Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Journal of Computer and System Sciences 72 (2006) 660–689

JOURNAL OF COMPUTER AND SYSTEM SCIENCES

www.elsevier.com/locate/jcss

LWPP and WPP are not uniformly gap-definable ☆

Holger Spakowski [a,1], Rahul Tripathi [b,*,2]

[a] *Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, 40225 Düsseldorf, Germany*
[b] *Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA*

Received 21 May 2004; received in revised form 23 November 2005

Available online 21 February 2006

Abstract

Resolving an issue open since Fenner, Fortnow, and Kurtz raised it in [S. Fenner, L. Fortnow, S. Kurtz, Gap-definable counting classes, J. Comput. System Sci. 48 (1) (1994) 116–148], we prove that LWPP is not uniformly gap-definable and that WPP is not uniformly gap-definable. We do so in the context of a broader investigation, via the polynomial degree bound technique, of the lowness, Turing hardness, and inclusion relationships of counting and other central complexity classes.

Keywords: Complexity classes; Gap-definability; Turing hardness; Polynomial degree bounds; Relativization theory

☆ A preliminary version of this paper appeared in [H. Spakowski, R. Tripathi, Degree bounds on polynomials and relativization theory, in: Proceedings of the 3rd IFIP International Conference on Theoretical Computer Science, Kluwer, 2004, pp. 105–118. [38]].

* Corresponding author.

E-mail addresses: spakowsk@cs.uni-duesseldorf.de (H. Spakowski), tripathi@cse.usf.edu (R. Tripathi).

1 Research supported in part by a grant from the DAAD and by the DFG under grants RO 1202/9-1 and RO 1202/9-3. Work done in part while visiting the University of Rochester.

2 Research supported in part by grants NSF-INT-9815095/DAAD-315-PPP-gü-ab and NSF-CCF-0426761. Most of this work was done while the author was affiliated with the Department of Computer Science at the University of Rochester, Rochester, NY 14627, USA.

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2006.01.002

1. Introduction

1.1. Background

Fenner, Fortnow, and Kurtz [16] introduced the function class GapP as a natural extension of the class #P. While #P functions are defined by the number of accepting paths of nondeterministic polynomial-time Turing machines, functions in GapP are defined by the difference between the number of accepting and rejecting paths of nondeterministic polynomial-time Turing machines. Fenner, Fortnow, and Kurtz [16] observed that many important counting classes (e.g., PP, $\mathrm{C_{=}P}$, $\mathrm{Mod}_k\mathrm{P}$) can be defined in terms of GapP functions. They called such classes gap-definable.

Informally speaking, a gap-definable counting class is a collection of all sets such that, for any set in the class, the membership of a string in the set depends (in a way particular to the class) on the gap (difference) between the number of accepting and rejecting paths produced by some nondeterministic polynomial-time Turing machine associated with the set. (See Section 2.2 for the definition of classes and Fig. 1 for the inclusion relationships between the classes mentioned here.) Gap-definable classes such as LWPP and AWPP are, for instance, interesting because of their relevance to quantum computing: LWPP is the best known classical upper bound for EQP (a quantum analog of P) and AWPP is the best known classical upper bound for BQP (a quantum analog of BPP) [19]. Thus the investigation of gap-definable classes may shed light on the structure of the quantum classes EQP and BQP. The gap-definable class SPP is low for several counting classes including PP, $\mathrm{C_{=}P}$, and $\mathrm{Mod}_k\mathrm{P}$, and the gap-definable class LWPP is low for PP and $\mathrm{C_{=}P}$ [16]. Because of this lowness property, SPP and LWPP are useful in understanding the structural complexity of counting classes PP and $\mathrm{C_{=}P}$. SPP is known to contain an important natural problem—the graph isomorphism problem [3]. Arvind and Vinodchandran [4] and Vinodchandran [45] showed that many group-theoretic computational problems are in SPP or LWPP. Since SPP and LWPP are considered as weak complexity classes, the classification of the graph isomorphism problem and certain group-theoretic computational problems into SPP or LWPP supports the belief that these problems are unlikely to be complete for NP.

A formal definition of gap-definability is given in terms of GapP functions and disjoint sets $A, R \subseteq \Sigma^* \times \mathbb{Z}$. (See Section 3 for the definition of gap-definability.) Based on the mechanism of relativizing this definition, Fenner, Fortnow, and Kurtz [16] suggested two ways of defining gap-definability for a relativizable class: *uniform* and *nonuniform* gap-definability. A relativizable class is uniformly gap-definable if it is gap-definable in every relativized world, where the choice of A and R is fixed and independent of the oracle. On the other hand, a relativizable class is nonuniformly gap-definable if it is gap-definable in every relativized world, where the choice of A and R depends on the oracle. Some examples of uniformly gap-definable counting classes are PP, $\mathrm{C_{=}P}$, $\mathrm{Mod}_k\mathrm{P}$, and SPP, and examples of classes that are nonuniformly gap-definable but were not known previously to be uniformly gap-definable are LWPP and WPP [16]. The proof of nonuniform gap-definability of LWPP and WPP given by Fenner, Fortnow, and Kurtz [16] required, given any oracle $\mathcal{O}$, an RE-immune set relative to $\mathcal{O}$ in order to define the sets A and R for these classes. Subsequently, Fenner, Fortnow, and Li [18] showed that A and R can be chosen such that $A \cup R$ is recursive. Fenner, Fortnow, and Kurtz [16] showed that SPP is low for every uniformly gap-definable class; whether SPP is low for LWPP or WPP remained open.

This paper resolves the open issues, raised by Fenner, Fortnow, and Kurtz [16], on whether LWPP is uniformly gap-definable and whether WPP is uniformly gap-definable. We prove that none of the classes LWPP and WPP are uniformly gap-definable. Thus LWPP and WPP are natural counting classes, which are nonuniformly gap-definable but are not uniformly gap-definable. This makes both LWPP and WPP special compared to other known natural gap-definable counting classes. Our proof that both LWPP and WPP are not uniformly gap-definable is in the context of a broader investigation using the polynomial degree bound technique. Among other results, we apply this proof technique to resolve an open question by Hemaspaandra, Ramachandran, and Zimand [28], and to extend the results by Hemaspaandra, Jain, and Vereshchagin [26].

1.2. The proof technique

In this paper, we use degree bounds of polynomials representing (not necessarily boolean) functions in constructing relativized worlds. Polynomials have been used in obtaining lower bounds for constant depth circuits [1,36], proving upper bounds on the power of complexity classes [40,41], proving closure properties of counting classes [11], proving bounds on the number of queries to compute a boolean function in the quantum black-box computing model [7], and in the construction of oracles in complexity theory [13,17,39]. See Beigel [8] and Regan [31] for nice surveys on the application of polynomials in circuit complexity and computational complexity theory.

In relativization theory, the technique of using degree bounds of polynomials has been extensively used in constructing oracles that separate complexity classes. We give some examples. Beigel [9] used a degree lower bound of a univariate polynomial to show that the set $L = \{x10^k \mid |x| \text{ is even and } k \in \mathbb{N}^+\}$ (called ODD-MAX-BIT in [9]) cannot be recognized by *perceptrons*[3] of polylogarithmic order, subexponential weight, and quasi-polynomial size. Using this result, he constructed an oracle relative to which $\mathrm{P}^{\mathrm{NP}} \nsubseteq \mathrm{PP}$. Aspnes et al. [5] showed that any low, i.e. polylog(n), degree polynomial fails to *sign* represent[4] the parity function on n bits with at least some constant probability when the input bits are chosen uniformly at random. So they were able to show that relative to a random oracle, PP $\neq$ PSPACE with probability one. Tarui [39] proved that if a low degree polynomial evaluates to zero on a certain large collection of inputs over a boolean domain, then the polynomial itself must be a zero polynomial. He used this result in constructing an oracle relative to which $\mathrm{BPP} \nsubseteq \mathrm{P}^{\mathrm{C_=P}}$. Recently, de Graaf and Valiant [13] made use of the

[3] A perceptron is a depth 2 circuit with a threshold gate at the root and AND-gates at the input level. The *order* of a perceptron is the maximum fanin of its AND-gates, its *weight* is the maximum absolute value of the weights on the inputs to the threshold gate, and its *size* is the number of AND-gates it contains.

