A PARAMETERIZED HALTING PROBLEM, Δ_0 TRUTH AND THE MRDP THEOREM

YIJIA CHEN, MORITZ MÜLLER, AND KEITA YOKOYAMA

Abstract. We study the parameterized complexity of the problem to decide whether a given natural number n satisfies a given Δ_0-formula $\varphi(x)$; the parameter is the size of φ. This parameterization focusses attention on instances where n is large compared to the size of φ. We show unconditionally that this problem does not belong to the parameterized analogue of AC^0. From this we derive that certain natural upper bounds on the complexity of our parameterized problem imply certain separations of classical complexity classes. This connection is obtained via an analysis of a parameterized halting problem. Some of these upper bounds follow assuming that $I\Delta_0$ proves the MRDP theorem in a certain weak sense.

§1. Introduction.

1.1. Parameterized complexity. While classical complexity theory measures computational resources by functions in the input length n alone, parameterized complexity theory additionally takes into account a *parameter* k associated with inputs.[1] The motivation is to focus attention on inputs with relatively small parameter $k \ll n$, namely, one asks for algorithms that are efficient on such inputs. If "efficient" means polynomial time, this leads to the class FPT: decidable problems that admit a polynomial time algorithm that is correct on inputs satisfying $g(k) \leqslant n$ for some computable $g : \mathbb{N} \to \mathbb{N}$, or equivalently, an algorithm correct on all inputs with runtime $f(k) \cdot n^{O(1)}$ for some computable $f : \mathbb{N} \to \mathbb{N}$. If "efficient" means AC^0, it leads to the class para-AC^0. Here, and throughout, by AC^0 we mean *dlogtime uniform* AC^0.

Many problems have natural parameters in the sense that the focus on inputs with relatively small parameters is practically or theoretically well motivated. Two examples:

p-Halt	
Instance:	$n \in \mathbb{N}$ in unary and a nondeterministic Turing machine $\mathbb{M}$.
Parameter:	$\lvert\mathbb{M}\rvert$, the size of $\mathbb{M}$.
Problem:	Does $\mathbb{M}$ accept the empty input in at most n steps?

Received November 1, 2022.

2020 *Mathematics Subject Classification.* Primary 03F20, 03H15, 68Q15, 68Q19, 68Q27.

Key words and phrases. bounded arithmetical truth, parameterized halting, descriptive complexity, weak arithmetic, MRDP theorem.

[1] All required definitions will be given precisely in Section 2.

0022-4812/00/0000-0000
DOI:10.1017/jsl.2024.44

p-Δ_0-TRUTH	
Instance:	$n \in \mathbb{N}$ in unary and a Δ_0-formula $\varphi(x)$.
Parameter:	$\lvert\varphi\rvert$, the size of φ.
Problem:	$\mathbb{N} \models \varphi(n)$?

The parameterized complexity of both problems is wide open. Before entering their discussion we note a special property: the problems are *almost tally*, in that inputs are long strings of 1's padded with relatively short binary strings. This concept is key to the approach taken here.

1.2. The parameterized halting problem. The importance of p-HALT is derived from its close connections to central problems in proof complexity and descriptive complexity theory [10]: the hypotheses that a certain logic considered by Gurevich [24] does not capture PTIME, and that p-optimal propositional proof systems do not exist [30] are both equivalent to the hypothesis that p-HALT cannot be decided in time $n^{f(k)}$ where $k := |\mathbb{M}|$ and $f : \mathbb{N} \to \mathbb{N}$ is any function.[2] So far, however, such algorithms have been ruled out only under a certain very strong *non-standard* complexity-theoretic hypothesis and only for computable f [9, 10]. Thus, lower bounds on p-HALT are poorly understood and of fundamental interest.

A seemingly modest and natural starting point is the following.

CONJECTURE 1.1. p-HALT $\notin$ para-AC^0.

This conjecture is highly plausible and might appear to be within reach because AC^0 is well-understood and, in particular, [11] establishes (unconditional) para-AC^0 lower bounds for many well-studied parameterized problems. It deserves some genuine interest because its failure implies that AC^0, or equivalently, $(+, \times)$-*invariant* FO is captured by some logic. However, we failed to prove the conjecture after years of attempts and only now understand why: it implies that nondeterministic exponential time NE is distinct from the linear time hierarchy LINH. This connection can be further tightened by considering the following variant of p-HALT:

p-HALT$_=$	
Instance:	$n \in \mathbb{N}$ in unary and a nondeterministic Turing machine $\mathbb{M}$.
Parameter:	$\lvert\mathbb{M}\rvert$.
Problem:	Does $\mathbb{M}$ accept the empty input in *exactly* n steps?

Accepting in exactly n steps means that there exists an accepting computation that has exactly n steps. While the classical problems underlying p-HALT$_=$ and p-HALT are easily seen to be equivalent (see Example 3.6), we shall see that their parameterized versions behave quite differently. In fact, p-HALT$_=$ appears to be harder than p-HALT, and is the hardest among all almost tally problems in para-NP, the parameterized analogue of NP. We refer to Section 7 for a discussion. We show:

[2] [12] gives a direct proof of the equivalence of the first two hypotheses.

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

THEOREM 1.2.

(i) p-HALT$_=$ $\in$ para-AC^0 *if and only if* NE $\subseteq$ LINH.
(ii) p-HALT$_=$ $\in$ para-AC^0 *implies* p-HALT $\in$ para-AC^0.

1.3. Δ_0 truth. Deciding the truth of Δ_0 formulas is a fundamental problem of mathematical logic. The choice of the parameter shifts attention to inputs where n is much larger than $|\varphi|$. This is a natural focus. Classical work of Paris and Dimitracopolous [34] took n to be nonstandard and related the complexity of truth definitions for Δ_0-formulas to the complexity-theoretic hypotheses that LINH or PH does not collapse. Wilkie proved a weak version of the former hypothesis by showing that p-Δ_0-TRUTH restricted to quantifier-free formula inputs can be decided in space $f(k) + O(\log n)$ where $k := |\varphi|$ is the parameter and $f : \mathbb{N} \to \mathbb{N}$ a computable function [36, proof of Lemma 3.1]. The straightforward algorithm decides p-Δ_0-TRUTH in space $f(k) \cdot \log n$. Can it be decided in space $f(k) + O(\log n)$? Maybe with nondeterminism? Can it be decided in time $f(k) \cdot n^{O(1)}$, i.e., is it in FPT? Maybe with nondeterminism, i.e., is it in para-NP?

Our main result (Theorem 4.3) shows that such *upper bounds* on the parameterized complexity of p-Δ_0-TRUTH imply *lower bounds* in classical complexity theory. Notably,

THEOREM 1.3. *If* p-Δ_0-TRUTH $\in$ para-NP, *then* NE $\not\subseteq$ LINH.

The proof relies on our analysis of p-HALT$_=$ and the following unconditional lower bound:

THEOREM 1.4. p-Δ_0-TRUTH $\notin$ para-AC^0.

The proof is based on diagonalization or, more specifically, the undefinability of truth. Furthermore, it relies on the classical result [6] of descriptive complexity theory that, roughly speaking, equates (uniform) AC^0 and first-order logic with built-in arithmetic.

1.4. The MRDP theorem. Theorem 1.3 yields some information concerning the provability of the Matiyasevich-Robinson-Davis-Putnam (MRDP) theorem (see [14] for an account) in bounded arithmetic. This theorem states that Σ_1-definable sets are *Diophantine* and it is a long standing open problem whether it is provable in $I\Delta_0$, i.e., Peano arithmetic with induction restricted to Δ_0-formulas.

Wilkie observed [36] that a positive answer would imply NP = co-NP. Gaifman and Dimitracopoulos [22] showed that adding exponentiation suffices: $I\Delta_0 + \forall x \exists y\ 2^x = y$ does prove MRDP. Kaye [27] proved MRDP using only induction for bounded existential formulas plus an axiom stating the totality of a suitable function of exponential growth. It is asked in [22, page 188] whether $I\Delta_0$ plus the totality of $x^{\log x}$, or of $x^{\log\log x}$ etc. proves MRDP. A positive answer would imply that $I\Delta_0$ *proves* MRDP *for small numbers*: this would mean that the equivalence of any Δ_0-formula $\varphi(\bar{x})$ to some Diophantine formula is proved in $I\Delta_0$ for all $\bar{x}$ of logarithmic order. Model-theoretically, the equivalence holds in any $I\Delta_0$-model for all $\bar{x}$ from the initial segment of numbers x such that 2^x exists, while proof-theoretically, we allow an $I\Delta_0$-proof to use exponentiation, but only once. Such limited use of exponentiation has been studied in bounded arithmetic [29].

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

We show that Theorem 1.3 implies:

THEOREM 1.5. *If $I\Delta_0$ proves MRDP for small numbers, then* NE $\not\subseteq$ LINH.

1.5. AC^0-bi-immunity. Could Conjecture 1.1 be false? We give further evidence for its truth by establishing a connection to the existence of AC^0-bi-immune sets in NP. Recall a problem Q is AC^0-*immune* if it does not have an infinite subset in AC^0; if additionally, also the complement of Q is AC^0-immune, then Q is AC^0-*bi-immune*.

THEOREM 1.6. *If* NP *contains an* AC^0*-bi-immune problem, then* p-HALT $\notin$ para-AC^0.

It is a standard hypothesis that NP contains even P-bi-immune problems and this follows from the measure hypothesis [31]. Whether NP contains at least AC^0-bi-immune problems has been asked once it was realized [1, 23] that deterministic time hierarchy theorems hold with bi-immunity (or, equivalently [5], almost everywhere) while this is open for nondeterministic time [1, 21]. While Zimand [37] obtained some partial positive answers, Allender and Gore [2] showed that this has different answers relative to different oracles.[3] This indicates that also refuting Conjecture 1.1 might be non-trivial.

1.6. Outline. Much of the technical work consists in connecting the dots between results of various subareas of logic and complexity, namely classical, parameterized and descriptive complexity theory and formal arithmetic. Section 2 reviews the results we need and fixes our notation. The technicalities are somewhat subtle, in particular, the move from p-HALT to p-HALT$_=$ is crucial. Section 3 introduces almost tally problems and proves Theorem 1.2 and various variants of it. Section 4 proves Theorem 1.4. This together with the results in Section 3 implies Theorem 1.3 and various variants. Section 5 derives (a strengthening of) Theorem 1.5. Section 6 proves Theorem 1.6. The final section discusses the role of uniformity, and exhibits the different behaviours of our parameterized problems p-HALT, p-HALT$_=$ and p-Δ_0-TRUTH.

§2. Preliminaries. Standard monographs are [3, 32] for classical complexity theory, [15, 16, 20] for parameterized complexity theory, [25, 28] for formal arithmetic, and [17, 26] for descriptive complexity theory.

2.1. Classical complexity. A *(classical) problem* is a subset of $\{0,1\}^*$, the set of finite binary strings. The length of a binary string $x \in \{0,1\}^*$ is denoted $|x|$. For $n \in \mathbb{N}$ we let 1^n denote the binary string consisting of n many 1's. We use multitape Turing machines with alphabet $\{0,1\}$ as our basic model of computation. When considering *dlogtime* Turing machines, i.e., deterministic machines running in time $O(\log n)$, it is understood that they access their input via an address tape (see, e.g., [6]). As usual, P and NP denote deterministic and nondeterministic polynomial time $n^{O(1)}$, and E and NE denote deterministic and nondeterministic exponential time with linear exponent $2^{O(n)}$. The *linear time hierarchy* LINH is the set of problems acceptable by alternating Turing machines in linear time $O(n)$ with $O(1)$

[3] [2] studies AC^0-immunity but their oracle constructions can be adapted to AC^0-bi-immunity.

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

alternations. LINSPACE and NLINSPACE denote deterministic and nondeterministic linear space $O(n)$. Clearly,

$$\mathsf{LINH} \subseteq \mathsf{LINSPACE} \subseteq \mathsf{NLINSPACE} \subseteq \mathsf{E} \subseteq \mathsf{NE}.$$

Following [6] we define (dlogtime uniform) AC^0 as the set of problems decided by AC^0-*circuit families* $(\mathsf{C}_n)_{n\in\mathbb{N}}$:

- C_n is a circuit (with $\wedge, \vee, \neg$ gates and unbounded fan-in) with n variables, size $\leqslant n^c$ and depth[4] $\leqslant d$, where $c, d \in \mathbb{N}$ are two constants independent of n;
- there is a dlogtime Turing machine which given $\langle 1^n, i, b\rangle$ where $n, i \in \mathbb{N}$ and $b \in \{0,1\}$ decides whether the ith bit of the binary encoding of C_n is b.

