

# A PARAMETERIZED HALTING PROBLEM, $\Delta_0$ 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  $\Delta_0$ -formula  $\varphi(x)$ ; the parameter is the size of  $\varphi$ . This parameterization focusses attention on instances where  $n$  is large compared to the size of  $\varphi$ . 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.<sup>1</sup> 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  $g(k) \leq n$  for some computable  $g : \mathbb{N} \rightarrow \mathbb{N}$ , or equivalently, an algorithm correct on all inputs with runtime  $f(k) \cdot n^{O(1)}$  for some computable  $f : \mathbb{N} \rightarrow \mathbb{N}$ . If “efficient” means  $AC^0$ , it leads to the class  $\text{para-}AC^0$ . 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:*  $|\mathbb{M}|$ , 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.

<sup>1</sup>All required definitions will be given precisely in Section 2.

© The Author(s), 2024. Published by Cambridge University Press on behalf of The Association for Symbolic Logic.  
 0022-4812/00/0000-0000  
 DOI:10.1017/jsl.2024.44



$p$ - $\Delta_0$ -TRUTH

*Instance:*  $n \in \mathbb{N}$  in unary and a  $\Delta_0$ -formula  $\varphi(x)$ .

*Parameter:*  $|\varphi|$ , the size of  $\varphi$ .

*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  $f : \mathbb{N} \rightarrow \mathbb{N}$  is any function.<sup>2</sup> 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<sup>0</sup>.

This conjecture is highly plausible and might appear to be within reach because AC<sup>0</sup> is well-understood and, in particular, [11] establishes (unconditional) para-AC<sup>0</sup> lower bounds for many well-studied parameterized problems. It deserves some genuine interest because its failure implies that AC<sup>0</sup>, 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<sub>=</sub>

*Instance:*  $n \in \mathbb{N}$  in unary and a nondeterministic Turing machine  $\mathbb{M}$ .

*Parameter:*  $|\mathbb{M}|$ .

*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<sub>=</sub> 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<sub>=</sub> 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:

<sup>2</sup>[12] gives a direct proof of the equivalence of the first two hypotheses.

THEOREM 1.2.

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

**1.3.  $\Delta_0$  truth.** Deciding the truth of  $\Delta_0$  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 Dimitracopoulos [34] took  $n$  to be nonstandard and related the complexity of truth definitions for  $\Delta_0$ -formulas to the complexity-theoretic hypotheses that  $\text{LINH}$  or  $\text{PH}$  does not collapse. Wilkie proved a weak version of the former hypothesis by showing that  $p\text{-}\Delta_0\text{-TRUTH}$  restricted to quantifier-free formula inputs can be decided in space  $f(k) + O(\log n)$  where  $k := |\varphi|$  is the parameter and  $f : \mathbb{N} \rightarrow \mathbb{N}$  a computable function [36, proof of Lemma 3.1]. The straightforward algorithm decides  $p\text{-}\Delta_0\text{-TRUTH}$  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  $\text{FPT}$ ? Maybe with nondeterminism, i.e., is it in  $\text{para-NP}$ ?

Our main result (Theorem 4.3) shows that such *upper bounds* on the parameterized complexity of  $p\text{-}\Delta_0\text{-TRUTH}$  imply *lower bounds* in classical complexity theory. Notably,

THEOREM 1.3. *If  $p\text{-}\Delta_0\text{-TRUTH} \in \text{para-NP}$ , then  $\text{NE} \not\subseteq \text{LINH}$ .*

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

THEOREM 1.4.  *$p\text{-}\Delta_0\text{-TRUTH} \notin \text{para-AC}^0$ .*

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)  $\text{AC}^0$  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  $\Sigma_1$ -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  $\Delta_0$ -formulas.

Wilkie observed [36] that a positive answer would imply  $\text{NP} = \text{co-NP}$ . 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  $\Delta_0$ -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].

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  $p\text{-HALT} \notin \text{para-}AC^0$ .*

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.<sup>3</sup> 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  $p\text{-HALT}$  to  $p\text{-HALT}_=$  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  $p\text{-HALT}$ ,  $p\text{-HALT}_=$  and  $p\text{-}\Delta_0\text{-TRUTH}$ .

**§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  $O(1)$

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

alternations. LINSPEC and NLINSPEC denote deterministic and nondeterministic linear space  $O(n)$ . Clearly,

$$\text{LINH} \subseteq \text{LINSPEC} \subseteq \text{NLINSPEC} \subseteq \text{E} \subseteq \text{NE}.$$

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

- $C_n$  is a circuit (with  $\wedge, \vee, \neg$  gates and unbounded fan-in) with  $n$  variables, size  $\leq n^c$  and depth<sup>4</sup>  $\leq d$ , 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  $i$ th 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

$$\langle x, y \rangle := x_0x_0 \dots x_{|x|-1}x_{|x|-1}01y_0y_0 \dots y_{|y|-1}y_{|y|-1}, \tag{1}$$

