

ON THE CONSISTENCY OF STRONGER LOWER BOUNDS FOR NEXP

NEIL THAPEN

Institute of Mathematics, Czech Academy of Sciences
e-mail address: thapen@math.cas.cz

ABSTRACT. It was recently shown by Atserias, Buss and Müller that the standard complexity-theoretic conjecture $\text{NEXP} \not\subseteq \text{P/poly}$ is consistent with the relatively strong bounded arithmetic theory V_2^0 , which can prove a substantial part of complexity theory. We observe that their approach can be extended to show that the stronger conjectures $\text{NEXP} \not\subseteq \text{EXP/poly}$ and $\text{NEXP} \not\subseteq \text{coNEXP}$ are consistent with a stronger theory, which includes every true universal number-sort sentence.

1. INTRODUCTION

The bounded arithmetic hierarchy $S_2^1 \subseteq T_2^1 \subseteq S_2^2 \subseteq \dots$, with union T_2 , is a well-studied family of first-order theories that plausibly captures the kind of reasoning one can do if one is limited to using concepts in the polynomial hierarchy [Bus85]. It has long been of interest how much of mathematics can be carried out in this setting [PWW88]. Many results in complexity theory can be formalized in it, at low levels in the hierarchy; some more recent examples are the PCP theorem, Toda’s theorem, the Schwartz-Zippel lemma and many circuit lower bounds [Pic15, BKZ15, AT25, MP20].

Now consider a theory T , such as T_2 or some fragment of it, which is known to formalize a substantial part of complexity theory, and take a conjecture $\mathcal{C}$ that you would like to prove. If you can show that $\mathcal{C}$ is unprovable in T , then one interpretation of this is that the methods which work for large parts of complexity theory are not enough to prove $\mathcal{C}$, and you need to try something new; see [PS21] for some work in this direction. On the other hand, if the negation $\neg\mathcal{C}$ is unprovable in T , this shows that $\mathcal{C}$ is at least consistent with T and thus with a large part of complexity theory. Concretely, there is a well-behaved structure (a model of T) which satisfies many of the complexity-theoretic properties of the real world and in which $\mathcal{C}$ is true. Interpreted optimistically, this is a partial result in the direction of showing $\mathcal{C}$ is true in the real world [Kra95].

The T_2^i hierarchy is not able to reason naturally about complexity phenomena at the level of PSPACE, EXP or above, since by design it is limited to working with polynomial-length strings. Already [Bus85] introduced a stronger hierarchy $V_2^0 \subseteq V_2^1 \subseteq \dots$ of two-sorted or “second-order” theories which also work with larger objects which we will call here sets, but which could just as well be called exponential-length strings. The base theory V_2^0 in the hierarchy is a conservative extension of T_2 , and the next level V_2^1 can prove many basic

Supported by the Czech Academy of Sciences (RVO 67985840) and GAČR grant 23-04825S.



properties of EXP, such as that every exponential-time machine (even with an oracle) has a computation.

It was recently shown in [ABM24] that the standard complexity-theoretic conjecture $\text{NEXP} \not\subseteq \text{P/poly}$ is consistent with V_2^0 . That is,

Theorem 1.1 [ABM24]. $V_2^0 \not\vdash \text{NEXP} \subseteq \text{P/poly}$.

The authors suggest this is the best currently available evidence for the truth of the conjecture. The purpose of this note is to show that their proof of Theorem 1.1, based on the well-known unprovability of the pigeonhole principle in (relativized) T_2 or V_2^0 , can be extended to show that the stronger conjecture $\text{NEXP} \not\subseteq \text{EXP/poly}$ is consistent with a stronger theory:

Theorem 1.2. $V_2^0 + \forall \tilde{\Sigma}_1^{1,b}(\mathbb{N}) \not\vdash \text{NEXP} \subseteq \text{EXP/poly}$.

See Definition 1.4 below for the precise definition of $\forall \tilde{\Sigma}_1^{1,b}(\mathbb{N})$, but it is roughly all sentences $\forall \vec{x} \varphi(\vec{x})$ which are true in $\mathbb{N}$, where $\vec{x}$ are first-order and $\varphi(\vec{x})$ is a NEXP property. It thus contains every true Π_1 number-sort sentence, including a fortiori all Π_1 number-sort consequences of V_2^1 , and also contains the axiom that every exponential-time machine has a computation on every input.