[4] A sign representation of a function $f:\{1,-1\}^N \to \{1,-1\}$ is a polynomial $p \in \mathbb{R}[y_1, y_2, \ldots, y_N]$ such that for all $y_1, y_2, \ldots, y_N \in \{1,-1\}$, $\mathrm{sign}(p(y_1, y_2, \ldots, y_N)) = \mathrm{sign}(f(y_1, y_2, \ldots, y_N))$. Note that any boolean function on N variables can be represented as a function from $\{1,-1\}^N$ to $\{1,-1\}$, where each bit $b \in \{0,1\}$ is replaced by $(-1)^b$.

degree of a representing polynomial over the field $\mathbb{Z}_p$, for prime p, to obtain a relativized separation of EQP (the quantum analog of P) from $\mathrm{Mod}_p\mathrm{P}$.

Beigel, Buhrman, and Fortnow [10] and Fenner et al. [17] showed that degree bounds of polynomials can be used to obtain relativized collapses as well. In particular, Beigel, Buhrman, and Fortnow [10] used polynomials to construct an oracle $\mathcal{A}$ such that $\mathrm{P}^{\mathcal{A}} = \bigoplus \mathrm{P}^{\mathcal{A}}$ and $\mathrm{NP}^{\mathcal{A}} = \mathrm{EXP}^{\mathcal{A}}$, and Fenner et al. [17] showed that relative to an $\mathcal{SP}$-generic oracle, AWPP (a class defined in Section 2) equals P. We apply the polynomial degree bound technique to notions such as relativized lowness, nonexistence of Turing-hard sets in some relativized world, and relativized separations.

1.3. Our contributions

Fenner, Fortnow, and Kurtz [16] showed that SPP is low for every uniformly gap-definable class (see Section 3 for the definition of uniform and nonuniform gap-definability). Thus SPP is low for each of PP, $\mathrm{C_{=}P}$, $\mathrm{Mod}_k\mathrm{P}$, and itself. Both LWPP and WPP are known to be nonuniformly gap-definable and, prior to this paper, it was an open question whether or not these classes are uniformly gap-definable [16] as well. Thus Fenner, Fortnow, and Kurtz [16] asked whether SPP is also low for LWPP or WPP. We give a relativized answer to their question by exhibiting an oracle relative to which even $\mathrm{UP} \cap \mathrm{coUP}$ is not low for LWPP as well as for WPP. As a consequence of this oracle construction and an observation relating the issues of uniform gap-definability and lowness of SPP, we get the result that LWPP and WPP are not uniformly gap-definable. This resolves an open question raised by Fenner, Fortnow, and Kurtz [16].

The existence of complete sets in a class is a topic of interest in complexity theory. Though classes such as NP, $\mathrm{C_{=}P}$, and PP possess polynomial-time many-one complete sets, for several other natural classes such as UP and BPP, no complete set (under any weak enough to be interesting notion of reducibility) is known. This motivates the investigation of completeness for these promise classes in relativized worlds. That line of research was pursued in several papers [24,26,35]. In particular, Hemaspaandra, Jain, and Vereshchagin [26] showed that there is an oracle relative to which $\mathrm{UP} \cap \mathrm{coUP}$, UP, FewP, and Few have no polynomial-time Turing complete sets. The existence of a relativized world where promise classes such as SPP, LWPP, WPP, and AWPP do not have (polynomial-time many-one or Turing) complete sets remained unresolved [28]. We use a lower bound on the approximate degree of a boolean function given by Nisan and Szegedy [29] to construct a relativized world in which AWPP has no polynomial-time Turing hard sets for $\mathrm{UP} \cap \mathrm{coUP}$. As a corollary, we obtain that none of the classes SPP, LWPP, WPP, and AWPP have polynomial-time Turing complete sets in some relativized world. This settles an open question by Hemaspaandra, Ramachandran, and Zimand [28], and extends one of the main results by Hemaspaandra, Jain, and Vereshchagin [26]. Using a similar technique, we construct another relativized world where AWPP has no polynomial-time Turing hard sets for ZPP.

Certain classes are known to be weak in some relativized worlds while their composition with themselves lead to powerful classes in every relativized world. $\mathrm{C_{=}P}$ is a class that is immune to RP in a relativized world [37], but its composition with itself, i.e. $\mathrm{C_{=}P}^{\mathrm{C_{=}P}}$, contains the polynomial hierarchy in every relativized world. (In fact,

$\mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}[1]} \subseteq \mathrm{UP}^{\mathrm{C}_=\mathrm{P}} \subseteq \mathrm{C}_=\mathrm{P}^{\mathrm{C}_=\mathrm{P}}$.) Since ZPP $\nsubseteq$ WPP in some relativized world and, relative to an oracle, WPP is not self-low [37], it is worth investigating whether WPP, a class similar to $\mathrm{C}_=\mathrm{P}$, behaves in the same way as $\mathrm{C}_=\mathrm{P}$ when composed with itself. We use properties of low degree multilinear polynomials to construct an oracle world in which ZPP is not contained in $\mathrm{WPP}^{\mathrm{WPP}}$, thus falsifying this intuition. We also use a lower bound result on the degree of a univariate polynomial (by Ehlich and Zeller [14] and Rivlin and Cheney [32]) to construct an oracle relative to which NP $\cap$ coNP $\nsubseteq$ AWPP.

The proof technique that we use are applicable to classes that are not known to be gap-definable. For instance, we use the degree lower bound of polynomials in constructing a relativized world where MIP $\cap$ coMIP has no polynomial-time Turing hard sets for ZPP. This result can be seen as an extension of a result by Hemaspaandra, Jain, and Vereshchagin [26], which states that relative to an oracle, IP $\cap$ coIP has no polynomial-time Turing hard sets for ZPP.

2. Preliminaries

2.1. Notations

Let $\mathbb{N}^+$, $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{Z}$ denote the set of positive integers, rational numbers, real numbers, and integers, respectively. Our alphabet is $\Sigma = \{0, 1\}$. For any $A \subseteq \Sigma^*$ and $n \in \mathbb{N}^+$, let $A^{=n}$ denote the set of strings of length n in A and $A^{\leqslant n}$ denote the set of strings of length at most n in A. For every $n \in \mathbb{N}^+$, let $[n] =_{\mathrm{df}} \{1, 2, \ldots, n\}$. Let $\langle \ldots \rangle$ be a multiarity, easily computable, and invertible pairing function. If $A, B \subseteq \Sigma^*$, then define $A \oplus B = \{0w \mid w \in A\} \cup \{1w \mid w \in B\}$. For any set X of variables and for any polynomial $p \in \mathbb{R}[X]$, $\deg(p)$ denotes the total degree of p.

For standard notions in complexity theory, such as complexity classes, classes known to be in between P and NP, reductions, etc., we refer the reader to the textbook by Hemaspaandra and Ogihara [27]. For any nondeterministic Turing machine N, $A \subseteq \Sigma^*$, and $x \in \Sigma^*$, we use the shorthand $N^A(x)$ for "the computation of N with oracle A on input x." For any deterministic oracle transducer M, $A \subseteq \Sigma^*$, and $x \in \Sigma^*$, we denote by $M^A(x)$ the value computed by M with oracle A on input x. Throughout the paper, polynomials bounding the running time of machines are monotonically increasing. We assume that the computation paths of an oracle Turing machine include the answers from the oracle. Given a nondeterministic Turing machine N, computation path ρ, and $x \in \Sigma^*$, let $\mathrm{sign}(N, x, \rho) = +1$ if ρ is an accepting path of $N(x)$, and let $\mathrm{sign}(N, x, \rho) = -1$ if $N(x)$ rejects along ρ. Let $\#\mathrm{acc}_{N^A}(x)$ ($\#\mathrm{rej}_{N^A}(x)$) denote the number of accepting (respectively, rejecting) paths of $N^A(x)$. For any oracle NPTM N and $A \subseteq \Sigma^*$, $\mathrm{gap}_{N^A} : \Sigma^* \to \mathbb{Z}$ is defined as follows: For all $x \in \Sigma^*$, $\mathrm{gap}_{N^A}(x) = \#\mathrm{acc}_{N^A}(x) - \#\mathrm{rej}_{N^A}(x)$.