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

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

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

**2.2. Parameterized complexity.** A *parameterized problem* is a pair  $(Q, \kappa)$  of an underlying classical problem  $Q \subseteq \{0, 1\}^*$  and a *parameterization*  $\kappa : \{0, 1\}^* \rightarrow \mathbb{N}$  mapping an instance  $x \in \{0, 1\}^*$  to its *parameter*  $\kappa(x) \in \mathbb{N}$ . We follow [18] and require that  $\kappa$  is computable by an  $\text{AC}^0$ -circuit family  $(C_n)_{n \in \mathbb{N}}$ . That is, for all  $x \in \{0, 1\}^*$ , besides  $|x|$  inputs the circuit  $C_{|x|}$  has  $|x|$  outputs and computes  $\text{bin}(\kappa(x))$ , 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  $\{ \langle 1^n, \mathbb{M} \rangle \mid \text{the nondeterministic Turing machine } \mathbb{M} \text{ accepts the empty input in at most } n \text{ steps} \}$  and a parameterization  $\kappa$  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-P = FPT contains the parameterized problems  $(Q, \kappa)$  that are *fixed-parameter tractable*, i.e., decidable in deterministic time  $f(\kappa(x)) \cdot |x|^{O(1)}$

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

for some computable  $f : \mathbb{N} \rightarrow \mathbb{N}$ . 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<sup>0</sup>. It is characterized as follows:

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

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

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

- *there are a computable  $g : \mathbb{N} \rightarrow \mathbb{N}$  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  $i$ th bit of the binary encoding of  $C_{n,k}$  is  $b$ .*
- (iii)  *$Q$  is decidable and there are a computable  $h : \mathbb{N} \rightarrow \mathbb{N}$  and an AC<sup>0</sup>-circuit family  $(C_n)_{n \in \mathbb{N}}$  such that for all  $x \in \{0, 1\}^*$  with  $|x| \geq h(\kappa(x))$  we have*

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

According to the terminology of [19], (iii) states that  $(Q, \kappa)$  is eventually in AC<sup>0</sup>.

**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  $L_{\text{ar}}$ -structure*, denoted  $\mathbb{N}$ , has universe  $\mathbb{N}$  and interprets the symbols in the obvious way. Every  $L_{\text{ar}}$ -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  $\varphi$  from their binary encodings, so  $|p|$  and  $|\varphi|$  denote the lengths of these encodings. Writing  $\varphi(\bar{x})$  for a formula  $\varphi$  means that *all* free variables of  $\varphi$  are among  $\bar{x}$ . A *sentence* is a formula without free variables.

A  $\Delta_0$ -*formula* is an  $L_{\text{ar}}$ -formula obtained from atomic formulas, Boolean connectives, and bounded quantifiers  $\exists x < p, \forall x < p$  where  $p$  is an  $L_{\text{ar}}$ -term not involving  $x$ ; e.g.,  $\exists x < p \varphi$  stands for  $\exists x (x < p \wedge \varphi)$ .  $\Sigma_1$ - and  $\Pi_1$ -*formulas* are obtained from  $\Delta_0$ -formulas by existential and universal quantification, respectively.

**THEOREM 2.3 (MRDP).** *For every  $\Delta_0$ -formula  $\varphi(\bar{x})$  there are  $L_{\text{ar}}$ -terms  $p(\bar{x}, \bar{y}), q(\bar{x}, \bar{y})$  such that*

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

Gödel showed that computable functions are  $\Sigma_1$ -definable. The MRDP theorem improves this to an existential definition:

**COROLLARY 2.4.** *For every computable  $f : \mathbb{N} \rightarrow \mathbb{N}$  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  $n \geq 2$ , and therefore consider the relational version

$$L_{ar}^r$$

of  $L_{ar}$  where  $+$ ,  $\times$  are ternary relation symbols. The standard  $L_{ar}^r$ -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  $L_{ar}^r$ -structure with universe  $[n]$ , simply denoted  $n$ , is the substructure of  $\mathbb{N}$  with universe  $[n]$ , i.e., it interprets the symbols in  $L_{ar}^r$  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  $\varphi$  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  $L_{ar}^r$ -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  $L_{ar}^r$ -formula  $\varphi_f(\bar{x}, y, \bar{z})$ . Indeed, it is straightforward to express an  $L_{ar}$ -term (in)equality by an existential  $L_{ar}^r$ -formula.

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

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

i.e.,  $\mathcal{S}(x) = ([n], +^{[n]}, \times^{[n]}, 0^{[n]}, 1^{[n]}, <^{[n]}, ONE^x)$ . 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  $L_{ar}^r \cup \{ONE\}$ -sentence  $\varphi$  such that for every  $x \in \{0, 1\}^*$  with  $|x| \geq 2$ :*

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

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

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

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



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  $w \geq 1$ . 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  $L_{ar}^r$ -structure  $n^d$  is interpretable in the standard  $L_{ar}^r$ -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  $L_{ar}^r$ -formula  $BIT(x, y)$  such that for every  $n > 1$  and all  $i, j \in [n]$ :*

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

**§3.  $p$ -Halt and NE versus LINH.** In this section we first introduce a workable notion of reduction that preserves para-AC<sup>0</sup>, then prove Theorem 1.2, then introduce almost tally problems and show  $p$ -HALT<sub>=</sub> 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  $(Q, \kappa)$  to another  $(Q', \kappa')$  is a reduction  $r : \{0, 1\}^* \rightarrow \{0, 1\}^*$  from  $Q$  to  $Q'$  such that  $\kappa' \circ r \leq f \circ \kappa$  for some computable function  $f : \mathbb{N} \rightarrow \mathbb{N}$ .