Furthermore by a slightly different argument we show (with a caveat that the result depends on a choice to formalize coNEXP without parameters – see the discussion at the end of the paper):

Theorem 1.3. $V_2^0 + \forall \tilde{\Sigma}_1^{1,b}(\mathbb{N}) \not\vdash \text{NEXP} \subseteq \text{coNEXP}$.

These two theorems could be taken as evidence towards the conjectures $\text{NEXP} \not\subseteq \text{EXP/poly}$ and $\text{NEXP} \not\subseteq \text{coNEXP}$ being true; on the other hand, the theorems are themselves not so difficult to prove from the pigeonhole principle lower bound, as was already observed in [ABM24], and it is tempting to conclude that this work rather shows that V_2^0 , even when significantly strengthened, is not really equipped to reason nontrivially about set-sort quantification.

This work also suggests a slightly different perspective on one of the observations of [ABM24], that $\text{NEXP} \not\subseteq \text{P/poly}$ is consistent with the considerable amount of complexity theory that can be formalized in T_2 . By our construction, the argument of [ABM24] can be made to show that the conjecture is consistent with *all* true statements of complexity theory that can be written as Π_1 number-sort statements, since these are included for free in $\forall \tilde{\Sigma}_1^{1,b}(\mathbb{N})$.

Does this mean that the prior work on formalizing complexity theory inside T_2 is irrelevant for the implications of these consistency results? I would say no, since to talk about NEXP we are working with set-sort objects, as this is how we choose to formalize exponential-length computations. The work on formalizing complexity still holds in the presence of such objects, in that (as far as I know) it all relativizes and is still valid for reasoning about machines which have access to these objects as oracles, while our theory $\forall \tilde{\Sigma}_1^{1,b}(\mathbb{N})$ has nothing to say about such a situation.

We have not been able to combine Theorems 1.2 and 1.3 into the natural next step, the consistency of $\text{NEXP} \not\subseteq \text{coNEXP/poly}$ with our theory. This would be particularly interesting as $\text{NEXP} \subseteq \text{coNEXP/poly}$ is in fact true, as can be shown by a census argument reported in [BFS09] (although it should be kept in mind that all our constructions rely on a false statement about NEXP, namely the existence of a function violating the pigeonhole principle,

being consistent with our theory). We comment on one of the difficulties here at the end of the paper.

Our proofs are self-contained and do not rely on [ABM24], but we assume some knowledge of bounded arithmetic. We give an overview below to fix notation, generally following [ABM24]. For details see e.g. [Bus85, Kra95, Bus98].

Bounded arithmetic in the style of [Bus85] has variables $x, y, \dots$ ranging over numbers, which we identify when convenient with binary strings. It has the first-order language $x \leq y$, 0 , 1 , $x + y$, $x \cdot y$, $\lfloor x/2 \rfloor$, $x \# y$, $|x|$ and built-in equality $x = y$. Here $|x|$ is the length of x in binary, and the smash function $\#$ is a weak form of exponentiation defined so that $|x \# y| = |x| \cdot |y|$, so the main effect of the totality of smash is that lengths are closed under polynomials. We will use the abbreviation $x \in \text{Log}$ for $\exists y(x = |y|)$ and will write expressions like 2^x for such numbers, so as in [ABM24] a formula of the form e.g. $\forall x \in \text{Log}(\dots 2^{x^2} \dots)$ stands for $\forall x \forall y(x = |y| \rightarrow \dots y \# y \dots)$.

A bounded formula is one in which every quantifier is bounded, of the form $\forall x < t$ or $\exists y < t$ for some term t . We write Σ_∞^b for the set of all bounded formulas. By Π_1 above we mean the set of universal closures of Σ_∞^b formulas.

The theory T_2 [Bus85] consists of a set of axioms BASIC, which are bounded formulas fixing the basic properties of the language, and the induction scheme Σ_∞^b -IND, that is, the axiom

$$\varphi(0) \wedge \forall y < z [\varphi(y) \rightarrow \varphi(y + 1)] \rightarrow \varphi(z)$$

for every Σ_∞^b formula φ , which may include other parameters. The theory T_2 is essentially the same as $I\Delta_0 + \Omega_1$ [PWW88].

