The Bulletin of Symbolic Logic
Volume 23, Number 4, December 2017

INCOMPLETENESS IN THE FINITE DOMAIN

PAVEL PUDLÁK

Abstract. Motivated by the problem of finding finite versions of classical incompleteness theorems, we present some conjectures that go beyond **NP** $\neq$ **coNP**. These conjectures formally connect computational complexity with the difficulty of proving some sentences, which means that high computational complexity of a problem associated with a sentence implies that the sentence is not provable in a weak theory, or requires a long proof. Another reason for putting forward these conjectures is that some results in proof complexity seem to be special cases of such general statements and we want to formalize and fully understand these statements. Roughly speaking, we are trying to connect syntactic complexity, by which we mean the complexity of sentences and strengths of the theories in which they are provable, with the semantic concept of complexity of the computational problems represented by these sentences.

We have introduced the most fundamental conjectures in our earlier works [27, 33–35]. Our aim in this article is to present them in a more systematic way, along with several new conjectures, and prove new connections between them and some other statements studied before.

§1. Introduction.

Gödel's incompleteness theorem is undoubtedly one of the most important theorems in logic. It speaks about absolute provability, i.e., about proofs without any restriction on their length. The question whether there is a "finite" or "feasible" version of the incompleteness theorem, where the complexity of proofs is bounded, has certainly intrigued many people, but very little has been published about it. With the advent of computers and theories developed for them, in particular complexity theory, the question about a finite version of the incompleteness theorem became even more interesting. The concept of polynomial time computations turned out to be the most important concept in complexity theory. The distinction between functions decidable in polynomial time and those computable only in exponential time plays a similar role as the distinction between computable and noncomputable in computability theory. The successful use of polynomial bounds suggested that one should also study which theorems have polynomial length proofs. A natural version of a finite incompleteness theorem was formulated by Harvey Friedman in 1979. Let $Con_T(\bar{n})$ be a natural formalization of the statement *"there is no derivation of contradiction of length n from the axioms of T"*. Friedman proved a lower bound of the form n^{ε} for some $\varepsilon > 0$ and asked whether such sentences have proofs in

Received May 18, 2017.
2010 *Mathematics Subject Classification.* 03F20, 03D15, 68Q15.
Key words and phrases. finite consistency, propositional proof systems, disjoint NP-sets, search problems.

1079-8986/17/2304-0002
DOI:10.1017/bsl.2017.32

T of polynomial (in n) lengths [14].[1] It turned out that the answer to his question is yes [29], but it is still possible, and seems very plausible, that for natural variations of this question there are no polynomial length proofs. Namely, this should be true if we ask about the lengths of proofs of $Con_T(\bar{n})$ in a theory S sufficiently *weaker* than T. However, proving such a claim must be extremely difficult, because it implies $\mathbf{P} \neq \mathbf{NP}$ (and even more than that).

Our motivation for studying such problems is the fundamental question: *what is the connection between logical strength of theories and computational complexity*? which is basically what the field of *proof complexity* is about. Here we refer to proof complexity in a broader sense that also includes the study of first order theories called bounded arithmetic. Since there is a close connection between propositional proof systems and first order theories, we view these two concepts as nonuniform and uniform versions of the same concept.

To give an example of a connection between theories and computational complexity, let us consider Buss's Witnessing Theorem [7]. This theorem states that one can construct polynomial time algorithms from proofs of certain sentences in the theory S^1_2 (see Theorem 2.2 below). This is an instance of a general phenomenon: if a theory is weak, the provably total functions have small computational complexity. Such theorems have been proven for a number of other theories and complexity classes. Another connection is the Feasible Interpolation Theorem of Krajíček [25]. According to this theorem, one can construct circuits from proofs of certain tautologies in various proof systems, in particular, in resolution; the circuits separate two sets of Boolean vectors defined by the tautology. A high level form of these results is that if something is provable in a weak formal system, i.e., the logical strength of the system is bounded, we can give bounds on some computational problems associated with the systems. If we state it contrapositively it suggests that increasing strength of logical formal systems is correlated with increasing complexity of the associated computational tasks. Thus a more specific question is: *find general principles of which these results are special instances.*

Connection between proofs and computations have been extensively studied in constructive mathematics in the context of intutionistic logic. There are also results that show interesting connections of proofs in the intutionistic calculus with computational complexity. For example, Buss and Mints [9] proved that given an intuitionistic proof of a disjunction $\phi \vee \psi$ in propositional logic (say, in the sequent calculus), one can find *in polynomial time* a proof of either ϕ or ψ. A more general theorem, a version of a realizability theorem for intuitionistic propositional calculus, was proved in [10]. However, the problems we are going to consider in this article have not been studied in the context of intuitionistic logic and also in this article we will only use classical logic.

[1] Here $\bar{n}$ denotes a *binary numeral*, a term of length $O(\log n)$ that represents n. Therefore the lower bound is nontrivial.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

The general principles that we study are connected with notoriously open and probably very difficult problems in computational complexity theory, so we cannot prove or disprove them with the currently available means. They can only be stated as hypotheses or conjectures without any formal supporting evidence. There are, essentially, two reasons for stating some sentences as conjectures. First, we believe that some basic theorems of proof theory should also hold true with suitable bounds on the lengths of proofs. The prime example is the Second Incompleteness Theorem discussed above. Second, some results in proof complexity and bounded arithmetic seem to follow a general pattern. For example, as we noted above, polynomial time computations are associated with the theory S^1_2 by a witnessing theorem. If we take S^2_2, which we believe is a stronger theory, then the corresponding function class is $\mathbf{P}^{\mathbf{NP}}$,[2] which we believe is a larger class than $\mathbf{P}$.[3] The form of this result suggests that S^2_2 requires more complex functions.

Alternatively, we can view the proposed conjectures as axioms. In fact, $\mathbf{NP} \neq \mathbf{P}$ has been treated as an axiom "to prove" hardness of various problems. In many cases $\mathbf{NP} \neq \mathbf{P}$ does not suffice and therefore a number of stronger hypotheses have been proposed. For example, the *Exponential Time Hypothesis* of Impagliazzo and Paturi [19] and its variants are used to determine the time complexity of concrete polynomial time computable problems. In the theory of approximate algorithms several conjectures have been proposed in order to show nonapproximability of certain problems.

Although we have to treat the most interesting statements only as hypotheses, there are some interesting problems that we can study and solve with the currently available means. These are problems about relationships among various conjectures. In particular, we would like to know whether there is one general principle that would cover all instances, or an infinite hierarchy of principles. If there is a hierarchy, is it linear, or does it branch? If it branches, is there a natural classification of conjectures? We will address some questions of this kind in this article. Furthermore, one can study relativizations of these conjectures. Several results about relativizations have been proven, but much more is needed. We will mention some more concrete problems at the and of the article.

We are primarily interested in these questions, because we want to understand the essence of fundamental problems. However, there is also a practical aspect of this research. The general conjectures suggest what specific problems in proof complexity we should study. Then we can "test" the conjectures on weak formal systems for which we do have means to prove results connecting them with computational complexity. In fact the main Conjectures CON and TFNP represent what researchers in proof complexity believe is likely to be true.

All conjectures that we consider in this article state something about unprovability, although they often have a natural equivalent version stated

[2] This was also proved by Buss in [7].

[3] We are not able to prove it formally, because a formal proof would give us $\mathbf{P}^{\mathbf{NP}} \neq \mathbf{P}$, which is equivalent to $\mathbf{NP} \neq \mathbf{P}$.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

in purely complexity-theoretical terms. The "finite domain" in the title refers to the fact that the lengths of computations and lengths of proofs of instances of the problems that we consider are at most exponential, hence there is a *finite* bound on them. Perhaps, a more precise term would be "exponential domain". In previous presentations of this topic, in particular in [33, 34], we used the term *"feasible incompleteness"*, which should be understood as *"being incomplete with respect to feasible proofs"*. In [33, 34] we also stated *the feasible incompleteness thesis*, which is an informal statement saying that unprovability of a sentence in a weak formal system may be caused by high computational complexity of a computational problem naturally associated with the sentence.

This article is partly a survey, but a large part consists of new results, or results that have not been published with full proofs. Specifically, the main conjectures were already presented in [34], but connections between them were only sketched there.

Here is how the article is organized. After two introductory sections, in Section 3, we recall a conjecture about finite consistencies and introduce a new conjecture about finite reflection principles. In Section 4 we present another important conjecture about total polynomial search problems. We discuss equivalent and stronger statements based on propositional proof systems and disjoint **NP** and **coNP** pairs of sets in Section 5. We introduce a classification of conjectures in Section 6 and show that uniform conjectures can be stated as statements about unprovability, which suggests a way towards general conjectures. Section 7 is about the role of reductions in the statements of conjectures. We conclude the article with some open problems.

§2. Preliminaries.

2.1. Theories. In this article we will use the word *"theory"* for a set of axioms that is decidable in polynomial time (i.e., for each formula we can decide in polynomial time in the length of the formula whether or not it is an axiom). This implies that given a sentence ϕ and a string of symbols d, it is possible also to decide in polynomial time if d is a proof of ϕ. Furthermore, we will only consider *consistent arithmetical theories* that use a fixed finite set of function and relation symbols representing functions and relations on the natural numbers. We use fragments of arithmetic (as those theories are called), because one can easily refer to standard formalizations of basic syntactical concepts. Being able to formalize syntactical concepts, such as first order formulas and proofs, is the essential property of the theories that we need.

Furthermore, we need that theories be *sufficiently strong*, because we need the formalizations of basic properties of syntactical concepts and computations to to be provable in the theories we will use. As usual, we will ensure that a theory is sufficiently strong by assuming that it contains a particular fixed basis theory. We will use Buss's theory S^1_2 for this purpose. Theory S^1_2 is one of the fragments of Bounded Arithmetic S_2 defined by Buss [7] (see also [17, 24]). Formally, it is not a fragment of PA (Peano Arithmetic), because it is formalized in a slightly richer language, but it is interpretable in it.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

Definition 2.1. We denote by $\mathcal{T}$ the class of all consistent arithmetical theories that extend Buss's theory S^1_2 by a set of axioms that is decidable in polynomial time.

For lack of a good name, we will only use the symbol $\mathcal{T}$ to denote this class of theories.

2.2. Bounded arithmetic. We will briefly describe S^1_2. It is very convenient to use S^1_2, but it should be noted that essentially all results and conjectures do not depend on the particular choice of the base theory. (This also concerns the formalization of the class **P**; the particular formalization that we use is also not essential.)

S^1_2 is a basic fragment of S_2 and it has similar relation to S_2 as $I\Sigma_1$ (Peano Arithmetic with induction restricted to Σ_1 formulas) to Peano Arithmetic. In S_2 (and so in S^1_2) the standard language of arithmetic is enriched by the symbols $\lfloor x/2 \rfloor, |x|, x\#y$. The intended interpretation of $\lfloor x/2 \rfloor$ is clear; this symbol is used in induction axioms. $|x|$ is the length of the binary representation of x if $x > 0$ and $|0| = 0$. The interpretation of $x\#y$ is $2^{|x|.|y|}$. Note that we do not have exponentiation in S_2, so # has to be a primitive symbol. Also note that the length of the binary representation of $2^{|m|.|n|}$ is roughly the product of the lengths of m and n. One can easily show that if $t(x_1, \dots, x_k)$ is a term in the language of S_2, then $|t(n_1, \dots, n_k)| \leq p(|n_1|, \dots, |n_k|)$ for some polynomial, and, vice versa, if $f(x_1, \dots, x_k)$ is a function that increases the lengths of input numbers at most polynomially, then there exists a term t such that $f(x_1, \dots, x_k) \leq t(x_1, \dots, x_k)$.

The theory S_2 is axiomatized by a finite set of axioms BASIC that fix the intended interpretation of symbols and by induction axioms

$$\alpha(\bar{0}) \wedge \forall x(\alpha(\lfloor x/2 \rfloor) \rightarrow \alpha(x)) \rightarrow \forall x.\alpha(x), \tag{1}$$

for all bounded formulas α. S^i_2 is S_2 with the axiom schema restricted to Σ^b_i formulas; we will define the classes Σ^b_i below.

2.3. Formulas and complexity classes. By a bounded formula, we mean a formula in which quantified variables are bounded by terms in the language of S_2. As we noted above, $x \leq t(y_1, \dots, y_k)$ implies that the *length* of x is polynomially bounded by the lengths of $y_1, \dots, y_k$. Sometimes we will also need to bound *the number itself* by a polynomial in the lengths of some other numbers. For example, we may need to bound the number of steps of an algorithm that has as an input the binary representation of a number x. In such cases we use *sharp bounds* which are bounds of the form $x \leq |s(y_1, \dots, y_k)|$. Since the outer function symbol in the term is $|\dots|$, x is polynomially bounded by the *lengths* of $y_1, \dots, y_k$. *Sharply bounded quantifiers* are bounded quantifiers with sharp bounds.

The hierarchy of bounded formulas Σ^b_n, Π^b_n, $n = 1, 2, \dots$, is defined by counting alternations of bounded quantifiers while *ignoring sharply bounded quantifiers*. In particular, prenex formulas that use bounded existential quantifiers and arbitrary sharply bounded quantifiers are Σ^b_1. So in Σ^b_1

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

(and similarly in higher classes) bounded existential quantifiers may alternate with sharply bounded universal quantifiers. This complication can be avoided by slightly extending S^1_2 with more function symbols and axioms. If we do this, then we can move all sharply bounded universal quantifiers after the bounded existential ones. The Σ^b_n formulas where all sharply bounded quantifiers are after all bounded quantifiers are called *strict*-Σ^b_n, or $\hat{\Sigma}^b_n$ formulas. $\hat{\Pi}^b_n$ formulas are defined similarly.