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

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

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



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  $\kappa$ -eventually definable where  $\kappa$  maps  $\langle 1^n, x \rangle$  to  $|x|$  (both functions map arguments that are not of the required form to, say, 0).

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

Let  $(n, num(x))$  be the expansion of the standard  $L_{ar}^r$ -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  $\varphi$  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.  $\dashv$

EXAMPLE 3.3. Let  $P : \{0, 1\}^* \rightarrow \{0, 1\}^*$  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  $\kappa$ -eventually definable where  $\kappa$  maps  $\langle 1^n, x \rangle$  to  $|x|$  (both functions map arguments that are not of the required form to, say, 0).

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

- a quantifier-free  $L_{ar}^r$ -formula  $\varphi(x, y, \bar{z})$  according to Remark 2.5,
- a computable  $f : \mathbb{N} \rightarrow \mathbb{N}$  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  $h : \mathbb{N} \rightarrow \mathbb{N}$  such that  $h(|x|) > num(x), num(P(x)), f(num(x))$  for all  $x \in \{0, 1\}^*$ .

Assume  $n \geq h(|x|)$ . Then  $\mathcal{S}(\langle 1^n, x \rangle)$  interprets the expansion  $(n, \ell)$  of the standard structure  $n$  by a constant  $c$  interpreting  $\ell := num(x)$ . In  $(n, \ell)$  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.  $\dashv$

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

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

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  $|r(x)| \geq |x|^{1/c}$  for all  $x \in \{0, 1\}^*$  with  $|x| \geq n_0$ . Define  $g : \mathbb{N} \rightarrow \mathbb{N}$  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  $\kappa$ -eventually definable. To verify this let  $x \in \{0, 1\}^*$  satisfy  $|x| \geq g(k)$  where  $k := \kappa(x)$ . Then  $|r(x)| \geq |x|^{1/c} \geq h'(f(k)) \geq h'(\kappa'(r(x)))$  using that  $h'$  is nondecreasing. Hence  $\mathcal{S}(r(x))^{I'} \cong \mathcal{S}(r'(r(x)))$ . As  $|x| \geq g(k) \geq h(k)$ , we conclude  $\mathcal{S}(x)^I \cong \mathcal{S}(r(x))$ , which implies  $\mathcal{S}(x)^{I' \circ I} \cong \mathcal{S}(r'(r(x)))$ .  $\dashv$

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

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

PROOF. Let  $P : \{0, 1\}^* \rightarrow \{0, 1\}^*$  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  $|\mathbb{M}'| \geq |\mathbb{M}|$ ) and thus gives a reduction as desired.  $\dashv$

REMARK 3.7. A parameterized problem  $(Q, \kappa)$  is in  $\text{para-AC}^0$  if and only if  $Q$  is decidable and there is an eventually definable reduction from  $(Q, \kappa)$  to a trivial problem, say,  $(Q_0, \kappa_0)$  for  $Q_0$  the set of strings starting with 0 and  $\kappa_0$  is constantly 0.

It is straightforward to check that this reducibility is transitive and preserves membership in  $\text{para-AC}^0$ :

LEMMA 3.8. Assume that there is an eventually definable reduction from  $(Q, \kappa)$  to  $(Q', \kappa')$ .

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

PROOF. (i) follows from Lemma 3.4. (ii) follows from (i) and Remark 3.7.  $\dashv$

**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 $_{=} \in \text{FPT}$  if and only if  $\text{NE} \subseteq \text{E}$ .

Theorem 1.2(i) is a  $\text{para-AC}^0$ -analogue of this result.

THEOREM 1.2.

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

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

To prove (i), first assume  $\text{NE} \subseteq \text{LINH}$  and let  $Q$  be the classical problem underlying  $p\text{-HALT}_=$  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,  $Q \in \text{NE}$ , so by assumption and Proposition 2.1(iii) we have  $un(Q) \in \text{AC}^0$ . Recall

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

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  $p\text{-HALT}_=$ . It is an honest parameterized reduction to  $(un(Q), \kappa)$  where  $\kappa$  maps  $1^{num(\langle bin(n), \mathbb{M} \rangle)}$  to  $|\mathbb{M}|$ . Since  $(un(Q), \kappa) \in \text{para-AC}^0$ , Lemma 3.8 implies  $p\text{-HALT}_= \in \text{para-AC}^0$ .

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

As  $Q \in \text{NE}$  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,  $p\text{-HALT}_= \in \text{para-AC}^0$  implies that the classical problem

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

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

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

$$Q \subseteq \{ \langle 1^n, x \rangle \mid n \in \mathbb{N}, x \in \{0, 1\}^* \}$$

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

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

Theorem 1.2(ii) holds not only for  $p\text{-HALT}$  but for every almost tally problem in para-NP. In fact,  $p\text{-HALT}_=$  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  $p\text{-HALT}_=$ .*

PROOF. Let  $(Q, \kappa) \in \text{para-NP}$  be almost tally. The identity is a parameterized reduction from  $(Q, \kappa)$  to its re-parameterization  $(Q, \kappa')$  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  $c \in \mathbb{N}, f : \mathbb{N} \rightarrow \mathbb{N}$  is a computable function, and  $k := |x|$ .