In *two-sorted* bounded arithmetic we add to the language new variables $X, Y, \dots$ of the *set sort*, representing bounded sets, and a relation $x \in X$ between the number and set sorts. For $\vec{X}$ a tuple of set variables, we write $\Sigma_\infty^b(\vec{X})$ for the set of bounded formulas which may now include these variables, and $T_2(\vec{X})$ for BASIC + $\Sigma_\infty^b(\vec{X})$ -IND (this is technically slightly different from the way similar notation is used in [ABM24]). Here we do not allow any quantification over set-sort variables, so these really play the role of undefined “oracle” predicate symbols. A theory of this form is sometimes called *relativized*.

Note that, for simplicity, we do not formally enforce that these sets are bounded. For example we do not introduce a number-valued symbol $|X|$ for the maximum element of X , or any axiom like $\forall x(x \in X \rightarrow x \leq |X|)$. The bounds on the number-sort in $\Sigma_\infty^b(\vec{X})$ formulas guarantee that they only access a bounded part of the sets $\vec{X}$, and it does not matter what these sets contain beyond this bound.

We write set-sort quantification as e.g. $\exists X$ rather than $\exists_2 X$. We will rely on capitalization to distinguish the sorts of variables. A $\Sigma_0^{1,b}$ formula is a formula in the two-sorted language which can freely use set-sort variables, which contains no set-sort quantifiers, and in which every number-sort quantifier is bounded. A $\Sigma_1^{1,b}$ formula can further contain set-sort existential quantifiers (but not inside negations). The theory V_2^0 extends T_2 by the bounded comprehension and induction schemes for $\Sigma_0^{1,b}$ formulas. The theory V_2^1 further adds induction for $\Sigma_1^{1,b}$ formulas.

We introduce the ad-hoc notation $\tilde{\Sigma}_1^{1,b}$ for formulas of the form $\exists \vec{X} \varphi(\vec{X}, \vec{z})$ where φ is $\Sigma_0^{1,b}$ and contains no set-sort variables other than $\vec{X}$, that is, where φ is $\Sigma_\infty^b(\vec{X})$. These express precisely the NEXP predicates.

Definition 1.4. The theory $\forall\tilde{\Sigma}_1^{1,b}(\mathbb{N})$ consists of every *true* sentence of the form $\forall\vec{z}\exists\vec{X}\varphi(\vec{X}, \vec{z})$, where φ is $\Sigma_\infty^b(\vec{X})$.

Here “true” means true in the standard model, that is, in the two-sorted structure where the number-sort is $\mathbb{N}$ and the set-sort is all bounded subsets of $\mathbb{N}$. Note that these sentences do not contain any universal set-sort quantifiers.

2. NEXP AND EXP/poly

Let $\text{comp}(e, t, W)$ be a $\Sigma_\infty^b(W)$ formula expressing that W is a computation of the universal Turing machine on input e running for time t in space t . We may think of the first t^2 entries of W as encoding of a $t \times t$ grid, where the i th column describes the contents of the tape at time i ; we also use part of W to encode the sequence of states and head positions.

We take Exp to be the axiom $\forall e \forall t \exists W \text{comp}(e, t, W)$ expressing that every exponential-time machine has a computation¹. This captures exponential-time because we think of a number x as taking an exponential *value* compared to its binary *length* $|x|$. For example, to express instead that every polynomial-time machine has a computation we would replace $\text{comp}(e, t, W)$ with $\text{comp}(e, |t|, W)$; furthermore in this case instead of W we could use a number-sort variable w bounded by roughly $t \# t$ to encode the computation.

The formula Exp is $\forall\tilde{\Sigma}_1^{1,b}$, without any set-sort parameters or universal set-sort quantifiers. It is true in the standard model, and so is included in $\forall\tilde{\Sigma}_1^{1,b}(\mathbb{N})$. If we allowed set-sort parameters as oracle inputs to the machine, Exp would become a much stronger axiom. It would imply, for example, that every exponential-size circuit has a computation, and in fact would be enough to prove the pigeonhole principle, destroying our main construction below.

We start by proving a slightly simpler version of Theorem 1.2:

Proposition 2.1. $V_2^0 + \text{Exp} \not\vdash \text{NEXP} \subseteq \text{EXP/poly}$.