Here, for binary strings $x = x_0 \dots x_{|x|-1}$ and $y = y_0 \dots y_{|y|-1}$ we use the standard pairing

$$\langle x, y\rangle := x_0 x_0 \dots x_{|x|-1} x_{|x|-1} 01 y_0 y_0 \dots y_{|y|-1} y_{|y|-1}, \tag{1}$$

and similarly for more arguments. The above definition is somewhat sensitive to the choice of the binary encoding of a circuit. An appropriate choice would be to encode C_n by the list of strings in the *direct connection language* corresponding to n; we refer to [6] for details.

For $n \in \mathbb{N}$ we let $bin(n) \in \{0,1\}^*$ denote the binary expansion of n; it has length $\lceil \log(n+1)\rceil$ for $n > 0$. For $x \in \{0,1\}^*$ let $num(x)$ be the natural number with binary expansion $1x$, i.e., $bin(num(x)) = 1x$. For a problem Q let

$$un(Q) := \left\{ 1^{num(x)} \mid x \in Q \right\}.$$

The last statement of the following is [2, Proposition 5], and the first two are trivial:

PROPOSITION 2.1 [2]. *Let Q be a problem. Then*:

(i) $Q \in \mathsf{NE}$ *if and only if* $un(Q) \in \mathsf{NP}$.
(ii) $Q \in \mathsf{E}$ *if and only if* $un(Q) \in \mathsf{P}$.
(iii) $Q \in \mathsf{LINH}$ *if and only if* $un(Q) \in \mathsf{AC}^0$.

2.2. Parameterized complexity. A *parameterized problem* is a pair (Q, κ) of an *underlying* classical problem $Q \subseteq \{0,1\}^*$ and a *parameterization* $\kappa : \{0,1\}^* \to \mathbb{N}$ mapping an instance $x \in \{0,1\}^*$ to its *parameter* $\kappa(x) \in \mathbb{N}$. We follow [18] and require that κ is computable by an AC^0-circuit family $(\mathsf{C}_n)_{n\in\mathbb{N}}$. That is, for all $x \in \{0,1\}^*$, besides $|x|$ inputs the circuit $C_{|x|}$ has $|x|$ outputs and computes $bin(\kappa(x))$, possibly padded with leading 0's to length $|x|$. It is a technical assumption satisfied by almost all parameterized problems of interest. For example, p-HALT has underlying classical problem $\{\langle 1^n, \mathbb{M}\rangle \mid$ the nondeterministic Turing machine $\mathbb{M}$ accepts the empty input in at most n steps$\}$ and a parameterization κ that maps strings of the form $\langle 1^n, \mathbb{M}\rangle$ to $|\mathbb{M}|$ and other strings to, say, 0.

The para-operator [19] turns a classical complexity class into a parameterized one (the most important intractable parameterized classes are not of this form, however). The class para-$\mathsf{P} = \mathsf{FPT}$ contains the parameterized problems (Q, κ) that are *fixed-parameter tractable*, i.e., decidable in deterministic time $f(\kappa(x)) \cdot |x|^{O(1)}$

[4]We assume $\neg$ gates are in front of inputs and not counted in depth; e.g., CNFs have depth 2.

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

for some computable $f : \mathbb{N} \to \mathbb{N}$. Similarly, para-NP denotes nondeterministic time $f(\kappa(x)) \cdot |x|^{O(1)}$ (for any computable f), para-L denotes deterministic space $f(\kappa(x)) + O(\log |x|)$, and para-NL denotes nondeterministic such space. Clearly,

$$\text{para-L} \subseteq \text{para-NL} \subseteq \text{FPT} \subseteq \text{para-NP}.$$

The central parameterized class in this paper is para-AC^0. It is characterized as follows:

PROPOSITION 2.2 [11]. *Let (Q, κ) be a parameterized problem. The following are equivalent*:

(i) $(Q, \kappa) \in$ para-AC^0.
(ii) *There is a family $(\mathsf{C}_{n,k})_{n,k\in\mathbb{N}}$ of circuits such that*:
  – *there are a computable $f : \mathbb{N} \to \mathbb{N}$ and $c, d \in \mathbb{N}$ such that for all $n, k \in \mathbb{N}$ the circuit $\mathsf{C}_{n,k}$ has n inputs, size at most $f(k) \cdot n^c$, and depth at most d;*
  – *for all $x \in \{0,1\}^*$ we have*

$$x \in Q \iff \mathsf{C}_{|x|,\kappa(x)}(x) = 1;$$

  – *there are a computable $g : \mathbb{N} \to \mathbb{N}$ and a deterministic Turing machine which given as input $\langle 1^n, 1^k, i, b\rangle$ where $n, k, i \in \mathbb{N}$ and $b \in \{0,1\}$ decides in time $g(k) + O(\log n)$ whether the ith bit of the binary encoding of $\mathsf{C}_{n,k}$ is b.*
(iii) *Q is decidable and there are a computable $h : \mathbb{N} \to \mathbb{N}$ and an AC^0-circuit family $(\mathsf{C}_n)_{n\in\mathbb{N}}$ such that for all $x \in \{0,1\}^*$ with $|x| \geqslant h(\kappa(x))$ we have*

$$x \in Q \iff \mathsf{C}_{|x|}(x) = 1.$$

According to the terminology of [19], (iii) states that (Q, κ) is *eventually in* AC^0.

2.3. Formal arithmetic. We let $L_{\text{ar}} := \{+, \times, 0, 1, <\}$ be the language of arithmetic with binary function symbols $+, \times$, constants $0, 1$ and a binary relation symbol $<$. The *standard L_{ar}-structure*, denoted $\mathbb{N}$, has universe $\mathbb{N}$ and interprets the symbols in the obvious way. Every L_{ar}-term p computes a polynomial with coefficients in $\mathbb{N}$ and of total degree at most $|p|$. We do not distinguish terms p or formulas φ from their binary encodings, so $|p|$ and $|\varphi|$ denote the lengths of these encodings. Writing $\varphi(\bar{x})$ for a formula φ means that *all* free variables of φ are among $\bar{x}$. A *sentence* is a formula without free variables.

A *Δ_0-formula* is an L_{ar}-formula obtained from atomic formulas, Boolean connectives, and bounded quantifiers $\exists x{<}p$, $\forall x{<}p$ where p is an L_{ar}-term not involving x; e.g., $\exists x{<}p\ \varphi$ stands for $\exists x(x{<}p \wedge \varphi)$. *$\Sigma_1$- and Π_1-formulas* are obtained from Δ_0-formulas by existential and universal quantification, respectively.

THEOREM 2.3 (MRDP). *For every Δ_0-formula $\varphi(\bar{x})$ there are L_{ar}-terms $p(\bar{x}, \bar{y}), q(\bar{x}, \bar{y})$ such that*

$$\mathbb{N} \models \forall \bar{x}\big(\varphi(\bar{x}) \leftrightarrow \exists \bar{y}\ p(\bar{x}, \bar{y}){=}q(\bar{x}, \bar{y})\big).$$

Gödel showed that computable functions are Σ_1-definable. The MRDP theorem improves this to an existential definition:

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

COROLLARY 2.4. *For every computable $f : \mathbb{N} \to \mathbb{N}$ there is a quantifier-free L_{ar}-formula $\varphi_f(x, y, \bar{z})$ such that for every $n, m \in \mathbb{N}$*

$$f(n) = m \iff \mathbb{N} \models \exists \bar{z}\ \varphi_f(n, m, \bar{z}).$$

We are mainly concerned with finite arithmetical structures with universe

$$[n] := \{0, \dots, n-1\}$$

for some $n \in \mathbb{N}$ with $n \geqslant 2$, and therefore consider the relational version

$$L^{\mathrm{r}}_{\mathrm{ar}}$$

of L_{ar} where $+$, $\times$ are ternary relation symbols. The standard $L^{\mathrm{r}}_{\mathrm{ar}}$-structure with universe $\mathbb{N}$, also denoted $\mathbb{N}$, interprets $+$, $\times$ by the graphs of addition and multiplication, respectively. For $n \in \mathbb{N}$ with $n > 1$, the standard $L^{\mathrm{r}}_{\mathrm{ar}}$-structure with universe $[n]$, simply denoted n, is the substructure of $\mathbb{N}$ with universe $[n]$, i.e., it interprets the symbols in $L^{\mathrm{r}}_{\mathrm{ar}}$ by $+^{[n]} := \{(k, \ell, m) \in [n]^3 \mid k + \ell = m\}$, $\times^{[n]} := \{(k, \ell, m) \in [n]^3 \mid k \cdot \ell = m\}$, $0^{[n]} := 0$, $1^{[n]} := 1$ and $<^{[n]} := \{(k, \ell) \in [n]^2 \mid k < \ell\}$.

Let $\varphi^{<y}$ be obtained from φ by replacing all quantifiers $\exists z$, $\forall z$ by $\exists z{<}y$, $\forall z{<}y$. For $\bar{n} = (n_0, \dots, n_{k-1}) \in \mathbb{N}^k$ and $n \in \mathbb{N}$ write $\bar{n} < n$ to express $n_i < n$ for all $i < k$. For every $L^{\mathrm{r}}_{\mathrm{ar}}$-formula $\varphi(\bar{x})$ with $1, \bar{n} < n$ we have

$$\mathbb{N} \models \varphi^{<n}(\bar{n}) \iff n \models \varphi(\bar{n}).$$

REMARK 2.5. Corollary 2.4 holds for a quantifier-free $L^{\mathrm{r}}_{\mathrm{ar}}$-formula $\varphi_f(\bar{x}, y, \bar{z})$. Indeed, it is straightforward to express an L_{ar}-term (in)equality by an existential $L^{\mathrm{r}}_{\mathrm{ar}}$-formula.

2.4. Descriptive complexity. A binary string $x = x_0 \cdots x_{n-1} \in \{0, 1\}^*$ of length $n > 1$ is often identified with the *string structure* $\mathcal{S}(x)$ defined as the $L^{\mathrm{r}}_{\mathrm{ar}} \cup \{ONE\}$-expansion of the standard $L^{\mathrm{r}}_{\mathrm{ar}}$-structure n that interprets the unary relation symbol ONE by

$$ONE^x := \{i \in [n] \mid x_i = 1\},$$

i.e., $\mathcal{S}(x) = \big([n], +^{[n]}, \times^{[n]}, 0^{[n]}, 1^{[n]}, <^{[n]}, ONE^x\big)$. We shall work with the following descriptive characterization of (dlogtime uniform) AC^0:

THEOREM 2.6 [6]. *A problem Q is in AC^0 if and only if there is an $L^r_{ar} \cup \{ONE\}$-sentence φ such that for every $x \in \{0, 1\}^*$ with $|x| \geqslant 2$:*

$$x \in Q \iff \mathcal{S}(x) \models \varphi.$$

This result and Proposition 2.2(iii) imply what is to be our working definition of para-AC^0: the parameterized problems that are *eventually definable*.

COROLLARY 2.7. *Let (Q, κ) be a parameterized problem with decidable Q. Then (Q, κ) is in* para-AC^0 *if and only if (Q, κ) is eventually definable: there are a computable $h : \mathbb{N} \to \mathbb{N}$ and an $L^r_{ar} \cup \{ONE\}$-sentence φ such that for all $x \in \{0, 1\}^*$ with $|x| \geqslant h(\kappa(x))$:*

$$x \in Q \iff \mathcal{S}(x) \models \varphi.$$

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

In descriptive complexity the role of reductions is played by interpretations. Let L, L' be languages consisting of relation symbols and constants. Let $w \in \mathbb{N}$ with $w \geqslant 1$. An *interpretation I of L' in L (of width w)* is given by an L-formula $\varphi_{\text{uni}}(\bar{x})$, an L-formula $\varphi_R(\bar{x}_0, \dots, \bar{x}_{r-1})$ for each r-ary relation symbol $R \in L'$, and an L-formula $\varphi_c(\bar{x})$ for every constant $c \in L'$; here, $\bar{x}$, $\bar{x}_i$ are w-tuples of variables. Given an L-structure A define the L'-structure A^I as follows. It has universe $A^I := \{\bar{a} \in A^w \mid A \models \varphi_{\text{uni}}(\bar{a})\}$, interprets an r-ary $R \in L'$ by $\{(\bar{a}_0, \dots, \bar{a}_{r-1}) \in (A^I)^r \mid A \models \varphi_R(\bar{a}_0, \dots, \bar{a}_{r-1})\}$, and a constant $c \in L'$ by the unique $\bar{a} \in A^I$ satisfying $\varphi_c(\bar{x})$ in A. If this uniqueness is violated or if the universe A^I is empty, then A^I is not defined. If $B \cong A^I$ for some I, we say B is *interpretable* in A. The following is standard.