In order to simplify formulas, we will sometimes use quantifiers with a superscript $\forall^p, \exists^p$ to indicate that the lengths of the quantified variables are polynomially bounded in the formula that follows. For example, $\forall x \exists^p y.\phi(x,y)$ means that $\phi(x,y)$ is equivalent to the formula $|y| \leq p(|x|) \wedge \phi(x,y)$ for some polynomial $p(x)$ (or, equivalently, to $y \leq t(x) \wedge \phi(x,y)$ for some S_2 term t).

The subsets of $\mathbb{N}$ that are in **NP** are precisely those that are definable by Σ^b_1 formulas. Similarly, other classes from the hierarchy of formulas Σ^b_n, Π^b_n define corresponding complexity classes Σ^p_n, Π^p_n from the *Polynomial Hierarchy*.

For **P**, there is no simple definition of a class of formulas. Formulas from the class $\Sigma^b_0 (= \Pi^b_0)$ have only sharply bounded quantifiers. These bounds imply that they define sets and relations computable in polynomial time, but we cannot define all sets in **P** by such formulas. The standard approach is to extend the language by function symbols for every polynomial time algorithm as it is in Cook's theory PV [11].[4] This requires also adding infinitely many axioms specifying the intended interpretation of each function symbol. In this article we will use a different approach, one that does not need an infinite number of function symbols and axioms. To this end we will use *Buss's Witnessing Theorem.*

THEOREM 2.2 ([7]). *Let $\phi(x,y) \in \Sigma^b_1$ and suppose that $S^1_2 \vdash \forall x \exists y.\phi(x,y)$. Then there exists a polynomial time computable function f such that $\mathbb{N} \models \forall x.\phi(x, f(x))$. Moreover, f is definable by a Σ^b_1 formula.*

The definability of f means that there exists a Σ^b_1 formula $\psi(x,y)$ such that

$$S^1_2 \vdash \forall x \exists! y.\psi(x,y) \wedge \forall x \forall y(\psi(x,y) \rightarrow \phi(x,y)).$$

A formula $\sigma(x)$ is Δ^b_1 provably in a theory T if $\sigma(x) \in \Sigma^b_1$ and, for some $\pi(x) \in \Pi^b_1$, T proves the sentence $\forall x.\sigma(x) \equiv \pi(x)$. By Buss's Witnessing Theorem, the provability of the equivalence in S^1_2 ensures that $\sigma(x)$ defines a set in **P**. We should stress that it is essential that the proof is in S^1_2. The equivalence $\mathbb{N} \models \forall x.\sigma(x) \equiv \pi(x)$ in general only ensures that $\sigma(x)$ defines a set in **NP** $\cap$ **coNP** which is believed to be larger than **P**.

Thus we will formalize polynomial decidable sets and relations by formulas that are Δ^b_1 provably in S^1_2. In the rest of this article Δ^b_1 *will always mean: Δ^b_1 provably in S^1_2*. Polynomial time computable functions will be formalized by Σ^b_1 formulas $\psi(x,y)$ such that $S^1_2 \vdash \forall x \exists! y.\psi(x,y)$. One can show that

[4] The relation of PV to S^1_2 is similar to the relation of Primitive Recursive Arithmetic to $I\Sigma_1$.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

polynomial time computable functions are provably in S^1_2 closed under composition and a form of recursion called *limited recursion on notation* [7, 11, 24]. Because of these properties, one can formalize syntax in a natural way in S^1_2.

2.4. Proofs. We will assume that proofs are formalized in a standard Hilbert-style proof system for first order logic. We will view the proofs as strings of formulas such that each formula is either an axiom (logical or an axiom of the theory in question) or is derived from previous formulas by an application of a deduction rule. The particular choice of the system makes little difference, because various calculi for first order logic *polynomially simulate* each other, which means that there are polynomial time computable transformations of proofs from one calculus to the other. However, it is important that the graph of the proof is a general directed acyclic graph, not just a tree, because transforming a general proof into the tree form may increase the length exponentially.

We could also use sequent calculi, but only the calculi with the cut rule present. Cut-elimination may increase the length more than any elementary function.

We will assume that formulas and proofs are encoded by binary strings. The *length of a proof* is the length of the string representing the proof. For the *Gödel number of a formula, or a proof*, we add 1 at the beginning of the string and take the number that it represents in binary notation.

A proof in a theory T will simply be called a *T-proof*.

2.5. Notation. A *binary numeral* is a suitably chosen closed term $\bar{n}$ whose value is n and whose length is $O(\log n)$. For example, we can represent a number with binary representation $a_1a_2\dots a_k$ by the term

$$(\dots((a_1\cdot\bar{2})+a_2)\cdot\bar{2}\dots\quad\dots+a_{k-1})\bar{2}+a_k,$$

where $\bar{2}$ is $SS(0)$.

Our computation model is the standard Turing machine, where the inputs are words in the alphabet $\Sigma := \{0,1\}$. When computing with numbers, we assume the binary representation. We will use numbers instead of binary strings when we formalize computations. For $n \in \mathbb{N}$, we denote by $|n|$ the length of the binary representation of n (the same symbol as is in S_2).

We will denote by $Pr_T(x,y)$ a natural formalization of the relation *"x is a T-proof of y"*. We will assume that basic properties of this relation are provable in S^1_2. Furthermore, we will assume that the following fact is true.

FACT 2.3. *If m is a Gödel number of a T-proof of a sentence with a Gödel number n, then there exists an S^1_2-proof of $Pr_T(\bar{m},\bar{n})$ whose length is bounded by a polynomial in $|m|$ and $|n|$.*

The main numerical parameter will be denoted by n. When we say *"polynomial length"* without mentioning the argument of the polynomial, we will always mean *"length bounded by $p(n)$ for some polynomial p"*.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

§3. The basic paradigm – finite consistency.

3.1. Finite consistency. Let $T \in \mathcal{T}$. We will denote by $Con_T(x)$ a formula expressing (in a natural way) the fact that there is no T-proof of contradiction of length x. In particular, we will need $Con_T(\bar{n})$, for $n \in \mathbb{N}$. The question mentioned in the introduction is:

QUESTION 3.1. *What is the length of the shortest T-proof of $Con_T(\bar{n})$?*

Using the analogy with Gödel's incompleteness theorem, it is natural to conjecture that the proof must be long, specifically, not polynomial in n. Friedman also proved a lower bound n^ε for some $\varepsilon > 0$.[5] This lower bound was improved to $\Omega(n/\log^2 n)$ for a proof system with Rosser's *C-rule*

$$\frac{\exists x.\phi(x)}{\phi(c)},$$

where c is a new constant [30]. This rule enables one to refer to an element satisfying ϕ without having to mention ϕ. The same asymptotic bound is probably true for some other systems where this rule can be simulated. In particular, in natural deduction systems, we can start with an assumption $\phi(y)$ and argue about y without having to repeat the assumption in each proof line.

The idea of the proofs of these lower bounds is to adapt the original proof of Gödel to the finite setting. Thus instead of the original diagonal formula, one uses a formula $\delta(\bar{n})$ with intended meaning *"I do not have a T-proof of length $\leq n$"*. One can easily prove that $\delta_T(\bar{n})$ is true and any proof of it must be longer than n. Then one proves that $\delta_T(\bar{n})$ can be derived from $Con_T(\bar{n})$ by a short proof. This is essentially the same as in the proof of Gödel's theorem, except that one has to prove good upper bounds on the lengths of proofs of certain true sentences. The shorter proofs one is able to find, the larger the lower bound is.

In [29] a polynomial *upper bound* was proved for finitely axiomatized sequential theories. In [30] the bound was improved to a linear upper bound for finitely axiomatized sequential theories and proofs using the C-rule. Sequential theories are, roughly speaking, theories in which one can code any finite sequence of elements of the universe (see [17] for the definition). Already very weak fragments of arithmetic and set theory are known to be sequential. This bound is based on partial truth definitions. In the standard proofs of the consistency of a theory T (without any bound on the lengths of proofs), one uses a truth definition for all formulas. Since in proofs of bounded length only formulas of bounded complexity can occur, it suffices to use a partial truth definition that defines truth only for sentences of limited complexity. The fact that partial truth definitions exist is well-known. However, to obtain such bounds one has to carefully estimate the length of the formulas and the lengths of proofs of particular statements. A polynomial upper bound can also be proved for theories axiomatized by a schema of a particular form, which includes Peano Arithmetic and Zermelo-Fraenkel Set Theory [29].

[5] Note that the length of the sentence $Con_T(\bar{n})$ is $O(\log n)$.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

To sum up the discussion above we state the bounds explicitly, but for the sake of simplicity we will only use theories from $\mathcal{T}$.

THEOREM 3.2 ([14, 29, 30]). (1) *For every theory $T \in \mathcal{T}$, there exists $\varepsilon > 0$ such that the length of the shortest T-proof of $Con_T(\bar{n})$ is at least n^ε.*

(2) *If, moreover, T is sequential and finitely axiomatized, then there are T-proofs of $Con_T(\bar{n})$ lengths of polynomial in n.*

In spite of the polynomial upper bound, we still believe that the incompleteness phenomenon of Gödel's theorem should manifest itself also in the finite domain—manifest not only by the n^ε lower bound. We conjecture that if T is sufficiently *stronger* than a theory S, then S-proofs of $Con_T(\bar{n})$ cannot be polynomially bounded. Concerning Gödel's Theorem, in my opinion, the fact that T does not prove its own consistency is not important—having a proof of consistency in a theory that we do not a priori believe is consistent would be useless. What is important is the consequence of Gödel's Theorem that there is no theory that could prove the consistency of all other consistent theories. The paradigm for our conjecture is this corollary, not the theorem itself.

Since it is not clear how much stronger T must be, we proposed the following conjecture in [29]:

CONJECTURE 3.3 (CON^N). *For every $S \in \mathcal{T}$, there exists $T \in \mathcal{T}$ such that the lengths of S-proofs of $Con_T(\bar{n})$ cannot be bounded by a polynomial in n.*[6]

Of course, we would also like to know how much stronger T must be than S so that there are no polynomial length S-proofs of $Con_T(n)$. It has been conjectured that it suffices that T proves the consistency of S, i.e., the following seems to be true:

CONJECTURE 3.4 (CON^{N+}). *For every $S, T \in \mathcal{T}$, if T proves Con_S, then S-proofs of $Con_T(\bar{n})$ cannot be bounded by a polynomial in n.*

The following observation suggests that the assumption about provability of the consistency may be the right choice.

PROPOSITION 3.5. *Let $T_0^{Con} := T$ and $T_{k+1}^{Con} := T_k^{Con} + Con_{T_k^{Con}}$ for $k \in \mathbb{N}$. Suppose that for every true theory $T \in \mathcal{T}$, T proves $Con_{T+Con_T}(\bar{n})$ by proofs of polynomial length. Then every true theory $T \in \mathcal{T}$ proves $Con_{T_k^{Con}}(\bar{n})$ by proofs of polynomial length for every $k \in \mathbb{N}$.*

In plain words, if there are polynomial length T-proofs of $Con_{T+Con_T}(\bar{n})$ for every true theory $T \in \mathcal{T}$, then there are polynomial length T-proofs of bounded consistencies of theories essentially stronger than $T + Con_T$. The proposition is an immediate corollary of the following lemma.

LEMMA 3.6. *Suppose that R proves $Con_s(\bar{n})$ by proofs of polynomial length and S proves $Con_T(\bar{n})$ by proofs of polynomial length. Then R proves $Con_T(\bar{n})$ by proofs of polynomial length. (Polynomial bounds are in terms of n.)*

PROOF-SKETCH. Instead of polynomially bounded proofs of bounded consistency statements, suppose that $R \vdash Con_S$ and $S \vdash Con_T$. Arguing in R,

[6] The superscript N stands for "nonuniform", whose meaning will be explained in Section 6.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

suppose that $\neg Con_T$. By the Σ-completeness of S, we have $S \vdash \neg Con_T$, hence S is inconsistent, contrary to the assumption $R \vdash Con_S$.[7] To prove the lemma just restate the proof with bounded consistency statements and polynomial bounds on the proofs. ⊣

It is well-known [13] that if T is stronger (proves more sentences) than S, then some sentences provable in both theories have much shorter proofs in T compared to the proofs in S. This may suggest that it would suffice to make T just a little stronger than S, namely, to add any true unprovable Π_1 sentence, in order to ensure that S-proofs of $Con_T(\bar{n})$ do not have polynomial length proofs. However, recently Pavel Hrubeš proved, using a Rosser-type selfreferential sentence, that in general this is not the case [personal communication]. His result is even stronger than the mere refutation of such a strengthening of Conjecture CON^{N+}.

Theorem 3.7 (P. Hrubeš, unpublished). *Let $S \in \mathcal{T}$ be a sequential finitely axiomatized theory and let $S \subseteq T \in \mathcal{T}$. Then there exists a true Π_1 sentence ϕ such that ϕ is not provable in T, yet the lengths of S-proofs of $Con_{S+\phi}(\bar{n})$ can be bounded by a polynomial in n.*

Proof. By the fixpoint theorem, one can construct a sentence ϕ such that

$$S \vdash \phi \equiv \forall x(Pr_T(x, \bar{\phi}) \rightarrow \exists y(|y| \leq 2^{|x|} \wedge Pr_S(y, \neg\bar{\phi}))), \tag{2}$$

where $Pr_T(u, v)$ (and $Pr_S(u, v)$) are natural formalizations of the relation u is a T-proof (respectively S-proof) of v. This is the well-known Rosser sentence, except that we use the bound $|y| \leq 2^{|x|}$ instead of the usual one $y < x$ and two proof relations instead of one. One can easily prove

$$T \nvdash \phi, \ S \nvdash \neg\phi, \text{ and } \mathbb{N} \models \phi,$$

in the same way as it is done for the standard Rosser sentence (cf. [17]). The idea of the proof of the theorem is to adapt the proof of $S \nvdash \neg\phi$ so that it gives a polynomial length S-proof of $Con_S(\overline{p(n)}) \rightarrow \neg\exists x(Pr_S(x, \neg\bar{\phi}) \wedge |x| \leq \bar{n})$ for some polynomial p. Note that the consequent of the implication is essentially $Con_{S+\phi}(\bar{n})$. Hence we get the statement of the theorem from Theorem 3.2(2).