To formalize EXP/poly we introduce a formula $\text{acc}_{\text{EXP}}(e, t)$ to express that the universal deterministic Turing machine, run for time and space t on input e , accepts. Note that over $V_2^0 + \text{Exp}$ this can be written equivalently as a $\Sigma_1^{1,b}$ formula and as a $\Pi_1^{1,b}$ formula,

$$\begin{aligned} \exists W, \text{comp}(e, t, W) \wedge \text{“}W \text{ is accepting”} \quad \text{and} \\ \forall W, \text{comp}(e, t, W) \rightarrow \text{“}W \text{ is accepting”}, \end{aligned}$$

since V_2^0 proves that any two computations of the same machine are equal. Here “ W is accepting” means simply that the bit of W coding the output of the computation is 1.

Let φ be any formula and let $c \in \mathbb{N}$. We define a sentence α_φ^c expressing that φ is in EXP/poly with exponent c , where we use the same c to control the amount of advice and the length of the computation:

$$\alpha_\varphi^c := \forall n \in \text{Log} \exists e < 2^{n^c} \forall x < 2^n, \text{acc}_{\text{EXP}}((e, x), 2^{n^c}) \leftrightarrow \varphi(x).$$

What this actually says is that φ is definable by the universal deterministic machine, with a suitable time bound, using advice e that depends on the length of the input. Strictly speaking, being in EXP/poly means being accepted by *some* exponential time machine with

¹Our axiom Exp should not be confused with the complexity class EXP , or with the standard notation exp for the axiom asserting that exponentiation is total on the number sort.

advice, not just the universal machine, but V_2^0 is strong enough to prove that the universal machine can simulate any other machine using an appropriate code and time bound.²

A language is in **NEXP** if and only if it is definable by a $\tilde{\Sigma}_1^{1,b}$ formula, and we will formalize **NEXP** by treating it as the class of predicates defined by such formulas. Again we could be more strict and insist that being in **NEXP** means being definable by the $\tilde{\Sigma}_1^{1,b}$ formula “there is an accepting computation of M on this input” for some non-deterministic machine M , but this would make no difference to our argument. In particular, it is easy to construct an M such that the formula above is equivalent in V_2^0 to the formula “ $\exists Y \neg \text{PHP}(x, Y)$ ” which we use below.

We can now state precisely what we mean for a theory T to prove that $\text{NEXP} \subseteq \text{EXP}/\text{poly}$: we mean that for every $\tilde{\Sigma}_1^{1,b}$ formula φ , there is an exponent $c \in \mathbb{N}$ such that $T \vdash \alpha_\varphi^c$. Thus to prove Proposition 2.1, we need to show that there is a $\tilde{\Sigma}_1^{1,b}$ -formula φ such that $V_2^0 + \text{Exp} + \{\neg \alpha_\varphi^c : c \in \mathbb{N}\}$ is consistent.

Our proof uses a similar approach to [ABM24]. We first prove a lemma that gives us a model of $V_2^0 + \text{Exp}$ in which the pigeonhole principle fails at some size a . Then we show that, if Proposition 2.1 were false, we would be able to prove the pigeonhole principle in this model by induction. Below, $\text{PHP}(x, R)$ is the Σ_∞^b formula expressing that R is *not* the graph of an injection from x to $x-1$. We use the notation Z^e to mean “the e th set coded by Z ” in the standard way of coding many sets into one, that is, we use $x \in Z^e$ to mean $(x, e) \in Z$.

Lemma 2.2. *For any $k \in \mathbb{N}$ there is a model M , with $n \in \text{Log}$ and relations R and Z on M such that*

$$M \models T_2(R, Z) + \forall e < 2^{n^k} \text{comp}(e, 2^{n^k}, Z^e) + \neg \text{PHP}(2^n, R).$$

Proof. Suppose not. Then there is some $k \in \mathbb{N}$ such that

$$T_2(R, Z) \vdash \forall n \in \text{Log}, \left[\exists e < 2^{n^k} \neg \text{comp}(e, 2^{n^k}, Z^e) \right] \vee \text{PHP}(2^n, R).$$

Let us write a for 2^n . By the Paris-Wilkie translation of relativized bounded arithmetic into propositional logic ([PW85] or see [Kra95]), for each n there is a constant-depth Frege refutation π_a , of size quasipolynomial in a , of the propositional formula

$$\langle \forall e < 2^{n^k} \text{comp}(e, 2^{n^k}, Z^e) \rangle \wedge \langle \neg \text{PHP}(a, R) \rangle$$

where expressions in angled brackets represent translations of first-order into propositional formulas, using a as a size parameter. The two conjuncts are in disjoint propositional variables, standing for the bits of Z on the left and the bits of R on the right.