LEMMA 2.8. *Let I an interpretation of L' in L of width w and I' an interpretation of L'' in L' of width w'. Further let A be an L-structure such that A^I is defined.*

(i) *For every L'-formula $\varphi(x, y, \dots)$ there is an L-formula $\varphi^I(\bar{x}, \bar{y}, \dots)$ where $\bar{x}, \bar{y}, \dots$ are w-tuples of variables such that for all $\bar{a}, \bar{b}, \dots \in A^I$:*

$$A^I \models \varphi(\bar{a}, \bar{b}, \dots) \iff A \models \varphi^I(\bar{a}, \bar{b}, \dots).$$

(ii) *There is an interpretation $I' \circ I$ of L'' in L of width $w \cdot w'$ such that if $(A^I)^{I'}$ is defined, then so is $A^{I' \circ I}$ and*

$$A^{I' \circ I} \cong (A^I)^{I'}.$$

The following is folklore, and a proof can be found in [35, Appendix].

LEMMA 2.9. *Let $d \in \mathbb{N}$.*

(i) *For every $n > 1$ the standard L^r_{ar}-structure n^d is interpretable in the standard L^r_{ar}-structure n. In fact, there is an interpretation I_d of width d such that $n^d \cong n^{I_d}$ for every $n > 1$, and the isomorphism maps each $a < n^d$ to the length d representation of a in base n.*

(ii) *There is an L^r_{ar}-formula $BIT(x, y)$ such that for every $n > 1$ and all $i, j \in [n]$:*

$$n \models BIT(i, j) \iff \text{the } j\text{th bit of } bin(i) \text{ is } 1.$$

§3. p-Halt and NE versus LINH. In this section we first introduce a workable notion of reduction that preserves para-AC^0, then prove Theorem 1.2, then introduce almost tally problems and show p-HALT$_=$ is the hardest among them in para-NP, and finally consider some generalizations and variants that will be instrumental later in Section 4 for the proof of Theorem 1.3 and its variants.

3.1. Eventually definable reductions. A *parameterized* reduction from a parameterized problem (Q, κ) to another (Q', κ') is a reduction $r : \{0,1\}^* \to \{0,1\}^*$ from Q to Q' such that $\kappa' \circ r \leqslant f \circ \kappa$ for some computable function $f : \mathbb{N} \to \mathbb{N}$.

DEFINITION 3.1. Let κ be a parameterization. A function $r : \{0,1\}^* \to \{0,1\}^*$ is *κ-eventually definable* if there are a computable $h : \mathbb{N} \to \mathbb{N}$ and an interpretation I such that

$$\mathcal{S}(x)^I \cong \mathcal{S}(r(x))$$

for all $x \in \{0,1\}^*$ with $|x| \geqslant h(\kappa(x))$.

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

EXAMPLE 3.2. The function

$$\langle 1^n, x\rangle \mapsto 1^{num(\langle bin(n), x\rangle)},$$

where $n \in \mathbb{N}$, $x \in \{0,1\}^*$ is κ-eventually definable where κ maps $\langle 1^n, x\rangle$ to $|x|$ (both functions map arguments that are not of the required form to, say, 0).

PROOF. Note $num(\langle bin(n), x\rangle) < 2^{|\langle bin(n),x\rangle|+1} \leqslant 2^{O(\log n+|x|)}$. Choose a constant $d \in \mathbb{N}$ and a computable $h : \mathbb{N} \to \mathbb{N}$ such that $num(\langle bin(n), x\rangle) < n^d$ and $num(x) < n$ whenever $n \geqslant h(|x|)$. It suffices to describe an interpretation of $\mathcal{S}(1^{num(\langle bin(n),x\rangle)})$ in $\mathcal{S}(\langle 1^n, x\rangle)$ whenever $n \geqslant h(|x|)$. It will be clear that the interpretation does not depend on n, x.

Let $(n, num(x))$ be the expansion of the standard $L^{\mathrm{r}}_{\mathrm{ar}}$-structure n that interprets a new constant by $num(x) \in [n]$. This is interpretable in $\mathcal{S}(\langle 1^n, x\rangle)$ using BIT. By Lemma 2.9, also $(n^d, num(x))$ is interpretable in $\mathcal{S}(\langle 1^n, x\rangle)$. But this structure defines (n and) $num(\langle bin(n), x\rangle) \in [n^d]$ using BIT. Thus, $\mathcal{S}(1^{num(\langle bin(n),x\rangle)})$ is interpretable in $\mathcal{S}(\langle 1^n, x\rangle)$ as claimed.

Finally, note there is a sentence φ that is true exactly in structures of the desired form $\mathcal{S}(\langle 1^n, x\rangle)$ for $n \in \mathbb{N}$ and $x \in \{0,1\}^*$. It is easy to modify the above interpretation to produce a structure isomorphic to $\mathcal{S}(0)$ given a structure that is not of the desired form. ⊣

EXAMPLE 3.3. Let $P : \{0,1\}^* \to \{0,1\}^*$ be computable. The function

$$\langle 1^n, x\rangle \mapsto \langle 1^n, P(x)\rangle,$$

where $n \in \mathbb{N}$, $x \in \{0,1\}^*$ is κ-eventually definable where κ maps $\langle 1^n, x\rangle$ to $|x|$ (both functions map arguments that are not of the required form to, say, 0).

PROOF. Let $p : \mathbb{N} \to \mathbb{N}$ be computable with $p(num(x)) = num(P(x))$ for all $x \in \{0,1\}^*$. We choose:

- a quantifier-free $L^{\mathrm{r}}_{\mathrm{ar}}$-formula $\varphi(x, y, \bar{z})$ according to Remark 2.5,
- a computable $f : \mathbb{N} \to \mathbb{N}$ so that for all $\ell \in \mathbb{N}$

$$\mathbb{N} \models \exists \bar{z} < f(\ell)\ \varphi(\ell, p(\ell), \bar{z}),$$

- and a computable $h : \mathbb{N} \to \mathbb{N}$ such that $h(|x|) > num(x), num(P(x)), f(num(x))$ for all $x \in \{0,1\}^*$.

Assume $n \geqslant h(|x|)$. Then $\mathcal{S}(\langle 1^n, x\rangle)$ interprets the expansion (n, ℓ) of the standard structure n by a constant c interpreting $\ell := num(x)$. In (n, ℓ) the formula $\exists \bar{z} \varphi(c, y, \bar{z})$ defines $p(\ell) = num(P(x))$. Using BIT, thus $\mathcal{S}(\langle 1^n, x\rangle)$ interprets $\mathcal{S}(\langle 1^n, P(x)\rangle)$.

Again it is easy to modify this interpretation to produce a structure isomorphic to $\mathcal{S}(0)$ given a structure that is not of the desired form. ⊣

Recall, a function $r : \{0,1\}^* \to \{0,1\}^*$ is *honest* if $|r(x)| \geqslant |x|^{\Omega(1)}$.

LEMMA 3.4. *Assume that $r, r' : \{0,1\}^* \to \{0,1\}^*$ are κ- and κ'-eventually definable, respectively, that $\kappa' \circ r \leqslant f \circ \kappa$ for some computable $f : \mathbb{N} \to \mathbb{N}$, and that r is honest. Then $r' \circ r$ is κ-eventually definable.*

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

PROOF. Choose I, h for r and I', h' for r' according to Definition 3.1. We can assume that h' is nondecreasing. Choose $n_0, c \in \mathbb{N}$ such that $|r(x)| \geqslant |x|^{1/c}$ for all $x \in \{0,1\}^*$ with $|x| \geqslant n_0$. Define $g : \mathbb{N} \to \mathbb{N}$ by

$$g(k) := \max\{(h'(f(k)))^c, h(k), n_0\}.$$

We claim that $I' \circ I$ and g witness that $r' \circ r$ is κ-eventually definable. To verify this let $x \in \{0,1\}^*$ satisfy $|x| \geqslant g(k)$ where $k := \kappa(x)$. Then $|r(x)| \geqslant |x|^{1/c} \geqslant h'(f(k)) \geqslant h'(\kappa'(r(x)))$ using that h' is nondecreasing. Hence $\mathcal{S}(r(x))^{I'} \cong \mathcal{S}(r'(r(x)))$. As $|x| \geqslant g(k) \geqslant h(k)$, we conclude $\mathcal{S}(x)^I \cong \mathcal{S}(r(x))$, which implies $\mathcal{S}(x)^{I' \circ I} \cong \mathcal{S}(r'(r(x)))$. ⊣

DEFINITION 3.5. Let (Q, κ) and (Q', κ') be parameterized problems. An *eventually definable reduction from* (Q, κ) *to* (Q', κ') is a parameterized reduction from (Q, κ) to (Q', κ') that is honest and κ-eventually definable.

EXAMPLE 3.6. There is an eventually definable reduction from p-HALT to p-HALT$_=$.

PROOF. Let $P : \{0,1\}^* \to \{0,1\}^*$ map a nondeterministic Turing machine $\mathbb{M}$ to another $\mathbb{M}'$ that simulates $\mathbb{M}$ and, if $\mathbb{M}$ accepts, then $\mathbb{M}'$ nondeterministically makes any number of steps before it halts and accepts; strings x not encoding machines are mapped to themselves. This is clearly a parameterized reduction. By Example 3.3, $\langle 1^n, x\rangle \mapsto \langle 1^n, P(x)\rangle$ is eventually definable.

Recall that this function outputs 0 on strings y not of the desired form $\langle 1^n, x\rangle$, i.e., the interpretation produces a structure isomorphic to $\mathcal{S}(0)$. We change the interpretation to output $\mathcal{S}(y)$ in this case. This ensures honesty (we can assume $|\mathbb{M}'| \geqslant |\mathbb{M}|$) and thus gives a reduction as desired. ⊣

REMARK 3.7. A parameterized problem (Q, κ) is in para-AC^0 if and only if Q is decidable and there is an eventually definable reduction from (Q, κ) to a trivial problem, say, (Q_0, κ_0) for Q_0 the set of strings starting with 0 and κ_0 is constantly 0.

It is straightforward to check that this reducibility is transitive and preserves membership in para-AC^0:

LEMMA 3.8. *Assume that there is an eventually definable reduction from* (Q, κ) *to* (Q', κ').

(i) *If there is an eventually definable reduction from* (Q', κ') *to* (Q'', κ''), *then there is one from* (Q, κ) *to* (Q'', κ'').
(ii) *If* $(Q', \kappa') \in$ para-AC^0 *and* Q *is decidable, then* $(Q, \kappa) \in$ para-AC^0.

PROOF. (i) follows from Lemma 3.4. (ii) follows from (i) and Remark 3.7. ⊣

3.2. The complexity of p-Halt$_=$. It is known that the question whether p-HALT$_=$ is fixed-parameter tractable is closely related to the relationship of E and NE:

THEOREM 3.9 [4, 7]. p-HALT$_=$ $\in$ FPT *if and only if* NE $\subseteq$ E.

Theorem 1.2(i) is a para-AC^0-analogue of this result.

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

THEOREM 1.2.

(i) p-HALT$_=$ $\in$ para-AC^0 *if and only if* NE $\subseteq$ LINH.
(ii) p-HALT$_=$ $\in$ para-AC^0 *implies* p-HALT $\in$ para-AC^0.

PROOF. (ii) follows from Example 3.6 and Lemma 3.8. Alternatively, let $(\mathsf{C}_{n,k})_{n,k}$ witness p-HALT$_=$ $\in$ para-AC^0 according to Proposition 2.2(b). Then $(\bigvee_{m\leqslant n} \mathsf{C}^n_{m,k})_{n,k}$ witnesses p-HALT $\in$ para-AC^0 where $\mathsf{C}^n_{m,k}$ checks its input has the form $\langle 1^n, x\rangle$ for some $x \in \{0,1\}^k$ and then runs $\mathsf{C}_{m,k}$ on $\langle 1^m, x\rangle$.