In the rest of this proof we will use the symbol $\vdash^*$ to denote provability by a proof of length polynomial in n.

Lemma 3.8. *$S \vdash^* \neg\exists x(Pr_T(x, \bar{\phi}) \wedge 2^{|x|} < \bar{n})$, or equivalently, $S \vdash^* \forall x(Pr_T(x, \bar{\phi}) \rightarrow 2^{|x|} \geq \bar{n})$.*

Proof-sketch. The number of numbers x such that $2^{|x|} < n$ is at most $n - 1$. Thus S can verify that each of them is not a T-proof of ϕ using a proof whose total length is polynomial in n. We are using the fact (not provable in S) that $T \nvdash \phi$. ⊣

Now we continue with the proof of the theorem. We will abbreviate the formula $\exists y(Pr_S(y, x) \wedge |y| \leq z)$ by $Prl_S(z, x)$. We have

$$S \vdash^* Prl_S(\bar{n}, \overline{\neg\phi}) \rightarrow Prl_S(\bar{m}_2, \ulcorner(Prl_S(\bar{n}, \overline{\neg\phi}))\urcorner), \tag{3}$$

[7] A high-level proof is: Con_S is the Π_1-reflection principle for S and Con_T is a Π_1 sentence. Hence $R \vdash Con_T$.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

where m_2 is polynomially bounded by n. This follows by the principle: given a proof of length $\leq n$, S can prove that such a proof exists by a proof of length polynomial in n (see Fact 2.3). Observe that according to (2)

$$S \vdash^* (\forall x(Pr_T(x, \bar{\phi}) \to 2^{|x|} \geq \bar{n}) \wedge Prl_S(\bar{n}, \overline{\neg\phi})) \to \phi.$$

(This is in fact provable by a logarithmic length proof.) From Lemma 3.8, we get

$$S \vdash^* Prl_S(\bar{n}, \overline{\neg\phi}) \to \phi.$$

Again, by formalizing this proof,

$$S \vdash^* Prl_S(\bar{m}_2, \ulcorner Prl_S(\bar{n}, \overline{\neg\phi})\urcorner) \to Prl_S(\bar{m}_3, \bar{\phi}),$$

for some m_3 polynomially bounded by n. By (3), this reduces to

$$S \vdash^* Prl_S(\bar{n}, \overline{\neg\phi}) \to Prl_S(\bar{m}_3, \bar{\phi}).$$

Since the antecedent says that $\neg\phi$ is provable by a proof of length $\leq n$, we get

$$S \vdash^* Prl_S(\bar{n}, \overline{\neg\phi}) \to \neg Con_S(\bar{m}),$$

for some m polynomially bounded by n. Since $Con_S(\bar{m})$ has an S-proof of length polynomial in m, we get a polynomial length S-proof of the negation of the antecedent, hence also of $Con_{S+\phi}(\bar{n})$. ⊣

More recently, Emil Jeřábek came up with an alternative proof. His idea is to use a sentence ϕ such that both ϕ and $\neg\phi$ are interpretable in S. It is well-known that such sentences exist for every sufficiently strong theory S (see Theorem 4.5(5) in [17]).[8] These sentences are also modifications of the classical Rosser sentences. Then we only need:

Lemma 3.9. *Let $S, T \in \mathcal{T}$ be sequential finitely axiomatized theories. If T is interpretable in S, then there exists a polynomial p such that*

$$S^1_2 \vdash \forall x(Con_S(p(x)) \to Con_T(x)).$$

Proof-sketch. Suppose $S, T \in \mathcal{T}$ are sequential finitely axiomatized theories and T is interpretable in S. Given a proof of contradiction in T we can translate the proof into a proof of contradiction in S. The translation is explicit and simple, therefore it can be formalized in S^1_2. ⊣

Continuing the proof of Jeřábek, this implies that S proves $Con_{S+\phi}(\bar{n})$ and $Con_{S+\neg\phi}(\bar{n})$ by proof of polynomial lengths, provided that S is finite and sequential and we can use Theorem 3.2(2). On the other hand, for every consistent T, either $T \nvdash \phi$ or $T \nvdash \neg\phi$. ⊣

Since $S + \neg Con_s$ is interpretable in S (see Theorem 4.5(1) in [17]), we also get the following result.

Proposition 3.10. *Let $S \in \mathcal{T}$ be a sequential finitely axiomatized theory and let $I\Sigma_1 \subseteq S$.*[9] *Then S proves $Con_{S+\neg Con_s}(\bar{n})$ by proofs of polynomial (in n) lengths.*

[8] In [17] the assumption is that $I\Sigma_1 \subseteq S$, but we believe that it could be reduced to $S^1_2 \subseteq S$.

[9] Again, we can surely use a much weaker assumption than $I\Sigma_1 \subseteq S$.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

Finally, we observe that if T is not finitely axiomatized, then it is possible that it does not prove the sentences $Con_T(\bar{n})$ by proofs of polynomial (in n) lengths. I am indebted to Fedor Pakhomov for suggesting this problem.

Proposition 3.11. *Suppose* CON^N *is true. Then for every* $S \in \mathcal{T}$*, there exists* S'*,* $S \subseteq S' \in \mathcal{T}$ *such that the lengths of* S'*-proofs of* $Con_{S'}(\bar{n})$ *cannot be bounded by a polynomial in* n*.*

Proof. Let $S \in \mathcal{T}$ be given. Let $T \in \mathcal{T}$ be such that the lengths of S-proofs of $Con_T(\bar{n})$ cannot be bounded by a polynomial in n. Such a T exists if we assume CON^N. Define

$$S' := S \cup \{\neg Pr_T(\bar{N}, \bot) \mid N \in \mathbb{N}\}.$$

Suppose that S' proves $Con_{S'}(\bar{n})$ by a proof of length $p(n)$ for some polynomial p. Such a proof can only use a polynomial number of the axioms of the form $\neg Pr_T(\bar{N}, \bot)$ with $|N| \leq p(n)$ (while there are exponentially many such axioms). But S can prove each of these axioms by a proof of polynomial size. Hence

$$S \vdash Con_{S'}(\bar{n})$$

by a polynomial size proof. Furthermore, for a suitable polynomial q,

$$S \vdash Con_{S'}(\overline{q(n)}) \rightarrow Con_T(\bar{n})$$

by a polynomial (in fact, even polylogarithmic) length proof. To prove this claim, one only needs to formalize the following argument in S:

> "Suppose there exists a proof P of contradiction in T of length $\leq n$. Then it is possible to prove in S that $Pr_T(\bar{P}, \bot)$ holds true by a proof of length polynomial in $|P|$, i.e., also polynomial in n. Hence we would get a proof of contradiction in S' of length $q(n)$."

Thus we would get polynomial length S-proofs of $Con_T(\bar{n})$ contrary to our assumption. ⊣

3.2. A finite reflection principle. Recall that the sentences expressing consistency of a theory T are special cases of *reflection principles* (see [40]). There are many versions of reflection principles. Here we will focus on the uniform Σ_1-reflection principles.

The *uniform* Σ_1*-reflection principle for* T*,* $\Sigma_1 RFN_T$, is the following schema for all Σ_1 sentences $\sigma(x)$ with one free variable x

$$\forall x \forall u (Pr_T(u, \ulcorner \sigma(\dot{x}) \urcorner) \rightarrow \sigma(x)),$$

where $\ulcorner \sigma(\dot{x}) \urcorner$ denotes the function that assigns the Gödel number of the formula $\sigma(\bar{x})$ to a given x, and $Pr_T(u, v)$ says that u is a proof of v in T. The principle is true if T is Σ_1-sound, i.e., T does not prove a false Σ_1 sentence. This schema can be axiomatized by a *single sentence* using a partial truth definition for Σ_1 formulas (a universal Σ_1 formula). Therefore the principle is called *"uniform"* and abbreviated with capital letters.

In order to get a meaningful finite version of $\Sigma_1 RFN_T$ we have to make a couple of modifications. We start by defining a finite Σ_1^b reflection principle for one formula.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

Definition 3.12. Let T be a theory, let $\alpha(x)$ be a Σ_1^b formula and let $n \in \mathbb{N}$. Then $\Sigma_1^b Rfn_T^{\alpha}(\bar{n})$ will denote the sentence:

$$\forall u, x, |u| \leq \bar{n}, |x| \leq \bar{n} \left(Pr_T(u, \ulcorner\alpha(\bar{x})\urcorner) \to \alpha(x)\right).$$

Having defined the reflection principle for one formula, we can study the schema, i.e., the set of sentences $\Sigma_1^b Rfn_T^{\alpha}(\bar{n})$ for all Σ_1^b formulas, but it is more interesting to have a single sentence for every n from which all instances are derivable by short proofs. To this end we need a universal Σ_1^b formula. One can construct a Σ_1^b formula μ_1 such that for every Σ_1^b formula $\alpha(x)$ there exist a natural number e and a polynomial p such that

$$|z| \geq p(|x|) \to (\alpha(x) \equiv \mu_1(\bar{e}, x, z)) \tag{4}$$

is provable in S_2^1 (see [17], page 336). The sentences that we are going to define are essentially $\Sigma_1^b Rfn_T^{\mu_1}(\bar{n})$.

Definition 3.13. The *finite uniform Σ_1^b principle* is the sequence of sentences $\Sigma_1^b RFN_T(\bar{n})$, $n \in \mathbb{N}$, defined by

$$\forall e, u, x, z(|e|, |u|, |x|, |z| \leq \bar{n} \wedge Pr_T(u, \ulcorner\mu_1(\bar{e}, \bar{x}, \bar{z})\urcorner) \to \mu_1(e, x, z)).$$

Lemma 3.14. *For every Σ_1^b formula $\alpha(x)$, there exist polynomials q and r such that S_2^1-proofs of the sentences*

$$\Sigma_1^b RFN_T(\overline{q(n)}) \to \Sigma_1^b Rfn_T^{\alpha}(\bar{n})$$

can be constructed in time $r(|n|)$.

Proof. Let $e \in \mathbb{N}$ and p be such that (4) is provable in S_2^1. Let $n \in \mathbb{N}$ be such that $n \geq |e|$ and let $m = p(n)$. The following argument can be done in S_2^1.

> Suppose $|u|, |x| \leq n$ and $Pr_T(u, \ulcorner\alpha(\bar{x})\urcorner)$. Then we also have $|u|, |x| \leq m$ and, since (4) is provable in T, we have $Pr_T(u', \ulcorner\mu_1(\bar{e}, \bar{x}, \overline{2^m})\urcorner)$ for some u'. The proof u' is constructed from u using the proof of (4) in T, which adds only a constant to the length and a small part in which this sentence is instantiated for the numerals $\bar{x}$ and $\overline{2^m}$. This makes the proof u' at most polynomially longer than m. Let m' be this polynomial bound. Applying $\Sigma_1^b RFN_T(\overline{m'})$, we get $\mu_1(\bar{e}, \bar{x}, \overline{2^m})$. Then using (4) in S_2^1, we finally get $\alpha(\bar{x})$.

Now we only need to observe that the above S_2^1 proof was explicitly constructed and the number of steps and the length of the formulas involved are of length polynomial in $|n|$. ⊣

Corollary 3.15. *Let $S, T \in \mathcal{T}$. Suppose that*

1. *$T \vdash \forall x.\phi(x)$, where $\phi \in \Sigma_1^b$, and*
2. *S-proofs of the sentences $\Sigma_1^b RFN_T(\bar{n})$ can be constructed in polynomial time in n.*

Then S-proofs of the sentences $\phi(\bar{m})$ can be constructed in time $r(|m|)$ for some polynomial r.

Proof. Since $\forall x.\phi(x)$ is provable in T, the sentences $\phi(\bar{m})$ have T-proofs of length bounded by $q(|m|)$ for some polynomial q. This is provable in S_2^1, so also in S. According to the assumption about S and by Lemma 3.14, one can construct in polynomial time proofs of $\Sigma^b_1 Rfn^\phi_T(\overline{q(|m|)})$ in polynomial time in $|m|$. Thus we get S-proofs of $\phi(\bar{m})$ in polynomial time. $\dashv$

Using $\Sigma^b_1 RFN_T(\bar{n})$, we can state a conjecture similar to our conjecture about $Con_T(\bar{n})$.

Conjecture 3.16 (RFN^N_1). *For every $S \in \mathcal{T}$, there exists $T \in \mathcal{T}$ such that the lengths of S-proofs of $\Sigma^b_1 RFN_T(\bar{n})$ cannot be bounded by $p(n)$ for any polynomial p.*

When α is $0 = 1$, then $\Sigma^b_1 Rfn^\alpha_T(\bar{n})$ is equivalent to $Con_T(\bar{n})$ and this equivalence has an S^1_2-proof of length polynomial in $|n|$. Thus, by Lemma 3.14, there exists a polynomial q such that $\Sigma^b_1 RFN_T(\overline{q(n)}) \to Con_T(\bar{n})$ has an S^1_2-proof of length polynomial in $|n|$. Consequently, Conjecture CON^N implies Conjecture RFN^N_1.

We will prove that Conjecture RFN^N_1 implies **NP**$\neq$**coNP**.

Proposition 3.17. *If* **NP**=**coNP**, *then there exists $S \in \mathcal{T}$ such that for all $T \in \mathcal{T}$, the lengths of S-proofs of $\Sigma^b_1 RFN_T(\bar{n})$ can be bounded by $p(n)$ for some polynomial p.*

Proof. The basic idea is to take some base theory and add all sentences of the form $\Sigma^b_1 RFN_T(\bar{n})$ that are true as axioms, regardless whether or not T is consistent. Assuming **NP**=**coNP**, it is possible to test these sentences in nondeterministic polynomial time for each T. Since the polynomial bound is different for different theories T we have to apply padding.