We now observe that there is an assignment α to the bits of Z that satisfies the left-hand conjunct, which we can construct by actually running the exponential time computations on each input e and recording them as sequences of bits. Once we restrict π_a by α , what is left is a quasipolynomial-sized constant-depth refutation of $\langle \neg \text{PHP}(a, R) \rangle$, which is impossible by [KPW95, PBI93]. $\square$

We derive Proposition 2.1 from the lemma.

Proof of Proposition 2.1. Let $\varphi(x)$ be the formula $\exists Y \neg \text{PHP}(x, Y)$. We will show that $V_2^0 + \text{Exp} + \{\neg \alpha_\varphi^c : c \in \mathbb{N}\}$ is consistent. Suppose this is not the case. Then there is some $c \in \mathbb{N}$ such that $V_2^0 \vdash \text{Exp} \rightarrow \alpha_\varphi^c$.

²Precisely, it proves that given a computation of a machine M we can construct a suitable computation of the universal machine simulating M .

Writing a for 2^n , we have that α_φ^c is $\forall n \in \text{Log } \Phi(a)$ where

$$\Phi(a) := \exists e < 2^{n^c} \forall x < 2^n, \text{acc}_{\text{EXP}}((e, x), 2^{n^c}) \leftrightarrow \varphi(x)$$

expresses that φ is in EXP/poly at length n . We can move the quantifier $\forall n \in \text{Log}$ outside the implication $\text{Exp} \rightarrow \alpha_\varphi^c$ and apply Parikh's theorem ([Par71] or see [Bus85]) to obtain that for some k , which we may assume is larger than c ,

$$V_2^0 \vdash \forall n \in \text{Log}, \left[\forall e, t < 2^{n^k} \exists W \text{comp}(e, t, W) \right] \longrightarrow \Phi(a). \quad (2.1)$$

Now let M, n, R, Z be as given by Lemma 2.2, so that M satisfies simultaneously $T_2(R, Z)$, $\forall e < 2^{n^k} \text{comp}(e, 2^{n^k}, Z^e)$ and $\neg\text{PHP}(2^n, R)$. Since comp and $\neg\text{PHP}$ do not contain any universal set-sort quantifiers, we may assume without loss of generality that M is a model of V_2^0 , since we can make it into such a model by adding to it every bounded set definable in M by a $\Sigma_\infty^b(Z, R)$ formula with number-sort parameters. Then in particular $M \models \forall e, t < 2^{n^k} \exists W \text{comp}(e, t, W)$, since every such W is encoded inside Z^e , so $M \models \Phi(a)$ by (2.1).

Thus we have $M \models V_2^0 + \neg\text{PHP}(a, R)$ and simultaneously, expanding $\Phi(a)$, that for some $d \in M$,

$$M \models \forall x < a, \text{acc}_{\text{EXP}}((d, x), 2^{n^c}) \leftrightarrow \exists Y \neg\text{PHP}(x, Y).$$

Since Z uniformly contains all computations running in time up to 2^{n^k} , we can replace $\text{acc}_{\text{EXP}}((d, x), 2^{n^c})$ with an equivalent $\Sigma_\infty^b(Z)$ formula $\theta(Z, d, x)$ where θ simply looks up in Z the final state of the computation on input (d, x) . Since induction holds for $\Sigma_\infty^b(Z)$ formulas in M , and $M \models \forall Y \text{PHP}(x, Y)$ for every standard x , we conclude by induction in M that there is some $x < a$ in M with $\forall Y \text{PHP}(x, Y)$ but $\exists Y' \neg\text{PHP}(x + 1, Y')$. But in V_2^0 , given a relation Y' failing PHP at $x + 1$ we can, by changing at most two pigeons, construct a relation Y failing PHP at x , as in [ABM24]. So this is a contradiction. $\square$

We go on to show the full Theorem 1.2, which replaces $V_2^0 + \text{Exp}$ with the stronger theory $V_2^0 + \forall \tilde{\Sigma}_1^{1,b}(\mathbb{N})$. Let us observe in passing that $V_2^0 + \text{Exp}$ already captures a nontrivial amount of the strength of V_2^1 , namely it proves all the Π_1 number-sort consequences of V_2^1 [KNT11] and in fact every $\forall \tilde{\Sigma}_1^{1,b}$ consequence of V_2^1 [BB14].