To prove (i), first assume NE $\subseteq$ LINH and let Q be the classical problem underlying p-HALT$_=$ but with input n encoded in binary:

Q	
Instance:	$n \in \mathbb{N}$ in *binary* and a nondeterministic Turing machine $\mathbb{M}$.
Problem:	Does $\mathbb{M}$ accept the empty input in exactly n steps?

Clearly, $Q \in$ NE, so by assumption and Proposition 2.1(iii) we have $un(Q) \in$ AC^0. Recall

$$un(Q) = \left\{ 1^{num(\langle bin(n), \mathbb{M}\rangle)} \;\middle|\; \begin{array}{l}\text{the nondeterministic Turing machine } \mathbb{M} \\ \text{accepts the empty input in exactly } n \text{ steps}\end{array} \right\}.$$

By Example 3.2 the map $\langle 1^n, \mathbb{M}\rangle \mapsto 1^{num(\langle bin(n),\mathbb{M}\rangle)}$ is eventually definable with respect to the parameterization of p-HALT$_=$. It is an honest parameterized reduction to $(un(Q), \kappa)$ where κ maps $1^{num(\langle bin(n),\mathbb{M}\rangle)}$ to $|\mathbb{M}|$. Since $(un(Q), \kappa) \in$ para-AC^0, Lemma 3.8 implies p-HALT$_=$ $\in$ para-AC^0.

Conversely, assume p-HALT$_=$ $\in$ para-AC^0. Let $Q \subseteq \{0,1\}^*$ be a problem in NE. To show that $Q \in$ LINH, it suffices to prove $un(Q) \in$ AC^0 again by Proposition 2.1(iii).

As $Q \in$ NE there is a constant $c \in \mathbb{N}$ and a nondeterministic Turing machine $\mathbb{M}$ accepting Q that on input x halts in time at most $num(x)^c - 2|x| - 2$. Consider the nondeterministic Turing machine $\mathbb{M}^*$ that started with the empty input runs as follows:

1. guess $y \in \{0,1\}^*$
2. simulate $\mathbb{M}$ on y
3. **if** $\mathbb{M}$ rejects, **then** reject
4. make dummy steps such that so far the total running time is $num(y)^c$
5. accept.

Line 1 takes exactly $2|y| + 2$ many steps by moving the head forth and back on some tape, so the dummy steps in line 4 are possible. Since *num* is injective, we have

$$x \in Q \iff \mathbb{M}^* \text{ accepts the empty input tape in exactly } num(x)^c + 1 \text{ many steps.} \tag{2}$$

Since $\mathbb{M}^*$ is a fixed machine, p-HALT$_=$ $\in$ para-AC^0 implies that the classical problem

$$Q' := \left\{ 1^n \;\middle|\; \mathbb{M}^* \text{ accepts the empty input tape in exactly } n+1 \text{ many steps} \right\}$$

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

is in AC^0. Choose a first-order sentence φ for Q' according to Theorem 2.6. Lemma 2.9 gives an interpretation I such that $\mathcal{S}(1^n)^I \cong \mathcal{S}(1^{n^c})$ for all $n > 1$. Then $1^{n^c} \in Q'$ is equivalent to $\mathcal{S}(1^n) \models \varphi^I$. Thus the r.h.s. in (2) is equivalent to $\mathcal{S}(1^{num(x)}) \models \varphi^I$ provided $num(x) > 1$, i.e., x is non-empty. The l.h.s. in (2) is equivalent to $1^{num(x)} \in un(Q)$. Thus φ^I witnesses that $un(Q) \in \mathsf{AC}^0$ according to Theorem 2.6. ⊣

REMARK 3.10. The direction from left to right only required an AC^0-circuit family for instances of p-HALT$_=$ with the fixed machine $\mathbb{M}^*$. This implies that the assertions in Theorem 1.2(i) are equivalent to p-HALT$_= \in \mathsf{XAC}^0$ (see Definition 7.1).

3.3. Almost tally problems. Recall that a classical problem $Q \subseteq \{0,1\}^*$ is *tally* if $Q \subseteq \{1\}^*$. All parameterized problems mentioned in the introduction are almost tally in the following sense:

DEFINITION 3.11. A parameterized problem (Q, κ) is *almost tally* if

$$Q \subseteq \left\{ \langle 1^n, x \rangle \;\middle|\; n \in \mathbb{N}, x \in \{0,1\}^* \right\}$$

and there is a computable $f : \mathbb{N} \to \mathbb{N}$ such that for all $n \in \mathbb{N}$, $x \in \{0,1\}^*$

$$|x| \leqslant f(\kappa(\langle 1^n, x \rangle)).$$

Theorem 1.2(ii) holds not only for p-HALT but for every almost tally problem in para-NP. In fact, p-HALT$_=$ is the hardest almost tally problem in para-NP:

LEMMA 3.12. *For every almost tally problem in* para-NP *there is an eventually definable reduction to* p-HALT$_=$.

PROOF. Let $(Q, \kappa) \in$ para-NP be almost tally. The identity is a parameterized reduction from (Q, κ) to its re-parameterization (Q, κ') where $\kappa'(\langle 1^n, x \rangle) := |x|$ for all $n \in \mathbb{N}$, $x \in \{0,1\}^*$. Hence, the identity is an eventually definable reduction. We can therefore assume that $\kappa = \kappa'$.

Let $\mathbb{M}$ be a nondeterministic Turing machine that accepts Q and on input $\langle 1^n, x \rangle$ runs in time at most $f(k) \cdot n^c$ where $c \in \mathbb{N}$, $f : \mathbb{N} \to \mathbb{N}$ is a computable function, and $k := |x|$.

Define $g : \mathbb{N}^2 \to \mathbb{N}$ by

$$g(m,k) := m^{c+1} + 2m + 2k + 2.$$

For $x \in \{0,1\}^*$ with $k := |x|$, consider the nondeterministic Turing machine $\mathbb{M}_x$ that on the empty input runs as follows:

1. nondeterministically write $\langle 1^m, x \rangle$ for some $m \in \mathbb{N}$
2. simulate $\mathbb{M}$ on $\langle 1^m, x \rangle$
3. **if** $\mathbb{M}$ does not halt or rejects, **then** reject
4. make dummy steps such that so far the total running time is $g(m,k)$
5. accept.

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

Step 1 can be implemented to take exactly $2 + 2m + 2 + 2k$ many steps (recall (1)), so the dummy steps in line 4 are possible if $m > f(k)$. Note that for each k, the function $m \mapsto g(m,k)$ is injective. Thus, if $n > f(k)$, we have

$$\langle 1^n, x\rangle \in Q \iff \langle 1^{g(n,k)+1}, \mathbb{M}_x\rangle \in p\text{-}\textsc{Halt}_=.$$

Using Example 3.3, one easily constructs an eventually definable reduction that maps $\langle 1^n, x\rangle$ to $\langle 1^{g(n,k)+1}, \mathbb{M}_x\rangle$. ⊣

It is straightforward to infer from Proposition 2.1 that NE ⊆ LINH if and only if every tally problem in NP is in AC^0. We don't know of a similarly easy proof of the following parameterized variant of this observation. Instead, our proof relies on our analysis of p-$\textsc{Halt}_=$:

Corollary 3.13. NE ⊆ LINH *if and only if every almost tally problem in* para-NP *is in* para-AC^0.

Proof. The l.h.s. is equivalent to p-$\textsc{Halt}_= \in$ para-AC^0 by Theorem 1.2(i). And by Lemmas 3.8 and 3.12, p-$\textsc{Halt}_= \in$ para-AC^0 is equivalent to the r.h.s. ⊣

3.4. Variants. For the optimistic reader, Corollary 3.13 gives an approach to separate NE from LINH. From this perspective, it is of interest to ask whether finding an almost tally problem outside para-AC^0 but in a natural subclass of para-NP implies stronger separations of natural complexity classes. We verify the following variants of Corollary 3.13:

Lemma 3.14.

(i) E ⊆ LINH *if and only if every almost tally problem in* FPT *is in* para-AC^0.
(ii) NLINSPACE ⊆ LINH *if and only if every almost tally problem in* para-NL *is in* para-AC^0.
(iii) LINSPACE ⊆ LINH *if and only if every almost tally problem in* para-L *is in* para-AC^0.

Proof. The proof of (i) is analogous to the proof of Corollary 3.13 using the subproblem of p-$\textsc{Halt}_=$ where the input machine $\mathbb{M}$ is deterministic. Similarly the proof of (iii) is analogous to the proof of (ii). We show how (ii) is proved by modifying the proof of Corollary 3.13.

Consider the following variant of p-Halt:

p-$\textsc{Halt}^*_=$	
Instance:	$n, m \in \mathbb{N}$ in unary with $n \leqslant m$ and a nondeterministic Turing machine $\mathbb{M}$.
Parameter:	$\lvert\mathbb{M}\rvert$.
Problem:	Does $\mathbb{M}$ accept the empty input in *exactly* n steps and space at most $\lfloor \log m \rfloor$?

Here, the space $\lfloor \log m \rfloor$ of a run bounds all work tapes together, that is, if c_i is the maximal cell number visited on work tape i, then $\sum_i c_i \leqslant \lfloor \log m \rfloor$.

It is clear that this problem is in para-NL.

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

CLAIM 1. *p-HALT$^*_=$ $\in$ para-AC^0 if and only if* NLINSPACE $\subseteq$ LINH.

PROOF OF CLAIM 1. Assume NLINSPACE $\subseteq$ LINH and let Q be the classical problem underlying p-HALT$^*_=$ but with the inputs n, m encoded in binary. Clearly, $Q \in$ NLINSPACE $\subseteq$ LINH, so $un(Q) \in$ AC^0 by Proposition 2.1(iii). Similarly as Example 3.2 one sees that

$$\langle 1^n, 1^m, \mathbb{M}\rangle \mapsto 1^{num(\langle bin(n), bin(m), \mathbb{M}\rangle)}$$

is eventually definable. Then p-HALT$^*_=$ $\in$ para-AC^0 follows as in Theorem 1.2(i).

Conversely, assume p-HALT$^*_=$ $\in$ para-AC^0 and let $Q \in$ NLINSPACE. Choose a nondeterministic Turing machine $\mathbb{M}$ accepting Q that on input $x \in \{0,1\}^*$ runs in time at most

$$\frac{num(x)^c}{10c(|x|+2)} - 10c(|x|+2) - |x|$$

and uses space at most $c \cdot |x|$; here $c \in \mathbb{N}$ is a suitable constant. Define $\mathbb{M}^*$ as in the proof of Theorem 1.2 but with the following implementation details. For the simulation in line 2, first initialize a length $c(|y|+2)$ binary counter using exactly $10c(|y|+2)$ steps, and increase it using exactly $10c(|y|+2)$ many steps for each simulated step of $\mathbb{M}$. In line 4 continue increasing the counter in this way until it reaches $num(y)^c/(10c(|y|+2))$. For long enough y, the binary representation of this number can be computed in time at most $num(y)$ and space $O(|y|)$ (where the constant in the O-notation depends on c). This computation can be done in parallel to the simulation in lines 2 and 4. Hence, line 5 completes exactly $num(y)^c + 1$ steps, and uses space at most $d \cdot |y|$ for a suitable $d \geqslant c$.

Thus, we arrive at the following variant of (2). For long enough $x \in \{0,1\}^*$:

$$x \in Q \iff \mathbb{M}^* \text{ accepts the empty input in exactly } num(x)^c + 1 \text{ many steps and space at most } \left\lfloor \log(num(x)^d) \right\rfloor.$$

Our assumption p-HALT$^*_=$ $\in$ para-AC^0 implies that the classical problem

$$Q' := \left\{ \langle 1^n, 1^m\rangle \;\middle|\; n \leqslant m \text{ and } \mathbb{M}^* \text{ accepts the empty input in exactly } n+1 \text{ many steps and space at most } \lfloor \log m \rfloor \right\}$$

is in AC^0. Now $un(Q) \in$ AC^0 (and hence $Q \in$ LINH) follows as in Theorem 1.2(i) using an interpretation I such that $\mathcal{S}(1^n)^I \cong \mathcal{S}(\langle 1^{n^c}, 1^{n^d}\rangle)$. ⊣

CLAIM 2. *For every almost tally problem in* para-NL *there is an eventually definable reduction to p-HALT$^*_=$.*

PROOF OF CLAIM 2. Let $(Q, \kappa) \in$ para-NL be almost tally and $\mathbb{M}$ be a nondeterministic Turing machine that accepts Q and that on input $\langle 1^n, x\rangle$ runs in time at most $f(k) \cdot n^c$ and space at most $f(k) + c \cdot \log n$ where $c \in \mathbb{N}$, $f : \mathbb{N} \to \mathbb{N}$ is a computable function, and $k := \kappa(\langle 1^n, x\rangle)$. We can assume $k = |x|$ (see the proof of Lemma 3.12).