Define  $g : \mathbb{N}^2 \rightarrow \mathbb{N}$  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.

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

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

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  $\text{NE} \subseteq \text{LINH}$  if and only if every tally problem in NP is in  $\text{AC}^0$ . 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  $p\text{-HALT}_=$ :

**COROLLARY 3.13.** *NE  $\subseteq$  LINH if and only if every almost tally problem in para-NP is in para- $\text{AC}^0$ .*

**PROOF.** The l.h.s. is equivalent to  $p\text{-HALT}_= \in \text{para-AC}^0$  by Theorem 1.2(i). And by Lemmas 3.8 and 3.12,  $p\text{-HALT}_= \in \text{para-AC}^0$  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- $\text{AC}^0$  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)  $\text{E} \subseteq \text{LINH}$  if and only if every almost tally problem in FPT is in para- $\text{AC}^0$ .
- (ii)  $\text{NLINSPACE} \subseteq \text{LINH}$  if and only if every almost tally problem in para-NL is in para- $\text{AC}^0$ .
- (iii)  $\text{LINSPEC} \subseteq \text{LINH}$  if and only if every almost tally problem in para-L is in para- $\text{AC}^0$ .

**PROOF.** The proof of (i) is analogous to the proof of Corollary 3.13 using the subproblem of  $p\text{-HALT}_=$  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  $p\text{-HALT}_=$ :

$p\text{-HALT}_=^*$

*Instance:*  $n, m \in \mathbb{N}$  in unary with  $n \leq m$  and a nondeterministic Turing machine  $\mathbb{M}$ .

*Parameter:*  $|\mathbb{M}|$ .

*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  $\sum_i c_i \leq \lfloor \log m \rfloor$ .

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

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

PROOF OF CLAIM 1. Assume  $\text{NLINSPACE} \subseteq \text{LINH}$  and let  $Q$  be the classical problem underlying  $p\text{-HALT}_{\leq}^*$  but with the inputs  $n, m$  encoded in binary. Clearly,  $Q \in \text{NLINSPACE} \subseteq \text{LINH}$ , so  $un(Q) \in \text{AC}^0$  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  $p\text{-HALT}_{\leq}^* \in \text{para-AC}^0$  follows as in Theorem 1.2(i).

Conversely, assume  $p\text{-HALT}_{\leq}^* \in \text{para-AC}^0$  and let  $Q \in \text{NLINSPACE}$ . 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  $d \geq c$ .

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

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

Our assumption  $p\text{-HALT}_{\leq}^* \in \text{para-AC}^0$  implies that the classical problem

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

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

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

PROOF OF CLAIM 2. Let  $(Q, \kappa) \in \text{para-NL}$  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  $c \in \mathbb{N}$ ,  $f : \mathbb{N} \rightarrow \mathbb{N}$  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

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

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

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

It now suffices to show that  $p\text{-HALT}_{=}^* \in \text{para-AC}^0$  if and only if every almost tally problem in  $\text{para-NL}$  is in  $\text{para-AC}^0$ . The forward direction follows from Claim 2 and Lemma 3.8. And if  $p\text{-HALT}_{=}^* \notin \text{para-AC}^0$ , then we get an almost tally problem in  $\text{para-NL} \setminus \text{para-AC}^0$  by rewriting inputs  $\langle 1^n, 1^m, \mathbb{M} \rangle$  of  $p\text{-HALT}_{=}^*$  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  $p\text{-DHALT} \notin \text{para-AC}^0$ , then  $E \not\subseteq \text{LINH}$ .*

*p*-DHALT

*Instance:*  $n \in \mathbb{N}$  in unary and a deterministic Turing machine  $\mathbb{M}$ .

*Parameter:*  $|\mathbb{M}|$ .

*Problem:* Does  $\mathbb{M}$  accept the empty input in at most  $n$  steps?

**§4. On the parameterized complexity of  $p\text{-}\Delta_0\text{-Truth}$ .** Recall, the problem  $p\text{-}\Delta_0\text{-TRUTH}$  asks whether a given  $n \in \mathbb{N}$  in unary satisfies a given  $\Delta_0$ -formula  $\varphi(x)$ , parameterized by the length of  $\varphi$ . Further recall that  $\Delta_0$  refers to the language  $L_{\text{ar}}$  with function symbols  $+$ ,  $\cdot$  and contains the  $L_{\text{ar}}$ -formulas with quantifiers bounded by  $L_{\text{ar}}$ -terms.

This section first observes that  $p\text{-}\Delta_0\text{-TRUTH}$  is “the same” as a basic parameterized model-checking problem, uses this to prove the lower bound  $p\text{-}\Delta_0\text{-TRUTH} \notin \text{para-AC}^0$  (Theorem 1.4), and finally, based on the previous section, infers consequences from *upper bounds* on the parameterized complexity of  $p\text{-}\Delta_0\text{-TRUTH}$ , including Theorem 1.3.