2.2. Complexity classes

We define the following complexity classes relevant to this paper.

Definition 2.1.

(1) [16,23] $\mathrm{GapP} = \{g \mid (\exists\, \mathrm{NPTM}\, N)[g = \mathrm{gap}_N]\}$.

(2) [16,23,30] SPP $= \{L \mid (\exists g \in \mathrm{GapP})\ (\forall x \in \Sigma^*)\ [g(x) \in \{0,1\} \wedge (x \in L \Leftrightarrow g(x) = 1)]\}$.
(3) [16] LWPP $= \{L \mid (\exists g \in \mathrm{GapP})\ (\exists h \in \mathrm{FP}\colon\ 0 \notin \mathrm{range}(h))\ (\forall x \in \Sigma^*)\ [g(x) \in \{0, h(0^{|x|})\} \wedge x \in L \Leftrightarrow g(x) = h(0^{|x|})]\}$.
(4) [16] WPP $= \{L \mid (\exists g \in \mathrm{GapP})\ (\exists h \in \mathrm{FP}\colon\ 0 \notin \mathrm{range}(h))\ (\forall x \in \Sigma^*)\ [g(x) \in \{0, h(x)\} \wedge x \in L \Leftrightarrow g(x) = h(x)]\}$.

SPP is an acronym of Stoic PP, WPP is an acronym of Wide PP, and LWPP is an acronym of Length-dependent Wide PP.

The counting class AWPP ("Almost WPP") was introduced by Fenner et al. [17]. The original definition of AWPP included the amplification property. Later, Fenner [15] gave a simplified definition for this class (see Theorem 2.3). We will only need the definition of AWPP due to Fenner in this paper.

Definition 2.2. [17] A language L is in AWPP if and only if for every polynomial $r(\cdot)$, there exist a GapP function g and a polynomial $p(\cdot)$ such that, for all $x \in \Sigma^*$,

$$x \in L \quad \Rightarrow \quad 1 - 2^{-r(|x|)} \leqslant \frac{g(x)}{2^{p(|x|)}} \leqslant 1, \quad \text{and}$$
$$x \notin L \quad \Rightarrow \quad 0 \leqslant \frac{g(x)}{2^{p(|x|)}} \leqslant 2^{-r(|x|)}.$$

Theorem 2.3. [15] *A language L is in* AWPP *if and only if there exist a* GapP *function g and a polynomial $p(\cdot)$ such that, for all $x \in \Sigma^*$,*

$$x \in L \quad \Rightarrow \quad \frac{2}{3} \leqslant \frac{g(x)}{2^{p(|x|)}} \leqslant 1, \quad \textit{and}$$
$$x \notin L \quad \Rightarrow \quad 0 \leqslant \frac{g(x)}{2^{p(|x|)}} \leqslant \frac{1}{3}.$$

We refer to any pair (N^A, M^A), where N is a nondeterministic polynomial-time oracle Turing machine, M is deterministic polynomial-time oracle transducer, and $A \subseteq \Sigma^*$, as an LWPP^A pair or a WPP^A pair, depending on the context. For any nondeterministic polynomial-time oracle Turing machine N, polynomial $q(\cdot)$, and $A \subseteq \Sigma^*$, we refer to (N^A, q) as an AWPP^A pair. We introduce the following notations.

- If (N^A, M^A) is an LWPP^A pair, then $L(N^A, M^A) =_{\mathrm{df}} \{x \in \Sigma^* \mid \mathrm{gap}_{N^A}(x) = M^A(0^{|x|})\}$.
- If (N^A, M^A) is a WPP^A pair, then $L(N^A, M^A) =_{\mathrm{df}} \{x \in \Sigma^* \mid \mathrm{gap}_{N^A}(x) = M^A(x)\}$.
- If (N^A, q) is an AWPP^A pair, then $L(N^A, q) =_{\mathrm{df}} \{x \in \Sigma^* \mid \mathrm{gap}_{N^A}(x)/2^{q(|x|)} \in [2/3, 1]\}$.

We define a predicate "valid" as follows.

- (N^A, M^A) is a valid LWPP^A pair if and only if for each $x \in \Sigma^*$, $M^A(0^{|x|}) \neq 0$ and $\mathrm{gap}_{N^A}(x) \in \{0, M^A(0^{|x|})\}$.

- (N^A, M^A) is a valid WPP^A pair if and only if for each $x \in \Sigma^*$, $M^A(x) \neq 0$ and $\mathrm{gap}_{N^A}(x) \in \{0, M^A(x)\}$.
- (N^A, q) is a valid AWPP^A pair if and only if for each $x \in \Sigma^*$, $\mathrm{gap}_{N^A}(x)/2^{q(|x|)} \in [0, 1/3] \cup [2/3, 1]$.

An interactive proof system [6,22] is a computational model consisting of a probabilistic polynomial-time verifier V interacting with an infinitely powerful prover P to decide the membership of a string in a set. The verifier and the prover interact using a protocol and at the end of it, the verifier either accepts or rejects. A generalization of this proof system, proposed by Ben-Or, Goldwasser, Kilian, and Wigderson [12], involves more than a single prover and is referred to as multiprover interactive proof system. A formal definition of a k-prover interactive proof system for a set L is as follows.

Definition 2.4. [6,12,22] For any $k \geqslant 1$, a set L has a k-prover interactive proof system if there is a probabilistic polynomial-time verifier V that interacts with k provers such that, for each $x \in \Sigma^*$, the following conditions hold:

(1) If $x \in L$, then there is a set of k provers $P_1, P_2, \ldots, P_k$ such that $\mathrm{Prob}[P_1, P_2, \ldots, P_k,$ and V on x accept$] \geqslant 1 - 2^{-|x|}$.
(2) If $x \notin L$, then for any set of k provers $P'_1, P'_2, \ldots, P'_k$, $\mathrm{Prob}[P'_1, P'_2, \ldots, P'_k,$ and V on x accept$] \leqslant 2^{-|x|}$.

Here the probability is over the random coin tosses done by V. IP (MIP) is the class of all sets that have 1-prover interactive proof systems (respectively, k-prover interactive proof systems for some $k \geqslant 1$).

It can be shown that if a set L has a k-prover interactive proof system for some k, then L also has a 2-prover interactive proof system [12]. Even in the case when the number of provers are polynomially related with the input length, the computational power of such a multiprover proof system is known to be no more than that of a 2-prover proof system.

The inclusion relationship between classes considered in this paper is summarized in Fig. 1.

2.3. Polynomial encoding

In our proofs, we use an encoding of the behavior of a nondeterministic polynomial-time oracle Turing machine on an input relative to some finite set, where the set can be viewed as a source of a possible oracle extension at some stage of the oracle construction. This encoding is defined in terms of a multilinear polynomial with integer coefficients over variables representing the strings in the set. The formal description of the polynomial encoding is given as follows.