For $x \in \{0,1\}^*$ with $k := |x|$, define the nondeterministic Turing machine $\mathbb{M}_x$ as in the proof of Lemma 3.12 but with a different g (chosen below) and line 1 changed

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

to nondeterministically write some $m \in \mathbb{N}$ in binary in exactly $2\lceil\log(m+1)\rceil + 2$ steps. The simulation in line 2 is done as in the previous claim maintaining a length $(c+1)\lceil\log(m+1)\rceil$ binary counter. It further maintains the position of $\mathbb{M}$'s head on the input tape which we can assume to be at most $|\langle 1^m, x\rangle| + 1$ and uses it to compute the currently scanned bit. Counter and position are updated for each simulated step of $\mathbb{M}$. If $k < m$, then one step of $\mathbb{M}$ is simulated in exactly $10c\lceil\log(m+1)\rceil$ steps. In line 4 the binary counter is updated until it reaches m^{c+1}. Hence line 4 is completed after exactly $g(m,k) := m^{c+1} \cdot 10c\lceil\log(m+1)\rceil + 2\lceil\log(m+1)\rceil + 2$ steps. The dummy steps in line 4 are possible if $m > f(k)$. In this case the computation takes space at most $d \log m$ for suitable $d \in \mathbb{N}$. Thus, if $n > f(k)$, we have

$$\langle 1^n, x\rangle \in Q \iff \langle 1^{g(n,k)+1}, 1^{n^d}, \mathbb{M}_x\rangle \in p\text{-}\textsc{Halt}^*_=.$$

Similarly as seen in the proof of Lemma 3.12, this implies the claim. ⊣

It now suffices to show that $p\text{-}\textsc{Halt}^*_= \in \mathsf{para\text{-}AC}^0$ if and only if every almost tally problem in para-NL is in para-AC^0. The forward direction follows from Claim 2 and Lemma 3.8. And if $p\text{-}\textsc{Halt}^*_= \notin \mathsf{para\text{-}AC}^0$, then we get an almost tally problem in para-NL $\setminus$ para-AC^0 by rewriting inputs $\langle 1^n, 1^m, \mathbb{M}\rangle$ of $p\text{-}\textsc{Halt}^*_=$ to $\langle 1^{\langle n,m\rangle}, \mathbb{M}\rangle$ where $\langle n, m\rangle$ is a pairing function on $\mathbb{N}$. ⊣

We find it worthwhile to explicitly point out the following direct corollary concerning the parameterized halting problem for *deterministic* Turing machines:

COROLLARY 3.15. *If* $p\text{-}\textsc{DHalt} \notin \mathsf{para\text{-}AC}^0$, *then* $\mathsf{E} \not\subseteq \mathsf{LINH}$.

p-DHALT	
Instance:	$n \in \mathbb{N}$ in unary and a deterministic Turing machine $\mathbb{M}$.
Parameter:	$\lvert\mathbb{M}\rvert$.
Problem:	Does $\mathbb{M}$ accept the empty input in at most n steps?

§4. On the parameterized complexity of p-Δ_0-Truth. Recall, the problem p-Δ_0-TRUTH asks whether a given $n \in \mathbb{N}$ in unary satisfies a given Δ_0-formula $\varphi(x)$, parameterized by the length of φ. Further recall that Δ_0 refers to the language L_{ar} with function symbols $+, \cdot$ and contains the L_{ar}-formulas with quantifiers bounded by L_{ar}-terms.

This section first observes that p-Δ_0-TRUTH is "the same" as a basic parameterized model-checking problem, uses this to prove the lower bound p-Δ_0-$\textsc{Truth} \notin \mathsf{para\text{-}AC}^0$ (Theorem 1.4), and finally, based on the previous section, infers consequences from *upper bounds* on the parameterized complexity of p-Δ_0-TRUTH, including Theorem 1.3.

4.1. Model-checking arithmetic. Recall $L^{\mathrm{r}}_{\mathrm{ar}}$ is the relational version of the language of arithmetic L_{ar}. We observe that p-Δ_0-TRUTH is "the same" as the parameterized model-checking problem for first-order logic over finite standard $L^{\mathrm{r}}_{\mathrm{ar}}$-structures:

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

p-MC$(L^{\mathrm{R}}_{\mathrm{AR}})$	
Instance:	$n > 1$ in unary and an $L^{\mathrm{r}}_{\mathrm{ar}}$-sentence φ.
Parameter:	$\lvert\varphi\rvert$.
Problem:	$n \models \varphi$?

LEMMA 4.1. *There is a computable function that maps every Δ_0-formula $\varphi(x)$ to an L^r_{ar}-sentence ψ such that for all $n \in \mathbb{N}$ with $n > 1$:*

$$\mathbb{N} \models \varphi(n) \iff n \models \psi. \tag{3}$$

Further, there is a computable function that maps every L^r_{ar}-sentence ψ to a Δ_0-formula $\varphi(x)$ such that (3) holds all $n \in \mathbb{N}$ with $n > 1$.

PROOF. For the second assertion define $\varphi(x)$ as $\psi^{<x}$ with atoms rewritten in the functional language L_{ar}. The first assertion is folklore, see [22, Proposition 2.2]. We give a brief sketch for completeness. It is routine to compute, given a Δ_0-formula $\varphi(\bar{x})$, a constant $c_\varphi > 1$ and an $L^{\mathrm{r}}_{\mathrm{ar}}$-formula $\psi_0(\bar{x})$ such that

$$\mathbb{N} \models \varphi(\bar{n}) \iff \mathbb{N} \models \psi_0^{<m}(\bar{n})$$

for all $\bar{n}, m \in \mathbb{N}$ with $m \geqslant \max\{\bar{n}, 2\}^{c_\varphi}$. Hence, for $n > 1$, the truth of $\varphi(n)$ is equivalent to $n^{c_\varphi} \models \psi_0(n)$. Since the number n is definable in the standard $L^{\mathrm{r}}_{\mathrm{ar}}$-structure n^{c_φ} (as the minimal element whose c_φth power does not exist), we can replace $\psi_0(n)$ by some sentence ψ_1. Then set $\psi := \psi_1^{I_{c_\varphi}}$ for the interpretation I_{c_φ} from Lemma 2.9. ⊣

4.2. A lower bound. In this subsection we prove the following.

THEOREM 1.4. p-Δ_0-TRUTH $\notin$ para-AC^0.

We fix a proper elementary extension M of the standard $L^{\mathrm{r}}_{\mathrm{ar}}$-model $\mathbb{N}$, and a nonstandard element $a \in M \setminus \mathbb{N}$. We let $<^M$ denote the interpretation of $<$ in M. We need a simple lemma:

LEMMA 4.2. *Let $f : \mathbb{N} \to \mathbb{N}$ be a computable function. Then there is an L^r_{ar}-formula $\chi_f(x, y)$ such that for every $k \in \mathbb{N}$ and every $b \in M$:*

$$f(k) = b \iff M \models \chi_f^{<a}(k, b).$$

PROOF. Choose $\varphi_f(x, y, \bar{z})$ according to Remark 2.5, and set $\chi_f(x, y) := \exists \bar{z} \varphi_f(x, y, \bar{z})$. In particular, φ_f is quantifier-free, so $\chi_f^{<a} = \exists \bar{z}{<}a\ \varphi_f$. Let $k \in \mathbb{N}$ and $b \in M$.

If $b = f(k)$, then $\mathbb{N} \models \varphi_f(k, b, \bar{m})$ for some $\bar{m} \in \mathbb{N}^{|\bar{z}|}$ since $\chi_f(x, y)$ defines f. Then $\bar{m} <^M a$ and $M \models \varphi_f(k, b, \bar{m})$, so $M \models \chi_f^{<a}(k, b)$.

If $M \models \chi_f^{<a}(k, b)$, then both $M \models \chi_f(k, b)$ and $M \models \chi_f(k, f(k))$. But $\chi_f(x, y)$ defines a function in $\mathbb{N}$ and hence in M (by elementarity of the extension), so $b = f(k)$. ⊣

Some notation: for $n \in \mathbb{N}$ define the $L^{\mathrm{r}}_{\mathrm{ar}}$-formula "$x{=}n$" by "$x{=}0$" := $x{=}0$ and "$x{=}(n+1)$" := $\exists y(\text{"}y{=}n\text{"} \wedge +(y, 1, x))$. For an $L^{\mathrm{r}}_{\mathrm{ar}}$-formula $\varphi(y, \bar{x})$ set $\varphi(\underline{n}, \bar{x}) :=$

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

$\exists y(\text{“}y{=}n\text{”} \wedge \varphi(y, \bar{x}))$; we understand $\varphi^{<z}(\underline{n}, \bar{x})$ as $\big(\varphi(\underline{n}, \bar{x})\big)^{<z}$. If $n < m$, then both $(\text{“}x{=}n\text{”})^{<m}$ and “$x{=}n$” define n in $\mathbb{N}$, so $\varphi^{<m}(\underline{n}, \bar{x})$ and $\varphi^{<m}(n, \bar{x})$ are equivalent in $\mathbb{N}$. In particular, for every $n \in \mathbb{N}$:

$$M \models \forall \bar{x}\big(\varphi^{<a}(\underline{n}, \bar{x}) \leftrightarrow \varphi^{<a}(n, \bar{x})\big). \tag{4}$$

Proof of Theorem 1.4. For contradiction, assume otherwise, so $p\text{-MC}(L^{\mathrm{r}}_{\mathrm{ar}}) \in \mathsf{para\text{-}AC}^0$ by Lemma 4.1. By Corollary 2.7, there is an increasing computable function $h : \mathbb{N} \to \mathbb{N}$ and a sentence sat such that for every $n \in \mathbb{N}$ and every $L^{\mathrm{r}}_{\mathrm{ar}}$-sentence φ with $n > h(num(\varphi))$ we have

$$n \models \varphi \iff \mathcal{S}\big(\langle 1^n, \varphi\rangle\big) \models sat. \tag{5}$$

⊣

For $k < n$, let (n, k) denote the expansion of the standard $L^{\mathrm{r}}_{\mathrm{ar}}$-structure n that interprets a constant c by k. It is clear that there is an interpretation I (independent of n, φ) such that $(n, num(\varphi))^I \cong \mathcal{S}(\langle 1^n, \varphi\rangle)$ for all φ with $num(\varphi) < n$. Replacing in sat^I the constant c by a new variable x gives an $L^{\mathrm{r}}_{\mathrm{ar}}$-formula $true(x)$ such that for $n > h(num(\varphi)) \geqslant num(\varphi)$:

$$\begin{aligned} \mathcal{S}\big(\langle 1^n, \varphi\rangle\big) \models sat &\iff n \models true\big(num(\varphi)\big) \\ &\iff \mathbb{N} \models true^{<n}\big(num(\varphi)\big), \end{aligned} \tag{6}$$

where $\mathbb{N}$ is the standard $L^{\mathrm{r}}_{\mathrm{ar}}$-model. Since $h : \mathbb{N} \to \mathbb{N}$ is computable, there is an $L^{\mathrm{r}}_{\mathrm{ar}}$-formula “$h(x) < y$” with the obvious meaning. Note the l.h.s. of (5) is equivalent to $\mathbb{N} \models \varphi^{<n}$. Combining (5) and (6) we get that $\mathbb{N}$ satisfies the universal closure of

$$\text{“}h(num(\varphi)) < y\text{”} \to \big(\varphi^{<y} \leftrightarrow true^{<y}(num(\varphi))\big)$$

for every $L^{\mathrm{r}}_{\mathrm{ar}}$-sentence φ. But $M \models \text{“}h(num(\varphi)) < a\text{”}$, hence

$$M \models \varphi^{<a} \leftrightarrow true^{<a}(num(\varphi)) \tag{7}$$

for every $L^{\mathrm{r}}_{\mathrm{ar}}$-sentence φ. As stated in [34, proof of Proposition 3] this contradicts Tarski's undefinability of truth. We include the details as they are omitted in [34].

