Meta-Mathematics of Computational Complexity Theory

Igor C. Oliveira

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 $P = NP$. 

Contents

1 Introduction 2

2 Preliminaries 3

2.1 Complexity Theory 3

2.2 Theories of Bounded Arithmetic 4

2.2.1 $PV_1$ 4

2.2.2 $S_2^1$, $T_2^1$, and Beyond 4

2.2.3 $APC_1$ 6

3 Auxiliary Definitions and Results 6

3.1 Witnessing Theorems 6

3.2 Bounded Arithmetic and Propositional Proofs 7

3.3 Cuts of Models of Bounded Arithmetic 8

4 The Strength of Bounded Arithmetic 9

4.1 Formalization of Results from Algorithms and Complexity 9

4.2 Concrete Example: Subbotovskaya’s Formula Lower Bound in $PV_1$ 10

5 Unprovability of Complexity Bounds 14

5.1 Unprovability of Upper Bounds 14

5.1.1 LEARN-Uniform Circuits and Unprovability 14

5.1.2 $P = NP$ and Propositional Proof Complexity 17

5.2 Unprovability of Lower Bounds 18

5.2.1 Average-Case Circuit Lower Bounds 18

5.2.2 Extended Frege Lower Bounds 20

5.3 Connection Between Upper Bounds and Lower Bounds 22

6 Additional Recent Developments 23

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

2Department 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 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 \( I\Delta_0 \), 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 \( APC_1 \) [Jef04, Jef05, Jef07], 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 \( S^2_2 \) and \( T^2_2 \) [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 \((\ast)\), 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 $\text{SIZE}[s]$ 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 $f : \mathbb{N}^k \to \mathbb{N}$ 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 $\text{FP}$ denotes the set of polynomial time functions. While the definition of polynomial time refers to a machine model, $\text{FP}$ 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:

\[
\begin{align*}
  c(x) & \triangleq 0, & s(x) & \triangleq x + 1, & a(x) & \triangleq \lfloor x/2 \rfloor, & d(x) & \triangleq 2 \cdot x, & \pi_i^k(x_1, \ldots, x_k) & \triangleq x_i, & x \# y & \triangleq 2^{|x| \cdot |y|}, \\
  x \leq y & \triangleq \begin{cases} 
    1 & \text{if } x \leq y \\
    0 & \text{otherwise},
  \end{cases} & \text{Choice}(x, y, z) & \triangleq \begin{cases} 
    y & \text{if } x > 0 \\
    z & \text{otherwise}.
  \end{cases}
\end{align*}
\]

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

\[
\begin{align*}
  f(\vec{x}, 0) & = g(\vec{x}) \\
  f(\vec{x}, y) & = h(\vec{x}, y, f(\vec{x}, \lfloor y/2 \rfloor)) \\
  f(\vec{x}, y) & \leq k(\vec{x}, y)
\end{align*}
\]

for every sequence $(\vec{x}, y)$ of natural numbers. Cobham [Cob65] proved that $\text{FP}$ 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 $\mathcal{T}_C$ (i.e., when $\forall x \exists! y \psi(x, y)$ is provable, for certain formulas $\psi$) is computable within the corresponding complexity class $C$ (i.e., the function $y = f(x)$ is in $C$). 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 $\text{PV}_1$ 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: $\text{PV}_1$, $S^1_2$, $T^1_2$, and $\text{APC}$. We assume basic familiarity with first-order logic. Results claimed below without reference can be found in [Kra95].

### 2.2.1 $\text{PV}_1$

$\text{PV}_1$ [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 $\text{PV}_1$, denoted $L_{\text{PV}_1}$, contains a function symbol for each polynomial-time algorithm $f: \mathbb{N}^k \rightarrow \mathbb{N}$ (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.

$\text{PV}_1$ 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 $\text{PV}_1$ admits a formulation where all axioms are universal formulas (i.e., $\forall \bar{x} \phi(\bar{x})$, where $\phi$ is free of quantifiers). In other words, $\text{PV}_1$ is a *universal theory*.

While the details of the definition of $\text{PV}_1$ are fairly technical (see, e.g., the longer overview in [CLO24b] or the exposition in [Kra95]), such details are often not needed. In particular, $\text{PV}_1$ has an equivalent formalization that does not require Cobham’s result [Jef06].

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

While $\text{PV}_1$ 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 $L_B$ 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 $x\#y = 2^{\lfloor |x| \cdot |y| \rfloor}$).

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,

\[
\forall y \leq t(\bar{x}) \, \varphi(\bar{x}, y) \equiv \forall y \,(y \leq t(\bar{x}) \rightarrow \varphi(\bar{x}, y)), \quad \text{and} \quad \\
\exists y \leq t(\bar{x}) \, \varphi(\bar{x}, y) \equiv \exists y \,(y \leq t(\bar{x}) \land \varphi(\bar{x}, y)).
\]

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 $\Sigma^b_i$ and $\Pi^b_i$ according to the quantifier structure of the sentence, ignoring the appearance of sharply bounded quantifiers. For instance, if $\varphi \in \Sigma^b_0$ and $\psi \equiv \exists y \leq t(\bar{x}) \varphi(y, \bar{x})$, 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 \( P(\bar{x}) \) is in \( \Sigma^p_i \) (the \( i \)-th level of the polynomial hierarchy) if and only if there is a \( \Sigma^b_i \)-formula that agrees with it over \( \mathbb{N} \).