Proof of Theorem 1.2. We want to show that $V_2^0 + \forall \tilde{\Sigma}_1^{1,b}(\mathbb{N}) + \{-\alpha_\varphi^c : c \in \mathbb{N}\}$ is consistent, where $\varphi(a)$ is again the formula $\exists Y \neg\text{PHP}(a, Y)$. Suppose not. Then by compactness, and using the fact that in V_2^0 we can combine any finite number of $\forall \tilde{\Sigma}_1^{1,b}$ formulas into one formula, there is $c \in \mathbb{N}$ and a single $\Sigma_\infty^b(U)$ formula $\theta(e, U)$ such that

$$V_2^0 \vdash \text{Exp} \wedge \forall e \exists U \theta(e, U) \rightarrow \alpha_\varphi^c$$

and $\forall e \exists U \theta(e, U)$ is true in the standard model. We include Exp here to guarantee natural behaviour of the relation acc_{EXP} which occurs in α_φ^c .

We now imitate the proof of Proposition 2.1. As in that proof, we can use Parikh's theorem to bound, by some term in $a = 2^n$, the values of e for which we need to witness θ . That is, we have for some $k > c$ that

$$V_2^0 \vdash \forall n \in \text{Log}, \left[\forall e, t < 2^{n^k} \exists W \text{comp}(e, t, W) \wedge \forall e < 2^{n^k} \exists U \theta(e, U) \right] \longrightarrow \Phi(a).$$

Finally we strengthen Lemma 2.2 to give a model $M \models T_2(R, Z, U)$ which, in addition to the conditions in Lemma 2.2, also satisfies $\forall e < 2^{n^k} \theta(e, U^e)$. We can do this using the same argument that we used before for the axiom Exp . Namely, we can always find an assignment

to the variables for U which satisfies the propositional translation $\langle \forall e < 2^{n^k} \theta(e, U^e) \rangle$, because the sentence $\forall e \exists U \theta(e, U)$ is true in the standard model. $\square$

3. NEXP AND coNEXP

We prove Theorem 1.3, that $V_2^0 + \forall \tilde{\Sigma}_1^{1,b}(\mathbb{N}) \not\vdash \text{NEXP} \subseteq \text{coNEXP}$. The argument is slightly different as it does not rely so directly on induction, and it is not clear if a similar argument can work for $\text{coNEXP}/\text{poly}$.

Proof of Theorem 1.3. We will again take the formula $\exists Y \neg \text{PHP}(x, Y)$ as our relation in NEXP. We will show that the theory does not prove it is in coNEXP . Suppose for a contradiction that it does, by which we mean that there is a $\Sigma_\infty^b(S)$ formula $\chi(x, S)$, with no other free variables, such that

$$V_2^0 + \forall \tilde{\Sigma}_1^{1,b}(\mathbb{N}) \vdash \forall x, \exists Y \neg \text{PHP}(x, Y) \leftrightarrow \forall S \chi(x, S) \quad (3.1)$$

(but see the discussion at the end of this section for a different formalization, which allows a parameter in χ and for which our argument does not work).

As in the proof of Theorem 1.2, we can use compactness to replace $\forall \tilde{\Sigma}_1^{1,b}(\mathbb{N})$ with a single true $\forall \tilde{\Sigma}_1^{1,b}$ sentence $\forall e \exists U \theta(e, U)$, then move this to the right-hand side, inside the scope of $\forall x$, and use Parikh's theorem to bound e . We also only take the left-to-right direction of the equivalence in (3.1). We obtain in this way, for some $k \in \mathbb{N}$, that

$$V_2^0 \vdash \forall x, \left[\forall e < 2^{|x|^k} \exists U \theta(e, U) \wedge \exists Y \neg \text{PHP}(x, Y) \right] \rightarrow \forall S \chi(x, S).$$