The function which for every $L^{\mathrm{r}}_{\mathrm{ar}}$-formula $\varphi(x)$ maps $num(\varphi)$ to $num(\varphi(\underline{num(\varphi)}))$ is computable. So by Lemma 4.2, there is a formula $sub(x, y)$ such that for every formula $\varphi(x)$ and every $b \in M$:

$$b = num(\varphi(\underline{num(\varphi)})) \iff M \models sub^{<a}(num(\varphi), b). \tag{8}$$

Define $\chi(x) := \forall y\big(sub(x, y) \to \neg true(y)\big)$ and $\theta := \chi(\underline{num(\chi)})$, and note

$$num(\theta) = num(\chi(\underline{num(\chi)})). \tag{9}$$

We arrive at the desired contradiction:

$$\begin{aligned} M \models \theta^{<a} &\iff M \models \forall y{<}a\big(sub^{<a}(num(\chi), y) \to \neg true^{<a}(y)\big) && \text{by (4)} \\ &\iff \text{for all } b <^M a : M \models sub^{<a}(num(\chi), b) \to \neg true^{<a}(b) \\ &\iff M \models \neg true^{<a}(num(\theta)) && \text{by (8) and (9)} \\ &\iff M \not\models \theta^{<a} && \text{by (7).} \end{aligned}$$

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

4.3. Upper bounds. Based on our analysis of halting problems in Section 3, we now see that various *upper bounds* on the complexity of p-Δ_0-Truth imply separations of classical complexity classes from LINH. This is our main result. The first assertion is Theorem 1.3:

Theorem 4.3.

(i) *If* p-Δ_0-Truth $\in$ para-NP, *then* NE $\not\subseteq$ LINH.
(ii) *If* p-Δ_0-Truth $\in$ FPT, *then* E $\not\subseteq$ LINH.
(iii) *If* p-Δ_0-Truth $\in$ para-NL, *then* NLINSPACE $\not\subseteq$ LINH.
(iv) *If* p-Δ_0-Truth $\in$ para-L, *then* LINSPACE $\not\subseteq$ LINH.

Proof. Since p-Δ_0-Truth is an almost tally problem, (i) follows from Theorem 1.4 and Corollary 3.13. The other assertions follow using Lemma 3.14. ⊣

§5. Provability of the MRDP theorem. In this section we prove:

Theorem 1.5. *If* $I\Delta_0$ *proves MRDP for small numbers, then* NE $\not\subseteq$ LINH.

In fact, we show that Theorem 1.3 implies a stronger statement for all computably enumerable Π_1-theories—up to logical equivalence, $I\Delta_0$ is a Π_1-theory. Here, a *theory* is a set of sentences, and a Π_1-*theory* is a set of Π_1-sentences. The proof uses Parikh's theorem [33]:

Theorem 5.1. *Let* T *be a* Π_1*-theory and* $\varphi(\bar{x}, \bar{y})$ *a* Δ_0*-formula. If* T *proves* $\exists \bar{y}\ \varphi(\bar{x}, \bar{y})$, *then* T *proves* $\exists \bar{y}{<}p(\bar{x})\ \varphi(\bar{x}, \bar{y})$ *for some term* $p(\bar{x})$.

Definition 5.2. A theory T *proves* MRDP if for every Δ_0-formula $\varphi(\bar{x})$ there are L_{ar}-terms $p(\bar{x}, \bar{y})$ and $q(\bar{x}, \bar{y})$ such that T proves

$$\varphi(\bar{x}) \leftrightarrow \exists \bar{y}\ p(\bar{x}, \bar{y}){=}q(\bar{x}, \bar{y}).$$

As mentioned in the introduction it is a long standing open problem whether $I\Delta_0$ proves MRDP and it is known that adding exponentiation suffices. Intuitively, the following concept asks whether MRDP can be proved using exponentiation only once.

Definition 5.3. A theory T *proves* MRDP *for small numbers* if for every $k \in \mathbb{N}$ and every Δ_0-formula $\varphi(\bar{x}) = \varphi(x_0, \ldots, x_{k-1})$ there are L_{ar}-terms $p(\bar{x}, \bar{y})$ and $q(\bar{x}, \bar{y})$ such that T proves

$$\textstyle\bigwedge_{i<k} 2^{x_i} {\leqslant} z \rightarrow \Big(\varphi(\bar{x}) \leftrightarrow \exists \bar{y}\ p(\bar{x}, \bar{y}){=}q(\bar{x}, \bar{y})\Big). \tag{10}$$

Here, $2^x {\leqslant} z$ stands for a well-known Δ_0-formula [25, Section V.3(c)]. The following strengthens Theorem 1.5:

Theorem 5.4. *Let* T *be a true* Π_1*-theory. Moreover, assume that* T *is computably enumerable. If* T *proves MRDP for small numbers, then* p-Δ_0-Truth $\in$ para-NP *and thus* NE $\not\subseteq$ LINH.

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

PROOF. Assume T proves (10) for $\varphi(x)$, and hence

$$2^x \leqslant z \wedge \varphi(x) \to \exists \bar{y}\ p(x,\bar{y}) = q(x,\bar{y}).$$

By Theorem 5.1 $\exists \bar{y}$ can be replaced by $\exists \bar{y} < r(x,z)$ for some term $r(x,z)$. But since T proves (10) for $\varphi(x)$, T proves

$$2^x \leqslant z \to \big(\varphi(x) \leftrightarrow \exists \bar{y} < r(x,z)\ p(x,\bar{y}) = q(x,\bar{y})\big).$$

Since T is computably enumerable, such terms p, q, r can be computed from φ. Given an instance $\langle 1^n, \varphi\rangle$ of p-Δ_0-TRUTH, compute p, q, r as above, guess $\bar{m} < r(n, 2^n)$ and check $p(n,\bar{m}) = q(n,\bar{m})$. Note the length of the guess $\bar{m}$ is $O(|r| \cdot \ell \cdot n)$ where ℓ is the length of the tuple $\bar{y}$. The check can be done in time $(|p| \cdot |q| \cdot |r| \cdot n)^{O(1)}$.

It follows that p-Δ_0-TRUTH $\in$ para-NP. Now apply Theorem 1.3. ⊣

It would be interesting to find variants of this result that infer p-Δ_0-TRUTH $\in$ FPT or p-Δ_0-TRUTH $\in$ para-NL from certain provabilities of MRDP or other arithmetical statements. Note this implies stronger separations of complexity classes by Theorem 4.3.

§6. p-Halt and a universal AC^0-easy set in NP. Recall, a problem Q is AC^0-bi-immune if neither Q nor its complement contain an infinite subset in AC^0. In this section we prove the following.

THEOREM 1.6. *If* NP *contains an* AC^0*-bi-immune problem, then* p-HALT $\notin$ para-AC^0.

We use the following technical lemma stating, roughly, that every computable function is dominated by a computable injection which is AC^0-invertible.

LEMMA 6.1. *Let* $f : \mathbb{N} \to \mathbb{N}$ *be computable. Then there is an increasing* $h : \mathbb{N} \to \mathbb{N}$ *with the following properties.*

(i) $h(n) \geqslant f(n^2)$ *for every* $n \in \mathbb{N}$.
(ii) $1^n \mapsto 1^{h(n)}$ *is computable in time* $h(n)^{O(1)}$.
(iii) *There is an* L^r_{ar}*-sentence* φ_h *such that for every* $x \in \{0,1\}^*$ *with* $|x| > 1$:

$$\mathcal{S}(x) \models \varphi_h \iff x = 1^{h(n)} \text{ for some } n \in \mathbb{N}.$$

(iv) *There is an* L^r_{ar}*-formula* $\varphi(x)$ *that defines* n *in* $\mathcal{S}(1^{h(n)})$ *for every* $n > 1$.

PROOF. Given a deterministic Turing machine $\mathbb{M}$ and $x \in \{0,1\}^*$ we let $y_{\mathbb{M},x} \in \{0,1\}^*$ encode the computation of $\mathbb{M}$ on x. This encoding can be chosen so that:

(a) $x \mapsto y_{\mathbb{M},x}$ is computable in time $|y_{\mathbb{M},x}|^{O(1)}$.
(b) $\big\{\langle x, y_{\mathbb{M},x}\rangle \bigm| x \in \{0,1\}^*\big\} \in \mathsf{AC}^0$.

Now, let $\mathbb{M}_f$ be a Turing machine that computes $1^n \mapsto 1^{f(n)}$. Let $\mathbb{M}$ be the machine that on input 1^n runs $\mathbb{M}_f$ on 1^{i^2} for every $i \leqslant n$. Define the *increasing* function $h : \mathbb{N} \to \mathbb{N}$ by

$$h(n) = num\big(\langle 1^n, y_{\mathbb{M},1^n}\rangle\big). \tag{11}$$

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

Clearly, the string $y_{\mathbb{M}_f,1^{n^2}}$ encoding the computation of $\mathbb{M}_f$ on input 1^{n^2} has length at least $f(n^2)$. Similarly, $|y_{\mathbb{M},1^n}| \geqslant f(n^2)$. Thus $h(n) \geqslant f(n^2)$ for every $n \in \mathbb{N}$, i.e., (i) holds.

(ii) holds by (a). To show (iii), Theorem 2.6 and (b) imply that there is an $L^{\mathrm{r}}_{\mathrm{ar}}$-sentence φ that holds precisely in the string structures of the form $\mathcal{S}\big(bin(h(n))\big)$ for $n \in \mathbb{N}$. Using *BIT*, there is an interpretation I such that $\mathcal{S}(1^m)^I \cong \mathcal{S}(bin(m))$ for every $m > 1$, so $\varphi_h := \varphi^I$ holds precisely in the string structures of the form $\mathcal{S}(1^{h(n)})$ for $n \in \mathbb{N}$ (we have $h(n) > 1$ for all $n \in \mathbb{N}$).

Trivially, n is definable in $\mathcal{S}(bin(h(n)))$, so (iv) follows using the interpretation I above. ⊣

Theorem 1.6 is an easy consequence of the following stronger result, and we view it as good evidence for the truth of Conjecture 1.1.

THEOREM 6.2. *Assume* p-HALT $\in$ para-AC^0. *Then there is an infinite tally problem* X *such that for every* $Q \in \mathsf{NP}$ *we have* $Q \cap X \in \mathsf{AC}^0$.

Proof of Theorem 1.6 from Theorem 6.2: Assume p-HALT $\in$ para-AC^0 and let $Q \in \mathsf{NP}$. Let X be as stated in Theorem 6.2. Then either $Q \cap X$ or $(\{0,1\}^* \setminus Q) \cap X$ is infinite. By Theorem 6.2 they are both in AC^0; indeed, $(\{0,1\}^* \setminus Q) \cap X = (\{0,1\}^* \cap X) \setminus (Q \cap X)$ is in AC^0 because both $\{0,1\}^* \cap X$ and $Q \cap X$ are. Hence, Q is not AC^0-bi-immune.

PROOF OF THEOREM 6.2. By Corollary 2.7 there is a computable increasing function $f : \mathbb{N} \to \mathbb{N}$ and an $L^{\mathrm{r}}_{\mathrm{ar}}$-sentence φ such that for every $\langle 1^n, \mathbb{M}\rangle$ with $n \geqslant f(|\mathbb{M}|)$:

$$\mathcal{S}\big(\langle 1^n, \mathbb{M}\rangle\big) \models \varphi \iff \mathbb{M} \text{ accepts the empty input tape in at most } n \text{ steps.} \quad (12)$$

Now let $h : \mathbb{N} \to \mathbb{N}$ be the increasing function as stated in Lemma 6.1. In particular, there is a deterministic Turing machine $\mathbb{M}_h$ and a constant $c \geqslant 1$ such that on input 1^m the machine $\mathbb{M}_h$ outputs the string $1^{h(m)}$ in time $h(m)^c$. The desired set X is defined by

$$X := \big\{1^{h(m)} \mid m > 1\big\}.$$

By Lemma 6.1(iii) the sentence φ_h witnesses $X \in \mathsf{AC}^0$ according to Theorem 2.6.

Now let $Q \subseteq \{0,1\}^*$ be a problem in NP. In particular, there is a nondeterministic Turing machine $\mathbb{M}_Q$ accepting Q and a constant $d \geqslant 1$ such that $\mathbb{M}_Q$ on x runs in time $|x|^d$.

