputs, $b \in \{0,1\}$.

1 Let $s(n) \triangleq n^{3/2}$. If $\mathsf{Size}(F) \geq s(n)$ or $\neg\mathsf{Valid\text{-}Fla}(F)$ **return** *"error"*;

2 If $\mathsf{Size}(F) = 0$, F computes a constant function $b_F \in \{0,1\}$. In this case, **return** *the n-bit string* $y_n^b \triangleq y_1^b 0^{n-1}$ *such that* $\oplus^b(y_1^b 0^{n-1}) \neq b_F$;

3 Let $\widetilde{F} \triangleq \mathsf{Normalize}(1^n, F)$;

// $\widetilde{F}$ satisfies the conditions in the proof of [Juk12, Claim 6.9], $\mathsf{Size}(\widetilde{F}) \leq \mathsf{Size}(F)$, $\forall x \in \{0,1\}^n\ F(x) = \widetilde{F}(x)$.

4 Let $\rho \triangleq \mathsf{Find\text{-}Restriction}(1^n, \widetilde{F})$, where $\rho\colon [n] \to \{0,1,\star\}$ and $|\rho^{-1}(\star)| = n-1$;

// ρ restricts a suitable variable x_i to a bit c_i, as in [Juk12, Lemma 6.8].

5 Let $F' \triangleq \mathsf{Apply\text{-}Restriction}(1^n, \widetilde{F}, \rho)$. Moreover, let $b' \triangleq b \oplus c_i$ and $n' \triangleq n-1$;

// F' is an n'-bit formula; $\forall z \in \{0,1\}^{\rho^{-1}(\star)}\ F'(z) = \widetilde{F}(z \cup x_i \mapsto c_i)$.

6 Let $y_{n'}^{b'} \triangleq R(1^{n'}, F', b')$ and **return** *the n-bit string* $y_n^b \triangleq y_{n'}^{b'} \cup y_i \mapsto c_i$;

Algorithm 1: Refuter Algorithm $R(1^n, F, b)$ [CKK+24].

(The pseudocode presented above is only an informal specification of $R(1^n, F, b)$. As mentioned in Section 4.1, a completely formal proof in PV_1 would employ Cobham's formalism and would specify how $R(1^n, F, b)$ can be defined from previously defined algorithms (e.g., Apply-Restriction) via the allowed operations.)

We note that $R(1^n, F, b)$ runs in time polynomial in $n+|F|+|b|$ and that it is definable in $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$. Next, as an instructive example, we establish the correctness $R(1^n, F, b)$ in $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$ by *polynomial induction* (PIND) for Π^b_1-formulas, assuming that the subroutines appearing in the pseudocode of $R(1^n, F, b)$ satisfy the necessary properties (provably in $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$).

Lemma 4.2. *Let* $s(n) \triangleq n^{3/2}$. *Then* $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}}) \vdash \mathsf{Ref}_{R,s}$.

Proof. We consider the formula $\varphi(N)$ defined as

$$\forall F\, \forall n\, (n = |N| \wedge n \geq 1 \wedge \mathsf{Valid\text{-}Fla}(F) \wedge \mathsf{Size}(F) < s(n)) \to$$
$$(|y_n^0|_\ell = |y_n^1|_\ell = n \wedge F(y_n^0) \neq \oplus^0(y_n^0) \wedge F(y_n^1) \neq \oplus^1(y_n^1))\,,$$

where as before we use $y_n^0 \triangleq R(1^n, F, 0)$ and $y_n^1 \triangleq R(1^n, F, 1)$. Note that $\varphi(N)$ is a Π^b_1-formula. Below, we argue that

$$\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}}) \vdash \varphi(1) \quad \text{and} \quad \mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}}) \vdash \forall N\, \varphi(\lfloor N/2 \rfloor) \to \varphi(N)\,.$$

Then, by polynomial induction for Π^b_1-formulas (available in $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$) and using that $\varphi(0)$ trivially holds, it follows that $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}}) \vdash \forall N\, \varphi(N)$. In turn, this yields $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}}) \vdash \mathsf{Ref}_{R,s}$.

Base Case: $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}}) \vdash \varphi(1)$**.** In this case, for a given formula F and length n, the hypothesis of $\varphi(1)$ is satisfied only if $n = 1$, F is a valid description of a formula, and $\mathsf{Size}(F) = 0$. Let $y_1^0 \triangleq R(1, F, 0)$ and $y_1^1 \triangleq R(1, F, 1)$. We need to prove that

$$|y_1^0|_\ell = |y_1^1|_\ell = 1 \wedge F(y_1^0) \neq \oplus^0(y_1^0) \wedge F(y_1^1) \neq \oplus^1(y_1^1)\,.$$

Since $n = 1$ and $\mathsf{Size}(F) = 0$, F evaluates to a constant b_F on every input bit. The statement above is implied by Line 2 in the definition of $R(n, F, b)$.

(Polynomial) Induction Step: $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}}) \vdash \forall N\, \varphi(\lfloor N/2 \rfloor) \to \varphi(N)$**.** Fix an arbitrary N, let $n \triangleq |N|$, and assume that $\varphi(\lfloor N/2 \rfloor)$ holds. By the induction hypothesis, for every valid formula F' with $\mathsf{Size}(F') < n'^{3/2}$, where $n' \triangleq n - 1$, we have

$$|y^0_{n'}|_\ell = |y^1_{n'}|_\ell = n' \;\wedge\; F'(y^0_{n'}) \neq \oplus^0(y^0_{n'}) \;\wedge\; F'(y^1_{n'}) \neq \oplus^1(y^1_{n'})\,, \tag{1}$$

where $y^0_{n'} \triangleq R(1^{n'}, F', 0)$ and $y^1_{n'} \triangleq R(1^{n'}, F', 1)$.

Now let $n \geq 2$, and let F be a valid description of a formula over n-bit inputs with $\mathsf{Size}(F) < n^{3/2}$. By the size bound on F, $R(1^n, F, b)$ ignores Line 1. If $\mathsf{Size}(F) = 0$, then similarly to the base case it is trivial to check that the conclusion of $\varphi(N)$ holds. Therefore, we assume that $\mathsf{Size}(F) \geq 1$ and $R(1^n, F, b)$ does not stop at Line 2.

Consider the following definitions:

1. $\widetilde{F} \triangleq \mathsf{Normalize}(1^n, F)$ (Line 3),
2. $\rho \triangleq \mathsf{Find\text{-}Restriction}(1^n, \widetilde{F})$ (Line 4),
3. $F' \triangleq \mathsf{Apply\text{-}Restriction}(1^n, \widetilde{F}, \rho)$ (Line 5),
4. $n' \triangleq n - 1$ (Line 5),
5. $b' \triangleq b \oplus c_i$ (Line 5), where ρ restricts x_i to c_i,
6. $y^{b'}_{n'} \triangleq R(1^{n'}, F', b')$ (Line 6),
7. $y^b_n \triangleq y^{b'}_{n'} \cup y_i \mapsto c_i$ (Line 6),
8. $s \triangleq \mathsf{Size}(F)$, $\widetilde{s} \triangleq \mathsf{Size}(\widetilde{F})$, and $s' \triangleq \mathsf{Size}(F')$.

We rely on the provability in $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$ of the following statements about the subroutines of $R(1^n, F, b)$ (see [CKK+24]):

(*i*) $\widetilde{s} \leq s$,

(*ii*) $s' \leq \widetilde{s} \cdot (1 - 1/n)^{3/2}$,

(*iii*) $\forall x \in \{0,1\}^n\ \widetilde{F}(x) = F(x)$,

(*iv*) $\forall z \in \{0,1\}^{\rho^{-1}(\star)}\ F'(z) = \widetilde{F}(z \cup x_i \mapsto c_i)$.