Definition 2.5. Let N be a nondeterministic polynomial-time oracle Turing machine with running time $t(\cdot)$. Let $\mathcal{O}, \mathcal{T} \subseteq \Sigma^*$ be such that $\mathcal{O} \cap \mathcal{T} = \emptyset$, and let $x_1, x_2, \ldots, x_m$, where $m = \|\mathcal{T}\|$, be the lexicographic enumeration of strings in $\mathcal{T}$. For any $x \in \Sigma^*$, a polynomial

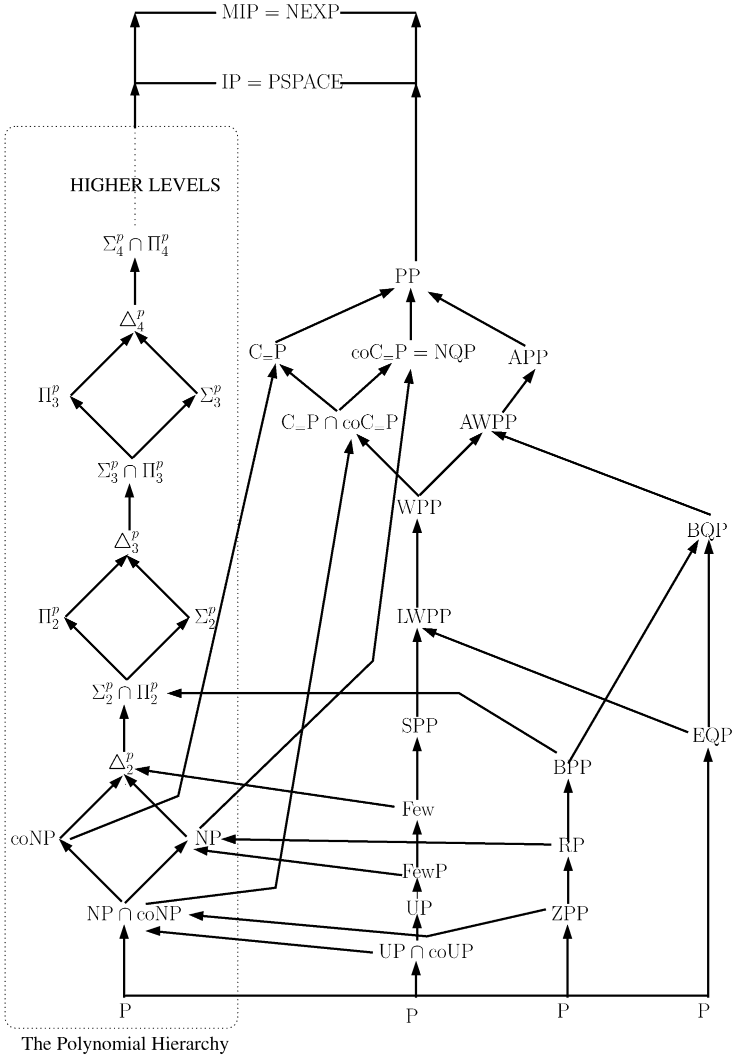

Fig. 1. Complexity graph G where a node represents a complexity class and a directed edge (U, V) in G represents the fact that "class U is known to be included in class V."

encoding of $N^{\mathcal{O}}(x)$ w.r.t. $\mathcal{T}$ is a multilinear polynomial $p \in \mathbb{Z}[y_1, y_2, \ldots, y_m]$ defined as follows: Call a computation path ρ of $N^{(\cdot)}(x)$ *allowable* if along ρ, all queries $q \in \mathcal{O}$ have a "yes" answer, all queries $q \notin \mathcal{O} \cup \mathcal{T}$ have a "no" answer, and no query $q \in \mathcal{T}$ is answered in a conflicting way. Let $x_{i_1}, x_{i_2}, \ldots, x_{i_\ell}$ be the distinct queries to strings in $\mathcal{T}$ along an allowable ρ. Create a monomial mono(ρ) that is the product of terms z_{i_k}, $k \in [\ell]$, where $z_{i_k} = y_{i_k}$ if x_{i_k} is answered "yes" and $z_{i_k} = (1 - y_{i_k})$ if x_{i_k} is answered "no" along ρ. Define

$$p(y_1, y_2, \ldots, y_m) = \sum_{\rho:\ \rho \text{ is allowable}} \operatorname{sign}(N, x, \rho) \cdot \operatorname{mono}(\rho).$$

The following proposition is evident from the definition of the polynomial encoding.

Proposition 2.6. *Let $p \in \mathbb{Z}[y_1, y_2, \ldots, y_m]$ be the polynomial encoding of $N^{\mathcal{O}}(x)$ w.r.t. $\mathcal{T}$. Then the polynomial $p(y_1, y_2, \ldots, y_m)$ has the following properties*:

(1) *for all $\mathcal{B} \subseteq \mathcal{T}$, $p(\chi_{\mathcal{B}}(x_1), \chi_{\mathcal{B}}(x_2), \ldots, \chi_{\mathcal{B}}(x_m)) = \mathrm{gap}_{N^{\mathcal{O}\cup\mathcal{B}}}(x)$, and*
(2) $\deg(p) \leqslant t(|x|)$.

Here N, $t(\cdot)$, $\mathcal{O}$, $\mathcal{T}$, m, and $x_1, x_2, \ldots, x_m$ are defined as in Definition 2.5.

3. Lowness and gap-definability

The low hierarchy within NP was introduced by Schöning [34] to study the inner structure of NP. Since the introduction of the low hierarchy, the concept of lowness has been generalized to arbitrary relativizable function and language classes. A set $L \subseteq \Sigma^*$ is said to be *low* for a relativizable class $\mathcal{C}$ if $\mathcal{C}^L \subseteq \mathcal{C}$. A class $\mathcal{C}_2$ is called *low* for a relativizable class $\mathcal{C}_1$ if $\mathcal{C}_1^{\mathcal{C}_2} \subseteq \mathcal{C}_1$.

Fenner, Fortnow, and Kurtz [16] introduced the notion of gap-definability to study the counting classes that can be defined using GapP functions alone. Since most of the well-known counting classes, such as PP, $\mathrm{C_{=}P}$, and $\mathrm{Mod}_k\mathrm{P}$, are gap-definable, any characterization for gap-definable classes carries over to these counting classes. For instance, it is known that SPP is low for every member of a particular collection of gap-definable classes, namely the collection of uniformly gap-definable classes. Thus it follows that SPP is low for the counting classes PP, $\mathrm{C_{=}P}$, and $\mathrm{Mod}_k\mathrm{P}$. The formal definition of gap-definability is given below.

Definition 3.1. [16] A class $\mathcal{C}$ is gap-definable if there exist disjoint sets $A, R \subseteq \Sigma^* \times \mathbb{Z}$ such that, for any $L \subseteq \Sigma^*$, $L \in \mathcal{C}$ if and only if there exists an NPTM N such that for all $x \in \Sigma^*$,

$$\begin{aligned} x \in L \quad &\Rightarrow \quad \bigl(x, \mathrm{gap}_N(x)\bigr) \in A, \quad \text{and} \\ x \notin L \quad &\Rightarrow \quad \bigl(x, \mathrm{gap}_N(x)\bigr) \in R. \end{aligned}$$

The class $\mathcal{C}$ is also denoted by $\mathrm{Gap}(A, R)$.

For relativizable classes, Fenner, Fortnow, and Kurtz [16] introduced two ways of defining gap-definability: uniform and nonuniform. A relativizable class $\mathcal{C}$ is said to be *uniformly gap-definable* if it is gap-definable w.r.t. any oracle with a fixed (independent of the oracle) choice of A and R. A relativizable class $\mathcal{C}$ is said to be *nonuniformly gap-definable* if it is gap-definable w.r.t. an oracle where the choice of A and R may depend on the oracle. Thus the choice of A and R may vary with different oracles in case of nonuniform gap-definability. We now give a definition that expresses the oracle (in)dependence of the pair (A, R) in the notion of gap-definability. In what follows, (A, R) is called an accepting pair if $A, R \subseteq \Sigma^* \times \mathbb{Z}$ and $A \cap R = \emptyset$.

Definition 3.2. [16]

(1) A relativizable class $\mathcal{C}$ is *gap-definable relative to an oracle $\mathcal{O}$ with accepting pair* (A, R) if for any $L \subseteq \Sigma^*$, $L \in \mathcal{C}^{\mathcal{O}}$ if and only if there exists an oracle NPTM N such that for all $x \in \Sigma^*$,

$$x \in L \quad \Rightarrow \quad \big(x, \mathrm{gap}_{N^{\mathcal{O}}}(x)\big) \in A, \quad \text{and}$$
$$x \notin L \quad \Rightarrow \quad \big(x, \mathrm{gap}_{N^{\mathcal{O}}}(x)\big) \in R.$$

(2) A relativizable class $\mathcal{C}$ is *uniformly gap-definable* if there is an accepting pair (A, R) such that for every oracle $\mathcal{O} \subseteq \Sigma^*$, it holds that $\mathcal{C}$ is gap-definable relative to $\mathcal{O}$ with accepting pair (A, R).
(3) A relativizable class $\mathcal{C}$ is *nonuniformly gap-definable* if for every oracle $\mathcal{O} \subseteq \Sigma^*$, there is an accepting pair (A, R) such that $\mathcal{C}$ is gap-definable relative to $\mathcal{O}$ with accepting pair (A, R).