Here is a sketch of a proof in more detail. Assume **NP**=**coNP**. Instead of just the sentences expressing $\Sigma^b_1 RFN_T(x)$, we will consider all formulas with one free variable x of the form

$$Q_1\, y_1, |y_1| \leq p_1(x) \dots Q_k\, y_k, |y_k| \leq p_k(x)\ \phi(y_1, \dots, y_k),$$

where $Q_i \in \{\forall, \exists\}$ and $\phi(y_1, \dots, y_k) \in \Delta^b_1$. Let Γ denote the class of such formulas. Our assumption implies that every formula $\phi(x) \in \Gamma$ is equivalent to a formula from this class in which all quantifiers are existential. Consequently, for every $\phi(x) \in \Gamma$, there exists a nondeterministic Turing machine M that accepts only true sentences of the form $\phi(\bar{n})$ and runs in time polynomial in n. With some additional (and tedious) work, one can show:

Case 1. There exists a nondeterministic Turing machine M that accepts true sentences of the form $\phi(\bar{n})$, for $\phi(x) \in \Gamma$ and $n \in \mathbb{N}$, such that for every $\phi(x) \in \Gamma$, there exists a polynomial p_ϕ such that on sentences $\phi(\bar{n})$ the machine always stops after $p_\phi(n)$ steps.

For every $\phi(\bar{n}) \in \Gamma$ and $n, m \in \mathbb{N}$, let a padded version of $\phi(\bar{n})$ be

$$\phi(\bar{n})_m := \phi(\bar{n}) \vee (\underbrace{0 = 1 \wedge \cdots \wedge 0 = 1}_{m}).$$

Clearly, $\phi(\bar{n})$ is derivable from $\phi(\bar{n})_m$ in predicate calculus using a proof whose length is polynomial (in fact, linear) in the length of the sentence

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

$\phi(\bar{n})_m$. Using a simple modification of the machine M from the previous claim, one can prove:

Case 2. There exists a nondeterministic *polynomial time* Turing machine M' that accepts only some true sentences of the form $\phi(\bar{n})_m$, for $\phi(x) \in \Gamma$, $n, m \in \mathbb{N}$ and such that for every $\phi(x) \in \Gamma$, there exists a polynomial p_ϕ such that for all $n \in \mathbb{N}$, M' accepts $\phi(\bar{n})_m$ for some $m \leq p_\phi(n)$.

Let S' be the theory axiomatized by the sentences accepted by M'. By construction, every true sentence of the form $\phi(\bar{n})$, $\phi(x) \in \Gamma$ has an S'-proof of length polynomial in n; in particular, this holds true for true sentences of the form $\Sigma_1^b RFN_T(\bar{n})$. The only issue is that the set of axioms is in **NP** and, maybe, not in **P**. This can also be fixed by padding: we can encode accepting computations into padding. E.g., by using $0 = 1$ and $1 = 0$, we can encode an arbitrary bit string and let a padded formula be accepted iff the padding encodes an accepting computation of M' on input $\phi(\bar{n})_m$. $\dashv$

The reason for introducing the conjecture about $\Sigma_1^b RFN$ is that it enables us to connect diverging branches of so far postulated conjectures, as we will see shortly. One can certainly study similar statements based on stronger reflection principles for classes of formulas $\Sigma_2^b, \Sigma_3^b, \dots$. The strength of these conjectures decreases with increasing indexes, so they are not interesting if we are looking for stronger conjectures. However, the study of these conjectures may reveal further interesting connections.

3.3. What is the finite Gödel theorem? We finish this section with a remark concerning the question what should be called the finite Gödel theorem. If Conjecture CON^N were proven true, we would certainly advocate calling it the finite Gödel theorem. However, one can also argue that the connection is different. Note that if T proves Con_S, then T-proofs of $Con_S(\bar{n})$ are very short; they are of logarithmic length in n, because the length of $Con_S(\bar{n})$ is logarithmic (recall that we are using binary numerals) and this sentence follows from Con_S by substitution (if we formalize Con_S as $\forall x. Con_S(x)$). Using this fact, we can derive Gödel's theorem from Friedman's lower bound n^ε on the lengths of T-proofs of $Con_T(\bar{n})$. So Friedman's lower bound can also be viewed as the finite Gödel theorem.

Proving Gödel's theorem in this roundabout way is certainly not natural, but in some cases it may be useful. Using estimates on finite consistency statements, we proved [31] that S_2 does not prove bounded consistency of the apparently weaker theory S_2^1, which ruled out an approach to the separation problem of these two theories. (Bounded consistency means that we only consider proofs in which all formulas are bounded.)

§4. Fast growing functions and hard search problems.

An important property of first-order theories studied in classical proof theory is their strength measured by the set of arithmetical sentences provable in them. Among the arithmetical sentences the most important role is played by Π_1 and Π_2 sentences. A proper Π_2 sentence, a sentence that is not equivalent to a Π_1 sentence, expresses the fact that some function is total. Specifically,

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

$\forall x \exists y.\phi(x, y)$, where ϕ is a bounded formula, can be interpreted as saying that there exists a computable function such that $\forall x.\phi(x, f(x))$. If we cannot write it equivalently using a formula $\forall x.\psi(x, y)$, where in ψ all quantifiers are bounded, then f has to grow faster than all functions defined by the terms of the theory. Moreover, for pairs of natural theories S and T with T essentially stronger than S, there are provably total computable functions in T that cannot be bounded by computable functions provably total in S. One can say that *"T proves the existence of larger numbers than S"*. This intuition can be made more precise using cuts of nonstandard models of arithmetic in which the arithmetical theories of S and T are satisfied: in general, T requires longer cuts than S.[10]

Remark 4.1. It is important to realize what "provably total" means. For a given theory and a computable function f, we can always find a Σ_1 definition for which the totality of f is not provable (e.g., given a defining formula $\phi(x, y)$, we can extend it by adding the consistency of T, i.e., $\phi(x, y) \wedge Con_T(x)$). So when we say that f is provably total, we mean that f is provably total *for some Σ_1 definition of f*.

4.1. Total polynomial search problems. We are interested in the exponential domain, which means that we only consider functions f such that the length of $f(x)$ is bounded by $p(|x|)$ for some polynomial p, so it does not make sense to compare the growth rate of the functions. Instead, we study the complexity of these functions. The class of sentences corresponding to Π_2 are $\forall\hat{\Sigma}^b_1$ sentences—the sentences starting with unbounded universal quantifier followed by a $\hat{\Sigma}^b_1$ sentence. Essentially, this class consists of sentences of the form

$$\forall x \exists y(|y| \leq p(|x|) \wedge \phi(x, y)), \tag{5}$$

where ϕ is a formalization of a polynomial time relation (i.e., $\phi \in \Delta^b_1$) and p is some polynomial. There is a computational task naturally associated with such sentences. Since this is important, we define it formally.

Definition 4.2. A *total polynomial search problem* is given by a pair (p, R), where p is a polynomial and R is a binary relation such that

1. R is decidable in polynomial time,
2. $\mathbb{N} \models \forall x \exists y(|y| \leq p(|x|) \wedge R(x, y))$.

The computational task is, for a given x, find y such that $|y| \leq p(|x|) \wedge R(x, y)$.

The class of all total polynomial search problems will be denoted by **TFNP**.[11] Here are two examples of **TFNP** problems.

Example 4.3. This example is based on the Pigeon-Hole Principle, which says that there is no one-to-one mapping from an $N + 1$-element set to an N-element set. The computational task associated with this principle is:

[10] Consider cuts that contain a fixed nonstandard element.

[11] The abbreviation **TFNP** is standard, but is rather misleading; the class is not a class of functions and it is not defined using **NP** relations. Therefore we used **TPS** in [34].

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

given a mapping from an $N+1$-element set to an N-element set, find a "collision", which is a pair $x \neq x'$ such that $f(x) = f(x')$. This problem is algorithmically trivial if the mapping is given as a list of pairs $(x, f(x))$. In this case N is less than the input length. However, if the problem is presented so that N is exponential in the input length, no polynomial time algorithm is known. Such a representation can be defined using Boolean circuits, or polynomial time algorithms that compute the function f. In fact, researchers in cryptography believe that the problem is hard even if the mapping is from $[N]$ to $[M]$ for M much smaller than N. These *hash functions* are used in various protocols.

A **TFNP** problem based on the Pigeon-Hole Principle can formally be defined as follows. Take a polynomial time computable function $f(r, x)$; think of f as a set of polynomial time computable functions of one variable x parametrized by r. Define a binary relation computable in polynomial time by

$$R(r, u) :\equiv (u \leq r \wedge f(r, u) \geq r) \vee \exists x, x' \leq r(u = (x, x') \wedge f(r, x) = f(r, x')).$$

In this formula, u is a witness of the fact that f does not map $\{0, \ldots, r\}$ into $\{0, \ldots, r-1\}$ or a witness of a collision. A polynomial bound on $|u|$ is determined by a polynomial bound on the lengths pairs of elements less than r.

Example 4.4. Our second example is based on the problem of factoring integers. Again the problem is nontrivial only if the number to be factored is presented in binary (decimal etc.) notation, in which case it is exponential in the input length. Since the search problem must have a solution for every number N, we have to distinguish the cases when N is prime and when it is composite. It is well-known that this is decidable in polynomial time. Formally, we define a binary relation computable in polynomial time by

$$Q(N, M) :\equiv N \text{ is prime } \vee (1 < M < N \wedge M \text{ divides } N).$$

The bound on M is simply $|M| \leq |N|$. A solution is any number M, $|M| \leq |N|$, if N is prime, or a proper factor if N is composite.

Having the concept of a total polynomial search problem, we can now replace the *growth rate* of functions by the *computational complexity* of finding solutions. Not surprisingly, the situation is much less clear than in the classical setting. Firstly, we can only hypothesize about the computational complexity of specific search problems. But this is what we expected and are ready to face. Secondly, we do not have a quantitative measure of complexity that we could apply to this kind of computational problems. We can distinguish problems for which the task is solvable in polynomial time from those for which it isn't, but some evidence suggests that there are also distinct classes of problems that are not solvable in polynomial time and have different complexity. To compare the complexity of different problems, we use reductions. Polynomial reductions are known for decision problems and used, in particular, in the theory of **NP** completeness. For **TFNP** there is also a natural concept of polynomial reduction. (Note that **TFNP** is not a class of decision problems, so we do need a different concept.)

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

Definition 4.5 ([20]). Let R and S be total polynomial search problems. We say that R is *polynomially reducible* to S if R can be solved in polynomial time using an oracle that gives solutions to S. We say that R and S are *polynomially equivalent* if there are polynomial reductions in both directions. We say that R is *many-one polynomially reducible* to S, if it is polynomially reducible using one query to the oracle for S.

Many-one polynomial reducibility can be equivalently defined by the condition: *there are functions f and g computable in polynomial time such that for all x and z,*

$$S(f(x),z) \;\Rightarrow\; R(x,g(x,z)),$$

were we are assuming that polynomial bounds on the lengths of numbers involved are implicit in the relations R and S.

Reductions enable us to study the structure of **TFNP** and define subclasses. We are interested in classes that are closed under polynomial reductions. One important class is **PHP**, the class of all **TFNP** problems reducible to an instance of the Pigeon-Hole Problem as described in Example 1 above. Several other classes were defined already in the seminal paper [20]. They enable one to show that a problem is probably not solvable in polynomial time. Specifically, if one proves that a problem is complete in one of the well-known classes, it implies that the problem is not solvable in polynomial time unless the class collapses to the bottom class consisting of all problems solvable in polynomial time.

From the point of view of computational complexity, it is natural to identify polynomially equivalent problems. However, we should bear in mind that from the point of view of a particular theory, two definition of the same problem may behave differently, as we noted above. We will consider definitions of **TFNP** by Δ_1^b formulas and for a given theory we will take "the best possible definition". Formally, this is defined as follows.

Definition 4.6. 1. A Δ_1^b *definition of a* **TFNP** *problem* (p,R) is a pair (q,ϕ) where q is a polynomial and ϕ is a Δ_1^b formula such that

$$\mathbb{N} \models \forall x,y((|y| \leq p(|x|) \wedge R(x,y)) \equiv (|y| \leq q(|x|) \wedge \phi(x,y))).$$

2. We say that $(p,P) \in$ **TFNP** is *provably total in a theory T*, if for some Δ_1^b definition (q,ϕ) of (p,P), T proves that

$$\forall x \exists y(|y| \leq q(|x|) \wedge \phi(x,y)).$$

3. The set of all $(p,P) \in$ **TFNP** provably total in T will be denoted by **TFNP**(T). The set of all $P \in$ **TFNP** polynomially reducible to some $Q \in$**TFNP**(T) will be denoted by **TFNP**$^*(T)$.

Note that according to our definition of the class Δ_1^b (in Section 2.3), the formula ϕ must be a Σ_1^b formula equivalent to a Π_1^b formula *provably in S_2^1* (to ensure that it defines a set in **P** it does not suffice to have a proof in T). On the other hand, we do *not* require that a problem P in **TFNP**$^*(T)$ is *provably* reducible to some $Q \in$**TFNP**(T). The difference between **TFNP**(T)

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

and **TFNP**$^*(T)$ is small; in fact, if we defined **TFNP** using **NP** relations (see $\overline{\mathbf{TFNP}}$ below), these classes would be the same.

To characterize low complexity theorems of fragments of arithmetic is an important problem studied in proof complexity. In particular, we are interested in sentences that are universal closures of Σ_1^b formulas. Naturally, we want to identify sentences that express the same fact. The best way to do that is to focus on provably total polynomial search problems. Provably total polynomial search problems of all fragments of bounded arithmetic S_2^i, $i = 1, 2, \dots$, have been characterized using combinatorial principles [1,2,36,39]. (S_2^i is S_2 with the induction schema (1) restricted to Σ_i^b formulas.) For S_2^1 they are all **TFNP** problems that are solvable in polynomial time (the lowest class in **TFNP**). The class of provably total problems of S_2^2 turned out to be surprisingly the class *Polynomial Local Search*, a class that had been introduced in [20].