The theories introduced by Buss share a common set \( \text{BASIC} \) 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** \( T^i_2 \). This is a theory in the language \( \mathcal{L}_B \) extending \( \text{BASIC} \) by the induction axiom \( \text{IND} \)

\[
\varphi(0) \land \forall x (\varphi(x) \rightarrow \varphi(x + 1)) \rightarrow \forall x \varphi(x)
\]

for all \( \Sigma^i_1 \)-formulas \( \varphi(a) \). The formula \( \varphi(a) \) may contain other free variables in addition to \( a \).

Informally, we say that \( T^i_2 \) admits induction for \( \text{NP} \) predicates. This definition can be extended to a theory that postulates induction for \( \Sigma^b_i \)-formulas, which gives rise to the theory \( T^i_2 \).

**Theory** \( S^i_2 \). This is a theory in the language \( \mathcal{L}_B \) extending \( \text{BASIC} \) by the polynomial induction axiom \( \text{PIND} \)

\[
\varphi(0) \land \forall x \ (\varphi(\lfloor x/2 \rfloor) \rightarrow \varphi(x)) \rightarrow \forall x \varphi(x)
\]

for all \( \Sigma^i_1 \)-formulas \( \varphi(a) \). The formula \( \varphi(a) \) may contain other free variables in addition to \( a \).

Analogously to \( T^i_2 \), we can define the theories \( S^i_2 \) via polynomial induction for \( \Sigma^b_i \)-formulas. It is known that \( \text{PV}_1 \) is essentially equivalent to \( T^0_2 \) under an appropriate vocabulary and axioms [Jeř06], and that \( S^1_2 \subseteq T^2_2 \subseteq S^2_2 + \Gamma \) for every \( i \geq 1 \).

When stating and proving results in \( S^i_2 \), 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 \( \Gamma \) be a set of \( \mathcal{L}_B \)-formulas. We say that a polynomial-time function \( f : \mathbb{N}^k \rightarrow \mathbb{N} \) is \( \Gamma \text{-definable} \) in \( S^i_2 \) if there is a formula \( \psi(\bar{x}, y) \in \Gamma \) for which the following conditions hold:

1. (i) For every \( a \in \mathbb{N}^k \), \( f(\bar{a}) = b \) if and only if \( \mathbb{N} \models \varphi(\bar{a}, b) \).
2. (ii) \( S^i_2 \vdash \forall \bar{x} (\exists y (\varphi(\bar{x}, y) \land \forall z (\varphi(\bar{x}, z) \rightarrow y = z))) \).

Every function \( f \in \text{FP} \) is \( \Sigma^b_1 \)-definable in \( S^i_2 \). By adding all functions in \( \text{FP} \) to the vocabulary of \( S^1_2 \) and by extending the axioms of \( S^1_2 \) with their defining equations, we obtain a theory \( S^1_2(\mathcal{L}_{\text{FP}}) \) that can refer to polynomial-time predicates using quantifier-free formulas. \( S^1_2(\mathcal{L}_{\text{FP}}) \) proves the polynomial induction scheme for both \( \Sigma^1_1 \)-formulas and \( \Pi^1_1 \)-formulas in the extended vocabulary. \( S^1_2(\mathcal{L}_{\text{FP}}) \) is conservative over \( S^1_2 \), in the sense that any \( \mathcal{L}_B \)-sentence provable in \( S^1_2(\mathcal{L}_{\text{FP}}) \) is also provable in \( S^1_2 \).

A \( \forall \Sigma^i_1 \)-sentence is simply a sentence \( \psi = \forall \bar{x} \varphi(\bar{x}) \) where \( \varphi \in \Sigma^i_1 \). Every \( \forall \Sigma^1_1 \)-sentence provable in \( S^1_2(\mathcal{L}_{\text{FP}}) \) is also provable in \( \text{PV}_1 \). In other words, \( S^1_2(\mathcal{L}_{\text{FP}}) \) is \( \forall \Sigma^1_1 \)-conservative over \( \text{PV}_1 \). On the other hand, it is known that if \( S^1_2(\mathcal{L}_{\text{FP}}) = \text{PV}_1 \), then the polynomial-time hierarchy collapses.
2.2.3 APC

In order to formalize probabilistic methods and randomized algorithms, Jeřábek [Jef04, Jef05, Jef07] formulated the theory $\text{APC}_1$ (this terminology is from [BKT14]) by extending $\text{PV}_1$ with the dual Weak Pigeonhole Principle ($\text{dWPHP}$) for $\text{PV}_1$ functions:

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

Informally, each sentence $\text{dWPHP}(f)$ 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 $\text{T}^2_2$ [MPW02], and consequently $\text{APC}_1 \subseteq \text{T}^2_2(\mathcal{L}_{\text{PV}})$.