Fenner, Fortnow, and Kurtz [16] proved that SPP is low for GapP. This implies that SPP is low for every *uniformly* gap-definable counting class, such as PP, $\mathrm{C_{=}P}$, $\bigoplus\mathrm{P}$, and SPP. It is easy to see that this result holds in every relativized world.

Theorem 3.3. [16] *If $\mathcal{C}$ is a uniformly gap-definable class, then for every $\mathcal{O} \subseteq \Sigma^*$, it holds that $\mathcal{C}^{\mathrm{SPP}^{\mathcal{O}}} = \mathcal{C}^{\mathcal{O}}$.*

In Theorem 3.6, we construct a relativized world in which UP ∩ coUP is not low for LWPP as well as for WPP. Since UP ∩ coUP ⊆ SPP in every relativized world, this also shows that relative to the same oracle, SPP is not low for either LWPP or WPP. Fenner, Fortnow, and Kurtz [16] proved that both LWPP and WPP are nonuniformly gap-definable. However, they leave open the question whether LWPP and WPP are uniformly gap-definable. From Theorems 3.3 and 3.6, we conclude that LWPP and WPP are not uniformly gap-definable.

We use a variant of the prime number theorem, stated in Lemma 3.4, in the proof of Theorem 3.6 to estimate the number of primes between two integers.

Lemma 3.4. [33] *For every $n \geqslant 17$, the number of primes less than or equal to n, i.e. $\pi(n)$, satisfies*

$$n/\ln n < \pi(n) < 1.25506n/\ln n.$$

The following lemma, Lemma 3.5, was used by Spakowski, Thakur, and Tripathi [37] to construct a relativized world in which WPP is not closed under polynomial-time Turing reductions. The same lemma is useful in proving Theorem 3.6.

Lemma 3.5. [37] *Let $N, p \in \mathbb{N}^+$ be such that p is a prime number and $p \leqslant N/2$. Let $s \in \mathbb{Z}[y_1, y_2, \ldots, y_N]$ be a multilinear polynomial with total degree* $\deg(s) < p$. *If for some $val \in \mathbb{Z}$, it holds that*

(1) $s(0,0,\ldots,0)=0$, *and*
(2) $s(y_1,y_2,\ldots,y_N)=val$, *for every* $y_1,y_2,\ldots,y_N\in\{0,1\}$ *with* $\sum_{i=1}^{N} y_i = p$,

then $p \mid val$.

Theorem 3.6. $(\exists\mathcal{A})$ $[\mathrm{LWPP}^{\mathrm{UP}^{\mathcal{A}}\cap\mathrm{coUP}^{\mathcal{A}}} \nsubseteq \mathrm{WPP}^{\mathcal{A}}]$.[5]

Proof. For any $B\subseteq\Sigma^*$, define the test language L_B by

$$L_B=\{0^n \mid \|B^{=2n}\|\neq 0\}.$$

We put certain constraints on the set B that guarantee L_B to be in $\mathrm{LWPP}^{\mathrm{UP}^B\cap\mathrm{coUP}^B}$. For each $n\in\mathbb{N}^+$, we say that B satisfies $\mathrm{Constraint}(B,n)$ if the following conditions hold:

(a) $B^{=2n+1}=\{0z\}$ for some $z\in\Sigma^{2n}$, and
(b) $B^{=2n+1}=\{0z\}\Rightarrow \|B^{=2n}\|\in\{0,\mathrm{rank}(z)\}$,

where $\mathrm{rank}(z)$ is the number of strings of length $|z|$ that are lexicographically less than or equal to z.

Claim 1. *If B satisfies* $\mathrm{Constraint}(B,n)$ *at each length n, then L_B is in* $\mathrm{LWPP}^{\mathrm{UP}^B\cap\mathrm{coUP}^B}$.

Proof. Let B satisfy $\mathrm{Constraint}(B,n)$ for every $n\in\mathbb{N}^+$. We will define $\mathcal{L}\subseteq\Sigma^*$, and oracle machines $\mathcal{N}$ and $\mathcal{M}$ that satisfy the following: (a) $\mathcal{L}\in\mathrm{UP}^B\cap\mathrm{coUP}^B$, (b) $(\mathcal{N}^{\mathcal{L}\oplus B},\mathcal{M}^{\mathcal{L}\oplus B})$ is a valid $\mathrm{LWPP}^{\mathcal{L}\oplus B}$ pair, and (c) $L(\mathcal{N}^{\mathcal{L}\oplus B},\mathcal{M}^{\mathcal{L}\oplus B})=L_B$. This will show that L_B is in $\mathrm{LWPP}^{\mathrm{UP}^B\cap\mathrm{coUP}^B}$. The set $\mathcal{L}$ is defined as follows:

$$\mathcal{L}=\{x \mid |x| \text{ is odd and } (\exists x')\,[|x'|=|x| \text{ and } \mathrm{rank}(x)\leqslant\mathrm{rank}(x') \text{ and } x'\in B]\}.$$

If B satisfies $\mathrm{Constraint}(B,n)$ for every $n\in\mathbb{N}^+$, then $\mathcal{L}\in\mathrm{UP}^B\cap\mathrm{coUP}^B$ since there is exactly one string $x'\in B$ at every odd length.

Let $\mathcal{N}'$ be a nondeterministic polynomial-time oracle Turing machine that, with access to the oracle B, on input x,

(1) if $x\notin 0^*$ then rejects x, and
(2) if $x\in 0^*$ then guesses a string x' of length $2|x|$ and accepts x' if and only if x' is in B.

Since $\#\mathrm{P}\subseteq\mathrm{GapP}$ in every relativized world, there exists a nondeterministic polynomial-time oracle Turing machine $\mathcal{N}$ such that for all $\mathcal{O}\subseteq\Sigma^*$ and $x\in\Sigma^*$, $\mathrm{gap}_{\mathcal{N}^{\mathcal{O}}}(x)=\#\mathrm{acc}_{\mathcal{N}'^{\mathcal{O}}}(x)$. Finally, we define the deterministic polynomial-time oracle transducer $\mathcal{M}$ that, with access to the oracle $\mathcal{L}\oplus B$, on input x,

(1) if $x\notin 0^*$ then outputs some nonzero value, say 1, and

[5] It is easy to see that $\mathrm{LWPP}^{\mathrm{UP}^{\mathcal{A}}\cap\mathrm{coUP}^{\mathcal{A}}}=\mathrm{LWPP}^{(\mathrm{UP}^{\mathcal{A}}\cap\mathrm{coUP}^{\mathcal{A}})\oplus\mathcal{A}}$.

(2) if $x \in 0^*$ then performs a binary search for the unique string $0w$, where $|w| = 2|x|$, in B by asking queries for the membership of strings of the form $0w'$, where $|w'| = 2|x|$, in $\mathcal{L}$. The machine $\mathcal{M}^{\mathcal{L}\oplus B}(0^n)$ finally outputs $\mathrm{rank}(w)$.