Here is another important conjecture.

Conjecture 4.7 (TFNP). *For every theory $T \in \mathcal{T}$ there exists a* **TFNP** *problem P that is not polynomially reducible to any* **TFNP** *problem provably total in T. Stated in symbols* **TFNP**$^*(T) \neq$ **TFNP**.[12]

The weaker statement **TFNP**$(T) \neq$ **TFNP**, in plain words, says that, for every theory $T \in \mathcal{T}$, there exists a total polynomial search problem (p, R) such that T cannot prove that the problem is total for any proper definition (definition by a Δ_1^b formula) of (p, R). This means that the unprovability in T is not caused by a particular way we define the problem, but by a semantic property of it that we imagine as high computational complexity. We state the conjecture in the stronger form, **TFNP**$^*(T) \neq$ **TFNP**, because **TFNP**(T) may not be closed under polynomial time reductions.

Let us compare this conjecture with the corresponding statement about fast growing recursive functions. One can easily prove by diagonalization that for every $T \in \mathcal{T}$, there exists a computable function f which grows faster than any computable function provably total in T. This means that for any computable function g provably total in T, there exists an n_0 such that $f(n) > g(n)$ for all $n \geq n_0$. Thus for any formalization of f by a Σ_1 formula T cannot prove that f is total. In the above conjecture, the condition that f cannot be bounded by provably total functions is replaced by the condition that a **TFNP** problem is not polynomially reducible to **TFNP** problems that are provably total in T.

All conjectures in this area can be stated in purely complexity theoretical terms. The above conjecture has an especially simple equivalent form, which we state now.

Conjecture 4.8 (equivalent to TFNP). *There is no complete problem in* **TFNP**, *i.e., there exist no* **TFNP** *problem to which all* **TFNP** *problems can be polynomially reduced.*

[12] We distinguish the complexity class **TFNP** and the conjecture about it TFNP by different fonts.

The proof of the equivalence of the versions is easy. To prove that the first version implies the second, suppose the second is false. Let P be a complete problem in **TFNP**. Then take a fragment of arithmetic and add the axiom that (a formalization of) P is total.

The converse implication follows immediately from the following fact.

LEMMA 4.9. *For every $T \in \mathcal{T}$, there exists a* **TFNP** *problem (p, P) such that all* **TFNP** *problems provably total in T are many-one polynomially reducible to (p, P).*

PROOF. The proof is based on the fact that one can effectively enumerate all problems in **TFNP**$^*(T)$. (Such proofs are routine and we include a proof here only because it demonstrates a method that can be applied in other similar situations, in particular, we will use it in Proposition 6.6.) The basic idea is to connect all provably total problems into one. We can recognize a definition of a provably total problem by finding a proof of the totality for this definition. A minor complication is that different provably total problems may require different polynomials as bounds on the witnesses and bounds in the Δ_1^b formulas defining them. This can easily be solved by suitable padding.

Now we present the argument in more detail. Recall that from the point of view of provability in a theory, it does not matter if we use Δ_1^b formulas or, more generally, Σ_1^b in the definition of the problems. So, for the sake of simplicity, we will enumerate Σ_1^b formulas.

Given a Σ_1^b formula $\psi(x)$, we say that $r(n)$ is a syntactic nondeterministic time bound for ψ if the bounds at quantifiers in the formula ensure that $\psi(x)$ is decidable by a nondeterministic Turing machine in time $r(n)$ where n is the length of x. Since ψ is a Σ_1^b formula, there always exists a polynomial r that is such a bound for ψ.

Let $T \in \mathcal{T}$ be given. We define a binary relation $R(u, v)$ by the following condition:

- if $u = (x', \phi, q, d, a)$ is a quintuple such that ϕ is a Σ_1^b formula, q is a polynomial, d is $|T|$-proof of $\forall x \exists y(|y| \leq q(|x|) \wedge \phi(x, y))$, and $|a| = r(|x'|)$, where r is a syntactic nondeterministic time bound for $\exists y(|y| \leq q(|x|) \wedge \phi(x, y))$, then $\phi(x', v)$.

Note that $\Phi := \forall x \exists y(|y| \leq q(|x|) \wedge \phi(x, y))$ is a Π_1 sentence and T is consistent and Σ_1-complete (since it contains S_2^1). Hence if T proves Φ, then Φ is true in $\mathbb{N}$.

The relation R is computable in nondeterministic polynomial time, because the condition on (x', ϕ, q, d, a) is a simple syntactical condition and if the condition is satisfied, $\phi(x', v)$ can be computed in nondeterministic polynomial time bounded by $|a|$. Further, for every u there exists some v, $|v| \leq |u|$, such that $R(u, v)$ holds true, because if the condition on (x', ϕ, q, d, a) is satisfied, then for every x' there exists v, $|v| \leq |a|$, that satisfies $\phi(x', v)$, and if the condition is not satisfied, then one can take $v = 0$.

The fact that we only know that R is computable in *nondeterministic* polynomial time is not a problem. Clearly, there exists a ternary relation P'

computable in polynomial time and a polynomial p' such that

$$R(u, v) \equiv \exists w(|w| \leq p'(|u|, |v|) \wedge P'(u, v, w)).$$

So we define

$$P(u, y) :\equiv \exists v, w(y = (v, w) \wedge P'(u, v, w))$$

and note that $|y| \leq p(|u|)$ for some polynomial p, because $|v| \leq |u|$ and $|w| \leq p'(|u|, |v|) \leq p'(|u|, |u|)$.

Let a **TFNP** problem (q, Q) be given and suppose that it is provably total in T. We have a Σ_1^b formula ϕ and a polynomial q that defines the problem and a T-proof of totality d for this representation. Also we have a nondeterministic polynomial time bound r for $\exists y(|y| \leq q(|x|) \wedge \phi(x, y))$. We define a reduction of (q, Q) to (p, P) by

$$x \mapsto f(x) := (x, \phi, q, d, 2^{r(|x|)}).$$

Given a witness (v, w) for $P(f(x), (v, w))$ we get a witness for $Q(x, v)$ simply by taking the first element from the pair (v, w). ⊣

We are indebted to to Emil Jeřábek for the following proposition.

Proposition 4.10 (E. Jeřábek, unpublished). *There exists a complete problem in* **TFNP** *w.r.t. polynomial reductions if and only if there exists a complete problem in* **TFNP** *w.r.t.* many-one *polynomial reductions.*

The proposition is an immediate corollary of the following lemma.

Lemma 4.11. *For every* **TFNP** *problem P, there exists a* **TFNP** *problem P' such that for every* **TFNP** *problem Q, if Q is polynomially reducible to P, then Q is many-one polynomially reducible to P'.*

Proof. Let P be given by a polynomial p and a binary relation R. We define a binary relation $R'(u, v)$ as follows. Interpret a string u as an encoding of a string x and an oracle Boolean circuit C. We will only allow oracle circuits that have $p(n)$ input bits for a possible oracle answer for each query of length n. Then $R'((x, C), v)$ will be defined to be true if v encodes a computation of C on input x with the oracle queries and answers to be pairs r, s such that $R(r, s)$ holds true; in other words, v encodes a computation of C that uses P as an oracle. Furthermore, $R'(u, v)$ is defined to be true, if u does not have the form described above. Clearly, R' defines a total problem: given (x, C), we can run C on input x using P as an oracle.

Suppose Q is reducible to P using a polynomial time query machine M. For each input x for the problem Q, we can construct in polynomial time an oracle Boolean circuit C that simulates computations of M on x. Given a string v such that $R'((x, C), v)$, we get an output string y of the computation of M that satisfies $Q(x, y)$, because M is a polynomial reduction of Q to R. So the reduction is given by the polynomial time functions $x \mapsto (x, C)$ and $v \mapsto y$, where y is the output of the computation encoded by v. ⊣

Furthermore, Jeřábek noted that we also get a conjecture equivalent to TFNP if we use the following modification. Let us denote by $\overline{\textbf{TFNP}}$ the class of search problems defined in the same way as **TFNP** except that the binary

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

relations are only required to be in **NP**.[13] Many-one polynomial reductions for $\overline{\textbf{TFNP}}$ are defined exactly in the same way as for **TFNP**.

Proposition 4.12. *There exists a complete problem in* **TFNP** *if and only if there exists a complete problem in* $\overline{\textbf{TFNP}}$.

Proof-hint. (1) Every problem P in **TFNP** is, by definition, also in $\overline{\textbf{TFNP}}$. (2) Let $Q \in \overline{\textbf{TFNP}}$. Let Q be given by a binary relation $\exists^p z.R(x, y, z)$. Then the binary relation R' defined by

$$R'(x, (y_1, y_2)) := R(x, y_1, y_2)$$

defines a problem in **TFNP**. ⊣

4.2. Some arguments supporting the conjecture. It is always difficult to justify a mathematical conjecture. Either the sentence is true, or it is false, but unlike in physics, in mathematics there are no experiments that may support one or the other. Thus the belief in a conjecture is based on subjective feelings. Here are our reasons why we believe that conjecture TFNP should be true.

1. Every **TFNP** problem is based on some mathematical principle that ensures that for every input there exists a solution. Although these principles are simple for the basic classes of **TFNP** problems, it seems likely that there is no universal mathematical principle that would work for every **TFNP** problem.
2. Combinatorial characterizations of provably total polynomial search problems have been obtained for some fragments of Bounded Arithmetic. The description of these combinatorial problems suggests that their strength increases with increasing strength of the theories.[14]
3. An oracle has been constructed relative to which the conjecture holds true [35].
4. The connection with search problems verifying the consistency of a theory that we describe below can also be viewed as a supporting argument.

4.3. Herbrand consistency search. Conjecture TFNP has another equivalent form in which the concept of consistency plays a key role. The well-known Herbrand theorem provides a "combinatorial" characterization of provability in predicate calculus (see, e.g., [8]). In particular one can characterize the consistency of theories. Let us consider logic without equality and the special case of universal sentences. According to Herbrand's theorem a universal sentence $\Phi := \forall x_1 \dots \forall x_k.\phi(x_1, \dots, x_k)$, where ϕ is quantifier free, is consistent if and only if for every family of terms τ_{ij}, $i = 1, \dots, n$,

[13] It would be more logical to use **TFP** for what is called **TFNP** and reserve **TFNP** for $\overline{\textbf{TFNP}}$.

[14] We only hypothesize that the strength of fragments S^i_2 of Bounded Arithmetic increases with increasing i, but this hypothesis is supported by a connection with the Polynomial Hierarchy in computational complexity [28].

$j = 1, \dots, k$,

$$\bigwedge_{i=1}^{n} \phi(\tau_{i1}, \dots, \tau_{ik}) \tag{6}$$

is satisfiable as a propositional formula. Sentences expressing consistency using standard proofs (in Hilbert-style, or Gentzen calculi) and sentences expressing consistency using Herbrand's theorem are equivalent provably in every theory that proves Herbrand's theorem. However, the corresponding restricted finite versions of consistency statements are essentially different because the transformation of standard proofs into sets of terms that witness provability in Herbrand's theorem is nonelementary. But here we are interested in a different aspect of Herbrand's theorem: the complexity of finding a satisfying assignment for (6). Since the formula, as a proposition, is always satisfiable when Φ is consistent, every consistent universal sentence defines a natural **TFNP** problem.

Definition 4.13. Let $\Phi := \forall x_1, \dots, x_k.\phi(x_1, \dots, x_k)$ be a consistent universal sentence. Then $HCS(\Phi)$, the *Herbrand Consistency Search for* Φ, is the following total polynomial search problem. Given terms τ_{ij} in the language of Φ, $i = 1, \dots, n$, $j = 1, \dots, k$, find a truth assignment to the atomic subformulas occurring in $\phi(\tau_{i1}, \dots, \tau_{ik})$, for $i = 1, \dots, n$, that makes $\bigwedge_{i=1}^{n} \phi(\tau_{i1}, \dots, \tau_{ik})$ true.

For simplicity, we define Herbrand consistency search only for universal sentences in this article, but using Skolemization, one can easily extend this definition to conjunctions of prenex formulas. In [35] we proved the following theorem.

Theorem 4.14. *For every total polynomial search problem P, there exists a consistent universal sentence Φ such that the problem P is many-one polynomially reducible to $HCS(\Phi)$.*

Using this theorem we can state Conjecture TFNP in the following equivalent form.

Conjecture 4.15 (equivalent to TFNP). *For every theory $T \in \mathcal{T}$ there exists a consistent universal sentence Φ such that $HCS(\Phi)$ is not polynomially reducible to any* **TFNP** *problem provably total in T, i.e., $HCS(\Phi) \notin$* **TFNP**$^*(T)$.

This form of the TFNP conjecture suggests a natural question: what is a sentence Φ that is likely not in **TFNP**$^*(T)$? The following could be an answer to this question.

Conjecture 4.16 (TFNP$^+$). *Suppose $T \in \mathcal{T}$ is axiomatized by a universal sentence. Then T does not prove that $HCS(T)$ is total for any formalization of it by a Δ_1^b formula.*

Note that if T is strong enough to prove Herbrand's theorem, then it does not prove the totality of $HCS(T)$ formalized in a natural way, because if it did, it would prove its own consistency. However, this does not exclude the possibility that it proves the totality for some contrived definition. Although we call it a conjecture, we are not very confident that it is true. But suppose

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

it were true and suppose that $S \in \mathcal{T}$ is axiomatized by a universal formula and T is a theory that proves Herbrand's Theorem and the consistency of S. Then we would have $HCS(S) \in \mathbf{TFNP}^*(T) \setminus \mathbf{TFNP}^*(S)$. Thus according to this conjecture, adding the consistency of a theory to itself produces more provably total polynomial search problems (at least for theories axiomatized by a universal formula).