By Items (*i*) and (*ii*) together with the bound $s < n^{3/2}$,

$$\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}}) \vdash s' \leq \widetilde{s} \cdot (1 - 1/n)^{3/2} \leq s \cdot (1 - 1/n)^{3/2} < n^{3/2} \cdot (1 - 1/n)^{3/2} = (n-1)^{3/2}\,.$$

Thus F' is a valid formula on n'-bit inputs of size $< n'^{3/2}$. By the first condition in the induction hypothesis (Equation (1)) and the definition of each y^b_n, we have $|y^0_n|_\ell = |y^1_n|_\ell = n$. Using the definitions listed above, the last two conditions in the induction hypothesis (Equation (1)), and Items (*iii*) and (*iv*), we derive in $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$ the following statements for each $b \in \{0,1\}$:

$$\begin{aligned} F'(y^{b'}_{n'}) &\neq \oplus^{b'}(y^{b'}_{n'})\,, \\ F(y^b_n) &= F'(y^{b'}_{n'})\,, \\ F(y^b_n) &\neq \oplus^{b'}(y^{b'}_{n'})\,. \end{aligned}$$

Therefore, using basic facts about the function symbols $\oplus^0$ and $\oplus^1$,

$$\oplus^{b'}(y^{b'}_{n'}) = \oplus^{b \oplus c_i}(y^{b'}_{n'}) = c_i \oplus (\oplus^b(y^{b'}_{n'})) = c_i \oplus (\oplus^b(y^b_n) \oplus c_i) = \oplus^b(y^b_n)\,.$$

These statements imply that, for each $b \in \{0,1\}$, $F(y^b_n) \neq \oplus^b(y^b_n)$. In other words, the conclusion of $\varphi(N)$ holds. This completes the proof of the induction step. □

As explained above, the provability of $\mathsf{Ref}_{R,s}$ in $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$ implies its provability in PV_1. Since $\mathsf{PV}_1 \vdash \mathsf{Ref}_{R,s} \to \mathsf{FLB}_s$, this completes the proof of Theorem 4.1. □

We have seen that a non-trivial formula size lower bound can be established in PV_1. More advanced circuit lower bounds are known to be provable assuming additional axioms extending PV_1 (e.g., [Kra95, Section 15.2] and [MP20]), but their provability in PV_1 (or equivalently, in $\mathsf{S}^1_2(\mathcal{L}_{\mathsf{PV}})$) is less clear.

Open Problem 4.3. *For each $d \geq 1$ and $\ell \geq 1$, can PV_1 prove that the parity function on n bits cannot be computed by depth-d circuits of size n^ℓ?*

Open Problem 4.4. *For each $\ell \geq 1$, is there a constant $k = k(\ell)$ such that PV_1 proves that every monotone circuit for the k-clique problem on n-vertex graphs must be of size at least n^ℓ?*

5 Unprovability of Complexity Bounds

The investigation of the unprovability of complexity bounds within theories of bounded arithmetic has a long and rich history. Much of the early work took place in the nineties, with significant results obtained by Razborov [Raz95a, Raz95b], Krajíček [Kra97], and other researchers. Since then, and in particular over the last decade, there has been renewed interest and progress in establishing unprovability results (see, e.g., [CK07, PS21, CKKO21, LO23, ABM23] and references therein).

In Section 5.1, we consider the unprovability of complexity upper bounds. The *unprovability* of an inclusion such as $\mathsf{NP} \subseteq \mathsf{SIZE}[n^k]$ is equivalent to the *consistency* of $\mathsf{NP} \not\subseteq \mathsf{SIZE}[n^k]$ with the corresponding theory. Such a consistency result establishes that, while we cannot confirm the separation is true in the standard model of natural numbers, we know it holds in a non-standard model of a theory so strong that complexity theory appears almost indistinguishable from the standard one. We stress that establishing the consistency of a lower bound is a necessary step towards showing that the lower bound is true. For this reason, the unprovability of upper bounds can be formally seen as progress towards showing unconditional complexity lower bounds.

In Section 5.2, we turn our attention to the unprovability of complexity lower bounds. This direction is partly driven by the desire to formally understand why proving complexity lower bounds is challenging, and to explore the possibility of a more fundamental underlying reason for this difficulty. Moreover, it might provide examples of hard sentences for logical theories and of hard propositional tautologies for proof systems. The investigation of the meta-mathematics of lower bounds has also found unexpected applications in algorithms and complexity (e.g., [CIKK16]).

Finally, in Section 5.3 we connect the two directions and explain how the unprovability of circuit lower bounds in PV_1 yields the unprovability of $\mathsf{P} = \mathsf{NP}$ in PV_1. The latter can be seen as a weakening of the P versus NP problem that considers the existence of feasible proofs that $\mathsf{P} = \mathsf{NP}$. This further motivates the investigation of the unprovability of lower bounds.

5.1 Unprovability of Upper Bounds

5.1.1 LEARN-Uniform Circuits and Unprovability

Cook and Krajíček [CK07] considered the provability of $\mathsf{NP} \subseteq \mathsf{SIZE}[\mathsf{poly}]$ in bounded arithmetic and obtained a number of conditional negative results. [KO17], building on techniques from [CK07], showed that for no integer $k \geq 1$ the theory PV_1 proves that $\mathsf{P} \subseteq \mathsf{SIZE}[n^k]$. Note that this is an unconditional result. Thus, for a natural theory capable of formalizing advanced results from complexity theory, such as the PCP Theorem, we can unconditionally rule out the provability of $\mathsf{P} \subseteq \mathsf{SIZE}[n^k]$. A slightly stronger model-theoretic formulation of the result of [KO17] appears in [BM20].

[BKO20] obtained results for stronger theories and ruled out the provability of infinitely often inclusions. In more detail, for an $\mathcal{L}_{\mathsf{PV}}$-function symbol h, consider the sentence

$$\mathsf{UB}_k^{i.o.}(h) \triangleq \forall 1^m\, \exists 1^n\, \exists C_n\, \forall x \left(n \geq m \wedge |C_n| \leq n^k \wedge (|x| \leq n \rightarrow \psi(n, C_n, x, h))\right),^{9}$$

where ψ is an open $\mathcal{L}_{\mathsf{PV}}$-formula stating that $h(x) \neq 0$ if and only if the evaluation of the circuit C_n on x (viewed as an n-bit string) is 1. In other words, $\mathsf{UB}_k^{i.o.}(h)$ states that the language defined by h (which is in P) admits circuits of size at most n^k on infinitely many input lengths n. [BKO20] showed that for each $k \geq 1$, there is an $\mathcal{L}_{\mathsf{PV}}$-function symbol h such that PV_1 does not prove $\mathsf{UB}_k^{i.o.}(h)$. Similarly, they established that $\mathsf{S}_2^1 \nvdash \mathsf{NP} \subseteq \mathsf{i.o.SIZE}[n^k]$ and $\mathsf{T}_2^1 \nvdash \mathsf{P}^{\mathsf{NP}} \subseteq \mathsf{i.o.SIZE}[n^k]$.

Building on these results, [CKKO21] introduced a modular framework to establish the unprovability of circuit upper bounds in bounded arithmetic using a learning-theoretic perspective. Next, we describe how their approach can be used to show a slightly weaker form of the result from [BKO20] described above. For an $\mathcal{L}_{\mathsf{PV}}$-function symbol h, we consider a sentence $\mathsf{UB}_{c,k}(h)$ stating that $L_h \in \mathsf{SIZE}[c \cdot n^k]$, where $x \in L_h$ if and only if $h(x) \neq 0$, i.e.,

$$\mathsf{UB}_{c,k}(h) \triangleq \forall 1^n\, \exists C_n\, \forall x \left(|C_n| \leq c \cdot n^k \wedge (|x| \leq n \rightarrow (\mathsf{Eval}(C_n, x, n) = 1 \leftrightarrow h(x) \neq 0))\right). \tag{2}$$

Our goal is to show that there is a function symbol h such that, for no choice of $c \geq 1$, PV_1 proves $\mathsf{UB}_{c,k}(h)$. (Note that in all results discussed in this section, we consider Log formalizations, as explained in Section 4.1.)

Overview of the Approach. Note that $\mathsf{UB}_{c,k}(h)$ claims the *existence* of circuits for L_h, i.e., it states a *non-uniform* upper bound. We explore the constructive aspect of PV_1 proofs, by extracting computational information from a PV_1-proof that such circuits exist. The argument has a *logical component*, where we extract from a proof of $\mathsf{UB}_{c,k}(h)$ a "LEARN-uniform" construction of a sequence $\{C_n\}_n$ of circuits for L_h, and a *complexity-theoretic component*, where we unconditionally establish that for each k LEARN-uniform circuits of this form do not exist for some h. Altogether, we get that for some h theory PV_1 does not prove $\mathsf{UB}_{c,k}(h)$ (no matter the choice of c).