Hence to get a contradiction it is enough to find a model M with an element a and relations R, S, U on M such that

$$M \models T_2(R, S, U) + \forall e < 2^{|a|^k} \theta(e, U^e) + \neg \text{PHP}(a, R) + \neg \chi(a, S).$$

Suppose there is no such M . Then as in Lemma 2.2 there are quasipolynomial-size refutations π_a of the propositional translation

$$\langle \forall e < 2^{|a|^k} \theta(e, U^e) \rangle \wedge \langle \neg \text{PHP}(a, R) \rangle \wedge \langle \neg \chi(a, S) \rangle.$$

We can construct an assignment to the U variables that satisfies the first conjunct $\langle \forall e < 2^{|a|^k} \theta(e, U^e) \rangle$ exactly as in the proof of Theorem 1.2. For the last conjunct $\langle \neg \chi(a, S) \rangle$, observe that in the standard model $\exists Y \neg \text{PHP}(x, Y)$ is false for all x . Therefore, by (3.1) and the fact that our theory $V_2^0 + \forall \tilde{\Sigma}_1^{1,b}(\mathbb{N})$ is sound, we have that $\forall x \exists S \chi(x, S)$ holds in the standard model, so there is an assignment to the S variables satisfying $\langle \neg \chi(a, S) \rangle$. Applying these two partial assignments to π_a we get quasipolynomial-size refutations of $\langle \neg \text{PHP}(a, R) \rangle$, which is impossible. $\square$

We discuss briefly what happens if you try to extend this argument to show unprovability of $\text{NEXP} \subseteq \text{coNEXP}/\text{poly}$. A natural formalization of this inclusion is: for every $\Sigma_\infty^b(Y)$ formula $\varphi(x, Y)$, there is a $\Sigma_\infty^b(S)$ formula $\chi(e, x, S)$ such that

$$\forall n \in \text{Log} \exists e \forall x < 2^n, \exists Y \varphi(x, Y) \leftrightarrow \forall S \chi(e, x, S). \quad (3.2)$$

Suppose this is provable, and let us ignore for simplicity the dependence of e on n . We can try to imitate the proof of Theorem 1.3. We put $\varphi(x, Y) := \neg \text{PHP}(x, Y)$, so that $\exists Y \varphi(x, Y)$ is always false in the standard model, and use the right-to-left direction of (3.2) to get some $e_0 \in \mathbb{N}$ for which $\forall x \exists S \neg \chi(e_0, x, S)$ is true in the standard model. We can then use this to

construct a model M with relations R, S satisfying, among other things, that $\varphi(a, R)$ and $\neg\chi(e_0, a, S)$. We would like M to in some way falsify the left-to-right direction of (3.2), but it does not, since that asserts that for some e we have $\exists Y\varphi(a, Y) \rightarrow \forall S\chi(e, a, S)$, and we only have that this fails for e_0 , which may be different from e .

Finally let us mention a more liberal formalization of **coNEXP**, for which we do not know how to prove Theorem 1.3. Our formulation of $T \vdash \text{NEXP} \subseteq \text{coNEXP}$ in Theorem 1.3 is: for every formula $\varphi(x)$ representing a **NEXP** predicate, there is a $\Sigma_\infty^b(S)$ formula $\chi(x, S)$ such that $T \vdash \forall x, \varphi(x) \leftrightarrow \forall S\chi(x, S)$. A weaker version of this allows χ to take an extra parameter e , and has the form: for every formula $\varphi(x)$ representing a **NEXP** predicate, there is a $\Sigma_\infty^b(S)$ formula $\chi(e, x, S)$ such that $T \vdash \exists e\forall x, \varphi(x) \leftrightarrow \forall S\chi(e, x, S)$. This is similar to our formalization of $T \vdash \text{NEXP} \subseteq \text{coNEXP}/\text{poly}$, except that e no longer varies with the binary length of x . We do not know how to prove Theorem 1.3 for this version, for the reason sketched in the previous paragraph.

It is helpful to think of $\chi(e, x, S)$ here as expressing acceptance by a universal **coNEXP** machine which takes a description of a machine e . In Theorem 1.3 we ask that there is some fixed, standard $e \in \mathbb{N}$ for which $\varphi(x)$ is equivalent to $\forall x\chi(e, x, S)$; in the weaker version the machine e may be different in different models of T , and may be nonstandard. We can call these different formalizations respectively **coNEXP without parameters** and **coNEXP with parameters**. For example Krajíček [Kra95, Section 7.6] takes the version of a complexity class with parameters as the more basic one; in the notation of [Kra95] the class without parameters in Theorem 1.3 would be denoted rather **coNEXP^{stan}**.