§5. Propositional proof systems, disjoint NP-pairs and disjoint coNP-pairs. So far we were concerned with first order theories. In this section we will show that one can also use other formal systems, namely, propositional proof systems, in order to state and study conjectures about incompleteness in the finite domain.

Let a language for classical propositional logic be fixed; say, we take connectives $\neg, \wedge, \vee$, and variables $p_1, p_2, \dots$. Let $TAUT$ be the set of all tautologies and SAT be the set of all satisfiable propositions. Following [12], we say that *a proof system is a polynomial time computable function P from* Σ^* *onto* **TAUT**.[15] If $P(w) = \phi$, we say that w is a proof of ϕ in the proof system P. This elegant definition captures three basic properties of proof systems:

1. the relation *"w is a proof of ϕ"* is decidable in polynomial time;
2. the system is sound;
3. the system is complete.

In the rest of this section the term "proof system" will always refer to "*propositional* proof system".

According to this definition, a proof can be any evidence that shows logical validity of a proposition. The standard formalizations of propositional calculus based on axioms and logical rules are systems from a special class of proof systems, called *Frege systems.*

We say that *a proof system P is polynomially bounded* if there exists a polynomial p such that every tautology ϕ has a P-proof of length at most $p(|\phi|)$. Since $TAUT$ is **coNP**-complete, the existence of a polynomially bounded proof system is equivalent to **NP**=**coNP**.

A weaker concept is length optimality. We say that *a proof system P is length-optimal* if for every proof system Q, there exists a polynomial p such that if ϕ has a proof of length n in P, then it has a proof of length at most $p(n)$ in Q. (Length-optimality is a nonuniform version of p-optimality that will be defined in Section 6.) In [27] we showed that Conjecture CON^N is equivalent to the following one.

Conjecture 5.1 (equivalent to CON^N). *There exists no length-optimal proof system.*

Why do we believe that this conjecture is true? An argument that we can give is based on a construction of proof systems used to prove that the two statements of Conjecture CON^N are equivalent. Given an arithmetical

[15] Recall that in this article Σ denotes $\{0, 1\}$, but in this definition it could be any finite alphabet of size at least 2.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

theory T, we can formalize the concept of a propositional tautology by some formula $\tau(x)$. For a given tautology t, we take its Gödel number n and treat any first order T-proof of $\tau(\bar{n})$ as a proof in a propositional proof system. Then it seem plausible that in stronger theories we can prove some tautologies by shorter proofs. Moreover, one can show that these proof systems are in a sense universal. So the fact that the logical strength of theories cannot be bounded is likely to be projected into these proof systems.

Another argument supporting the conjecture is from our experience with specific proof systems studied in proof complexity. Most systems are based on some class of formulas and deduction rules. If we enlarge the class of formulas then, usually, the system becomes stronger. For example, if we use quantified Boolean formulas instead of ordinary Boolean formulas, the system seems much stronger. For some weak systems, in particular, bounded depth Frege systems, this has actually been proven [21]. As, apparently, there is no limit on how strong expressive power formulas can have, we also believe that there is no limit on how efficient a proof system can be.

5.1. Disjoint NP pairs. In [37] Razborov defined the *canonical pair of a proof system P* to be the pair of sets $(PR(P), NSAT^*)$ where

$$\begin{aligned} PR(P) &:= \{(\phi, 2^m);\ \phi \text{ has a } P\text{-proof of length at most } m\},\\ NSAT^* &:= \{(\phi, 2^m);\ \neg\phi \text{ is satisfiable}\}. \end{aligned}$$

Note that this is a pair of two disjoint **NP** sets.

We say that *a disjoint* **NP** *pair (A, B) is polynomially reducible to a disjoint* **NP** *pair (C, D)* if there exists a polynomial time computable function f that maps A into C and B into D. We say that pairs (A, B) and (C, D) are *polynomially equivalent* if there are reductions between them in both directions.

It is not difficult to show that canonical pairs of proof systems are universal in the class of all disjoint **NP** pairs, which means that every disjoint **NP** pair (A, B) is polynomially reducible to the canonical pair of some proof system P. In fact, even more is true.

Proposition 5.2 ([15]). *For every disjoint* **NP** *pair (A, B), there exists a proof system whose canonical pair is polynomially* equivalent *to (A, B).*

Furthermore, if P and Q are proof systems and there exists a polynomial p such that for every tautology ϕ, if ϕ has a P-proof of length n, then ϕ has a Q-proof of length at most $p(n)$, then the canonical pair of P is polynomially reducible to the canonical pair of Q. Indeed, the mapping $(\phi, 2^n) \mapsto (\phi, 2^{p(n)})$ is such a reduction. Thus we get:

Proposition 5.3 ([22, 37]). *If P is a length-optimal proof system, then its canonical pair is a complete disjoint* **NP** *pair with respect to polynomial reductions (i.e., every disjoint* **NP** *pair is reducible to it).*[16]

Therefore the following conjecture is a strengthening of Conjecture CON^N.

[16] Razborov proved this fact for p-optimal proof systems (see Definition 6.1 below); Köbler, Messner and Torán improved it to length optimal proof systems.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

Conjecture 5.4 (DisjNP). *There exist no complete disjoint* **NP** *pair (with respect to polynomial reductions).*

Glaßer et al. [16] constructed an oracle relative to which there is no complete disjoint **NP**-pair. Other than that, we have little supporting evidence. A combinatorial characterization of the canonical pair has only been found for the resolution proof system. In [16] they also constructed an oracle relative to which there exists a complete disjoint **NP**-pair, but no length-optimal proof system exists, i.e., Conjecture DisjNP fails, but Conjecture CON^N holds true.

5.2. Disjoint coNP pairs. We now turn to disjoint **coNP** pairs. When comparing different disjoint **coNP**-pairs, one can use the same polynomial reduction as used for disjoint **NP**-pairs; hence one can also ask similar questions. In particular, are there disjoint **coNP** pairs inseparable by a set in **P**? Are there complete disjoint **coNP** pairs? We believe that the answer to the first question is yes, because we accept $\mathbf{NP} \cap \mathbf{coNP} \neq \mathbf{P}$ as a very likely fact. The answer to the second question is less clear, but we still lean to the negative answer.

Conjecture 5.5 (DisjCoNP). *There exist no complete disjoint* **coNP** *pair (with respect to polynomial reductions).*

An oracle relative to which the conjecture is true was recently constructed by Erfan Khaniki [private communication]. The next proposition states that Conjecture TFNP is a consequence of the above conjecture.

Proposition 5.6. *If there exists a complete* **TFNP** *problem, then there exists a complete disjoint* **coNP** *pair.*

The proposition follows from the two lemmas below and Proposition 4.10. First we need a definition.

Definition 5.7. Let a **TFNP** problem (p, R) be given. Assume that $R(x, y) \Rightarrow |y| = p(|x|)$. The *canonical disjoint* **coNP** *pair of* (p, R) is the pair (A_0, A_1) defined as follows. The elements of $A_0 \cup A_1$ are pairs (x, C) where x is an arbitrary binary string and C is a Boolean circuit with $p(|x|)$ bit-inputs and one bit-output. The sets A_0 and A_1 are defined by

$$(x, C) \in A_i \;\equiv\; \forall y(R(x, y) \to C(y) = i). \tag{7}$$

The condition that, for a given x, all elements y satisfying $R(x, y)$ have the same length is, clearly, not essential, because we can always pad the string y to the maximal length $p(|x|)$.

Lemma 5.8. *For every disjoint* **coNP** *pair* (B_0, B_1) *there exists a* **TFNP** *problem* (p, R) *such that* (B_0, B_1) *is polynomially reducible to the canonical disjoint* **coNP** *pair of* (p, R).

Proof. Let a disjoint **coNP** pair (B_0, B_1) be given. Suppose that B_is are defined by

$$x \in B_i \;\equiv\; \forall y(|y| \leq r_i(|x|) \to \beta_i(x, y))$$

for $i = 0, 1$, where β_i is computable in polynomial time and r_i is a polynomial. Let the binary relation R be defined by

$$R(x, z) \;\equiv\; \exists i \in \{0, 1\} \exists y(z = (i, y) \wedge |y| \leq r_i(|x|) \wedge \neg\beta_i(x, y)).$$

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

Since β_is are computable in polynomial time, so is also R and the length of every z satisfying $R(x,z)$ is polynomially bounded in the length of x. Furthermore, since B_0 and B_1 are disjoint, R is total. Again, by suitably padding z we may ensure that $R(x,z) \Rightarrow |z| = p(|x|)$ for some polynomial p. Let (A_0, A_1) be the canonical pair of (p, R). The pair (B_0, B_1) is reducible to (A_0, A_1) by the mapping

$$x \;\mapsto\; (x, C),$$

where C is a circuit such that $C(i,y) = 1-i$, because for this C, $(x, C) \in A_j$ iff $x \in B_j$. ⊣

Lemma 5.9. *Let (p, P) and (q, Q) be two* **TFNP** *problems such that $P(x,y) \Rightarrow |y| = p(|x|)$ and $Q(x,y) \Rightarrow |y| = q(|x|)$. Let (A_0, A_1), respectively (B_0, B_1), be their canonical* **coNP** *pairs and suppose that (p, P) is polynomially many-one reducible to (q, Q). Then (A_0, A_1) is reducible to (B_0, B_1).*

Proof. Let (p, P), (q, Q), and a polynomial many-one reduction (f, g) of (p, P) to (q, Q) be given. Let (A_0, A_1) and (B_0, B_1) be the canonical **coNP** pairs of (p, P) and (q, Q). We define a polynomial reduction of (A_0, A_1) to (B_0, B_1) as follows. For an input of the form (x, C) where C is a Boolean circuit, we put

$$h(x, C) = (f(x), D_x),$$

where D_x is a Boolean circuit with $q(|f(x)|)$ bit inputs such that for all y of length $q(|f(x)|)$,

$$D_x(y) = C(g(x,y)). \tag{8}$$

If an input z does not have the required form, we put $h(z) = 0$. We will check that this defines a polynomial reduction of (A_0, A_1) to (B_0, B_1). Let $(x, C) \in A_i$ and let y be any number such that $|y| = q(|f(x)|)$ and $Q(f(x), y)$. Since $P(x, g(x,y))$, we have $C(g(x,y)) = i$ by the definition of A_i. By (8), $D_x(y) = i$. This proves that $f(x) \in B_i$. ⊣

5.3. Multivalued functions. A class closely related to **TFNP** and the question whether there exists a complete problem in this class were studied by Beyersdorff, Köbler and Messner [5]. We need a couple of preliminary definitions.

A multivalued partial function f is called an **NP** *multivalued function* if it is computed by a nondeterministic polynomial time Turing machine M in the following sense. M stops in two possible states: ACCEPT and REJECT. For a given input value x, the values of f are those words on the output tape which appear when the state ACCEPT is reached. For a function $f \in$ **NPMV** we denote by $f\{x\}$ the set of all values for the input x. Thus f is total iff $f\{x\} \neq \emptyset$ for all x. The class of **NP** *multivalued functions* is denoted by **NPMV**. The class of *total* **NP** *multivalued functions* is denoted by $\mathbf{NPMV}_t$.

By their nature, $\mathbf{NPMV}_t$ functions are **TFNP** problems, but there is an essential difference in how one defines reduction. For $f, g \in$ **NPMV**, we

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

say that f is polynomially reducible to g if there exists a polynomial time computable function h such that for all x,

$$f\{x\} = g\{h(x)\}.$$

A relation to our Conjecture TFNP is given by the following proposition.

Proposition 5.10. *The existence of a complete function in* $\mathbf{NPMV}_t$ *implies the existence of a complete* **TFNP** *problem.*

Proof. Let g be a complete function in $\mathbf{NPMV}_t$. We can represent g using a polynomial time computable ternary relation as follows:

$$g\{x\} = \{y;\ \exists^p z.R(x,y,z)\}.$$

Recall that the superscript at the existential quantifier means that we tacitly assume that there exists a polynomial bound p such that $R(x,y,z)$ is satisfied only if the lengths of y and z are bounded by $p(|x|)$. Define

$$Q(x,u) := R(x,(u)_1,(u)_2).$$

We claim that Q defines a complete **TFNP** problem. Let $S(x,y)$ be a binary relation computable in polynomial time viewed as a **TFNP** problem (again, we tacitly assume an implicit polynomial bound on the length of y). Define a function $f \in \mathbf{NPMV}_t$ by

$$f\{x\} := \{y;\ S(x,y)\}.$$

Since f is reducible to the complete function g, there exists a polynomial time computable function h such that $f\{x\} = g\{h(x)\}$, which is equivalent to

$$\{y;\ S(x,y)\} = \{y;\ \exists z.R(h(x),y,z)\} = \{y;\ \exists z.Q(h(x),(y,z))\}.$$

Thus the pair of functions h,k, where $k(u) := (u)_1$, is a polynomial reduction of S to Q. ⊣

We do not know if the opposite implication holds true. Beyersdorff et al. [5] proved that if there exists a complete function in $\mathbf{NPMV}_t$, then there exists a complete disjoint **coNP** pair. This is now a consequence of Propositions 5.6 and 5.10.

§6. Classification of conjectures.

6.1. Uniform and nonuniform. A more natural way to compare proof systems than just comparing the lengths of proofs is polynomial simulation. This is a concept, introduced in [11], similar to polynomial reductions used in the theory of **NP**-completeness and those we used to compare **TFNP** problems.

Definition 6.1. We say that *a proof P system polynomially simulates a proof system Q* if there exists a polynomial time computable function f such that given a Q-proof d of ϕ, $f(d)$ is a P-proof of (the same) ϕ. We say that *a proof system P is p-optimal* if it polynomially simulates every proof system.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

Using this concept we can state a conjecture slightly weaker than Conjecture CON^N.