LEARN-uniform circuits. We will be interested in languages that can be efficiently learned with a bounded number of equivalence queries, in the following sense. For functions $s, q\colon \mathbb{N} \rightarrow \mathbb{N}$, we say that a language $L \subseteq \{0,1\}^*$ is in $\mathsf{LEARN\text{-}uniform}^{\mathsf{EQ}[q]}\ \mathsf{SIZE}[s]$ if there is a polynomial-time algorithm $A^{\mathsf{EQ}(L_n)}(1^n)$ that outputs a circuit of size at most $s(n)$ for L_n after making at most $q(n)$ equivalence queries to L_n, where $L_n = L \cap \{0,1\}^n$. The equivalence query oracle, given the description of an n-bit circuit D of size a most $s(n)$, replies "yes" if D computes L_n, or provides some counter-example w such that $D(w) \neq L_n(w)$.

Extracting LEARN-uniform circuits from PV_1 proofs. For convenience, write $\mathsf{UB}_{c,k}(h) = \forall 1^n\, \exists C_n\, \forall x\ \phi(1^n, C_n, x)$ in Equation (2), where $\phi(1^n, C_n, x)$ is free of quantifiers. Since PV_1 is a universal theory, under the assumption that $\mathsf{PV}_1 \vdash \mathsf{UB}_{c,k}(h)$, we can apply Theorem 3.2 (KPT Witnessing Theorem) to obtain the provability in PV_1 of the disjunction

$$\forall 1^n\, \forall x_1 \ldots \forall x_k \left(\phi(1^n, t_1(1^n), x_1) \vee \phi(1^n, t_2(1^n, x_1), x_2) \vee \cdots \vee \phi(1^n, t_k(1^n, x_1, \ldots, x_{k-1}), x_k)\right), \tag{3}$$

where $t_1, \ldots, t_k$ are $\mathcal{L}_{\mathsf{PV}}$-terms and $k = O(1)$. Most importantly, due to the soundness of PV_1, this statement is true over the standard model $\mathbb{N}$. Additionally, the terms in PV_1 correspond to polynomial-time

[9] Recall that 1^n is simply a convenient notation to refer to a variable n that is set to $|N|$ for some variable N.

algorithms. Next, we will discuss how to interpret Equation (3) over $\mathbb{N}$ as an interactive protocol and how this perspective leads to a LEARN-uniform construction.

The KPT Witnessing Theorem can be intuitively understood as follows [KPS90]. Consider a search problem $Q(1^n)$, where given the input 1^n, we need to find D such that $\forall x\, \phi(1^n, D, x)$. The problem $Q(1^n)$ can be solved using a k-round *Student-Teacher protocol*. In the first round, the student proposes $D_1 = t_1(1^n)$ as a solution to the search problem $Q(1^n)$. This solution is either correct, or there exists a counterexample w_1 such that $\neg\phi(1^n, t_1(1^n), w_1)$. The teacher then provides this counterexample value w_1, and the protocol moves to the next round. In each subsequent round $1 \leq i < k$, the student computes $D_i = t_i(1^n, w_1, \ldots, w_{i-1})$ based on the counterexamples $w_1, \ldots, w_{i-1}$ received in the previous rounds. This D_i is either a correct solution for $Q(1^n)$, in which case the problem is solved, or there is another counterexample w_i provided by the teacher such that $\neg\phi(1^n, t_i(1^n, w_1, \ldots, w_{i-1}), w_i)$. If the latter is the case, the protocol continues to the next round $i + 1$. The theorem guarantees that for every input 1^n, the student will successfully solve the search problem $Q(1^n)$ within some round $1 \leq i \leq k$.

From a PV_1 proof of a circuit upper bound for a language L, we can derive a Student-Teacher protocol for the search problem $Q(1^n)$ corresponding to Equation (3). In this protocol, the student proposes a candidate circuit D, and the teacher provides a counterexample w to D (an input w such that $D(w) \neq L(w)$) if one exists. (Note that $\phi(1^n, D, x)$ might not be true for other reasons, e.g., if $|D| > c \cdot n^k$, but in such cases there is no need to invoke the equivalence query oracle and we can proceed in the Student-Teacher protocol with, say, $w = 0^n$.) The student is guaranteed to succeed after at most k queries, regardless of the counterexamples provided by the teacher. Finally, for every input n, the student computes according to a constant number of fixed PV_1 terms $t_1, \ldots, t_k$. Since a PV_1 term is merely a composition of a finite number of PV_1 function symbols (polynomial-time algorithms), the student's computation runs in polynomial time. Therefore, from the provability in PV_1 of a non-uniform circuit upper bound for a language in P, we can extract a LEARN-uniform family of circuits for L.

Unconditional lower bound against LEARN-uniform circuits. The argument described above reduces the unprovability of upper bounds to a complexity-theoretic question with no reference to logic. To complete the proof, it is enough to show that for each k there is a language $L \in \mathsf{P}$ such that $L \notin \mathsf{LEARN\text{-}uniform}^{\mathsf{EQ}[O(1)]}\ \mathsf{SIZE}[O(n^k)]$. This unconditional lower bound against LEARN-uniform circuits is established in [CKKO21] by generalizing a lower bound from [SW14] against P-uniform circuits, which can be interpreted as LEARN-uniform constructions with $q = 0$ queries. Roughly speaking, [CKKO21] shows that one can eliminate each equivalence query using a small amount of non-uniform advice, and that the base case where no queries are present (as in [SW14]) can be extended to a lower bound against a bounded amount of advice.

This completes the sketch of the argument. The approach is fairly general and can be adapted to other theories. The strength of the theory affects the learning model against which one needs to obtain lower bounds (e.g., by increasing the number of queries or allowing randomized learners).

Open Problem 5.1. *Show that* S^1_2 *does not prove that* $\mathsf{P} \subseteq \mathsf{SIZE}[n^k]$.

In order to solve Open Problem 5.1, using the connection from [CKKO21] it is sufficient to show that $\mathsf{P} \not\subseteq \mathsf{LEARN\text{-}uniform}^{\mathsf{EQ}[q]}\ \mathsf{SIZE}[O(n^k)]$ for $q = \mathsf{poly}(n)$. In other words, this amounts to understanding the class of languages that admit circuits that can be produced with a polynomial number of equivalence queries.

Open Problem 5.2. *Show that* T^1_2 *does not prove that* $\mathsf{NP} \subseteq \mathsf{SIZE}[n^k]$.

5.1.2 $\mathsf{P} = \mathsf{NP}$ and Propositional Proof Complexity

Suppose that P is actually equal to NP. In this scenario, there exists a polynomial-time algorithm g (i.e., a PV_1 function symbol) that can find a satisfying assignment for any given satisfiable formula. In other words, if $\mathsf{Formula}(F, 1^n)$ denotes an $\mathcal{L}_{\mathsf{PV}}$-formula that checks if F is a valid description of a formula over n input bits, and $\mathsf{Sat}(F, x)$ is an $\mathcal{L}_{\mathsf{PV}}$-formula that checks if x satisfies the formula encoded by F, the sentence

$$\varphi_{\mathsf{P}=\mathsf{NP}}(g) \triangleq \forall 1^n \forall F \forall x \left((\mathsf{Formula}(F, 1^n) \wedge \mathsf{Sat}(F, x)) \rightarrow \mathsf{Sat}(F, g(F)) \right) \tag{4}$$

is true in the standard model $\mathbb{N}$.

Open Problem 5.3. *Show that for no polynomial-time function symbol g theory PV_1 proves the sentence $\varphi_{\mathsf{P}=\mathsf{NP}}(g)$.*

Equivalently, Open Problem 5.3 states that PV_1 (and by standard conservation results S^1_2) is consistent with $\mathsf{P} \neq \mathsf{NP}$. This means that either $\mathsf{P} \neq \mathsf{NP}$, as is commonly assumed, making the conjecture trivially true, or $\mathsf{P} = \mathsf{NP}$, but this cannot be proven using only polynomial-time concepts and reasoning. Therefore, Open Problem 5.3 represents a formal weakening of the conjecture that $\mathsf{P} \neq \mathsf{NP}$. The statement is known to follow from the purely combinatorial conjecture that the extended Frege propositional proof system is not polynomially bounded, which is a major open problem in proof complexity.