3 Auxiliary Definitions and Results

3.1 Witnessing Theorems

Suppose a sentence $\psi$ 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 $\psi$ 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

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

As an immediate consequence, if we apply Theorem 3.1 to $T \triangleq \text{PV}_1$, we obtain $\mathcal{L}_{\text{PV}}$-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 $\mathcal{L}$, $\phi$ 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

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

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 $\text{dWPHP}$ 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 $S^1_2(L_{PV})$ 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 $S^1_2(L_{PV})$. Second, this means that in some situations we can establish the provability of a sentence in $PV_1$ using the more convenient theory $S^1_2(L_{PV})$ (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 $L_{PV}$-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 $\pi_n$ 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 $\land, \lor$, and $\neg$. A Frege ($F$) 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. $F$ 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 ($eF$) proof system, we also allow repeated subformulas appearing in a proof to be abbreviated via new symbols.

**Cook’s Translation [Coo75].** Let $\varphi$ be a universal $L_{PV}$-sentence of the form $\varphi \triangleq \forall x \psi(x)$, where $\psi(x)$ is free of quantifiers. Cook [Coo75] established that if $\varphi$ 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 $eF$-proofs.
- Moreover, the existence of polynomial-size $eF$-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 $n$. 

---

7
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 $u \in I$,

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

Claim 3.4 can be proved by induction on the complexity of $\varphi$. 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

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

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 $\mathcal{L}_B$-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 $M' \supseteq_{cf} M$. 

8
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 [Jef05, LC11, Lè14], 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 [Jef07], 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 $L_{PV}$-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 $k$):

$$PV_1 \vdash \forall \varphi \left( \text{Valid-Fla}(\varphi) \land \exists y \text{ Sat}(\varphi, y) \rightarrow \text{Valid-}k\text{-CNF}(f(\varphi)) \land \exists z \text{ Sat}(f(\varphi), z) \right)$$

$$PV_1 \vdash \forall \varphi \left( \text{Valid-Fla}(\varphi) \land \forall y \neg\text{Sat}(\varphi, y) \rightarrow \text{Valid-}k\text{-CNF}(f(\varphi)) \land \forall z \text{ Value}_{\leq 1-\delta}(f(\varphi), z) \right)$$

where all the expressions are quantifier-free $L_{PV}$-formulas: Valid-Fla($x$) checks if $x$ is a valid Boolean formula, Valid-$k$-CNF($x$) checks if $x$ is a valid $k$-CNF, Sat($u, v$) checks if $v$ is a satisfying assignment for $u$, and 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 PV₁, which ensure that the theory “understands” how different algorithms are constructed from one another. In many cases, the verification that PV₁ proves the desired properties is straightforward but tedious, requiring some initial setup of basic capabilities of PV₁ (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 \( L_{PV} \)-sentence \( \text{FLB}_s \) stating that Boolean formulas for the parity function on \( n \) bits require at least \( s(n) \) leaves:

\[
\forall N \forall n \forall F \left( n = |N| \land n \geq 1 \land \text{Valid-Fla}(F) \land \text{Size}(F) < s(n) \rightarrow \exists x (|x| \leq n \land \text{Eval}(F, x) \neq \oplus(x)) \right),
\]

where we identify \( n \)-bit strings with natural numbers of length at most \( n \), and employ a well-behaved \( L_{PV} \)-function symbol \( \oplus \) such that PV₁ proves the basic properties of the parity function, e.g., PV₁ \( \vdash \oplus(x1) = 1 - \oplus(x) \).

Note that \( \text{FLB}_s \) is a \( \forall \Sigma^b_1 \)-sentence. Consequently, if PV₁ \( \vdash \text{FLB}_s \), 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 PV₁ 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 \( \text{poly}(N) = \text{poly}(2^n) \). 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 PV₁

In this section, we explore some details of a formalization in PV₁ 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 \( s(n) \equiv n^{3/2} \). Then PV₁ \( \vdash \text{FLB}_s \).

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 \( \text{Size}(F_1) \leq (1 - 1/n)^{3/2} \cdot \text{Size}(F) \).

---

\(^6\)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 \( F_\ell \) on \( k \)-bit inputs such that

\[
\text{Size}(F_\ell) \leq \text{Size}(F) \cdot (1 - 1/n)^{3/2} \cdot (1 - 1/(n-1))^{3/2} \ldots (1 - 1/(k+1))^{3/2} = \text{Size}(F) \cdot (k/n)^{3/2}.
\]

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

We present the argument in a more constructive way when formalizing the result in PV. 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 \( \oplus^b(x) \triangleq \oplus(x) + b \) (mod 2). In order to prove FLB in PV, we explicitly consider a polynomial-time function \( R(1^n, F, b) \) with the following property:\(^8\)

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

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 \( \text{Ref}_{R,s} \) described as follows:

\[
\forall 1^n \forall F \left( \text{Valid-Fla}(F) \land \text{Size}(F) < s(n) \rightarrow |y_n^0|_e = |y_n^1|_e = n \land F(y_n^0) \neq \oplus^0(y_n^0) \land F(y_n^1) \neq \oplus^1(y_n^1) \right),
\]