Define the nondeterministic Turing machine $\mathbb{M}_{Q,h,m}$ to run $\mathbb{M}_h$ on 1^m to produce output $1^{h(m)}$ and then run $\mathbb{M}_Q$ on $1^{h(m)}$. This machine runs in time

$$n(m) := h(m)^c + h(m)^d.$$

Choose a constant $e \in \mathbb{N}$ such that $m \geqslant |\mathbb{M}_h| + |\mathbb{M}_Q| + e$ implies $m^2 \geqslant |\mathbb{M}_{Q,h,m}|$. Then

$$n(m) \geqslant h(m) \geqslant f(m^2) \geqslant f(|\mathbb{M}_{Q,h,m}|).$$

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

Hence, by (12), for $m \geqslant |\mathbb{M}_h| + |\mathbb{M}_Q| + e$:

$$\begin{aligned} 1^{h(m)} \in Q &\iff \mathbb{M}_{Q,h,m} \text{ accepts the empty input in at most } n(m) \text{ steps} \\ &\iff \mathcal{S}(\langle 1^{n(m)}, \mathbb{M}_{Q,h,m}\rangle) \models \varphi. \end{aligned} \tag{13}$$

Lemma 6.1(iv) implies that there is an interpretation I such that for every $m \in \mathbb{N}$

$$\mathcal{S}(1^{h(m)})^I = \mathcal{S}\big(\langle 1^{n(m)}, \mathbb{M}_{Q,h,m}\rangle\big).$$

By Theorem 2.6 it suffices to show that for every $x \in \{0,1\}^*$ with $|x| \geqslant h(|\mathbb{M}_h| + |\mathbb{M}_Q| + e)$:

$$x \in Q \cap X \iff \mathcal{S}(x) \models \varphi_h \wedge \varphi^I.$$

Assume $x \in Q \cap X$. Then $x = 1^{h(m)}$ for some $m > 1$ and $\mathcal{S}(x) \models \varphi_h$. Since $|x| = h(m) \geqslant h(|\mathbb{M}_h| + |\mathbb{M}_Q| + e)$ and h is increasing, we have $m \geqslant |\mathbb{M}_h| + |\mathbb{M}_Q| + e$. Thus $x = 1^{h(m)} \in Q$ implies $\mathcal{S}(\langle 1^{n(m)}, \mathbb{M}_{Q,h,m}\rangle) \models \varphi$ by (13), and $\mathcal{S}(1^{h(m)}) \models \varphi^I$ follows.

Conversely, assume $S(x) \models \varphi_h \wedge \varphi^I$. By $\mathcal{S}(x) \models \varphi_h$, we have $x \in X$, so $x = 1^{h(m)}$ for some $m > 1$. By $\mathcal{S}(1^{h(m)}) \models \varphi^I$ we have $\mathcal{S}(\langle 1^{n(m)}, \mathbb{M}_{Q,h,m}\rangle) \models \varphi$. This implies $x = 1^{h(m)} \in Q$ by (13) because, as above, $m \geqslant |\mathbb{M}_h| + |\mathbb{M}_Q| + e$. ⊣

§7. Problem comparison.

7.1. The role of uniformity. Our proof of the lower bound p-Δ_0-TRUTH $\notin$ para-AC^0 (Theorem 1.4) makes crucial use of the uniformity condition in the definition of para-AC^0. To shed some light on this dependence, we relax the uniformity condition as follows.

DEFINITION 7.1. Let (Q,κ) be a parameterized problem and $d, k \in \mathbb{N}$. The *kth slice of* (Q,κ) is the classical problem $\{x \in Q \mid \kappa(x) = k\}$. The class XAC^0 contains (Q,κ) if and only if AC^0 contains every slice of (Q,κ). The class XAC^0_d contains (Q,κ) if and only if AC^0_d contains every slice of (Q,κ); here, AC^0_d denotes the class of problems decided by dlogtime uniform circuit families of polynomial size and depth d.

Clearly,

$$\text{para-}\mathsf{AC}^0 \subseteq \bigcup_{d\in\mathbb{N}} \mathsf{XAC}^0_d \subseteq \mathsf{XAC}^0 \tag{14}$$

and $\mathsf{XAC}^0_0 \not\subseteq$ para-AC^0 since it contains undecidable problems.

LEMMA 7.2. *Assume there is an eventually definable reduction from* (Q,κ) *to* (Q',κ').

(i) *If* $(Q',\kappa') \in \mathsf{XAC}^0$, *then* $(Q,\kappa) \in \mathsf{XAC}^0$.
(ii) *If* $(Q',\kappa') \in \bigcup_d \mathsf{XAC}^0_d$, *then* $(Q,\kappa) \in \bigcup_d \mathsf{XAC}^0_d$.

PROOF. Let r denote the reduction and choose f such that $\kappa' \circ r \leqslant f \circ \kappa$. Choose an interpretation I and a function h witnessing that r is eventually definable. To show (i), assume $(Q',\kappa') \in \mathsf{XAC}^0$. We show that for every $k \in \mathbb{N}$ the kth slice of Q is in AC^0.

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

Fix $k \in \mathbb{N}$ and let $x \in \{0,1\}^*$ with $\kappa(x) = k$. Since κ' is AC^0-computable, Theorem 2.6 implies that for every $k' \in \mathbb{N}$ there is a sentence $\chi_{k'}$ that is true in $\mathcal{S}(r(x))$ if and only if $\kappa'(r(x)) = k'$. For every $k' \in \mathbb{N}$ choose a sentence $\psi_{k'}$ that defines the k'th slice of Q' according to Theorem 2.6. If $|x| \geqslant h(k)$, then

$$\varphi := \bigvee_{k' \leqslant f(k)} (\chi_{k'} \wedge \psi_{k'})^I$$

is true in $\mathcal{S}(x)$ if and only if $r(x) \in Q'$, i.e., $x \in Q$. Translating φ gives an AC^0-family that decides the kth slice of Q on instances x with $|x| \geqslant h(k)$. This can be extended to the whole slice by hardwiring instances of length $< h(k)$.

For (ii) we assume there is $d \in \mathbb{N}$ such that every slice of Q' is in AC^0_d. Now, in Theorem 2.6, the quantifier alternation rank of φ depends only on the depth of the AC^0-family, and vice-versa; this follows from the proof of [6, Theorem 8.1]. In particular, all $\psi_{k'}$ and $\chi_{k'}$ have quantifier alternation rank $\leqslant d'$ for some d' that depends only on d. The depth of the AC^0-family translating the above φ is $\leqslant d''$ for some d'' depending only on d'. The hardwiring of instances of small length can be done by circuits of depth 2. Thus, $(Q, \kappa) \in \mathsf{XAC}^0_{d''}$. ⊣

The class XAC^0 is important in our context because it is a natural upper bound on p-Δ_0-Truth:

Proposition 7.3. p-Δ_0-Truth $\in \mathsf{XAC}^0$.

Proof. It suffices to show that for every Δ_0-formula $\varphi(x)$ the problem $\{1^n \mid \mathbb{N} \models \varphi(n)\}$ belongs to AC^0. But this problem is $un(Q)$ for $Q := \{x \in \{0,1\}^* \mid \mathbb{N} \models \varphi(num(x))\}$. Clearly $Q \in \mathsf{LINH}$, so $un(Q) \in \mathsf{AC}^0$ follows from Proposition 2.1. ⊣

We show that it is likely difficult to improve Theorem 1.4 to p-Δ_0-Truth $\notin \bigcup_{d \in \mathbb{N}} \mathsf{XAC}^0_d$. This somewhat artificial class also exhibits the different behaviors of the parameterized problems p-Halt, p-Halt$_=$, and p-Δ_0-Truth.

Theorem 7.4.

(i) p-Halt $\in \mathsf{XAC}^0_2$.
(ii) p-Halt$_=$ $\in \mathsf{XAC}^0_d$ *for some* $d \in \mathbb{N}$ *if and only if* $\mathsf{NE} \subseteq \mathsf{LINH}$.
(iii) p-Δ_0-Truth $\in \mathsf{XAC}^0_d$ *for some* $d \in \mathbb{N}$ *if and only if* LINH *collapses.*

Proof. (i) For fixed $k \in \mathbb{N}$, let $\mathbb{M}_{k,0}, \dots, \mathbb{M}_{k,\ell_k - 1}$ list all nondeterministic Turing machines of size k and let $n_{k,i}$ be the minimal n such that $\mathbb{M}_{k,i}$ accepts the empty input in n steps; if there is no such n, let $n_{k,i} := \infty$. Then, on instances $(1^n, \mathbb{M})$ with parameter $|\mathbb{M}| = k$, p-Halt is decided by the following family of simple Boolean functions:

$$F_{n,k}(x_0 \dots x_{n-1}, y_0 \dots y_{k-1}) = \bigvee_{\substack{i < \ell_k \text{ such} \\ \text{that } n_{k,i} \leqslant n}} \big(x_0 \dots x_{n-1} = 1^n \wedge y_0 \dots y_{k-1} = \mathbb{M}_{k,i}\big).$$

Observe that $F_{n,k}$ can be understood as a circuit of depth 2 and size $O(k \cdot \ell_k \cdot n)$.

(ii) By Remark 3.10, $\mathsf{NE} \subseteq \mathsf{LINH}$ is equivalent to both p-Halt$_=$ $\in$ para-AC^0 and p-Halt$_=$ $\in \mathsf{XAC}^0$. By (14) it is equivalent to p-Halt$_=$ $\in \bigcup_d \mathsf{XAC}^0_d$.

To see (iii), assume LINH collapses. Paris and Dimitracopolous [34, proof of Proposition 4] showed that this implies the following. There is an $L^{\mathrm{r}}_{\mathrm{ar}}$-formula

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

$\lambda(x, y)$ such that for every Δ_0-formula $\varphi(x)$ there are $c_\varphi, d_\varphi, e_\varphi \in \mathbb{N}$ such that for all $n \geqslant c_\varphi$

$$\mathbb{N} \models \varphi(n) \iff n^{d_\varphi} \models \lambda(n, e_\varphi).$$

For each fixed φ there is an AC^0-family that given 1^n decides whether n satisfies the r.h.s. The size of this family is bounded by n^{f_φ} for some $f_\varphi \in \mathbb{N}$ depending on φ, but the depth of this family is determined by the quantifier alternation rank of λ and, in particular, does not depend on φ. This implies p-Δ_0-Truth $\in \mathsf{XAC}^0_d$ for some $d \in \mathbb{N}$.

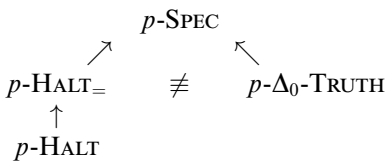
Conversely, assume p-Δ_0-Truth $\in \mathsf{XAC}^0_d$ and let $Q \in \mathsf{LINH}$. It is well known (see, e.g., [25, Chapter V, Lemma 2.13]) that there is a Δ_0-formula that is satisfied by $num(x)$ if and only if $x \in Q$. Fixing this formula in the input to p-Δ_0-Truth, the assumption implies that there is a dlogtime uniform circuit family $(C_n)_n$ of polynomial size and depth d such that for all $x \in \{0,1\}^*$:

$$x \in Q \iff C_{num(x)}(1^{num(x)}) = 1.$$

It suffices to show that, given x, the r.h.s. can be checked by an alternating machine in linear time with d alternations. This is straightforward by guessing a path through $C_{num(x)}$. For example, if the output gate is a $\vee$-gate, the machine existentially guesses an input gate g_1 to it, and if it is a $\wedge$-gate it universally guesses g_1. Depending on the type of g_1 it either existentially or universally guesses an input gate g_2 to g_1, and so on. When reaching (with g_{d-1} or earlier) an input gate or a negation thereof, the machine accepts or rejects, respectively. Each guess requires $O(|x|)$ bits. Checking that, e.g., g_2 is an input to g_1 can be done in time logarithmic in the size of $C_{num(x)}$, that is, in time $O(|x|)$. We omit further details. ⊣

7.2. Reducibilities. On the one hand p-Halt$_=$ might appear 'easier' than p-Δ_0-Truth in that the latter is not in para-AC^0 while this is unknown for the former. On the other hand, Theorem 7.4 might indicate that p-Halt$_=$ is 'harder' than p-Δ_0-Truth. Also recall from the introduction that p-Halt$_=$ is trivially in para-NP but not known to be solvable in time $n^{f(k)}$ while for p-Δ_0-Truth it is the other way around. The problems seem incomparable. In this subsection we verify this intuition for our notion of reducibility.

Saying that a (parameterized) problem is *reducible* to another means that there is an eventually definable reduction. Two problems are *equivalent* if they are reducible to one another. The picture is as follows: an arrow indicates reducibility, $\equiv$ means equivalence.