**4.1. Model-checking arithmetic.** Recall  $L_{\text{ar}}^r$  is the relational version of the language of arithmetic  $L_{\text{ar}}$ . We observe that  $p\text{-}\Delta_0\text{-TRUTH}$  is “the same” as the parameterized model-checking problem for first-order logic over finite standard  $L_{\text{ar}}^r$ -structures:



<p><math>p</math>-MC(<math>L_{AR}^R</math>)</p> <p><i>Instance:</i> <math>n &gt; 1</math> in unary and an <math>L_{ar}^r</math>-sentence <math>\varphi</math>.</p> <p><i>Parameter:</i> <math> \varphi </math>.</p> <p><i>Problem:</i> <math>n \models \varphi?</math></p>
--

LEMMA 4.1. *There is a computable function that maps every  $\Delta_0$ -formula  $\varphi(x)$  to an  $L_{ar}^r$ -sentence  $\psi$  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  $L_{ar}^r$ -sentence  $\psi$  to a  $\Delta_0$ -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  $\Delta_0$ -formula  $\varphi(\bar{x})$ , a constant  $c_\varphi > 1$  and an  $L_{ar}^r$ -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  $m \geq \max\{\bar{n}, 2\}^{c_\varphi}$ . 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  $L_{ar}^r$ -structure  $n^{c_\varphi}$  (as the minimal element whose  $c_\varphi$ th power does not exist), we can replace  $\psi_0(n)$  by some sentence  $\psi_1$ . Then set  $\psi := \psi_1^{I_{c_\varphi}}$  for the interpretation  $I_{c_\varphi}$  from Lemma 2.9. ⊣

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

THEOREM 1.4.  $p$ - $\Delta_0$ -TRUTH  $\not\leq$  para-AC<sup>0</sup>.

We fix a proper elementary extension  $M$  of the standard  $L_{ar}^r$ -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  $f : \mathbb{N} \rightarrow \mathbb{N}$  be a computable function. Then there is an  $L_{ar}^r$ -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,  $\varphi_f$  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  $L_{ar}^r$ -formula “ $x=n$ ” by “ $x=0$ ” :=  $x=0$  and “ $x=(n+1)$ ” :=  $\exists y (“y=n” \wedge +(y, 1, x))$ . For an  $L_{ar}^r$ -formula  $\varphi(y, \bar{x})$  set  $\varphi(\underline{n}, \bar{x}) :=$

$\exists y("y=n" \wedge \varphi(y, \bar{x}))$ ; we understand  $\varphi^{<z}(\underline{n}, \bar{x})$  as  $(\varphi(\underline{n}, \bar{x}))^{<z}$ . If  $n < m$ , then both  $("x=n")^{<m}$  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  $n \in \mathbb{N}$ :

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

**PROOF OF THEOREM 1.4.** For contradiction, assume otherwise, so  $p\text{-MC}(L_{\text{ar}}^r) \in \text{para-AC}^0$  by Lemma 4.1. By Corollary 2.7, there is an increasing computable function  $h : \mathbb{N} \rightarrow \mathbb{N}$  and a sentence  $\text{sat}$  such that for every  $n \in \mathbb{N}$  and every  $L_{\text{ar}}^r$ -sentence  $\varphi$  with  $n > h(\text{num}(\varphi))$  we have

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

⊥

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

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

where  $\mathbb{N}$  is the standard  $L_{\text{ar}}^r$ -model. Since  $h : \mathbb{N} \rightarrow \mathbb{N}$  is computable, there is an  $L_{\text{ar}}^r$ -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

$$"h(\text{num}(\varphi)) < y" \rightarrow (\varphi^{<y} \leftrightarrow \text{true}^{<y}(\text{num}(\varphi)))$$

for every  $L_{\text{ar}}^r$ -sentence  $\varphi$ . But  $M \models "h(\text{num}(\varphi)) < a"$ , hence

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

for every  $L_{\text{ar}}^r$ -sentence  $\varphi$ . 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  $L_{\text{ar}}^r$ -formula  $\varphi(x)$  maps  $\text{num}(\varphi)$  to  $\text{num}(\varphi(\text{num}(\varphi)))$  is computable. So by Lemma 4.2, there is a formula  $\text{sub}(x, y)$  such that for every formula  $\varphi(x)$  and every  $b \in M$ :

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

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

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

We arrive at the desired contradiction:

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

**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  $p\text{-}\Delta_0\text{-TRUTH}$  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  $p\text{-}\Delta_0\text{-TRUTH} \in \text{para-NP}$ , then  $\text{NE} \not\subseteq \text{LINH}$ .
- (ii) If  $p\text{-}\Delta_0\text{-TRUTH} \in \text{FPT}$ , then  $\text{E} \not\subseteq \text{LINH}$ .
- (iii) If  $p\text{-}\Delta_0\text{-TRUTH} \in \text{para-NL}$ , then  $\text{NLINSPACE} \not\subseteq \text{LINH}$ .
- (iv) If  $p\text{-}\Delta_0\text{-TRUTH} \in \text{para-L}$ , then  $\text{LINSPACE} \not\subseteq \text{LINH}$ .