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 \( |z|_e \) to denote the bitlength of \( z \). Our plan is to define \( R \) and show that \( \text{PV} \vdash \text{Ref}_{R,s} \). Note that this implies FLB in PV 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 \( \text{Ref}_{R,s} \) is a universal sentence and \( S_2^1(\text{LV}) \) is \( \forall \Sigma_1^b \)-conservative over \( \text{PV} \) (i.e., provability of such a sentence in \( S_2^1(\text{LV}) \) implies its provability in \( \text{PV} \)), it is sufficient to describe a formalization in the more convenient theory \( S_2^1(\text{LV}) \). For this reason, polynomial induction for NP and coNP predicates (admissible in \( S_2^1(\text{LV}) \); 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:

\(^7\)Actually, 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 \( \text{Eval}(F, x) \).
As an instructive example, we establish the correctness of the necessary properties (provably in (PIND) for operations.)

Proof. 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 PV1 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 \( S_2^1(\mathbb{L}_{PV}) \). Next, as an instructive example, we establish the correctness \( R(1^n, F, b) \) in \( S_2^1(\mathbb{L}_{PV}) \) by polynomial induction (PIND) for \( \Pi^1_n \)-formulas, assuming that the subroutines appearing in the pseudocode of \( R(1^n, F, b) \) satisfy the necessary properties (provably in \( S_2^1(\mathbb{L}_{PV}) \)).

**Lemma 4.2.** Let \( s(n) \triangleq n^{3/2} \). Then \( S_2^1(\mathbb{L}_{PV}) \vdash \text{Ref}_{R,s} \).

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

\[
\forall F \forall n \left( n = |N| \land n \geq 1 \land \text{Valid-Fla}(F) \land \text{Size}(F) < s(n) \right) \rightarrow \\
(\forall \ell (|y_n^0|_\ell = |y_n^1|_\ell = n \land F(y_n^0) \neq \oplus^0(y_n^0) \land F(y_n^1) \neq \oplus^1(y_n^1)) \),
\]

where 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 \( \Pi^1_n \)-formula. Below, we argue that

\[
S_2^1(\mathbb{L}_{PV}) \vdash \varphi(1) \quad \text{and} \quad S_2^1(\mathbb{L}_{PV}) \vdash \forall N \varphi(|N/2|) \rightarrow \varphi(N).
\]

Then, by polynomial induction for \( \Pi^1_n \)-formulas (available in \( S_2^1(\mathbb{L}_{PV}) \)) and using that \( \varphi(0) \) trivially holds, it follows that \( S_2^1(\mathbb{L}_{PV}) \vdash \forall N \varphi(N) \). In turn, this yields \( S_2^1(\mathbb{L}_{PV}) \vdash \text{Ref}_{R,s} \).

**Base Case:** \( S_2^1(\mathbb{L}_{PV}) \vdash \varphi(1) \). 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 \( \text{Size}(F) = 0 \). Let \( y_1^0 \triangleq R(1, F, 0) \) and \( y_1^1 \triangleq R(1, F, 1) \). We need to prove that

\[
|y_1^0|_\ell = |y_1^1|_\ell = 1 \land F(y_1^0) \neq \oplus^0(y_1^0) \land F(y_1^1) \neq \oplus^1(y_1^1).
\]

Since \( n = 1 \) and \( \text{Size}(F) = 0 \), \( F \) 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: $S^1_{\mathcal{L}_{PV}} \vdash \forall N. \varphi([N/2]) \rightarrow \varphi(N)$. Fix an arbitrary $N$, let $n \triangleq |N|$, and assume that $\varphi([N/2])$ holds. By the induction hypothesis, for every valid formula $F'$ with $\text{Size}(F') < n'^{3/2}$, where $n' \triangleq n - 1$, we have

$$|y_{n'}^0|_e = |y_{n'}^1|_e = n' \land F'(y_{n'}^0) \neq \emptyset^0(y_{n'}^0) \land F'(y_{n'}^1) \neq \emptyset^1(y_{n'}^1), \quad (1)$$

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

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

Consider the following definitions:

1. $\widetilde{F} \triangleq \text{Normalize}(1^n, F)$ (Line 3),
2. $\rho \triangleq \text{Find-Restriction}(1^n, \widetilde{F})$ (Line 4),
3. $F' \triangleq \text{Apply-Restriction}(1^n, \widetilde{F}, \rho)$ (Line 5),
4. $n' \triangleq n - 1$ (Line 5),
5. $b' \triangleq b \oplus c_i$ (Line 5), where $\rho$ restricts $x_i$ to $c_i$,
6. $y_{n'}^{b'} \triangleq R(1^{n'}, F', b')$ (Line 6),
7. $y_{b'}^b \triangleq y_{n'}^{b'} \cup y_1 \mapsto c_i$ (Line 6),
8. $s \triangleq \text{Size}(F)$, $\overline{s} \triangleq \text{Size}(\widetilde{F})$, and $s' \triangleq \text{Size}(F')$.