ACKNOWLEDGEMENT

I am grateful to Albert Atserias, Sam Buss and Moritz Müller for helpful discussions on this topic and for comments on earlier versions of this work, and to an anonymous referee for raising the issue of different formalizations of **coNEXP**.

REFERENCES

- [ABM24] Albert Atserias, Sam Buss, and Moritz Müller. On the consistency of circuit lower bounds for non-deterministic time. *Journal of Mathematical Logic*, page 2450023, 2024. doi:10.1142/S0219061324500235.
- [AT25] Albert Atserias and Iddo Tzameret. Feasibly constructive proof of Schwartz–Zippel lemma and the complexity of finding hitting sets. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 1096–1107, 2025. doi:10.1145/3717823.371827.
- [BB14] Arnold Beckmann and Samuel R Buss. Improved witnessing and local improvement principles for second-order bounded arithmetic. *ACM Transactions on Computational Logic*, 15(1):1–35, 2014. doi:10.1145/2559950.
- [BFS09] Harry Buhrman, Lance Fortnow, and Rahul Santhanam. Unconditional lower bounds against advice. In *Automata, Languages and Programming: 36th International Colloquium, ICALP 2009*, pages 195–209, 2009. doi:10.1007/978-3-642-02927-1_18.
- [BKZ15] Samuel Buss, Leszek Kołodziejczyk, and Konrad Zdanowski. Collapsing modular counting in bounded arithmetic and constant depth propositional proofs. *Transactions of the American Mathematical Society*, 367(11):7517–7563, 2015. doi:10.1090/S0002-9947-2015-06233-3.
- [Bus85] Samuel Buss. *Bounded arithmetic*. Princeton University, 1985.
- [Bus98] Samuel Buss. First-order proof theory of arithmetic. *Handbook of proof theory*, 137:79–147, 1998.
- [KNT11] Leszek Aleksander Kołodziejczyk, Phuong Nguyen, and Neil Thapen. The provably total NP search problems of weak second order bounded arithmetic. *Annals of Pure and Applied Logic*, 162(6):419–446, 2011. doi:10.1016/j.apal.2010.12.002.

- [KPW95] Jan Krajíček, Pavel Pudlák, and Alan Woods. An exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle. *Random structures & algorithms*, 7(1):15–39, 1995. doi:10.1002/rsa.3240070103.
- [Kra95] Jan Krajíček. *Bounded arithmetic, propositional logic and complexity theory*. Cambridge University Press, 1995.
- [MP20] Moritz Müller and Ján Pich. Feasibly constructive proofs of succinct weak circuit lower bounds. *Annals of Pure and Applied Logic*, 171(2):102735, 2020. doi:10.1016/j.apal.2019.102735.
- [Par71] Rohit Parikh. Existence and feasibility in arithmetic. *The Journal of Symbolic Logic*, 36(3):494–508, 1971. doi:10.2307/2269958.
- [PBI93] Toniann Pitassi, Paul Beame, and Russell Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational Complexity*, 3:97–140, 1993. doi:10.1007/BF01200117.
- [Pic15] Ján Pich. Logical strength of complexity theory and a formalization of the PCP theorem in bounded arithmetic. *Logical Methods in Computer Science*, 11, 2015. doi:10.2168/LMCS-11(2:8)2015.
- [PS21] Ján Pich and Rahul Santhanam. Strong co-nondeterministic lower bounds for NP cannot be proved feasibly. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 223–233, 2021. doi:10.1145/3406325.3451117.
- [PW85] Jeff Paris and Alex Wilkie. Counting problems in bounded arithmetic. In *Methods in Mathematical Logic: Proceedings of the 6th Latin American Symposium on Mathematical Logic, 1983*, pages 317–340. Springer, 1985. doi:10.1007/BFb0075316.
- [PWW88] J. Paris, A. Wilkie, and A. Woods. Provability of the pigeonhole principle and the existence of infinitely many primes. *The Journal of Symbolic Logic*, 53:1235–1244, 1988. doi:10.2307/2274618.