**PROOF.** Since  $p\text{-}\Delta_0\text{-TRUTH}$  is an almost tally problem, (i) follows from Theorem 1.4 and Corollary 3.13. The other assertions follow using Lemma 3.14.  $\dashv$

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

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

In fact, we show that Theorem 1.3 implies a stronger statement for all computably enumerable  $\Pi_1$ -theories—up to logical equivalence,  $I\Delta_0$  is a  $\Pi_1$ -theory. Here, a *theory* is a set of sentences, and a  $\Pi_1$ -*theory* is a set of  $\Pi_1$ -sentences. The proof uses Parikh’s theorem [33]:

**THEOREM 5.1.** *Let  $T$  be a  $\Pi_1$ -theory and  $\varphi(\bar{x}, \bar{y})$  a  $\Delta_0$ -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  $\Delta_0$ -formula  $\varphi(\bar{x})$  there are  $L_{\text{ar}}$ -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  $\Delta_0$ -formula  $\varphi(\bar{x}) = \varphi(x_0, \dots, x_{k-1})$  there are  $L_{\text{ar}}$ -terms  $p(\bar{x}, \bar{y})$  and  $q(\bar{x}, \bar{y})$  such that  $T$  proves

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

Here,  $2^x \leq z$  stands for a well-known  $\Delta_0$ -formula [25, Section V.3(c)]. The following strengthens Theorem 1.5:

**THEOREM 5.4.** *Let  $T$  be a true  $\Pi_1$ -theory. Moreover, assume that  $T$  is computably enumerable. If  $T$  proves MRDP for small numbers, then  $p\text{-}\Delta_0\text{-TRUTH} \in \text{para-NP}$  and thus  $\text{NE} \not\subseteq \text{LINH}$ .*

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

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

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

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

Since  $T$  is computably enumerable, such terms  $p, q, r$  can be computed from  $\varphi$ . Given an instance  $\langle 1^n, \varphi \rangle$  of  $p\text{-}\Delta_0\text{-TRUTH}$ , 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  $\ell$  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  $p\text{-}\Delta_0\text{-TRUTH} \in \text{para-NP}$ . Now apply Theorem 1.3. ⊣

It would be interesting to find variants of this result that infer  $p\text{-}\Delta_0\text{-TRUTH} \in \text{FPT}$  or  $p\text{-}\Delta_0\text{-TRUTH} \in \text{para-NL}$  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  $\text{AC}^0$ -easy set in NP.** Recall, a problem  $Q$  is  $\text{AC}^0$ -bi-immune if neither  $Q$  nor its complement contain an infinite subset in  $\text{AC}^0$ . In this section we prove the following.

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

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

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

- (i)  $h(n) \geq f(n^2)$  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  $L_{ar}^r$ -sentence  $\varphi_h$  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  $L_{ar}^r$ -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)  $\{\langle x, y_{\mathbb{M},x} \rangle \mid x \in \{0, 1\}^*\} \in \text{AC}^0$ .

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  $i \leq n$ . Define the increasing function  $h : \mathbb{N} \rightarrow \mathbb{N}$  by

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

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,  $|y_{\mathbb{M}, 1^n}| \geq f(n^2)$ . Thus  $h(n) \geq f(n^2)$  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  $L_{\text{ar}}^r$ -sentence  $\varphi$  that holds precisely in the string structures of the form  $\mathcal{S}(\text{bin}(h(n)))$  for  $n \in \mathbb{N}$ . Using BIT, there is an interpretation  $I$  such that  $\mathcal{S}(1^m)^I \cong \mathcal{S}(\text{bin}(m))$  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  $\mathcal{S}(\text{bin}(h(n)))$ , 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  $p\text{-HALT} \in \text{para-AC}^0$ . Then there is an infinite tally problem  $X$  such that for every  $Q \in \text{NP}$  we have  $Q \cap X \in \text{AC}^0$ .*

*Proof of Theorem 1.6 from Theorem 6.2:* Assume  $p\text{-HALT} \in \text{para-AC}^0$  and let  $Q \in \text{NP}$ . 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  $\text{AC}^0$ ; indeed,  $(\{0, 1\}^* \setminus Q) \cap X = (\{0, 1\}^* \cap X) \setminus (Q \cap X)$  is in  $\text{AC}^0$  because both  $\{0, 1\}^* \cap X$  and  $Q \cap X$  are. Hence,  $Q$  is not  $\text{AC}^0$ -bi-immune.

**PROOF OF THEOREM 6.2.** By Corollary 2.7 there is a computable increasing function  $f : \mathbb{N} \rightarrow \mathbb{N}$  and an  $L_{\text{ar}}^r$ -sentence  $\varphi$  such that for every  $\langle 1^n, \mathbb{M} \rangle$  with  $n \geq f(|\mathbb{M}|)$ :

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

Now let  $h : \mathbb{N} \rightarrow \mathbb{N}$  be the increasing function as stated in Lemma 6.1. In particular, there is a deterministic Turing machine  $\mathbb{M}_h$  and a constant  $c \geq 1$  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

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

By Lemma 6.1(iii) the sentence  $\varphi_h$  witnesses  $X \in \text{AC}^0$  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  $d \geq 1$  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  $m \geq |\mathbb{M}_h| + |\mathbb{M}_Q| + e$  implies  $m^2 \geq |\mathbb{M}_{Q,h,m}|$ . Then