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

- $(i)$ $\overline{s} \leq s$,
- $(ii)$ $s' \leq \overline{s} \cdot (1 - 1/n)^{3/2}$,
- $(i)$ $\forall x \in \{0, 1\}^n \overline{F}(x) = F(x)$,
- $(ii)$ $\forall z \in \{0, 1\}^{n - 1}(x) F'(z) = \overline{F}(z \cup x_i \mapsto c_i)$.

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

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

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 $y_{n'}^b$, we have $|y_{n'}^b|_e = |y_n|^e = n$. Using the definitions listed above, the last two conditions in the induction hypothesis (Equation (1)), and Items (iii) and (iv), we derive in $S^1_{\mathcal{L}_{PV}}$ the following statements for each $b \in \{0, 1\}$:

$$F'(y_{n'}^{b'}) \neq \emptyset^0(y_{n'}^{b'}),$$
$$F(y_{n'}^{b'}) = F'(y_{n'}^{b'}),$$
$$F(y_n^b) \neq \emptyset^b(y_n^b).$$

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

$$\emptyset^b(y_{n'}^{b'}) = \emptyset^{b \oplus c_i}(y_{n'}^{b'}) = c_i \oplus (\emptyset^b(y_{n'}^{b'})) = c_i \oplus (\emptyset^b(y_n^b) \oplus c_i) = \emptyset^b(y_n^b).$$

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

As explained above, the provability of $\text{Ref}_{R, s}$ in $S^1_{\mathcal{L}_{PV}}$ implies its provability in $\text{PV}_1$. Since $\text{PV}_1 \vdash \text{Ref}_{R, s} \rightarrow \text{FLB}_s$, this completes the proof of Theorem 4.1. \hfill \Box
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 $S^1_2(\mathcal{L}_{PV})$) 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 $n^\ell$?

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 $n^\ell$?

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 $NP \subseteq SIZE[n^k]$ is equivalent to the consistency of $NP \not\subseteq SIZE[n^k]$ 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 $P = NP$ 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 $P = NP$. 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 $NP \subseteq SIZE[\text{poly}]$ 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 $P \subseteq SIZE[n^k]$. 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 $P \subseteq SIZE[n^k]$. A slightly stronger model-theoretic formulation of the result of [KO17] appears in [BM20].
In more detail, for an \( L_{PV} \)-function symbol \( h \), consider the sentence

\[
UB^{i,o}_k(h) \triangleq \forall m \exists n \exists C_n \forall x \big( n \geq m \land |C_n| \leq n^k \land (|x| \leq n \rightarrow \psi(n, C_n, x, h)) \big),
\]

where \( \psi \) is an open \( L_{PV} \)-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, \( UB^{i,o}_k(h) \) 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 \( L_{PV} \)-function symbol \( h \) such that \( PV_1 \) does not prove \( UB^{i,o}_k(h) \). Similarly, they established that \( S^1_2 \not\subseteq NP \subseteq i.o.\text{SIZE}[n^k] \) and \( T^1_2 \not\subseteq P^{NP} \subseteq i.o.\text{SIZE}[n^k] \).

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 \( L_{PV} \)-function symbol \( h \), we consider a sentence \( UB_{c,k}(h) \) stating that \( L_h \in \text{SIZE}[c \cdot n^k] \), where \( x \in L_h \) if and only if \( h(x) \neq 0 \), i.e.,

\[
UB_{c,k}(h) \triangleq \forall n \exists C_n \forall x \big( |C_n| \leq c \cdot n^k \land (|x| \leq n \rightarrow (\text{Eval}(C_n, x, n) = 1 \iff h(x) \neq 0)) \big).
\]

Our goal is to show that there is a function symbol \( h \) such that, for no choice of \( c \geq 1 \), \( PV_1 \) proves \( UB_{c,k}(h) \). (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 \( UB_{c,k}(h) \) 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 \( UB_{c,k}(h) \) 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 \( UB_{c,k}(h) \) (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 \( s, q : \mathbb{N} \rightarrow \mathbb{N} \), we say that a language \( L \subseteq \{0,1\}^* \) is in LEARN-uniform\( ^{EQ[\ell]} \text{SIZE}[s] \) if there is a polynomial-time algorithm \( A^{EQ(L,n)}(1^n) \) 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 \( D(w) \neq L_n(w) \).

**Extraction of LEARN-uniform circuits from \( PV_1 \) proofs.** For convenience, write \( UB_{c,k}(h) = \forall 1^n \exists C_n \forall x \phi(1^n, C_n, x) \) 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 \( PV_1 \vdash UB_{c,k}(h) \), we can apply Theorem 3.2 (KPT Witnessing Theorem) to obtain the provability in \( PV_1 \) of the disjunction

\[
\forall 1^n \forall x_1 \ldots \forall x_k \left( \phi(1^n, t_1(1^n), x_1) \lor \phi(1^n, t_2(1^n, x_1), x_2) \lor \cdots \lor \phi(1^n, t_k(1^n, x_1, \ldots, x_{k-1}), x_k) \right),
\]

where \( t_1, \ldots, t_k \) are \( L_{PV} \)-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

---

9Recall 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 \( L \in P \) such that \( L \not\in \text{LEARN-uniform}^{\mathsf{EQ}O(1)} \mathsf{SIZE}[O(n^k)] \). This unconditional lower bound against LEARN-uniform circuits is established in [CKKO21] by generalizing a lower bound from [SW14] against \( \mathsf{P} \)-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 \( P \subseteq \mathsf{SIZE}[n^k] \).