Theorem 5.4 ([Coo75])**.** *Suppose that there is a sequence $\{F_n\}_{n \geq 1}$ of propositional tautologies of size polynomial in n that require $e\mathcal{F}$ proofs of size $n^{\omega(1)}$. Then there is no function symbol g such that PV_1 proves $\varphi_{\mathsf{P}=\mathsf{NP}}(g)$.*

Proof. Here we only provide a sketch of the proof. More details and extensions of the result can be found in the textbooks [Kra95, Kra19]. We establish that if $\mathsf{PV}_1 \vdash \varphi_{\mathsf{P}=\mathsf{NP}}(g)$ for some g, then every tautology has a polynomial size $e\mathcal{F}$ proof.

Recall the definitions and results from Section 3.2. For a propositional proof system P (described by an $\mathcal{L}_{\mathsf{PV}}$ function symbol), we consider an $\mathcal{L}_{\mathsf{PV}}$-sentence stating the soundness of P:

$$\mathsf{Sound}_P \triangleq \forall 1^n \forall F \exists \pi \left(\mathsf{Formula}(F, 1^n) \wedge \mathsf{Proof}_P(F, \pi) \right) \rightarrow \forall x \left(|x| \leq n \rightarrow \mathsf{Sat}(F, x) \right),$$

where $\mathsf{Proof}_P(F, \pi)$ states that π is a valid P-proof of F. Given g, we consider a proof system P_g defined as follows: Given a valid description of an n-bit propositional formula F and a candidate proof $\widetilde{\pi}$, P_g accepts $\widetilde{\pi}$ as a proof of F if and only if

$$g(\neg F) = \widetilde{\pi} \quad \text{and} \quad \neg \mathsf{Sat}(\neg F, \widetilde{\pi}),$$

where $\neg F$ represents the negation of F. Observe that for any tautology F, $\pi_F \triangleq g(\neg F)$ is a valid P_g-proof of F.

Note that $\mathsf{PV}_1 \vdash \mathsf{Sound}_{P_g}$, which follows from the provability of Equation (4) and the definition of P_g using g. Now consider the quantifier-free $\mathcal{L}_{\mathsf{PV}}$-formula

$$\psi \triangleq \neg \mathsf{Formula}(F, 1^n) \vee \neg \mathsf{Proof}_{P_g}(F, \pi) \vee |x| > n \vee \mathsf{Sat}(F, x).$$

The provability of $\forall 1^n \forall F \forall \pi \, \psi$ in PV_1 follows from the provability of Sound_{P_g}.

Using Cook's translation (Section 3.2), the sequence of propositional formulas $||\psi||_m$ admits $e\mathcal{F}$-proofs of polynomial size. Moreover, given an actual n-bit propositional formula F of polynomial size and the corresponding P_g-proof π_F (represented by fixed strings $\langle F \rangle$ and $\langle \pi_F \rangle$), one can show that there are polynomial

size $e\mathcal{F}$ proofs of both $||\mathsf{Formula}(\langle F\rangle, 1^n)||_{\mathsf{poly}(n)}$ and $||\mathsf{Proof}_{P_g}(\langle F\rangle, \langle \pi_F\rangle)||_{\mathsf{poly}(n)}$. (Intuitively, this follows by an evaluation of the expressions on these fixed inputs.) Since $e\mathcal{F}$ is closed under substitution, we can derive in $e\mathcal{F}$ with a polynomial size proof the formula $||\mathsf{Sat}(\langle F\rangle, x)||_{\mathsf{poly}(n)}$.

Finally, for every propositional formula $F(x)$ on n-bit inputs, it is possible to efficiently prove in $e\mathcal{F}$ the propositional formula $||\mathsf{Sat}(\langle F\rangle, x)||_{\mathsf{poly}(n)} \to F(x)$. (This can be established by a slightly more general structural induction on formulas F using information about $||\cdot||$ and $\langle\cdot\rangle$.) Overall, since $e\mathcal{F}$ is closed under implication, it follows from these derivations that there is a polynomial size $e\mathcal{F}$ proof of F. This completes the sketch of the proof of the result. □

Open Problem 5.3 would also follow from a proof that Buss's hierarchy of theories T^i_2 does not collapse [KPT91], another central problem in bounded arithmetic. More precisely, it is enough to obtain the following separation.

Open Problem 5.5. *Show that for some $i > j \geq 1$ we have $\mathsf{T}^i_2 \neq \mathsf{T}^j_2$.*

It is known that PV_1 proves that $\mathsf{P} = \mathsf{NP}$ if and only if it proves that $\mathsf{NP} = \mathsf{coNP}$. Consequently, a super-polynomial lower bound on the length of $e\mathcal{F}$ proofs also yields the consistency of $\mathsf{NP} \neq \mathsf{coNP}$ with PV_1.

Finally, we remark that the use of witnessing theorems alone (as done in Section 5.1.1) is probably not sufficient to settle Open Problem 5.3. This is because these theorems typically also hold when we extend the theory with all true universal statements. Thus an unprovability argument that only employs the witnessing theorem would establish unconditionally that each sentence $\varphi_{\mathsf{P}=\mathsf{NP}}(g)$ is false and therefore $\mathsf{P} \neq \mathsf{NP}$. Some researchers interpret this as evidence that the investigation of propositional proof complexity might be unavoidable. Another approach to Open Problem 5.3 is discussed in Section 5.3.

5.2 Unprovability of Lower Bounds

5.2.1 Average-Case Circuit Lower Bounds

In this section, we discuss the unprovability of strong average-case lower bounds in PV_1. We focus on an unprovability result from [PS21], stated and proved in a slightly stronger form in [LO23]. The proof is based on a technique introduced by [Kra11] and further explored in [Pic15a].

We consider an average-case separation of co-nondeterministic circuits against non-deterministic circuits of subexponential size. In more detail, we investigate the provability of a sentence $\mathsf{LB}^1(s_1, s_2, m, n_0)$ stating that, for every input length $n \geq n_0$, there is a co-nondeterministic circuit C of size $\leq s_1(n)$ such that, for every nondeterministic circuit D of size $\leq s_2(n)$, we have

$$\Pr_{x\sim\{0,1\}^n}\Big[C(x) = D(x)\Big] \;\leq\; 1 - \frac{m(n)}{2^n}.$$

Let $\mathsf{coNSIZE}[s(n)]$ and $\mathsf{NSIZE}[s(n)]$ refer to co-nondeterministic circuits and nondeterministic circuits of size $s(n)$, respectively. More formally, $\mathsf{LB}^1(s_1, s_2, m, n_0)$ is an $\mathcal{L}_{\mathsf{PV}}$-sentence capturing the following lower bound statement:

$$\begin{aligned}&\forall n \in \mathsf{LogLog} \text{ with } n \geq n_0\ \exists C \in \mathsf{coNSIZE}[s_1(n)]\ \forall D \in \mathsf{NSIZE}[s_2(n)]\\ &\exists m = m(n) \text{ distinct } n\text{-bit strings } x^1, \ldots, x^m \text{ s.t. } \mathsf{Error}(C, D, x^i) \text{ for all } i \in [m],\end{aligned}$$

where $\mathsf{Error}(C, D, x)$ means that the circuits C and D disagree on the input x. This statement can be seen as an average-case form of the $\mathsf{coNP} \not\subseteq \mathsf{NP/poly}$ conjecture if we let $s_1(n) = n^{O(1)}$, $s_2(n) = n^{\omega(1)}$, and

$m(n) = 2^n/n$. (Note that we consider in this section a LogLog formalization, according to the notation explained in Section 4.1.)