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

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

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

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

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

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

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

Assume  $x \in Q \cap X$ . Then  $x = 1^{h(m)}$  for some  $m > 1$  and  $\mathcal{S}(x) \models \varphi_h$ . Since  $|x| = h(m) \geq h(|\mathbb{M}_h| + |\mathbb{M}_Q| + e)$  and  $h$  is increasing, we have  $m \geq |\mathbb{M}_h| + |\mathbb{M}_Q| + e$ . 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  $\mathcal{S}(x) \models \varphi_h \wedge \varphi^I$ . 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,  $m \geq |\mathbb{M}_h| + |\mathbb{M}_Q| + e$ .  $\dashv$

### §7. Problem comparison.

**7.1. The role of uniformity.** Our proof of the lower bound  $p\text{-}\Delta_0\text{-TRUTH} \notin \text{para-AC}^0$  (Theorem 1.4) makes crucial use of the uniformity condition in the definition of  $\text{para-AC}^0$ . To shed some light on this dependence, we relax the uniformity condition as follows.

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

Clearly,

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

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

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

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

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

Fix  $k \in \mathbb{N}$  and let  $x \in \{0, 1\}^*$  with  $\kappa(x) = k$ . Since  $\kappa'$  is  $\text{AC}^0$ -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  $|x| \geq h(k)$ , then

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

is true in  $\mathcal{S}(x)$  if and only if  $r(x) \in Q'$ , i.e.,  $x \in Q$ . Translating  $\varphi$  gives an  $\text{AC}^0$ -family that decides the  $k$ th slice of  $Q$  on instances  $x$  with  $|x| \geq h(k)$ . 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  $\text{AC}_d^0$ . Now, in Theorem 2.6, the quantifier alternation rank of  $\varphi$  depends only on the depth of the  $\text{AC}^0$ -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  $\leq d'$  for some  $d'$  that depends only on  $d$ . The depth of the  $\text{AC}^0$ -family translating the above  $\varphi$  is  $\leq d''$  for some  $d''$  depending only on  $d'$ . The hardwiring of instances of small length can be done by circuits of depth 2. Thus,  $(Q, \kappa) \in \text{XAC}_{d''}^0$ .  $\dashv$

The class  $\text{XAC}^0$  is important in our context because it is a natural upper bound on  $p\text{-}\Delta_0\text{-TRUTH}$ :

PROPOSITION 7.3.  $p\text{-}\Delta_0\text{-TRUTH} \in \text{XAC}^0$ .

PROOF. It suffices to show that for every  $\Delta_0$ -formula  $\varphi(x)$  the problem  $\{1^n \mid \mathbb{N} \models \varphi(n)\}$  belongs to  $\text{AC}^0$ . But this problem is  $\text{un}(Q)$  for  $Q := \{x \in \{0, 1\}^* \mid \mathbb{N} \models \varphi(\text{num}(x))\}$ . Clearly  $Q \in \text{LINH}$ , so  $\text{un}(Q) \in \text{AC}^0$  follows from Proposition 2.1.  $\dashv$

We show that it is likely difficult to improve Theorem 1.4 to  $p\text{-}\Delta_0\text{-TRUTH} \notin \bigcup_{d \in \mathbb{N}} \text{XAC}_d^0$ . This somewhat artificial class also exhibits the different behaviors of the parameterized problems  $p\text{-HALT}$ ,  $p\text{-HALT}_=$ , and  $p\text{-}\Delta_0\text{-TRUTH}$ .

THEOREM 7.4.

- (i)  $p\text{-HALT} \in \text{XAC}_2^0$ .
- (ii)  $p\text{-HALT}_= \in \text{XAC}_d^0$  for some  $d \in \mathbb{N}$  if and only if  $\text{NE} \subseteq \text{LINH}$ .
- (iii)  $p\text{-}\Delta_0\text{-TRUTH} \in \text{XAC}_d^0$  for some  $d \in \mathbb{N}$  if and only if  $\text{LINH}$  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  $|\mathbb{M}| = k$ ,  $p\text{-HALT}$  is decided by the following family of simple Boolean functions:

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

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

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



$\lambda(x, y)$  such that for every  $\Delta_0$ -formula  $\varphi(x)$  there are  $c_\varphi, d_\varphi, e_\varphi \in \mathbb{N}$  such that for all  $n \geq c_\varphi$

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

For each fixed  $\varphi$  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  $n^{f_\varphi}$  for some  $f_\varphi \in \mathbb{N}$  depending on  $\varphi$ , but the depth of this family is determined by the quantifier alternation rank of  $\lambda$  and, in particular, does not depend on  $\varphi$ . This implies  $p\text{-}\Delta_0\text{-TRUTH} \in XAC^0_d$  for some  $d \in \mathbb{N}$ .