In order to solve Open Problem 5.1, using the connection from [CKKO21] it is sufficient to show that \( P \not\subseteq \text{LEARN-uniform}^{\mathsf{EQ}[q]} \mathsf{SIZE}[O(n^k)] \) for \( q = \text{poly}(n) \). 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 \( P = \text{NP} \) and Propositional Proof Complexity

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

\[
\varphi_{P=\text{NP}}(g) \triangleq \forall 1^n \forall F \forall x \left( (\text{Formula}(F, 1^n) \land \text{Sat}(F, x)) \rightarrow \text{Sat}(F, g(F)) \right)
\]

(4)

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

**Open Problem 5.3.** Show that for no polynomial-time function symbol \( g \) theory \( \mathcal{PV}_1 \) proves the sentence \( \varphi_{P=\text{NP}}(g) \).

Equivalently, Open Problem 5.3 states that \( \mathcal{PV}_1 \) (and by standard conservation results \( S^1 \)) is consistent with \( P \neq \text{NP} \). This means that either \( P \neq \text{NP} \), as is commonly assumed, making the conjecture trivially true, or \( P = \text{NP} \), 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 \( P \neq \text{NP} \). 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 \( eF \) proofs of size \( n^{\omega(1)} \). Then there is no function symbol \( g \) such that \( \mathcal{PV}_1 \) proves \( \varphi_{P=\text{NP}}(g) \).

**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 \( \mathcal{PV}_1 \vdash \varphi_{P=\text{NP}}(g) \) for some \( g \), then every tautology has a polynomial size \( eF \) proof.

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

\[
\text{Sound}_P \triangleq \forall 1^n \forall F \exists \pi \left( \text{Formula}(F, 1^n) \land \text{Proof}_P(F, \pi) \right) \rightarrow \forall x \left( |x| \leq n \rightarrow \text{Sat}(F, x) \right),
\]

where \( \text{Proof}_P(F, \pi) \) states that \( \pi \) 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 \( \bar{\pi} \), \( P_g \) accepts \( \bar{\pi} \) as a proof of \( F \) if and only if

\[
g(\neg F) = \bar{\pi} \quad \text{and} \quad \neg \text{Sat}(\neg F, \bar{\pi}),
\]

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

\[
\psi \triangleq \neg \text{Formula}(F, 1^n) \lor \neg \text{Proof}_{P_g}(F, \pi) \lor |x| > n \lor \text{Sat}(F, x).
\]

The provability of \( \forall 1^n \forall F \forall \pi \psi \) in \( \mathcal{PV}_1 \) follows from the provability of \( \text{Sound}_{P_g} \).

Using Cook’s translation (Section 3.2), the sequence of propositional formulas \( ||\psi||_m \) admits \( eF \)-proofs of polynomial size. Moreover, given an actual \( n \)-bit propositional formula \( F \) of polynomial size and the corresponding \( P_g \)-proof \( \pi_F \) (represented by fixed strings \( \langle F \rangle \) and \( \langle \pi_F \rangle \)), one can show that there are polynomial
size $eF$ proofs of both $\|\text{Formula}(\langle F \rangle, 1^n)\|_{\text{poly}(n)}$ and $\|\text{Proof}_{eF}(\langle F \rangle, \langle \pi F \rangle)\|_{\text{poly}(n)}$. (Intuitively, this follows by an evaluation of the expressions on these fixed inputs.) Since $eF$ is closed under substitution, we can derive in $eF$ with a polynomial size proof the formula $\|\text{SAT}(\langle F \rangle, x)\|_{\text{poly}(n)}$.

Finally, for every propositional formula $F(x)$ on $n$-bit inputs, it is possible to efficiently prove in $eF$ the propositional formula $\|\text{SAT}(\langle F \rangle, x)\|_{\text{poly}(n)} \rightarrow F(x)$. (This can be established by a slightly more general structural induction on formulas $F$ using information about $\|\cdot\|$ and $\langle \cdot \rangle$. Overall, since $eF$ is closed under implication, it follows from these derivations that there is a polynomial size $eF$ 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 $T^i_2 \neq T^j_2$.