In particular, we show unconditionally that p-Halt$_=$ and p-Δ_0-Truth are not equivalent and both are reducible to yet another almost tally problem of central importance to mathematical logic, namely the following parameterized version of the spectrum problem:

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

p-SPEC	
Instance:	$n \in \mathbb{N}$ in unary and a first-order sentence φ.
Parameter:	$\lvert\varphi\rvert$.
Problem:	Does φ have a model of size n?

Recall that having a model of size n means that n belongs to the spectrum of φ.

We start comparing p-HALT and p-HALT$_=$. By Example 3.6, p-HALT is reducible to p-HALT$_=$. Concerning the converse we have the following.

COROLLARY 7.5. *If* p-HALT$_=$ *is reducible to* p-HALT, *then* $\mathsf{NE} \subseteq \mathsf{LINH}$.

PROOF. By Theorem 7.4(i), p-HALT $\in \bigcup_d \mathsf{XAC}^0_d$. If p-HALT$_=$ is reducible to p-HALT, then p-HALT$_= \in \bigcup_d \mathsf{XAC}^0_d$ by Lemma 7.2(ii). This implies $\mathsf{NE} \subseteq \mathsf{LINH}$ by Theorem 7.4(ii). ⊣

Adapting a mode of speech from [8], call an almost tally problem (Q, κ) *slicewise monotone* if $(1^n, x) \in Q$ implies $(1^m, x) \in Q$ for all $x \in \{0,1\}^*$ and all $n, m \in \mathbb{N}$ with $n < m$. One can show that p-HALT is the hardest such problem in para-NP. This is an easy modification of the proof of Lemma 3.12 and strengthens [8, Proposition 11]:

COROLLARY 7.6. *Every almost tally problem in* para-NP *that is slicewise monotone is reducible to* p-HALT.

We turn to p-HALT$_=$ and p-Δ_0-TRUTH.

COROLLARY 7.7.

(i) *If* p-Δ_0-TRUTH *is reducible to* p-HALT$_=$, *then* $\mathsf{NE} \not\subseteq \mathsf{LINH}$.
(ii) *If* p-HALT$_=$ *is reducible to* p-Δ_0-TRUTH, *then* $\mathsf{NE} \subseteq \mathsf{LINH}$.
(iii) p-Δ_0-TRUTH *and* p-HALT$_=$ *are not equivalent*.

PROOF. (iii) follows from (i) and (ii). For (i), assume p-Δ_0-TRUTH is reducible to p-HALT$_=$. Then p-Δ_0-TRUTH $\in$ para-NP and $\mathsf{NE} \not\subseteq \mathsf{LINH}$ follows by Theorem 1.3.

For (ii), assume p-HALT$_=$ is reducible to p-Δ_0-TRUTH. Then p-HALT$_= \in \mathsf{XAC}^0$ by Proposition 7.3 and Lemma 7.2(i). This implies $\mathsf{NE} \subseteq \mathsf{LINH}$ by Remark 3.10. ⊣

Finally, we turn to p-SPEC:

PROPOSITION 7.8. *Both* p-HALT$_=$ *and* p-Δ_0-TRUTH *are reducible to* p-SPEC.

PROOF. It is straightforward to compute from a nondeterministic Turing machine $\mathbb{M}$ a first-order sentence $\varphi_{\mathbb{M}}$ that has a model of size n if and only if $\mathbb{M}$ accepts the empty input in exactly n steps.

Concerning p-Δ_0-TRUTH, by Lemma 4.1, it suffices to show that p-MC$(L^{\mathrm{r}}_{\mathrm{ar}})$ is reducible to p-SPEC: map an instance $(1^n, \varphi)$ of p-MC$(L^{\mathrm{r}}_{\mathrm{ar}})$ to $(1^n, \varphi \wedge \psi)$ where ψ is an $L^{\mathrm{r}}_{\mathrm{ar}}$-sentence whose finite models are exactly those isomorphic to some standard finite $L^{\mathrm{r}}_{\mathrm{ar}}$-structure. ⊣

Observe p-SPEC can be solved in nondeterministic time $n^{f(k)}$ for some computable $f : \mathbb{N} \to \mathbb{N}$ where $k := \lvert\varphi\rvert$ is the parameter. Can the parameter be moved out of the exponent? We find it worthwhile to explicitly point out the following direct corollary of the previous proposition and Theorem 1.3:

COROLLARY 7.9. *If* p-SPEC $\in$ para-NP, *then* $\mathsf{NE} \not\subseteq \mathsf{LINH}$.

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press

Acknowledgements. We thank the anonymous referees for their detailed comments. A partial conference version of this article appeared as [13].

Funding. Yijia Chen is supported by the National Natural Science Foundation of China (Project 62372291). Keita Yokoyama is partially supported by JSPS KAKENHI grant numbers JP19K03601, JP21KK0045, and JP23K03193.

REFERENCES

[1] E. Allender, R. Beigel, U. Hertrampf, and S. Homer, *Almost-everywhere complexity hierarchies for nondeterministic time*. ***Theoretical Computer Science***, vol. 115 (1993), no. 2, pp. 225–241.

[2] E. Allender and V. Gore, *On strong separations from* AC^0, ***Advances in Computational Complexity Theory*** (Jin-Yi Cai, editor), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 13, DIMACS/AMS, 1990, pp. 21–38.

[3] S. Arora and B. Barak, ***Computational Complexity–A Modern Approach***, Cambridge University Press, Cambridge, 2009.

[4] Y. Aumann and Y. Dombb, *Fixed structure complexity*, ***3rd International Workshop on Parameterized and Exact Computation (IWPEC'08)***, LNCS 5018, Springer, Berlin Heidelberg, New York, 2008, pp. 30–42.

[5] J. L. Balcázar and U. Schöning, *Bi-immune sets for complexity classes*. ***Mathematical Systems Theory***, vol. 18 (1985), no. 1, pp. 1–10.

[6] D. A. M. Barrington, N. Immerman, and H. Straubing, *On uniformity within* NC^1. ***Journal of Computer and System Sciences***, vol. 41 (1990), no. 3, pp. 274–306.

[7] Y. Chen and J. Flum, *A logic for PTIME and a parameterized halting problem*, ***Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, (LICS'09)***, IEEE Computer Society, Los Alamitos, 2009, pp. 397–406.

[8] ———, *On slicewise monotone parameterized problems and optimal proof systems for TAUT*, ***Proceedings of the 24th International Workshop Computer Science Logic (CSL'10)***, LNCS 6247, Springer, Berlin Heidelberg, New York 2010, pp. 200–214.

[9] ———, *On the complexity of Gödel's proof predicate*. ***The Journal of Symbolic Logic***, vol. 75 (2010), no. 1, pp. 239–254.

[10] ———, *From almost optimal algorithms to logics for complexity classes via listings and a halting problem*. ***Journal of the ACM***, vol. 59 (2012), no. 4, pp. 1–34.

[11] ———, *Some lower bounds in parameterized* AC^0. ***Information and Computation***, vol. 267 (2019), pp. 116–134.

[12] Y. Chen, J. Flum, and M. Müller, *A surprising relationship between descriptive complexity and proof complexity*. ***Bulletin of the EATCS, the Logic in Computer Science Column by Yuri Gurevich***, vol. 138 (2022), p. 2022.

[13] Y. Chen, M. Müller, and K. Yokoyama, *A parameterized halting problem, the linear time hierarchy, and the MRDP theorem*, ***Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'18)***, ACM, New York, 2018, pp. 235–244.

[14] M. Davis, *Hilbert's tenth problem is unsolvable*. ***The American Mathematical Monthly***, vol. 80 (1973), no. 3, pp. 233–269.

[15] R. G. Downey and M. R. Fellows, ***Parameterized Complexity***, Monographs in Computer Science, Springer, Berlin Heidelberg, New York, 1999.

[16] ———, ***Fundamentals of Parameterized Complexity***, Texts in Computer Science, Springer, 2013.

[17] H.-D. Ebbinghaus and J. Flum, ***Finite Model Theory***, Perspectives in Mathematical Logic, Springer, Berlin Heidelberg, New York, 1995.

[18] M. Elberfeld, C. Stockhusen, and T. Tantau, *On the space and circuit complexity of parameterized problems: Classes and completeness*. ***Algorithmica***, vol. 71 (2015), no. 3, pp. 661–701.

[19] J. Flum and M. Grohe, *Describing parameterized complexity classes*. ***Information and Computation***, vol. 187 (2003), no. 2, pp. 291–319.

[20] ———, ***Parameterized Complexity Theory***, Texts in Theoretical Computer Science. An EATCS Series, Springer, Berlin Heidelberg, New York, 2006.

[21] L. Fortnow and R. Santhanam, *New non-uniform lower bounds for uniform classes*, ***Proceedings of the 31st Conference on Computational Complexity (CCC'16)***, *LIPIcs*, 50, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2016, pp. 19:1–19:14.

[22] H. Gaifman and C. Dimitracopoulos, *Fragments of arithmetic and the MRDP theorem*, ***Logic and Algorithmic***, Monographie de L'Enseignement Mathematique, 30, Université de Genève, Geneva, 1982, pp. 187–206.

[23] J. G. Geske, D. T. Huynh, and J. I. Seiferas, *A note on almost-everywhere-complex sets and separating deterministic-time-complexity classes*. ***Information and Computation***, vol. 92 (1991), no. 1, pp. 97–104.

[24] Y. Gurevich, *Logic and the challenge of computer science*, ***Current Trends in Theoretical Computer Science***, (E. Börger, editor), Computer Science Press, New York, 1988, pp. 1–57.

[25] P. Hájek and P. Pudlák, ***Metamathematics of First-Order Arithmetic***, Perspectives in Mathematical Logic, Oxford University Press, Oxford, 1998. Second printing.

[26] N. Immerman, ***Descriptive Complexity***. Graduate Texts in Computer Science, Springer, Berlin Heidelberg, New York, 1999.

[27] R. Kaye, *Diophantine induction*. ***Annals of Pure and Applied Logic***, vol. 46 (1990), pp. 1–40.

[28] ———, ***Models of Peano Arithmetic***, Oxford Logic Guides, Springer, Berlin, 1991.

[29] J. Krajíček, *Exponentiation and second order bounded arithmetic*. ***Annals of Pure and Applied Logic***, vol. 48 (1990), no. 3, pp. 261–276.

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

[31] E. Mayordomo, *Almost every set in exponential time is p-bi-immune*. ***Theoretical Computer Science***, vol. 136 (1994), no. 2, pp. 487–506.

[32] C. H. Papadimitriou, ***Computational Complexity***, Addison-Wesley, Boston, 1994.

[33] R. Parikh, *Existence and feasibility in arithmetic*. ***The Journal of Symbolic Logic***, vol. 36 (1971), pp. 494–508.

[34] J. B. Paris and C. Dimitracopoulos, *Truth definitions for Δ_0 formulae*, ***Logic and Algorithmic***, Monographie de L'Enseignement Mathematique, 30, Université de Genève, Geneva, 1982, pp. 317–329.

[35] N. Schweikardt, *Arithmetic, first-order logic, and counting quantifiers*. ***ACM Transactions on Computational Logic***, vol. 6 (2005), no. 3, pp. 634–671.

[36] A. J. Wilkie, *Applications of complexity theory to Σ_0-definability problems in arithmetic*, ***Model Theory of Algebra and Arithmetic***, (L. Pacholski and J. Wierzejewski, editors), Lecture Notes in Mathematics 834, Springer, Berlin Heidelberg, New York, 1980, pp. 363–369.

[37] M. Zimand, *Large sets in* AC^0 *have many strings with low Kolmogorov complexity*. ***Information Processing Letters***, vol. 62 (1997), no. 3, pp. 165–170.

DEPARTMENT OF COMPUTER SCIENCE
SHANGHAI JIAO TONG UNIVERSITY
SHANGHAI, CHINA
E-mail: yijia.chen@cs.sjtu.edu.cn

FACULTY OF COMPUTER SCIENCE AND MATHEMATICS
UNIVERSITY OF PASSAU
PASSAU, GERMANY
E-mail: moritz.mueller@uni-passau.de

MATHEMATICAL INSTITUTE
TOHOKU UNIVERSITY
SENDAI, JAPAN
E-mail: keita.yokoyama.c2@tohoku.ac.jp

https://doi.org/10.1017/jsl.2024.44 Published online by Cambridge University Press