It can easily be verified that $(\mathcal{N}^{\mathcal{L}\oplus B}, \mathcal{M}^{\mathcal{L}\oplus B})$ is a valid $\mathrm{LWPP}^{\mathrm{UP}^B \cap \mathrm{coUP}^B}$ pair and $L(\mathcal{N}^{\mathcal{L}\oplus B}, \mathcal{M}^{\mathcal{L}\oplus B}) = L_B$. Thus the claim follows. □

We construct an oracle $\mathcal{A}$ such that, for each n, $\mathrm{Constraint}(\mathcal{A}, n)$ is true and $L_{\mathcal{A}} \notin \mathrm{WPP}^{\mathcal{A}}$. Let (N_i, M_i) be an enumeration of machine pairs where N_i is nondeterministic oracle Turing machine, M_i is a deterministic oracle transducer, and both N_i and M_i run in time $n^i + i$ on inputs of length n. The oracle $\mathcal{A}$ is constructed in stages. In each stage, the membership in $\mathcal{A}$ of strings of length $2n$ and $2n+1$ are decided for some $n \in \mathbb{N}^+$. Initially, $\mathcal{A} := \{0^{2m+1} \mid m \in \mathbb{N}^+\}$ and $n := 17$.

Stage i, $i \geqslant 1$: Choose n large enough so that $2^n > 4n^2(n^i + i)$, no string of length $2n$ or more is queried in the previous stages, and n is larger than the value of n in the previous stage. We diagonalize against nondeterministic polynomial-time oracle Turing machine N_i and deterministic polynomial-time oracle transducer M_i. Let $\mathcal{A} := \mathcal{A} - \{0^{2n+1}\}$ and let $val =_{\mathrm{df}} M_i^{\mathcal{A}}(0^n)$. Because of the condition $0 \notin \mathrm{range}(h)$ in the definition of WPP, we can assume that val is nonzero.

Let

$$S = \{w \mid w \in \Sigma^{2n} \text{ and } M_i^{\mathcal{A}}(0^n) \text{ does not query } w\} \\ \cup \{0w \mid w \in \Sigma^{2n} \text{ and } M_i^{\mathcal{A}}(0^n) \text{ does not query } 0w\}.$$

($\star$) Choose $B \subseteq S$ such that $\mathrm{Constraint}(B, n)$ is true and the following holds:

$$\|B^{=2n}\| \neq 0 \quad \text{and} \quad \mathrm{gap}_{N_i^{\mathcal{A}\cup B}}(0^n) \neq val, \quad \text{or}$$
$$\|B^{=2n}\| = 0 \quad \text{and} \quad \mathrm{gap}_{N_i^{\mathcal{A}\cup B}}(0^n) \neq 0.$$

We will show in Claim 2 that there is a set B satisfying ($\star$). Set $\mathcal{A} := \mathcal{A} \cup B$. Move to the next stage.

End of Stage

Clearly, the construction guarantees that $L_{\mathcal{A}} \notin \mathrm{WPP}^{\mathcal{A}}$. Thus it remains to show that a set B satisfying ($\star$) always exists.

Claim 2. *For every $i \geqslant 1$, there exists a set B satisfying* ($\star$).

Proof. Assume to the contrary that in some stage i, no set B satisfying ($\star$) exists. Then for every $B \subseteq S$ such that B satisfies $\mathrm{Constraint}(B, n)$, the following holds:

$$\|B^{=2n}\| \neq 0 \quad \Rightarrow \quad \mathrm{gap}_{N_i^{\mathcal{A}\cup B}}(0^n) = val, \quad \text{and}$$
$$\|B^{=2n}\| = 0 \quad \Rightarrow \quad \mathrm{gap}_{N_i^{\mathcal{A}\cup B}}(0^n) = 0.$$

Let $Z = \{z \in \Sigma^{2n} \mid \mathrm{rank}(z) \text{ is prime},\ 0z \in S,\ \text{and } 2^{n-2} \leqslant \mathrm{rank}(z) \leqslant 2^{n-1}\}$.

Fix an arbitrary element z from Z. Then for all $C \subseteq \Sigma^{2n} \cap S$, it holds that

$$\|C\| = \mathrm{rank}(z) \quad \Rightarrow \quad \mathrm{gap}_{N_i^{\mathcal{A} \cup C \cup \{0z\}}}\big(0^n\big) = val, \quad \text{and} \tag{1}$$

$$\|C\| = 0 \quad \Rightarrow \quad \mathrm{gap}_{N_i^{\mathcal{A} \cup C \cup \{0z\}}}\big(0^n\big) = 0. \tag{2}$$

Let $N =_{\mathrm{df}} \|\Sigma^{2n} \cap S\|$ and let $x_1, x_2, \ldots, x_N$ be the lexicographic enumeration of strings in $\Sigma^{2n} \cap S$. Let $s_z \in \mathbb{Z}[y_1, y_2, \ldots, y_N]$ be the polynomial encoding of $N_i^{\mathcal{A} \cup \{0z\}}(0^n)$ w.r.t. $\Sigma^{2n} \cap S$. From Proposition 2.6, it follows that the polynomial $s_z(y_1, y_2, \ldots, y_N)$ has the following properties:

- for all $C \subseteq \Sigma^{2n} \cap S$, it holds that $s_z(\chi_C(x_1), \chi_C(x_2), \ldots, \chi_C(x_N)) = \mathrm{gap}_{N_i^{\mathcal{A} \cup C \cup \{0z\}}}(0^n)$.
- $\deg(s_z) \leqslant n^i + i < \mathrm{rank}(z) < N/2$.

Statements (1) and (2), respectively, imply that

- for all $y_1, y_2, \ldots, y_N \in \{0, 1\}$ such that $\sum_{i=1}^{N} y_i = \mathrm{rank}(z)$, we have $s_z(y_1, y_2, \ldots, y_N) = val$, and
- $s_z(0, 0, \ldots, 0) = 0$.

It follows from Lemma 3.5 that $\mathrm{rank}(z) \mid val$.

Therefore, we have shown that for each $z \in Z$, $\mathrm{rank}(z) \mid val$. Hence by Lemma 3.4 and the fact that $2^n > 4n^2(n^i + i)$, $val \geqslant \prod_{z \in Z} \mathrm{rank}(z) \geqslant 2^{\|Z\|} \geqslant 2^{\pi(2^{n-1}) - \pi(2^{n-2}) - n^i - i} \geqslant 2^{2^{n-1}/n^2 - n^i - i} > 2^{n^i + i}$. However, $M_i^{(\cdot)}(0^n)$ runs in time $n^i + i$ and so $val \leqslant 2^{n^i + i}$. Thus we have a contradiction. This completes the proofs of Claim 2 and Theorem 3.6. □

Corollary 3.7. LWPP *and* WPP *are not uniformly gap-definable.*

Corollary 3.8. *There is a relativized world $\mathcal{A}$ such that*

(1) *for any class $\mathcal{C} \in \{\mathrm{UP} \cap \mathrm{coUP}, \mathrm{UP}, \mathrm{FewP}, \mathrm{Few}, \mathrm{SPP}, \mathrm{LWPP}\}$, $\mathcal{C}^{\mathcal{A}}$ is not low for* $\mathrm{LWPP}^{\mathcal{A}}$, *and*
(2) *for any class $\mathcal{C} \in \{\mathrm{UP} \cap \mathrm{coUP}, \mathrm{UP}, \mathrm{FewP}, \mathrm{Few}, \mathrm{SPP}, \mathrm{LWPP}, \mathrm{WPP}\}$, $\mathcal{C}^{\mathcal{A}}$ is not low for* $\mathrm{WPP}^{\mathcal{A}}$.

4. Robust hardness under Turing reducibilities

Complexity classes such as P, NP, coNP, PP, $\mathrm{C_{=}P}$, and $\mathrm{Mod}_k\mathrm{P}$ are *robust* in possessing polynomial-time many-one complete sets. That is, these complexity classes contain polynomial-time many-one complete sets in every relativized world. However, classes such as NP ∩ coNP, UP, and BPP lack polynomial-time many-one complete sets in some relativized worlds because of the built-in promises in their definitions [24,35]. The current section continues this exploration of complexity classes to gap-definable counting classes.

We prove that there exist relativized worlds where several gap-definable counting classes including AWPP, WPP, LWPP, and SPP lack polynomial-time Turing complete sets. We resolve an open question of Hemaspaandra, Ramachandran, and Zimand [28] and extend one of the main results of Hemaspaandra, Jain, and Vereshchagin [26]. The central technical tool used in the proofs of this section is a lower bound by Nisan and Szegedy [29] on the approximate degree of certain boolean functions.