It is known that $PV_1$ proves that $P = \text{NP}$ if and only if it proves that $\text{NP} = \text{coNP}$. Consequently, a super-polynomial lower bound on the length of $eF$ proofs also yields the consistency of $\text{NP} \neq \text{coNP}$ 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 $\varphi_{\text{NP}}$ is false and therefore $P \neq \text{NP}$. 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 $\text{LB}^1(s_1, s_2, m, n_0)$ 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

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

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

$$\forall n \in \text{LogLog} \quad \text{with } n \geq n_0 \quad \exists C \in \text{coNSIZE}[s_1(n)] \forall D \in \text{NSIZE}[s_2(n)]$$

$$\exists m = m(n) \text{ distinct } n\text{-bit strings } x^1, \ldots, x^m \text{ s.t. } \text{Error}(C, D, x^i) \text{ for all } i \in [m],$$

where $\text{Error}(C, D, x)$ means that the circuits $C$ and $D$ disagree on the input $x$. This statement can be seen as an average-case form of the $\text{coNP} \nsubseteq \text{NP/poly}$ conjecture if we let $s_1(n) = n^{O(1)}$, $s_2(n) = n^{o(1)}$, 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 $s_1(n) = n^d$, $s_2(n) = 2^{n^d}$, and $m(n) = 2^n/n$. Then PV$_1$ does not prove the sentence $LB^1(s_1, s_2, m, n_0)$.

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 PV$_1 \vdash LB^1(s_1, s_2, m, n_0)$ 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 $LB^1(s_1, s_2, m, n_0)$ implies the provability of a worst-case lower bound for coNSIZE[$n^d$] against NSIZE[$2^{n^d}$]. We formalize the latter by a sentence $LB^1_w(s_1, s_2, n_0)$.

(ii) Given any proof of $LB^1_w(s_1, s_2, n_0)$ 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 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 $\leq 2^n$ 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 $LB^1(s_1, s_2, m, n_0)$. 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 poly$(2^n)$. The example provided next shows that this is still useful.

**Lemma 5.7 ([CLO24a]).** Let $L \in$ 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 $L' \in$ NP and for every constant $\varepsilon > 0$, we have $L' \in$ DTIME[$2^{n^{\varepsilon}}$].

**Proof.** Suppose that $L \in$ 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 $n^{2\gamma'}$ 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 $B^\gamma$ decides $L'$. Since $D_x$ is a co-nondeterministic circuit over inputs of length $m \triangleq n^\gamma$ and has size at most $n^{2\gamma'} = n^{2\gamma'/\gamma} \leq m^{\log m}$ (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 $B^\gamma$ runs in time at most $O(2^{n^\gamma})$. Clearly, Steps 1–2 run in 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 $2^{n^\gamma}$ if we set $\gamma \leq \varepsilon/2$. Finally, since $L \in \text{NTIME}[n^d]$, the brute force computation in Step 4 can be performed in deterministic time $2^{O(\ell^d)}$ over an input of length $\ell$. Since $\ell = n^\gamma = |w|$ in our case, if $\gamma \leq \varepsilon/2d$ we get that Step 4 runs in time at most $2^{n^\gamma}$. Overall, if we set $\gamma \triangleq \varepsilon/2d$, it follows that $B^\gamma$ runs in time at most $O(2^{n^\gamma})$.

This completes the proof that $L' \in \text{DTIME}[2^{n^\gamma}]$.

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 $\text{LB}_1^1(s_1, s_2, m, n_0)$. 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 ($eF$) 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 $\text{PV}_1$ and its language $\mathcal{L}_\text{PV}$. We shall use the following $\mathcal{L}_\text{PV}$ formulas:

- $\text{Sat}(x, y)$: a quantifier-free formula formalizing that $y$ is a satisfying assignment of the Boolean formula $x$;
- $\text{Taut}(x) \triangleq \forall y \leq x \text{Sat}(x, y)$;
- $\text{Proof}_P(x, z)$: 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 $M \models \text{PV}_1$, and assume that $\phi \in M$ is a propositional formula. The following statements are equivalent:

(i) There is no $eF$-proof of $\phi$ in $M$:

$$M \models \forall z \neg \text{Proof}_{eF}(\phi, z).$$

(ii) There is an extension $M' \supseteq M$ (also a model of $\text{PV}_1$) in which $\phi$ is falsified:

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

The proof of Lemma 5.9 proceeds by compactness and uses that the correctness of the propositional translation from $\text{PV}_1$ to $eF$ (Section 3.2) is also provable in $\text{PV}_1$.

**Lemma 5.10.** Let $M$ be a nonstandard countable model of $\text{PV}_1$. Then it has a cofinal extension $M' \supseteq M$ (also a model of $\text{PV}_1$) such that every tautology in $M'$ has an $eF$-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 $M_i \models \text{PV}_1$ and $\text{PV}_1$ is universal, we also have $M' \models \text{PV}_1$.

We will need the following analogue of Lemma 3.6 for $\text{PV}_1$.

**Fact 5.11.** Let $M_0$ be a nonstandard countable model of $\text{PV}_1$. Then there is a (countable) cut $M$ of $M_0$ that is a model of $\text{PV}_1$ 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 $\text{PV}_1$. There is a model $M^*$ of $\text{PV}_1$ such that the following properties hold:

(i) Any tautology in $M^*$ has an $eF$-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 $eF$ lower bounds in $\text{PV}_1$ [KP89]). Consider the sentence

$$\Psi_{eF} \triangleq \forall x \exists \phi \geq x \left[ \text{Taut}(\phi) \land \forall \pi \left( |\pi| \leq |\phi|^{\log |\phi|} \rightarrow \neg \text{Proof}_{eF}(\phi, \pi) \right) \right].$$

The sentence $\Psi_{eF}$ is not provable in $\text{PV}_1$. 