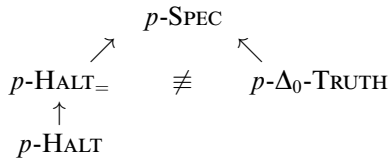
Conversely, assume  $p\text{-}\Delta_0\text{-TRUTH} \in XAC^0_d$  and let  $Q \in \text{LINH}$ . It is well known (see, e.g., [25, Chapter V, Lemma 2.13]) that there is a  $\Delta_0$ -formula that is satisfied by  $\text{num}(x)$  if and only if  $x \in Q$ . Fixing this formula in the input to  $p\text{-}\Delta_0\text{-TRUTH}$ , the assumption implies that there is a dlogtime uniform circuit family  $(C_n)_n$  of polynomial size and depth  $d$  such that for all  $x \in \{0, 1\}^*$ :

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

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  $C_{\text{num}(x)}$ . 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  $C_{\text{num}(x)}$ , that is, in time  $O(|x|)$ . We omit further details.  $\dashv$

**7.2. Reducibilities.** On the one hand  $p\text{-HALT}_=$  might appear ‘easier’ than  $p\text{-}\Delta_0\text{-TRUTH}$  in that the latter is not in  $\text{para-}AC^0$  while this is unknown for the former. On the other hand, Theorem 7.4 might indicate that  $p\text{-HALT}_=$  is ‘harder’ than  $p\text{-}\Delta_0\text{-TRUTH}$ . Also recall from the introduction that  $p\text{-HALT}_=$  is trivially in  $\text{para-NP}$  but not known to be solvable in time  $n^{f(k)}$  while for  $p\text{-}\Delta_0\text{-TRUTH}$  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  $p\text{-HALT}_=$  and  $p\text{-}\Delta_0\text{-TRUTH}$  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:

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

Recall that having a model of size  $n$  means that  $n$  belongs to the spectrum of  $\varphi$ .

We start comparing *p*-HALT and *p*-HALT<sub>=</sub>. By Example 3.6, *p*-HALT is reducible to *p*-HALT<sub>=</sub>. Concerning the converse we have the following.

**COROLLARY 7.5.** *If  $p$ -HALT<sub>=</sub> is reducible to  $p$ -HALT, then  $NE \subseteq LINH$ .*

**PROOF.** By Theorem 7.4(i),  $p$ -HALT  $\in \bigcup_d XAC_d^0$ . If *p*-HALT<sub>=</sub> is reducible to *p*-HALT, then  $p$ -HALT<sub>=</sub>  $\in \bigcup_d XAC_d^0$  by Lemma 7.2(ii). This implies  $NE \subseteq LINH$  by Theorem 7.4(ii).  $\dashv$

Adapting a mode of speech from [8], call an almost tally problem  $(Q, \kappa)$  *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<sub>=</sub> and *p*- $\Delta_0$ -TRUTH.

**COROLLARY 7.7.**

- (i) *If  $p$ - $\Delta_0$ -TRUTH is reducible to  $p$ -HALT<sub>=</sub>, then  $NE \not\subseteq LINH$ .*
- (ii) *If  $p$ -HALT<sub>=</sub> is reducible to  $p$ - $\Delta_0$ -TRUTH, then  $NE \subseteq LINH$ .*
- (iii)  *$p$ - $\Delta_0$ -TRUTH and  $p$ -HALT<sub>=</sub> are not equivalent.*

**PROOF.** (iii) follows from (i) and (ii). For (i), assume *p*- $\Delta_0$ -TRUTH is reducible to *p*-HALT<sub>=</sub>. Then *p*- $\Delta_0$ -TRUTH  $\in$  para-NP and  $NE \not\subseteq LINH$  follows by Theorem 1.3.

For (ii), assume *p*-HALT<sub>=</sub> is reducible to *p*- $\Delta_0$ -TRUTH. Then  $p$ -HALT<sub>=</sub>  $\in XAC^0$  by Proposition 7.3 and Lemma 7.2(i). This implies  $NE \subseteq LINH$  by Remark 3.10.  $\dashv$

Finally, we turn to *p*-SPEC:

**PROPOSITION 7.8.** *Both  $p$ -HALT<sub>=</sub> and  $p$ - $\Delta_0$ -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*- $\Delta_0$ -TRUTH, by Lemma 4.1, it suffices to show that *p*-MC( $L_{ar}^I$ ) is reducible to *p*-SPEC: map an instance  $(1^n, \varphi)$  of *p*-MC( $L_{ar}^I$ ) to  $(1^n, \varphi \wedge \psi)$  where  $\psi$  is an  $L_{ar}^I$ -sentence whose finite models are exactly those isomorphic to some standard finite  $L_{ar}^I$ -structure.  $\dashv$

Observe *p*-SPEC can be solved in nondeterministic time  $n^{f(k)}$  for some computable  $f : \mathbb{N} \rightarrow \mathbb{N}$  where  $k := |\varphi|$  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  $NE \not\subseteq LINH$ .*

**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. IMMERMANN, 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 slice-wise 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 Mathématique, 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. KRAJÍČEK 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  $\Delta_0$  formulae*, **Logic and Algorithmic**, Monographie de L'Enseignement Mathématique, 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  $\Sigma_0$ -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](mailto:yijia.chen@cs.sjtu.edu.cn)

FACULTY OF COMPUTER SCIENCE AND MATHEMATICS  
UNIVERSITY OF PASSAU  
PASSAU, GERMANY

*E-mail:* [moritz.mueller@uni-passau.de](mailto:moritz.mueller@uni-passau.de)

MATHEMATICAL INSTITUTE  
TOHOKU UNIVERSITY  
SENDAI, JAPAN

*E-mail:* [keita.yokoyama.c2@tohoku.ac.jp](mailto:keita.yokoyama.c2@tohoku.ac.jp)