If $f:\{0,1\}^N \to \{0,1\}$ is a boolean function and $p \in \mathbb{R}[y_1, y_2, \ldots, y_N]$ is a multilinear polynomial such that, for every $y_1, y_2, \ldots, y_N \in \{0,1\}$, $f(y_1, y_2, \ldots, y_N) = p(y_1, y_2, \ldots, y_N)$, then p is said to be a polynomial representing f *exactly*. If p is a smallest degree multilinear polynomial representing a boolean function f exactly, then we use $\deg(f)$ to denote $\deg(p)$, the total degree of p. We now give a definition of the notion of the *approximate* degree of a boolean function.

Definition 4.1. [29] Given a boolean function $f:\{0,1\}^N \to \{0,1\}$ and a polynomial $p \in \mathbb{R}[y_1, \ldots, y_N]$, we say that *p approximates f* if for every $y_1, \ldots, y_N \in \{0,1\}$, it holds that $|f(y_1, \ldots, y_N) - p(y_1, \ldots, y_N)| \leqslant 1/3$. The *approximate degree* of f, denoted by $\widetilde{\deg}(f)$, is the minimum integer d such that there is a polynomial of degree d that approximates f.

Nisan and Szegedy [29] obtained a $\Omega(\sqrt{N})$ lower bound on the degree and approximate degree of a restricted, though still quite general, boolean function. In particular, they showed that any boolean function, whose value is zero on the all-zero input but whose value is one on every boolean input vector with Hamming weight (the number of 1's in the boolean vector) one, has approximate degree at least $\sqrt{N/6}$. As a direct consequence of this, they obtained a $\Omega(\sqrt{N})$ lower bound on the approximate degree of the boolean OR function. (In fact, Nisan and Szegedy [29] also obtained a matching upper bound of $O(\sqrt{N})$ on the approximate degree of the OR function.)

Lemma 4.2. [29] *Let f be a boolean function on N inputs such that $f(0,0,\ldots,0) = 0$ and for every $x_1, x_2, \ldots, x_N \in \{0,1\}$ such that $\sum_{i\in[N]} x_i = 1$, $f(x_1, x_2, \ldots, x_N) = 1$. Then the following inequalities hold*:

$$\deg(f) \geqslant \sqrt{N/2} \quad \textit{and} \quad \widetilde{\deg}(f) \geqslant \sqrt{N/6}.$$

We use this result by Nisan and Szegedy [29] to prove Lemma 4.5, which is central to our relativization results involving the class AWPP.

When we speak about relativized Turing reductions, it is natural to ask whether the Turing reduction is allowed access to the oracle. We answer this question by giving two different definitions of relativized Turing reductions as in Definitions 4.3(1) and 4.3(2).

Definition 4.3.

(1) If $\mathcal{C}_1$ and $\mathcal{C}_2$ are relativizable classes, then for each $\mathcal{A} \subseteq \Sigma^*$, we say that $\mathcal{C}_1^{\mathcal{A}}$ has a $\leqslant_T^{p,\mathcal{A}}$-hard set for $\mathcal{C}_2^{\mathcal{A}}$ if there exists $L_1 \in \mathcal{C}_1^{\mathcal{A}}$ such that for every $L_2 \in \mathcal{C}_2^{\mathcal{A}}$,

$L_2 \in \mathrm{P}^{\mathcal{A} \oplus L_1}$. If $\mathcal{C}_1$ and $\mathcal{C}_2$ are the same class, then L_1 is referred to as a $\leqslant_T^{p,\mathcal{A}}$-complete set for $\mathcal{C}_1^{\mathcal{A}}$. In this case, we say that $\mathcal{C}_1^{\mathcal{A}}$ has a $\leqslant_T^{p,\mathcal{A}}$-complete set.

(2) If $\mathcal{C}_1$ and $\mathcal{C}_2$ are relativizable classes, then for each $\mathcal{A} \subseteq \Sigma^*$, we say that $\mathcal{C}_1^{\mathcal{A}}$ has a $\leqslant_T^{p}$-hard set for $\mathcal{C}_2^{\mathcal{A}}$ if there exists $L_1 \in \mathcal{C}_1^{\mathcal{A}}$ such that for every $L_2 \in \mathcal{C}_2^{\mathcal{A}}$, $L_2 \in \mathrm{P}^{L_1}$. If $\mathcal{C}_1$ and $\mathcal{C}_2$ are the same class, then L_1 is referred to as a $\leqslant_T^{p}$-complete set for $\mathcal{C}_1^{\mathcal{A}}$. In this case, we say that $\mathcal{C}_1^{\mathcal{A}}$ has a $\leqslant_T^{p}$-complete set.

However, Lemma 4.4 shows that the two notions, Definitions 4.3(1) and 4.3(2), of relativized polynomial-time Turing reductions are equivalent when dealing with hardness results. We note that the two notions lead to remarkably different effects as studied in [21,25].

Lemma 4.4. (See [26] for a similar lemma.) *If $\mathcal{C}_1$ and $\mathcal{C}_2$ are relativizable classes and if $\mathcal{C}_1$ is closed under join operation in every relativized world, then for every $\mathcal{A} \subseteq \Sigma^*$, $\mathcal{C}_1^{\mathcal{A}}$ has a $\leqslant_T^{p,\mathcal{A}}$-hard set for $\mathcal{C}_2^{\mathcal{A}}$ if and only if $\mathcal{C}_1^{\mathcal{A}}$ has a $\leqslant_T^{p}$-hard set for $\mathcal{C}_2^{\mathcal{A}}$.*

Proof. Let L be a set in $\mathcal{C}_1^{\mathcal{A}}$ that is $\leqslant_T^{p,\mathcal{A}}$-hard for $\mathcal{C}_2^{\mathcal{A}}$. Then for every $L' \in \mathcal{C}_2^{\mathcal{A}}$, $L' \in \mathrm{P}^{L \oplus \mathcal{A}}$. Since $\mathcal{C}_1^{\mathcal{A}}$ is closed under join operation and since $\mathcal{A} \in \mathcal{C}_1^{\mathcal{A}}$, it follows that $L \oplus \mathcal{A}$ is in $\mathcal{C}_1^{\mathcal{A}}$. Hence, $L \oplus \mathcal{A} \in \mathcal{C}_1^{\mathcal{A}}$ is $\leqslant_T^{p}$-hard for $\mathcal{C}_2^{\mathcal{A}}$.

The other direction also follows easily because for any $\mathcal{A} \subseteq \Sigma^*$, the $\leqslant_T^{p}$-hardness of a set for $\mathcal{C}_2^{\mathcal{A}}$ implies the hardness of the set under $\leqslant_T^{p,\mathcal{A}}$ reduction for $\mathcal{C}_2^{\mathcal{A}}$. □

The proof of Theorem 4.6, which is one of the main results of this section, uses Lemmas 4.4 and 4.5. We mention that Hemaspaandra, Jain, and Vereshchagin [26] proved, using a different combinatorial technique, that relative to an oracle, FewP contains no polynomial-time Turing hard set for UP ∩ coUP. Theorem 4.6 extends this result and implies that there is a relativized world where SPP has no polynomial-time many-one or Turing complete sets. That answers positively a question raised by Hemaspaandra, Ramachandran, and Zimand [28].

The following lemma is central to our oracle constructions involving the class AWPP.

Lemma 4.5. *Let $\mathcal{O} \subseteq \Sigma^*$ and let (N, q) be an arbitrary* AWPP *pair with polynomial p bounding the running time of N. Fix an arbitrary $x \in \Sigma^*$. Let C be a subset of Σ^* such that the following are true*:

(1) *$(N^{\mathcal{O} \cup A}, q)$ is a valid* $\mathrm{AWPP}^{\mathcal{O} \cup A}$ *pair for every $A \subseteq C$.*
(2) *$x \in L(N^{\mathcal{O} \cup \{\alpha\}}, q) \Leftrightarrow x \notin L(N^{\mathcal{O}}, q)$, for every $\alpha \in C$.*

Then $\|C\| \leqslant 6p(|x|)^2$.