Theorem 5.6 ([PS21, LO23]). *Let $d \geq 1$, $\delta > 0$, and $n_0 \geq 1$ be arbitrary parameters, and let $s_1(n) = n^d$, $s_2(n) = 2^{n^\delta}$, and $m(n) = 2^n/n$. Then PV_1 does not prove the sentence $\mathsf{LB}^1(s_1, s_2, m, n_0)$.*

In the remainder of this section, we provide some intuition about the proof of this result.

Overview of the Argument. Suppose, towards a contradiction, that $\mathsf{PV}_1 \vdash \mathsf{LB}^1(s_1, s_2, m, n_0)$ with parameters as above. The central idea of the argument is that establishing a strong complexity *lower bound* within bounded arithmetic leads to a corresponding complexity *upper bound*. These lower and upper bounds *contradict each other*. Consequently, this contradiction implies the unprovability of the lower bound statement. In a bit more detail, the argument proceeds as follows:

(*i*) The provability of the average-case lower bound sentence $\mathsf{LB}^1(s_1, s_2, m, n_0)$ implies the provability of a *worst-case* lower bound for $\mathsf{coNSIZE}[n^d]$ against $\mathsf{NSIZE}[2^{n^\delta}]$. We formalize the latter by a sentence $\mathsf{LB}^1_{\mathsf{wst}}(s_1, s_2, n_0)$.

(*ii*) Given any proof of $\mathsf{LB}^1_{\mathsf{wst}}(s_1, s_2, n_0)$ in PV_1, we extract a *complexity upper bound* for an *arbitrary* co-nondeterministic circuit $E_m(x)$ over an input x of length m and of size at most $\mathsf{poly}(m)$. More precisely, we show that there is a deterministic circuit B_m of size $\leq 2^{m^{o(1)}}$ such that $\Pr_{x \sim \{0,1\}^m}[E_m(x) = B_m(x)] \geq 1/2 + 2^{-m^{o(1)}}$.

(*iii*) We invoke an existing hardness amplification result to conclude that, on any large enough input length n, *every* co-nondeterministic circuit C_n of size $\leq n^d$ agrees with some nondeterministic circuit D_n of size $\leq 2^{n^\delta}$ on more than a $1 - 1/n$ fraction of the inputs.

Since PV_1 is a *sound* theory, i.e., every theorem of PV_1 is a true sentence, Item (*iii*) is in contradiction with the complexity lower bound stated in $\mathsf{LB}^1(s_1, s_2, m, n_0)$. Consequently, PV_1 does not prove this sentence.

The most interesting step of the argument is the proof of Item (*ii*). The key point is that the proof of a lower bound in PV_1 must be somewhat constructive, in the sense that it not only shows that every small circuit D fails to solve the problem but also produces a string w witnessing this fact. Below we give a simple example of its usefulness, showing a setting where a constructive lower bound yields an upper bound. Note that the application of a witnessing theorem to a LogLog formalization provides algorithms running in time $\mathsf{poly}(2^n)$. The example provided next shows that this is still useful.

Lemma 5.7 ([CLO24a]). *Let $L \in \mathsf{NP}$. Suppose that there is a uniform algorithm $R(1^n, D)$ such that, for every co-nondeterministic circuit D on n input variables and of size at most $n^{\log n}$, $R(1^n, D)$ runs in time $2^{O(n)}$ and outputs a string $w \in \{0,1\}^n$ such that $D(w) \neq L(w)$. Then, for every language $L' \in \mathsf{NP}$ and for every constant $\varepsilon > 0$, we have $L' \in \mathsf{DTIME}[2^{n^\varepsilon}]$.*

Proof. Suppose that $L \in \mathsf{NTIME}[n^d]$ for some $d \in \mathbb{N}$. Let M' be a nondeterministic machine that decides L' and runs in time at most $n^{c'}$, where $c' \in \mathbb{N}$. Let $\varepsilon > 0$ be an arbitrary constant. Let $\gamma = \gamma(d, \varepsilon) > 0$ be a small enough constant to be defined later. Finally, let R be the algorithm provided by the hypothesis of the lemma. We show that the following deterministic algorithm $B^\gamma(x)$ decides L' in time $O(2^{n^\varepsilon})$:

Input: $x \in \{0,1\}^n$ for some $n \geq 1$.

1 Compute the description of a co-nondeterministic circuit E' of size at most $n^{2c'}$ that decides the complement of L';
`// In other words,` $E'(u) = 1 - L'(u)$ `for every string` $u \in \{0,1\}^n$.

2 Produce the description of a co-nondeterministic circuit $D_x(y)$, where $y \in \{0,1\}^{n^\gamma}$, such that $D_x(y)$ ignores its input y and computes according to $E'(x)$;
`// While the length of` y `is smaller than the length of` u, D_x `and` E' `share the same nondeterministic input string, and` E' `sets` u `to be the fixed string` x.

3 Compute $w = R(1^{n^\gamma}, D_x) \in \{0,1\}^{n^\gamma}$;

4 Determine the bit $b = L(w)$ by a brute force computation, then **return** b;

Algorithm 2: Algorithm $B^\gamma(x)$ for deciding language L'.

First, we argue that B^γ decides L'. Since D_x is a co-nondeterministic circuit over inputs of length $m \triangleq n^\gamma$ and has size at most $n^{2c'} = m^{2c'/\gamma} \leq m^{\log m}$ (for a large enough m), $R(1^{n^\gamma}, D_x)$ outputs a string $w \in \{0,1\}^{n^\gamma}$ such that $L(w) = 1 - D_x(w)$. Consequently,

$$b = L(w) = 1 - D_x(w) = 1 - E'(x) = 1 - (1 - L'(x)) = L'(x) \ ,$$

i.e., the output bit of $B^\gamma(x)$ is correct.

Next, we argue that B^γ runs in time at most $O(2^{n^\varepsilon})$. Clearly, Steps 1–2 run in $\mathsf{poly}(n)$ time. Moreover, Step 3 runs in time $2^{O(n^\gamma)}$ under the assumption on the running time of $R(1^{n^\gamma}, D_x)$. This is at most 2^{n^ε} if we set $\gamma \leq \varepsilon/2$. Finally, since $L \in \mathsf{NTIME}[n^d]$, the brute force computation in Step 4 can be performed in deterministic time $2^{O(\ell^d)}$ over an input of length ℓ. Since $\ell = n^\gamma = |w|$ in our case, if $\gamma \leq \varepsilon/2d$ we get that Step 4 runs in time at most 2^{n^ε}. Overall, if we set $\gamma \triangleq \varepsilon/2d$, it follows that B^γ runs in time at most $O(2^{n^\varepsilon})$. This completes the proof that $L' \in \mathsf{DTIME}[2^{n^\varepsilon}]$. □

The proof of Item (*ii*) is significantly more sophisticated, since one does not get an algorithm R as above from a PV_1 proof of the lower bound sentence $\mathsf{LB}^1(s_1, s_2, m, n_0)$. The argument combines a witnessing theorem for sentences with more than four quantifier alternations and an ingenious technique from [Kra11] that relies on ideas from the theory of computational pseudorandomness.

Open Problem 5.8. *Strengthen the unprovability result from Theorem 5.6 in the following directions:*

(*a*) *show that it holds in the polynomial size regime, i.e., with $s_1(n) = n^a$ and for some $s_2(n) = n^b$;*

(*b*) *establish the unprovability of worst-case lower bounds against nondeterministic circuits;*

(*c*) *show the unprovability of average-case lower bounds against deterministic circuits;*

(*d*) *establish the same result with respect to a stronger theory.*

We refer to [LO23, CLO24a, CLO24b] for some related results and partial progress.

5.2.2 Extended Frege Lower Bounds

This section covers a result on the unprovability of super-polynomial size extended Frege ($e\mathcal{F}$) lower bounds in PV_1 [KP89] (see also [CU93, Bus90]). We refer to Section 3.2 for the necessary background. We will also need the definitions and results from Section 3.3.

We adapt the presentation from [Kra19]. Consider the theory PV_1 and its language $\mathcal{L}_{\mathsf{PV}}$. We shall use the following $\mathcal{L}_{\mathsf{PV}}$ formulas:

- $\mathsf{Sat}(x, y)$: a quantifier-free formula formalizing that y is a satisfying assignment of the Boolean formula x;
- $\mathsf{Taut}(x) \triangleq \forall y \leq x\, \mathsf{Sat}(x, y)$;
- $\mathsf{Proof}_P(x, z)$: a quantifier-free formula formalizing that z is a P-proof of x.