Conjecture 6.2 (CON). *There exists no p-optimal proof system.*

In [27] we proved that this conjecture is equivalent to the following uniform version of Conjecture CON^N.

Conjecture 6.3 (equivalent to CON). *For every $S \in \mathcal{T}$, there exists $T \in \mathcal{T}$ such that S-proofs of $Con_T(\bar{n})$ cannot be constructed in polynomial time in n.*

A uniform version of Conjecture RFN_1^N is obtained in the same way.

Conjecture 6.4 (RFN_1). *For every $S \in \mathcal{T}$, there exists $T \in \mathcal{T}$ such that S-proofs of $\Sigma_1^b RFN_T(\bar{n})$ cannot be constructed in polynomial time in n.*

Except for modifications of these conjectures, such as Conjecture CON^{N+}, we do not know of any other pair of uniform and nonuniform conjectures of the kind studied in this article. In particular, TFNP is apparently uniform, but we do not know if it has a nonuniform companion.

Note that $\mathbf{NP} \neq \mathbf{coNP}$ is implied by the nonuniform conjectures CON^N and RFN_1^N, while the uniform versions CON and RFN_1 are only known to imply $\mathbf{P} \neq \mathbf{NP}$. Thus we should also classify $\mathbf{NP} \neq \mathbf{coNP}$ as nonuniform and $\mathbf{P} \neq \mathbf{NP}$ as uniform. Then it may seem strange that according to this classification $\mathbf{NP} \neq \mathbf{coNP}$ should be a nonuniform conjecture, in spite of the fact that both $\mathbf{NP}$ and $\mathbf{coNP}$ are uniform complexity classes. But if we look at $\mathbf{NP} \neq \mathbf{coNP}$ from the point of view of proof complexity, then it is clearly a nonuniform version of $\mathbf{P} \neq \mathbf{NP}$. Just consider the following equivalent formulations of these conjectures:

- $\mathbf{P} \neq \mathbf{NP} \Leftrightarrow$ there exists a proof system P such that for every tautology τ a P-proof of τ *can be constructed in polynomial time*;
- $\mathbf{NP} \neq \mathbf{coNP} \Leftrightarrow$ there exists a proof system P such that every tautology τ *has a P-proof of polynomial length.*

However, although Conjecture DisjNP seems to be uniform, it does imply the nonuniform Conjecture CON^N (see Proposition 5.3). We do not have an explanation for this.

6.2. Logical complexity. We started with statements about finite consistency, statements that express facts about logic, and eventually arrived at statements about disjoint sets of certain complexity, statements from structural complexity theory that apparently have nothing to do with the main theme of incompleteness. But one should realize that expressing these conjectures using concepts from computational complexity theory is just a convenient way to state them. It seems that it should be possible to present all uniform conjectures as statements about unprovability of certain sentences in theories from the class $\mathcal{T}$. The following proposition shows how to state Conjecture CON in this way.

Proposition 6.5. *There exists a p-optimal proof system (for TAUT) if and only if there exists a theory $T \in \mathcal{T}$ such that for every proof system P there exists a definition of P by a Δ_1^b formula such that T proves the soundness of P represented by this formula.*

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

For the proof, see [34], pages 578–9. The next proposition shows how to express Conjecture DisjNP as a statement about unprovability of certain sentences.

PROPOSITION 6.6. *There exists a complete disjoint* **NP** *pair if and only if there exists a theory $T \in \mathcal{T}$ such that for every disjoint* **NP** *pair (B_0, B_1) there are Σ_1^b definitions of B_0 and B_1 for which T proves that they define disjoint sets.*

PROOF. Suppose that there exists a complete disjoint **NP** pair (A_0, A_1). Let $\exists^p y.\alpha_i(x, y)$ be Σ_1^b definitions of A_i, $i = 0, 1$. Define a theory T to be

$$S_2^1 + \forall x(\neg\exists^p y.\alpha_0(x, y) \vee \neg\exists^p y.\alpha_1(x, y)).$$

Let (B_0, B_1) be an arbitrary disjoint **NP** pair. Let $\exists^p y.\beta_i(x, y)$ be some Σ_1^b definitions of B_i, $i = 0, 1$. Since (A_0, A_1) is complete, there exists a polynomial time reduction f of (B_0, B_1) to (A_0, A_1). Consider the following definitions of B_i, $i = 0, 1$, by Σ_1^b formulas:

$$\exists^p y.\beta_i(x, y) \wedge \exists^p z.\alpha_i(f(x), z).$$

It is clear that they define the sets B_i correctly and that T proves that sets defined by these formulas are disjoint.

The proof of the converse implication is a standard argument that we have already presented in the proof of Lemma 4.9, so we will be very brief.

Let T be a theory with the property stated in the proposition. For $i = 0, 1$, let A_i be the set of tuples $(x, \beta_0, \beta_1, d, a)$ such that

- β_0 and β_1 are Σ_1^b formulas, d is a T-proof of the disjointness of the sets defined by β_0 and β_1, a is a nondeterministic time bound for β_0 and β_1, and $\exists^p y.\beta_i(x, y)$ holds true.

We leave to the reader to verify that these conditions define a disjoint **NP** pair and that every disjoint **NP** pair is polynomially reducible to it. $\dashv$

The nonexistence of a complete disjoint **coNP** pair, Conjecture DisjCoNP, can be expressed as a statement about provability in the same way. Conjecture TFNP was, in fact, introduced as a sentence about unprovability in theories in $\mathcal{T}$.

Thus a natural way to classify such conjectures is according to the logical complexity of sentences that are claimed to be unprovable. The two most important classes are $\forall\Pi_1^b$ and $\forall\Sigma_1^b$ (i.e., the sentences of the form: universally quantified Π_1^b and Σ_1^b formulas). Our uniform conjectures are classified as follows:

$\forall\Pi_1^b$ – CON, DisjNP;
$\forall\Sigma_1^b$ – RFN$_1$, TFNP, DisjCoNP.

6.3. Some related statements. Several concepts related to our conjectures have been studied. We will present some of these sentences here. We will call them conjectures, since we believe that they are true, but we do not have essentially any supporting argument for their truth.

We have observed that Conjecture CON^N can be strengthened to Conjecture DisjNP. Its uniform version, Conjecture CON, can, furthermore, be strengthened in a different way. Recall that **UP**, *unambiguous* **P**, is the class of languages that are accepted by polynomial time *nondeterministic* Turing machines that satisfy the property that for every accepted input, there is a *unique* accepting computation. Köbler, Messner and Torán [22] proved that if there exists a p-optimal proof system, then **UP** has a complete set with respect to many-one reductions. Hence the following is a strengthening of Conjecture CON.

Conjecture 6.7 (UP). *There is no complete set, with respect to many-one reductions, in* **UP**.

So far we only talked about proof systems for *TAUT*. In the same way one can define proof systems and polynomial simulations for any set. In particular, a *proof system for SAT* is a polynomial time computable function from Σ^* onto *SAT*. There is one essential difference between proof systems for *TAUT* and *SAT*—the latter does have polynomially bounded proof systems. In fact, the definition of *SAT* itself gives one such proof system; in this system any pair (ϕ, a), where a is a satisfying assignment of a formula ϕ is a proof (of the satisfiability of) ϕ. This is called the *standard* proof system for *SAT*.

Here is an example of a nonstandard proof system P for *SAT*. In P a proof of ϕ is either a pair (ϕ, a), where a is a satisfying assignment of ϕ, or it is ϕ itself in the case when ϕ is a proposition γ_n expressing, in a natural way, the fact that n is a composite number and n is a composite number. Note that in the standard proof system the proof of γ_n encodes a nontrivial factor of n. Hence, if the standard proof system p-simulated P, then factoring would be in polynomial time.

Beyersdorff et al. [5] proved that the existence of a p-optimal proof system for *SAT* implies the existence of a complete function in $\mathbf{NPMV}_t$. Hence, by our Proposition 5.10, it also implies the existence of a complete problem in **TFNP**. To put the conjecture about complete sets in *SAT* into context, we need the following proposition.

Proposition 6.8. *Let $S \in \mathcal{T}$ be a theory such that for every theory $T \in \mathcal{T}$, S-proofs of $\Sigma^b_1 RFN_T(\bar{n})$ can be constructed in polynomial time in n. Then there exists a p-optimal proof system for SAT.*

Proof. Let $sat(x, y)$ be a Δ^b_1 formula expressing the fact that y is a satisfying assignment of a propositional formula x. Suppose that S satisfies the assumption of the proposition. We define a proof system P for *SAT* by:

$$y \text{ is a } P\text{-proof of } x \Leftrightarrow y \text{ is an } S\text{-proof of } \exists z.sat(\bar{x}, z).$$

Given a proof system f for *SAT*, we take $T \in \mathcal{T}$ such that it proves the soundness of f, i.e.,

$$T \vdash \forall y \exists z.sat(f(y), z). \tag{9}$$

By Corollary 3.15, S-proofs of $\exists z.sat(f(\bar{d}), z)$ can be constructed in polynomial time for every d. Thus, given an f-proof d of $f(d)$, we can construct in polynomial time a proof in P. Hence P is a p-optimal proof system for *SAT*. ⊣

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

Conjectures DisjNP, DisjCoNP, and UP are related to our main conjectures CON and TFNP. Here is an example of a plausible conjecture that is apparently incomparable with CON and TFNP.

Conjecture 6.9 (NP∩coNP). *There is no complete set in* **NP** $\cap$ **coNP**.

Beyersdorff et al. [5] proved that if both *TAUT* and *SAT* have p-optimal proof systems, then there exists a complete set in **NP** $\cap$ **coNP**. Hence Conjecture NP∩coNP is above Conjecture RFN_1.

The implications between the most important uniform conjectures considered in this article are depicted in the figure below. Recall that CON is equivalent to the nonexistence of a p-optimal proof system for *TAUT*.

$$\begin{tikzcd}
 & & & & & \mathsf{DisjCoNP} \arrow[dl] \\
\mathsf{DisjNP} \arrow[dr] & \mathsf{UP} \arrow[d] & & & \mathsf{TFNP} \arrow[dl] & \\
 & \mathsf{CON} \arrow[dr] & \mathsf{NP} \cap \mathsf{coNP} \arrow[d] & \mathsf{SAT} \arrow[dl] & & \\
 & & \mathsf{CON} \vee \mathsf{SAT} \arrow[d] & & & \\
 & & \mathsf{RFN}_1 \arrow[d] & & & \\
 & & \mathbf{P} \neq \mathbf{NP} & & &
\end{tikzcd}$$

6.4. Towards general conjectures. We will focus on uniform conjectures, because the situation there seems to be clearer. We have seen that our uniform conjectures are statements about unprovability of particular sentences. The structure of these sentences is determined by

1. some class $\mathcal{C}$ of sentences,
2. associated with computational problems $\mathcal{P}$, and
3. some complexity hierarchy $\mathcal{H}$ of the associated problems.

The conjectures say that the more difficult the associated computational problem is, the more difficult is to prove the sentence.

Consider, for example, Conjecture DisjNP. In this conjecture we have sentences expressing that two sets defined by Σ_1^b sentences are disjoint. These sentences are of the form:

$$\forall x(\neg\phi(x) \vee \neg\psi(x)), \tag{10}$$

where ϕ and ψ are Σ_1^b sentences defining the two sets. These sentences are equivalent to universally quantified Π_1^b sentence, but they have a special form. For sentences of this form, a natural task is, for a given x, to decide which of the two $\neg\phi(x)$ or $\neg\psi(x)$ is true. The complexity hierarchy of the computational problems is defined using polynomial time reductions.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

Consider Conjecture CON. In the equivalent form of this conjecture given by Proposition 6.5, the sentences expressing that a propositional proof system P is sound are also universally quantified Π_1^b sentences. They have the form

$$\forall x, y, z(\mathit{proof}_P(x, y) \to \mathit{sat}(x, z)), \tag{11}$$

where $\mathit{proof}_P(x, y)$ is a Δ_1^b formula expressing that y is a P-proof of x. The structure of sentences (10) and (11) is similar (essentially, they are universally quantified disjunctions), but the length of y in the second formula is not polynomially bounded in the length of x. Furthermore, we use a different kind or reductions to define the hierarchy: in the first case, a reduction can map x to another element, but we do not care about the witnesses of the Σ_1^b formulas; in the second case, x does not change, but we map a witness y to another witness.

Ideally, we would like to state a general conjecture from which our current conjectures would follow as special cases. However, to be able to do that, we first need to fully understand what are the classes $\mathcal{C}$ whose sentences can be associated with computational tasks, what are the computational problems $\mathcal{P}$, and what are the complexity hierarchies $\mathcal{H}$. So far we only have examples.

§7. The role of reductions.

In Propositions 6.5 and 6.6 we saw that conjectures whose statements used reductions can be equivalently stated without referring to any concept of polynomial reduction. In this section we will explain how polynomial reductions naturally appear when we compare the logical strength of sentences.

When we are comparing sentences from some class $\mathcal{C}$, we do it with respect to some base theory T. Thus for some $\Phi, \Psi \in \mathcal{C}$, we are asking whether $T \vdash \Phi \to \Psi$. One can show that at least for a specific type of sentences and a specific theory T, the provability implies the existence of a reduction.

Let the base theory be S_2^1 and the sentences have the form $\forall x \exists^p y.\phi(x, y)$, where ϕ is Σ_1^b. We will show that provability of one sentence from the other implies the existence of a polynomial reduction of one **TFNP** problem to the other. The following is a well-known fact (see [18]), but we will still give a proof, because we want to argue that it can be generalized to stronger theories.