Proof. Suppose $\mathsf{PV}_1 \vdash \Psi_{e^F}$. Fix a countable nonstandard model $M$ of $\mathsf{PV}_1$. Let $M^*, a$, and $n \triangleq |a|$ be as in Corollary 5.12. Since $\Psi_{e^F}$ holds in $M^*$, there is a tautology $\phi \in M^*$ with $\phi \geq a$ and consequently $|\phi| \geq n$ such that $\phi$ does not have an $e^F$-proof of size $|\phi|^\log |\phi|$ in $M^*$. In particular, $\phi$ does not have a proof of size $n^{\log n}$. On the other hand, by the two properties of $M^*$ the formula $\phi$ has an $e^F$-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 $\mathsf{PV}_1$ does not prove $\Psi_{e^F}$. \hfill $\square$

**Open Problem 5.14.** Show that $\mathsf{PV}_1$ cannot prove fixed-polynomial size lower bounds on the length of $e^F$ proofs.

**Open Problem 5.15.** Establish the unprovability of the sentence $\Psi_{e^F}$ 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 $\mathsf{PV}_1$ function symbol $h$ and a circuit size parameter $k \in \mathbb{N}$, consider the sentence

$$\text{LB}^{a,e}_k(h) \triangleq \lnot \text{UB}^{i,o}_k(h),$$

where $\text{UB}^{i,o}_k(h)$ is the sentence defined in Section 5.1.1. The sentence $\text{LB}^{a,e}_k(h)$ 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 $\mathsf{PV}_1$ from the unprovability of lower bounds in $\mathsf{PV}_1$ [BKO20]).

If there exists $k \in \mathbb{N}$ such that for no function symbol $h$ theory $\mathsf{PV}_1$ proves the sentence $\text{LB}^{a,e}_k(h)$, then for no function symbol $f$ theory $\mathsf{PV}_1$ proves the sentence $\varphi_{\mathsf{P} = \mathsf{NP}}(f)$.

Theorem 5.16 shows that if $\mathsf{PV}_1$ does not prove $n^k$-size lower bounds for a language in $\mathsf{P}$, then $\mathsf{P} \neq \mathsf{NP}$ is consistent with $\mathsf{PV}_1$. Note that the hypothesis of Theorem 5.16 is weaker than the assumption that $\mathsf{PV}_1$ does not prove that $\mathsf{NP} \not\subseteq \text{SIZE}[n^k]$ for some $k$.

**Sketch of the proof of Theorem 5.16.** We proceed in the contrapositive. We formalize in $\mathsf{PV}_1$ the result that if $\mathsf{P} = \mathsf{NP}$, then for any parameter $k$, $\mathsf{P} \not\subseteq \text{i.o.}\text{SIZE}[n^k]$ (see, e.g., [Lip94, Theorem 3]). This result combines the collapse of $\mathsf{PH}$ to $\mathsf{P}$ with Kannan’s argument [Kan82] that $\mathsf{PH}$ 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 $\mathsf{PV}_1$.

We address this issue as follows. From the provability in $\mathsf{PV}_1$ that $\mathsf{P} = \mathsf{NP}$, it follows that for each $i \geq 1$ theory $T_{i}^2$ collapses to $\mathsf{PV}_1$ [KPT91]. Recall that the dual weak pigeonhole principle (dWPHP) for $L_{\mathsf{PV}}$-functions is provable in $T_{i}^2$. Define a $\mathsf{PV}_1$ 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 dWPHP($g$), we now derive in $\mathsf{PV}_1$ 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 $\mathsf{PV}_1$-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 $\mathsf{PV}_1$.

Given that the provability in $\mathsf{PV}_1$ that $\mathsf{P} = \mathsf{NP}$ implies the provability in $\mathsf{PV}_1$ that $\mathsf{PH}$ collapses to $\mathsf{P}$, it follows that $\varphi(x)$ is equivalent in $\mathsf{PV}_1$ to the language defined by some $L_{\mathsf{PV}}$-function $h$. In other words, $\mathsf{PV}_1 \vdash \text{LB}^{a,e}_k(h)$, which completes the proof of Theorem 5.16. \hfill \square
shows an example of a simple lower bound that is not provable in PV, 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 [Jef07]. [Kra21] showed that dWPHP cannot be proved in PV under the assumption that $P \subseteq \text{SIZE}[n^k]$ for some constant $k$. [ILW23] established the same unprovability result assuming sub-exponentially secure indistinguishability obfuscation and coNP $\not\subseteq$ i.o.AM. [ABM23] established the unprovability of $\text{NEXP} \subseteq \text{SIZE}[\text{poly}]$ in the theory of bounded arithmetic $V_2^0$ (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 $T_2^2$ implies that $P^{\text{NP}} \not\subseteq \text{SIZE}[n^{1+\epsilon}]$ for some $\epsilon > 0$. 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 $P = \text{NP}$ 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, Myktya 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