The following lemma is central to the unprovability result.

Lemma 5.9. *Let $M \models \mathsf{PV}_1$, and assume that $\phi \in M$ is a propositional formula. The following statements are equivalent:*

(*i*) *There is no $e\mathcal{F}$-proof of ϕ in M:*

$$M \models \forall z\, \neg \mathsf{Proof}_{e\mathcal{F}}(\phi, z)\,.$$

(*ii*) *There is an extension $M' \supseteq M$ (also a model of PV_1) in which ϕ is falsified:*

$$M' \models \exists y\, \mathsf{Sat}(\neg\phi, y)\,.$$

The proof of Lemma 5.9 proceeds by compactness and uses that the correctness of the propositional translation from PV_1 to $e\mathcal{F}$ (Section 3.2) is also provable in PV_1.

Lemma 5.10. *Let M be a nonstandard countable model of PV_1. Then it has a cofinal extension $M' \supseteq_{\mathsf{cf}} M$ (also a model of PV_1) such that every tautology in M' has an $e\mathcal{F}$-proof in M'.*

The proof of Lemma 5.10 iterates Lemma 5.9 while taking cuts to ensure that the limit extension $M' = \bigcup_i M_i$ (where $M_0 = M$) is cofinal in M. Since each $M_i \models \mathsf{PV}_1$ and PV_1 is universal, we also have $M' \models \mathsf{PV}_1$.

We will need the following analogue of Lemma 3.6 for PV_1.

Fact 5.11. *Let M_0 be a nonstandard countable model of PV_1. Then there is a (countable) cut M of M_0 that is a model of PV_1 and a length $n \in M$ such that for every $b \in M$ we have $M \models |b| \leq n^k$ for some standard number k.*

The next result is a consequence of Lemma 5.10 and Fact 5.11.

Corollary 5.12. *Let M be a nonstandard countable model of PV_1. There is a model M^* of PV_1 such that the following properties hold:*

(*i*) *Any tautology in M^* has an $e\mathcal{F}$-proof in M^*.*

(*ii*) *There is a nonstandard element $a \in M$ of length $n \triangleq |a|$ such that for any element $b \in M^*$ there is a standard number k such that $M^* \models |b| \leq n^k$.*

Theorem 5.13 (Unprovability of super-polynomial size $e\mathcal{F}$ lower bounds in PV_1 [KP89])**.** *Consider the sentence*

$$\Psi_{e\mathcal{F}} \triangleq \forall x\, \exists \phi \geq x\, [\mathsf{Taut}(\phi) \wedge \forall \pi\, (|\pi| \leq |\phi|^{\log|\phi|} \rightarrow \neg \mathsf{Proof}_{e\mathcal{F}}(\phi, \pi))]\,.$$

The sentence $\Psi_{e\mathcal{F}}$ is not provable in PV_1.

Proof. Suppose $\mathsf{PV}_1 \vdash \Psi_{e\mathcal{F}}$. Fix a countable nonstandard model M of PV_1. Let M^*, a, and $n \triangleq |a|$ be as in Corollary 5.12. Since $\Psi_{e\mathcal{F}}$ holds in M^*, there is a tautology $\phi \in M^*$ with $\phi \geq a$ and consequently $|\phi| \geq n$ such that ϕ does not have an $e\mathcal{F}$-proof of size $|\phi|^{\log|\phi|}$ in M^*. In particular, ϕ does not have a proof of size $n^{\log n}$. On the other hand, by the two properties of M^* the formula ϕ has an $e\mathcal{F}$-proof of size at most n^k for some standard number k. Finally, since n is nonstandard, we have $n^k \leq n^{\log n}$. This contradiction implies that PV_1 does not prove $\Psi_{e\mathcal{F}}$. □

Open Problem 5.14. *Show that* PV_1 *cannot prove fixed-polynomial size lower bounds on the length of* $e\mathcal{F}$ *proofs.*

Open Problem 5.15. *Establish the unprovability of the sentence* $\Psi_{e\mathcal{F}}$ *in theory* S^1_2.

5.3 Connection Between Upper Bounds and Lower Bounds

In this section, we explain a result from [BKO20] showing that the unprovability of $\mathsf{P} = \mathsf{NP}$ (Open Problem 5.3) is related to the unprovability of circuit lower bounds. For a PV_1 function symbol h and a circuit size parameter $k \in \mathbb{N}$, consider the sentence

$$\mathsf{LB}^{a.e.}_k(h) \triangleq \neg\mathsf{UB}^{i.o.}_k(h) ,$$

where $\mathsf{UB}^{i.o.}_k(h)$ is the sentence defined in Section 5.1.1. The sentence $\mathsf{LB}^{a.e.}_k(h)$ states that the language defined by h is hard on input length n for circuits of size n^k whenever n is sufficiently large.

Theorem 5.16 (Unprovability of $\mathsf{P} = \mathsf{NP}$ in PV_1 from the unprovability of lower bounds in PV_1 [BKO20])**.** *If there exists* $k \in \mathbb{N}$ *such that for no function symbol* h *theory* PV_1 *proves the sentence* $\mathsf{LB}^{a.e.}_k(h)$*, then for no function symbol* f *theory* PV_1 *proves the sentence* $\varphi_{\mathsf{P}=\mathsf{NP}}(f)$.

Theorem 5.16 shows that if PV_1 does not prove n^k-size lower bounds for a language in P, then $\mathsf{P} \neq \mathsf{NP}$ is consistent with PV_1. Note that the hypothesis of Theorem 5.16 is weaker than the assumption that PV_1 does not prove that $\mathsf{NP} \not\subseteq \mathsf{SIZE}[n^k]$ for some k.

Sketch of the proof of Theorem 5.16. We proceed in the contrapositive. We formalize in PV_1 the result that if $\mathsf{P} = \mathsf{NP}$, then for any parameter k, $\mathsf{P} \not\subseteq \mathsf{i.o.SIZE}[n^k]$ (see, e.g., [Lip94, Theorem 3]). This result combines the collapse of PH to P with Kannan's argument [Kan82] that PH can define languages that are almost-everywhere hard against circuits of fixed-polynomial size. Typically, proving this claim requires showing the existence of a truth table of size 2^n that is hard against circuits of size n^k. However, this result might not be provable in PV_1.

We address this issue as follows. From the provability in PV_1 that $\mathsf{P} = \mathsf{NP}$, it follows that for each $i \geq 1$ theory T^i_2 collapses to PV_1 [KPT91]. Recall that the dual weak pigeonhole principle (dWPHP) for $\mathcal{L}_{\mathsf{PV}}$-functions is provable in T^2_2. Define a PV_1 function symbol g that takes as input a circuit C of size n^k and outputs the lexicographic first n^{k+1} bits of the truth table computed by C. From $\mathsf{dWPHP}(g)$, we now derive in PV_1 that the prefix of some truth table is not computable by circuits of size n^k, if n is sufficiently large. We can implicitly extend this truth table prefix with zeroes and use the resulting truth table to define a PV_1-formula $\varphi(x)$ with a constant number of bounded quantifiers that defines a language L that is hard against circuits of size n^k, where the hardness is provable in PV_1.

Given that the provability in PV_1 that $\mathsf{P} = \mathsf{NP}$ implies the provability in PV_1 that PH collapses to P, it follows that $\varphi(x)$ is equivalent in PV_1 to the language defined by some $\mathcal{L}_{\mathsf{PV}}$-function h. In other words, $\mathsf{PV}_1 \vdash \mathsf{LB}^{a.e.}_k(h)$, which completes the proof of Theorem 5.16. □

[CLO24b] shows an example of a simple lower bound that is not provable in PV_1, under a plausible cryptographic assumption. This indicates that Theorem 5.16 might offer a viable approach towards a solution to Open Problem 5.3.