Proposition 7.1. *Suppose that* $\mathbb{N} \models \forall x \exists^p y.\phi(x, y) \wedge \forall u \exists^p v.\psi(u, v)$ *and*

$$S_2^1 \vdash \forall x \exists^p y.\phi(x, y) \to \forall u \exists^p v.\psi(u, v), \tag{12}$$

where ϕ *and* ψ *define polynomial time computable relations. Then the* **TFNP** *problem defined by* ψ *is polynomially reducible to the* **TFNP** *problem defined by* ψ.

Proof. This proposition is an immediate consequence of the following result (see [32]).

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

LEMMA 7.2. *If $S^1_2 \vdash \forall x \exists^p y \forall^p z.\alpha(x,y,z)$, where α is Π^b_0, then for a given x, one can compute y such that $\forall^p z.\alpha(x,y,z)$ using a polynomial time oracle Turing machine with any oracle that, for a given x and y such that $\exists^p z.\neg\alpha(x,y,z)$ holds true, produces some z such that $\neg\alpha(x,y,z)$ holds true.*

Write the implication in (12) in the following prenex form

$$\forall u \exists x \exists v \forall y (\phi(x,y) \to \psi(u,v)).$$

The quantifiers $\exists v, \forall y$ are bounded already in the original formula and $\exists x$ can be bounded by Parikh's theorem. By the lemma, there is a polynomial time Turing machine M that computes x and v from a given u using any oracle that whenever $\exists y(\phi(x,y) \wedge \neg\psi(u,v))$ holds true produces a witness for y. We want to use an oracle that only produces witnesses for $\exists y.\phi(x,y)$. Clearly, such an oracle suffices. If M asks a query (x,v) such that $\psi(u,v)$ is true, then we can stop, because we already have a witness for $\exists^p v.\psi(u,v)$. If no such query occurs during the computation of M, then the oracle must always produce a witness for $\exists y.\phi(x,y)$, hence we get x and v such that $\forall y(\phi(x,y) \to \psi(u,v))$ is true, which is equivalent to $\exists y.\phi(x,y) \to \psi(u,v)$. But the antecedent is always true, so we have $\psi(u,v)$. $\dashv$

If the base theory T is stronger than S^1_2, we believe that we nevertheless get some class of reductions that is probably stronger than polynomial time computable reductions, but still somewhat restricted so that the classes of **TFNP** equivalent with respect to these reductions do not completely collapse. These reductions should be defined using the provably total search problems of T. The idea is that the provably total polynomial search problems of S^1_2 are the problems solvable in polynomial time and this gives us reductions that are polynomial time computations with oracle queries to which we substitute solutions of the problem to which we are reducing the given problem. Similarly, if $\mathcal{S}$ is the class of provably total polynomial search problems of T, then provability in T should give us reductions that are problems from $\mathcal{S}$ with oracle queries. A special case of this appeared in [6] (not quite explicitly) where the theory was T^1_2 and the class of search problems was **PLS**. Although it may be interesting to study such reductions in general, we believe that they would give the same conjectures if used instead of polynomial reductions.

§8. Conclusions and open problems.

In this article we put forward the thesis that there exists a connection between the complexity of problems associated with first order sentences and their logical strength manifested as impossibility of proving them in weak theories. If we interpret this thesis in a broad sense, then the thesis is true; e.g., we cannot prove in a weak theory that some computation stops if the problem requires extremely long time to be solved. However, our argument here is that there may be such a connection already on the very low level, namely in the domain of problems solvable in nondeterministic polynomial time. Since the current state of research into such low complexity classes does not have the means to prove separations, we can only state and compare hypotheses about such a connection.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

There are two basic conjectures which have equivalent formulations and come in some flavors. The first one is about finite consistency statements and was proposed already a long time ago [27]. The second one is more recent and concerns provably total polynomial search problems. We showed how they are related to some weaker statements and some stronger ones. Some of these statements had already been studied before. There are still many problems that need to be solved if we want to fully understand this topic; some are of a fundamental nature, some are more specific. Some problems have already been mentioned in previous sections. Below we briefly mention some more.

1. The main problem, mentioned in Section 6.4, is to find a general conjecture about incompleteness and computational complexity. The conjectures we studied in this article should be special cases of it.
2. More specifically, propose a natural and plausible conjecture that implies the two main Conjectures CON and TFNP, or prove that one of these conjectures implies the other, or show that their relativizations are independent.
3. Construct oracles that show that relativized conjectures are different or show that they are equivalent for pairs of conjectures presented in this article. Apparently the only separation that is known is a separation of Conjectures CON and DisjNP, see [16].
4. In order to get more evidence for Conjecture TFNP, characterize provably total polynomial search problems in stronger systems of Bounded Arithmetic. The strongest theory for which a combinatorial characterization has been found is V_2^1, see [3, 23].
5. Characterize more canonical pairs of propositional proof systems in order to get more evidence for Conjecture DisjNP. A combinatorial characterization of the canonical pair has only been found for Resolution, see [4]. Characterize canonical pairs of some total polynomial search problems (as defined in this article) in order to get some evidence for Conjecture $\mathsf{DisjCoNP}$. Nothing is known in this direction.
6. We would also be interested in seeing connections between the nonexistence of complete problems in some probabilistic classes and our main conjectures. Köbler et al [22] proved that if $TAUT_2$ (or SAT_2) have a p-optimal proof system, then **BPP**, **RP** and **ZPP** have many-one complete problems. ($TAUT_2$ and SAT_2 are the sets of Π_2 and Σ_2 quantified Boolean tautologies.) But most researchers believe that these probabilistic classes do have complete problems, because they are in fact equal to **P**.
7. Another important subject are *proof-complexity generators* of Krajíček and Razborov (see [26, 38] and the references therein). One conjecture about proof-complexity generators states that they are hard for every proof system. It would be very interesting to find connections to our conjectures.

§9. **Acknowledgments.** The author is supported by the ERC Advanced Grant 339691 (FEALORA) and the institute grant RVO: 67985840. Part of this article was written while the author was visiting Simons Insitute in Berkeley, California. I would like to thank Pavel Hrubeš, Emil Jeřábek, Jan Krajíček and Neil Thapen for their useful comments on the draft of this article. I am also indebted to anonymous referees for pointing out many small errors and suggesting improvements of presentation.

REFERENCES

[1] A. Beckmann and S. R. Buss, *Polynomial local search in the polynomial hierarchy and witnessing in fragments of bounded arithmetic*. ***Journal of Mathematical Logic***, vol. 9 (2009), no. 1, pp. 103–138.

[2] ———, *Characterizing definable search problems in bounded arithmetic via proof notations*, ***Ways of Proof Theory*** (R. Schindler, editor), Ontos Series in Mathematical Logic, Walter de Gruyter GmbH, Berlin, 2010, pp. 65–134.

[3] ———, *Improved witnessing and local improvement principles for second-order bounded arithmetic*. ***ACM Transactions on Computational Logic***, vol. 15 (2014), no. 1, Article 2.

[4] A. Beckmann, P. Pudlák, and N. Thapen, *Parity games and propositional proofs*. ***ACM Transactions on Computational Logic***, vol 15 (2014), no. 2, Article 17.

[5] O. Beyersdorff, J. Köbler, and J. Messner, *Nondeterministic functions and the existence of optimal proof systems*. ***Theoretical Computer Science***, vol. 410 (2009), pp. 3839–3855.

[6] S. R. Buss, L. Kołodziejczyk, and N. Thapen, *Fragments of approximate counting*. ***Journal of Symbolic Logic***, vol. 79 (2014), no. 2, pp. 496–525.

[7] S. R. Buss, ***Bounded Arithmetic***, Bibliopolis, Naples, 1986.

[8] ———, *An introduction to proof theory*, ***Handbook of Proof Theory*** (S. R. Buss, editor), Elsevier, Amsterdam, 1998, pp. 1–78.

[9] S. R. Buss and G. Mints, *The complexity of the disjunction and existence properties in intuitionistic logic*. ***Annals of Pure and Applied Logic***, vol. 99 (1999), pp. 93–104.

[10] S. Buss and P. Pudlák, *On the computational content of intuitionistic propositional proofs*. ***Annals of Pure and Applied Logic***, vol. 109 (2001), pp. 49–64.

[11] S. A. Cook, *Feasibly constructive proofs and the propositional calculus*, ***Proceedings of the Seventh Annual ACM Symposium on Theory of Computing***, ACM, New York, 1975, pp. 83–97.

[12] S. A. Cook and R. A. Reckhow, *The relative efficiency of propositional proof systems*. ***Journal of Symbolic Logic***, vol. 44 (1979), no. 1, pp. 36–50.

[13] A. Ehrenfeucht and J. Mycielski, *Abbreviating proofs by adding new axioms*, this Bulletin, vol. 77 (1971), pp. 366–367.

[14] H. Friedman, *On the consistency, completeness and correctness*. Unpublished typescript, 1979.

[15] C. Glaßer, A. L. Selman, and L. Zhang, *Canonical disjoint NP-pairs of propositional proof systems*. ***Theoretical Computer Science***, vol. 370 (2007), no. 1–3, pp. 60–73.

[16] C. Glaßer, A. L. Selman, S. Sengupta, and L. Zhang, *Disjoint NP-pairs*. ***SIAM Journal on Computing***, vol. 33 (2004), no. 6, pp. 1369–1416.

[17] P. Hájek and P. Pudlák, ***Metamathematics of First Order Arithmetic***, ASL Perspectives in Logic, Springer-Verlag, Berlin, 1993.

[18] J. Hanika, *Herbrandizing search problems in bounded arithmetic*. ***Mathematical Logic Quarterly***, vol. 50 (2004), no. 6, pp. 577–586.

[19] R. Impagliazzo and R. Paturi, *The complexity of k-SAT*, ***Proceedings of the 14th Annual IEEE Conference on Computational Complexity***, IEEE Computer Society, Los Alamitos, CA, 1999, pp. 237–240.

[20] D. Johnson, C. Papadimitriou, and M. Yannakakis, *How easy is local search?* ***Journal of Computer and System Sciences***, vol. 37 (1988), pp. 79–100.

[21] J. Krajíček and R. Impagliazzo, *A note on conservativity relations among bounded arithmetic theories*. ***Mathematical Logic Quarterly***, vol. 48 (2002), no. 3, pp. 375–377.

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press

[22] J. Köbler, J. Messner, and J. Torán, *Optimal proof systems imply complete sets for promise classes*. ***Information and Computation***, vol. 184 (2003), pp. 71–92.

[23] L. Kołodziejczyk, P. Nguyen, and N. Thapen, *The provably total NP search problems of weak second order bounded arithmetic*. ***Annals of Pure and Applied Logic***, vol. 162 (2011), no. 6, pp. 419–446.

[24] J. Krajíček, ***Bounded Arithmetic, Propositional Logic, and Complexity Theory***, Encyclopedia of Mathematics and its Applications, vol. 60, Cambridge University Press, Cambridge, 1995.

[25] ———, *Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic*. ***Journal of Symbolic Logic***, vol. 62 (1997), no. 2, pp. 457–486.

[26] ———, *On the proof complexity of the Nisan-Wigderson generator based on a hard NP∩coNP function*. ***Journal of Mathematical Logic***, vol. 11 (2011), no. 1, pp. 11–27.

[27] J. Krajíček and P. Pudlák, *Propositional proof systems, the consistency of first order theories and the complexity of computations*. ***Journal of Symbolic Logic***, vol. 54 (1989), no. 3, pp. 1063–1079.

[28] J. Krajíček, P. Pudlák, and G. Takeuti, *Bounded arithmetic and polynomial hierarchy*. ***Annals of Pure and Applied Logic***, vol. 52 (1991), pp. 143–154.

[29] P. Pudlák, *On the length of proofs of finitistic consistency statements in first order theories*, ***Logic Colloquium 84*** (J. B. Paris, A. J. Wilkie, and G. M. Wilmers, editors), North Holland, Amsterdam, 1986, pp. 165–196.

[30] ———, *Improved bounds to the length of proofs of finitistic consistency statements*, ***Contemporary Mathematics, vol. 65***, American Mathematical Society, 1987, pp. 309–331.

[31] ———, *A note on bounded arithmetic*. ***Fundamenta Mathematicae***, vol. 136 (1990), no. 2, pp. 86–89.

[32] ———, *Some relations between subsystems of arithmetic and the complexity theory*, ***Logic from Computer Science, Proceedings of a Workshop held November 13-17, 1989*** (Y. N. Moschovakis, editor), Springer-Verlag, New York, 1992, pp. 499–519.

[33] ———, *Gödel and computations*. ***ACM SIGACT News***, vol. 37/4 (2006), pp. 13–21.

[34] ———, ***Logical Foundations of Mathematics and Computational Complexity, a Gentle Introduction***, Springer Monographs in Mathematics, Springer-Verlag, Heidelberg, 2013.

[35] ———, *On the complexity of finding falsifying assignments for Herbrand disjunctions*. ***Archive for Mathematical Logic***, vol. 54 (2015), no. 7, pp. 769–783.

[36] P. Pudlák and N. Thapen, *Alternating minima and maxima, nash equilibria and bounded arithmetic*. ***Annals of Pure and Applied Logic***, vol. 163 (2012), no. 5, pp. 604–614.

[37] A. A. Razborov, *On provably disjoint NP-pairs*. ECCC Technical report TR94-006, 1994.

[38] ———, *Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution*. ***Annals of Mathematics***, vol. 181 (2015), no. 2, pp. 415–472.

[39] A. Skelley and N. Thapen, *The provably total search problems of bounded arithmetic*. ***Proceedings of the London Mathematical Society***, vol. 103 (2011), no. 1, pp. 106–138.

[40] C. Smoryński, *The incompleteness theorems*, ***Handbook of Mathematical Logic*** (J. Barwise, editor), North-Holland, Amsterdam, 1977, pp. 821–865.

INSTITUTE OF MATHEMATICS
CZECH ACEDEMY OF SCIENCES
PRAGUE, CZECH REPUBLIC
E-mail: pudlak@math.cas.cz

https://doi.org/10.1017/bsl.2017.32 Published online by Cambridge University Press