6 Additional Recent Developments

The provability of the dual Weak Pigeonhole Principle (dWPHP) for polynomial-time functions is closely related to the provability of exponential circuit lower bounds for a language in deterministic exponential time [Jeř07]. [Kra21] showed that dWPHP cannot be proved in PV_1 under the assumption that $\mathsf{P} \subseteq \mathsf{SIZE}[n^k]$ for some constant k. [ILW23] established the same unprovability result assuming sub-exponentially secure indistinguishability obfuscation and $\mathsf{coNP} \not\subseteq \mathsf{i.o.AM}$.

[ABM23] established the unprovability of $\mathsf{NEXP} \subseteq \mathsf{SIZE}[\mathsf{poly}]$ in the theory of bounded arithmetic V^0_2 (not covered in this survey). Interestingly, their approach does not employ a witnessing theorem. It proceeds instead by simulating a comprehension axiom scheme assuming the provability of the upper bound sentence, eventually relying on an existing lower bound on the provability of the pigeonhole principle.

[CLO24b] systematically investigates the reverse mathematics of complexity lower bounds. They demonstrated that various lower bound statements in communication complexity, error-correcting codes, and for Turing machines are equivalent to well-studied combinatorial principles, such as the weak pigeonhole principle for polynomial-time functions and its variants. Consequently, complexity lower bounds can be regarded as fundamental axioms with significant implications. They use these equivalences to derive conditional results on the unprovability of lower bounds.

[CKK+24] investigates the provability of the circuit size hierarchy in bounded arithmetic, captured by a sentence CSH stating that for each $n \geq n_0$, there is a circuit of size n^a that does not admit an equivalent circuit of size n^b, where $a > b > 1$ and n_0 are fixed. They showed that CSH is provable in T^2_2, while its provability in T^1_2 implies that $\mathsf{P}^{\mathsf{NP}} \not\subseteq \mathsf{SIZE}[n^{1+\varepsilon}]$ for some $\varepsilon > 0$. Thus a better proof complexity upper bound for the circuit size hierarchy yields new circuit lower bounds.

[Kra24] offers a comprehensive reference on proof complexity generators, whose investigation is closely related to dWPHP and its provability in bounded arithmetic. The theory of proof complexity generators offers tautologies that serve as potential candidates for demonstrating super-polynomial extended Frege lower bounds and consequently the unprovability of $\mathsf{P} = \mathsf{NP}$ in PV_1.

We have not covered a number of results connected to the meta-mathematics of complexity lower bounds developed in the context of propositional proof complexity (see, e.g., [Raz15, Kra19, AR23, Kra24] and references therein). It is worth noting that results on the non-automatability of weak proof systems such as [AM20, dRGN+21] were made possible thanks to the investigation of the meta-mathematics of proof complexity.

Finally, several other recent papers have investigated directions connected to bounded arithmetic and the meta-mathematics of complexity theory, e.g., [PS22, Kha22, PS23, AKPS24, LLR24]. Due to space constraints, we are not able to cover all recent developments in this survey.

Acknowledgements. I would like to thank Jan Krajíček, Mykyta Narusevych, Ján Pich, and Dimitrios Tsintsilidas for their valuable comments and feedback on an earlier version of this survey. This work received support from the Royal Society University Research Fellowship URF\R1\191059; the UKRI Frontier Research Guarantee EP/Y007999/1; and the Centre for Discrete Mathematics and its Applications (DIMAP) at the University of Warwick.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[ABM23] Albert Atserias, Samuel R. Buss, and Moritz Müller. On the consistency of circuit lower bounds for non-deterministic time. In *Symposium on Theory of Computing* (STOC), pages 1257–1270, 2023.

[AKPS24] Noel Arteche, Erfan Khaniki, Ján Pich, and Rahul Santhanam. From proof complexity to circuit complexity via interactive protocols. In *International Colloquium on Automata, Languages, and Programming* (ICALP), 2024.

[AM20] Albert Atserias and Moritz Müller. Automating resolution is NP-hard. *J. ACM*, 67(5):31:1–31:17, 2020.

[AR23] Per Austrin and Kilian Risse. Sum-of-squares lower bounds for the minimum circuit size problem. In *Computational Complexity Conference* (CCC), pages 31:1–31:21, 2023.

[AW09] Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *Transactions on Computation Theory* (TOCT), 1(1), 2009.

[Bey09] Olaf Beyersdorff. On the correspondence between arithmetic theories and propositional proof systems – a survey. *Mathematical Logic Quarterly*, 55(2):116–137, 2009.

[BGS75] Theodore P. Baker, John Gill, and Robert Solovay. Relativizatons of the $\mathsf{P} =? \mathsf{NP}$ Question. *SIAM J. Comput.*, 4(4):431–442, 1975.

[BKKK20] Sam R. Buss, Valentine Kabanets, Antonina Kolokolova, and Michal Koucký. Expander construction in VNC^1. *Annals of Pure and Applied Logic*, 171(7):102796, 2020.

[BKO20] Jan Bydzovsky, Jan Krajíček, and Igor C. Oliveira. Consistency of circuit lower bounds with bounded theories. *Logical Methods in Computer Science*, 16(2), 2020.

[BKT14] Samuel R. Buss, Leszek A. Kołodziejczyk, and Neil Thapen. Fragments of approximate counting. *Journal of Symbolic Logic*, 79(2):496–525, 2014.

[BM20] Jan Bydzovsky and Moritz Müller. Polynomial time ultrapowers and the consistency of circuit lower bounds. *Arch. Math. Log.*, 59(1-2):127–147, 2020.

[Bus86] Samuel R. Buss. *Bounded Arithmetic*. Bibliopolis, 1986.

[Bus90] Samuel R. Buss. On model theory for intuitionistic bounded arithmetic with applications to independence results. In *Feasible Mathematics: A Mathematical Sciences Institute Workshop, Ithaca, New York, June 1989*, pages 27–47. Springer, 1990.

[Bus94] Samuel R. Buss. On herbrand's theorem. In *Selected Papers from the Logic and Computational Complexity International Workshop* (LCC), pages 195–209, 1994.

[Bus97] Samuel R. Buss. Bounded arithmetic and propositional proof complexity. In *Logic of Computation*, pages 67–121. Springer Berlin Heidelberg, 1997.

[CHO+22] Lijie Chen, Shuichi Hirahara, Igor C. Oliveira, Ján Pich, Ninad Rajgopal, and Rahul Santhanam. Beyond natural proofs: Hardness magnification and locality. *J. ACM*, 69(4):25:1–25:49, 2022.

[CIKK16] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In *Conference on Computational Complexity* (CCC), pages 10:1–10:24, 2016.

[CJSW21] Lijie Chen, Ce Jin, Rahul Santhanam, and Ryan Williams. Constructive separations and their consequences. In *Symposium on Foundations of Computer Science* (FOCS), 2021.

[CK07] Stephen A. Cook and Jan Krajíček. Consequences of the provability of $\mathsf{NP} \subseteq \mathsf{P/poly}$. *Journal of Symbolic Logic*, 72(4):1353–1371, 2007.

[CKK+24] Marco Carmosino, Valentine Kabanets, Antonina Kolokolova, Igor C. Oliveira, and Dimitrios Tsintsilidas. Provability of the circuit size hierarchy and its consequences. Preprint, 2024.

[CKKO21] Marco Carmosino, Valentine Kabanets, Antonina Kolokolova, and Igor C. Oliveira. Learn-uniform circuit lower bounds and provability in bounded arithmetic. In *Symposium on Foundations of Computer Science* (FOCS), 2021.

[CLO24a] Lijie Chen, Jiatu Li, and Igor C. Oliveira. On the unprovability of circuit size bounds in intuitionistic S^1_2. Preprint: arXiv:2404.11841, 2024.

[CLO24b] Lijie Chen, Jiatu Li, and Igor C. Oliveira. Reverse mathematics of complexity lower bounds. In *Symposium on Foundations of Computer Science* (FOCS), 2024.

[CN10] Stephen A. Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity*. Cambridge University Press, 2010.

[Cob65] Alan Cobham. The intrinsic computational difficulty of functions. *Proc. Logic, Methodology and Philosophy of Science*, pages 24–30, 1965.

[Coo75] Stephen A. Cook. Feasibly constructive proofs and the propositional calculus (preliminary version). In *Symposium on Theory of Computing* (STOC), pages 83–97, 1975.

[CU93] Stephen Cook and Alasdair Urquhart. Functional interpretations of feasibly constructive arithmetic. *Annals of Pure and Applied Logic*, 63(2):103–200, 1993.

[Din07] Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007.

[dRGN+21] Susanna F. de Rezende, Mika Göös, Jakob Nordström, Toniann Pitassi, Robert Robere, and Dmitry Sokolov. Automating algebraic proof systems is NP-hard. In *Symposium on Theory of Computing* (STOC), pages 209–222, 2021.

[Gay23] Azza Gaysin. Proof complexity of CSP. ArXiv e-Print arXiv:2201.00913, 2023.

[Gay24] Azza Gaysin. Proof complexity of universal algebra in a CSP dichotomy proof. ArXiv e-Print arXiv:2403.06704, 2024.

[HP93] Petr Hájek and Pavel Pudlák. *Metamathematics of first-order arithmetic*. Springer-Verlag, 1993.

[ILW23] Rahul Ilango, Jiatu Li, and Ryan Williams. Indistinguishability obfuscation, range avoidance, and bounded arithmetic. In *Symposium on Theory of Computing* (STOC), pages 1076–1089. ACM, 2023.

[Jeř04] Emil Jeřábek. Dual weak pigeonhole principle, boolean complexity, and derandomization. *Annals of Pure and Applied Logic*, 129(1-3):1–37, 2004.

[Jeř05] Emil Jeřábek. *Weak pigeonhole principle and randomized computation*. PhD thesis, Charles University in Prague, 2005.

[Jeř06] Emil Jeřábek. The strength of sharply bounded induction. *Mathematical Logic Quarterly*, 52(6):613–624, 2006.

[Jeř07] Emil Jeřábek. Approximate counting in bounded arithmetic. *Journal of Symbolic Logic*, 72(3):959–993, 2007.

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

[Kan82] Ravi Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control*, 55(1-3):40–56, 1982.

[Kha22] Erfan Khaniki. Nisan-Wigderson generators in proof complexity: New lower bounds. In *Computational Complexity Conference* (CCC), pages 17:1–17:15, 2022.

[KO17] Jan Krajíček and Igor C. Oliveira. Unprovability of circuit upper bounds in Cook's theory PV. *Logical Methods in Computer Science*, 13(1), 2017.

[KP89] Jan Krajíček and Pavel Pudlák. Propositional provability and models of weak arithmetic. In *CSL'89: Proceedings of the 3rd Workshop on Computer Science Logic*, pages 193–210, 1989.

[KPS90] Jan Krajíček, Pavel Pudlák, and Jiří Sgall. Interactive computations of optimal solutions. In *International Symposium on Mathematical Foundations of Computer Science* (MFCS), volume 452, pages 48–60, 1990.

[KPT91] Jan Krajíček, Pavel Pudlák, and Gaisi Takeuti. Bounded arithmetic and the polynomial hierarchy. *Annals of Pure and Applied Logic*, 52(1-2):143–153, 1991.

[Kra95] Jan Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1995.

[Kra97] Jan Krajícek. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *J. Symb. Log.*, 62(2):457–486, 1997.

[Kra11] Jan Krajícek. On the proof complexity of the Nisan-Wigderson generator based on a hard NP $\cap$ coNP function. *Journal of Mathematical Logic*, 11(1), 2011.

[Kra19] Jan Krajíček. *Proof Complexity*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2019.

[Kra21] Jan Krajíček. Small circuits and dual weak PHP in the universal theory of p-time algorithms. *ACM Transactions on Computational Logic* (TOCL), 22(2):1–4, 2021.

[Kra24] Jan Krajícek. *Proof Complexity Generators*. Monograph available at https://www.karlin.mff.cuni.cz/~krajicek/gdraft.html, 2024.

[LC11] Dai Tri Man Le and Stephen A. Cook. Formalizing randomized matching algorithms. *Log. Methods Comput. Sci.*, 8(3), 2011.

[Lip94] Richard J. Lipton. Some consequences of our failure to prove non-linear lower bounds on explicit functions. In *Structure in Complexity Theory Conference* (CCC), pages 79–87, 1994.

[LLR24] Jiawei Li, Yuhao Li, and Hanlin Ren. Meta-mathematics of resolution lower bounds: A TFNP perspective. Preprint, 2024.

[LO23] Jiatu Li and Igor C. Oliveira. Unprovability of strong complexity lower bounds in bounded arithmetic. In *Symposium on Theory of Computing* (STOC), 2023.

[Lê14] Dai Tri Man Lê. *Bounded Arithmetic and Formalizing Probabilistic Proofs*. PhD thesis, University of Toronto, 2014.

[McK10] Richard McKinley. A sequent calculus demonstration of Herbrand's theorem. *arXiv preprint arXiv:1007.3414*, 2010.

[MP20] Moritz Müller and Ján Pich. Feasibly constructive proofs of succinct weak circuit lower bounds. *Annals of Pure and Applied Logic*, 171(2), 2020.

[MPW02] Alexis Maciel, Toniann Pitassi, and Alan R. Woods. A new proof of the weak pigeonhole principle. *Journal of Computer and System Sciences*, 64(4):843–872, 2002.

[Oja04] Kerry Ojakian. *Combinatorics in Bounded Arithmetic*. PhD thesis, Carnegie Mellon University, 2004.

[Par71] Rohit Parikh. Existence and feasibility in arithmetic. *Journal of Symbolic Logic*, 36(3):494–508, 1971.

[Pic15a] Ján Pich. Circuit lower bounds in bounded arithmetics. *Annals of Pure and Applied Logic*, 166(1):29–45, 2015.

[Pic15b] Ján Pich. Logical strength of complexity theory and a formalization of the PCP theorem in bounded arithmetic. *Logical Methods in Computer Science*, 11(2), 2015.

[PS21] Ján Pich and Rahul Santhanam. Strong co-nondeterministic lower bounds for NP cannot be proved feasibly. In *Symposium on Theory of Computing* (STOC), pages 223–233, 2021.

[PS22] Ján Pich and Rahul Santhanam. Learning algorithms versus automatability of Frege systems. In *International Colloquium on Automata, Languages, and Programming* (ICALP), pages 101:1–101:20, 2022.

[PS23] Ján Pich and Rahul Santhanam. Towards $\mathrm{P} \neq \mathrm{NP}$ from extended Frege lower bounds. *Electron. Colloquium Comput. Complex.*, TR23-199, 2023.

[Pud06] Pavel Pudlák. Consistency and games - in search of new combinatorial principles. In V. Stoltenberg-Hansen and J. Väänänen, editors, *Logic Colloquium '03*, volume 24 of *Lecture Notes in Logic*, pages 244–281. ASL, 2006.

[PWW88] Jeff B. Paris, A. J. Wilkie, and Alan R. Woods. Provability of the pigeonhole principle and the existence of infinitely many primes. *J. Symb. Log.*, 53(4):1235–1244, 1988.

[Raz95a] Alexander A. Razborov. Bounded arithmetic and lower bounds in boolean complexity. In P. Clote and J. Remmel, editors, *Feasible Mathematics II*, pages 344—-386. Birkhäuser, 1995.

[Raz95b] Alexander A Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izvestiya: mathematics*, 59(1):205, 1995.

[Raz15] Alexander A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. *Annals of Mathematics*, pages 415–472, 2015.

[RR97] Alexander A. Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

[Sub61] Bella A. Subbotovskaya. Realization of linear functions by formulas using $+$, $\cdot$, $-$. In *Soviet Math. Dokl*, 1961.

[SW14] Rahul Santhanam and Ryan Williams. On uniformity and circuit lower bounds. *Computational Complexity*, 23(2):177–205, 2014.

[TC21] Iddo Tzameret and Stephen A. Cook. Uniform, integral, and feasible proofs for the determinant identities. *J. ACM*, 68(2):12:1–12:80, 2021.

[Woo81] Alan R. Woods. *Some problems in logic and number theory and their connections*. PhD thesis, University of Manchester, 1981.

[WP87] Alex J. Wilkie and Jeff B. Paris. On the scheme of induction for bounded arithmetic formulas. *Ann. Pure Appl. Log.*, 35:261–302, 1987.