Proof. W.l.o.g. assume that $x \notin L(N^{\mathcal{O}}, q)$. Let

$$C =_{\mathrm{df}} \{\alpha \in \Sigma^* \mid x \in L(N^{\mathcal{O} \cup \{\alpha\}}, q)\}.$$

To get a contradiction, suppose that $k =_{\mathrm{df}} \|C\| > 6p(|x|)^2$. Let $s \in \mathbb{Z}[y_1, y_2, \ldots, y_k]$ be the polynomial encoding of $N^{\mathcal{O}}(x)$ w.r.t. C. From Proposition 2.6 it is easy to see that s satisfies the following properties:

(1) For every $y_1, y_2, \ldots, y_k \in \{0, 1\}$, $s(y_1, y_2, \ldots, y_k)/2^{q(|x|)} \in [0, 1/3] \cup [2/3, 1]$.
(2) $s(0, 0, \ldots, 0)/2^{q(|x|)} \in [0, 1/3]$.
(3) $s(y_1, y_2, \ldots, y_k)/2^{q(|x|)} \in [2/3, 1]$ for every $y_1, y_2, \ldots, y_k \in \{0, 1\}$ with $\sum_{i=1}^{k} y_i = 1$.
(4) $\deg(s) \leqslant p(|x|)$.

Let f be the boolean function defined by

- $f(y_1, y_2, \ldots, y_k) = 0 \Leftrightarrow s(y_1, y_2, \ldots, y_k)/2^{q(|x|)} \in [0, 1/3]$, and
- $f(y_1, y_2, \ldots, y_k) = 1 \Leftrightarrow s(y_1, y_2, \ldots, y_k)/2^{q(|x|)} \in [2/3, 1]$.

Hence $f(0, 0, \ldots, 0) = 0$, and for every boolean vector $\vec{y}$ of Hamming weight 1, $f(\vec{y}) = 1$. It follows from Lemma 4.2 that $\widetilde{\deg}(f) \geqslant \sqrt{k/6}$. On the other hand, it is easy to see that polynomial s approximates f in the sense of Definition 4.1. Therefore $\widetilde{\deg}(f) \leqslant \deg(s) \leqslant p(|x|) < \sqrt{k/6}$. A contradiction. □

Theorem 4.6. *There exists an oracle $\mathcal{A}$ such that* $\mathrm{AWPP}^{\mathcal{A}}$ *has no $\leqslant_T^{p,\mathcal{A}}$-hard set for* $\mathrm{UP}^{\mathcal{A}} \cap \mathrm{coUP}^{\mathcal{A}}$.

Proof. Let (N_i, q_j, M_k) be an enumeration of tuples where N_i is a nondeterministic polynomial-time oracle Turing machine, q_j is a polynomial, and M_k is a deterministic polynomial-time oracle Turing machine. For each AWPP pair (N_i, q_j), we define our test language as follows:

$$L_{\langle i,j\rangle}(B) = \left\{0^n \mid n \text{ is a power of the } \langle i,j\rangle\text{th prime number and } \left\|B \cap 0\Sigma^n\right\| \neq 0\right\}.$$

Since AWPP is closed under join operation in every relativized world, by Lemma 4.4 it suffices to construct an oracle $\mathcal{A}$ such that $\mathrm{AWPP}^{\mathcal{A}}$ has no $\leqslant_T^p$-hard set for $\mathrm{UP}^{\mathcal{A}} \cap \mathrm{coUP}^{\mathcal{A}}$. The oracle $\mathcal{A}$ is constructed in stages. Initially, $\mathcal{A} := \{0\}^*$. In stage $\langle i, j, k\rangle$, we diagonalize against tuple (N_i, q_j, M_k) and modify oracle $\mathcal{A}$ at some length.

Stage $\langle i, j, k\rangle$: Let $r(\cdot)$ be a polynomial that bounds the running time of both N_i and M_k. Choose an integer n satisfying the following requirements: (a) n is a power of the $\langle i, j\rangle$th prime number, (b) $2^n > 6 \cdot r(n) \cdot r(r(n))^2$, (c) n is large enough so that n satisfies any promises made in the previous stages and no string of length greater than or equal to n is queried in any of the previous stages, and (d) n is larger than the value of n in the previous stage. Let $\mathcal{A} := \mathcal{A} - \{0^{n+1}\}$.

Consider $M_k(0^n)$ with oracle $L(N_i^{\mathcal{A}}, q_j)$. Let $\beta_1, \beta_2, \ldots, \beta_\ell$, where $0 \leqslant \ell \leqslant r(n)$, be the sequence of queries asked by $M_k(0^n)$ to the oracle $L(N_i^{\mathcal{A}}, q_j)$.

If there exists a set $B \subseteq \Sigma^{n+1}$ such that $(N_i^{\mathcal{A}\cup B}, q_j)$ is not a valid $\mathrm{AWPP}^{\mathcal{A}\cup B}$ pair, then set $\mathcal{A} := \mathcal{A} \cup B$. This may cause the test language $L_{\langle i,j\rangle}(\mathcal{A})$ not to be in $\mathrm{UP}^{\mathcal{A}} \cap \mathrm{coUP}^{\mathcal{A}}$. But this is no problem because $L_{\langle i,j\rangle}(\mathcal{A})$ is only defined to witness that the (now invalid)

$\mathrm{AWPP}^{\mathcal{A}}$ pair $(N_i^{\mathcal{A}}, q_j)$ does not constitute a $\leqslant_T^p$-hard set for $\mathrm{UP}^{\mathcal{A}} \cap \mathrm{coUP}^{\mathcal{A}}$. We can move to the next stage. But we have to make sure that $\mathrm{AWPP}^{\mathcal{A}}$ pair $(N_i^{\mathcal{A}}, q_j)$ does not become valid in later stages. Therefore, we promise to choose the value of n in the next stage to be larger than $r(|w|)$, where w is an arbitrary input string that makes $\mathrm{AWPP}^{\mathcal{A}}$ pair $(N_i^{\mathcal{A}}, q_j)$ invalid, and then move to the next stage.

Otherwise, we proceed with the following claim.

Claim 3. *There exists a string $z_0 \in 0\Sigma^n$ ($z_1 \in 1\Sigma^n$) that can be added to $\mathcal{A}$ without changing the answers of the* $\mathrm{AWPP}^{\mathcal{A}}$ *pair $(N_i^{\mathcal{A}}, q_j)$ to the queries $\beta_1, \beta_2, \ldots, \beta_\ell$, and hence without changing the acceptance behavior of $M_k(0^n)$.*

Let us assume that the claim is true. If $M_k(0^n)$ with oracle $L(N_i^{\mathcal{A}}, q_j)$ accepts, then set $\mathcal{A} := \mathcal{A} \cup \{z_1\}$. If $M_k(0^n)$ with oracle $L(N_i^{\mathcal{A}}, q_j)$ rejects, then set $\mathcal{A} := \mathcal{A} \cup \{z_0\}$. Move to the next stage.

End of Stage

It is easy to see that one of the following is true for each AWPP pair (N_i, q_j).

(1) $(N_i^{\mathcal{A}}, q_j)$ violates the promise of a valid $\mathrm{AWPP}^{\mathcal{A}}$ pair at some stage of oracle construction, or

(2) $L_{\langle i,j\rangle}(\mathcal{A})$ is in $\mathrm{UP}^{\mathcal{A}} \cap \mathrm{coUP}^{\mathcal{A}}$ but for each $k \in \mathbb{N}$, there exists $x \in \Sigma^*$ such that $x \in L_{\langle i,j\rangle}(\mathcal{A}) \Leftrightarrow x \notin L(M_k^{L(N_i^{\mathcal{A}}, q_j)})$. This ensures that in case $(N_i^{\mathcal{A}}, q_j)$ constitutes a valid $\mathrm{AWPP}^{\mathcal{A}}$ pair, then $L_{\langle i,j\rangle}(\mathcal{A}) \not\leqslant_T^p L(N_i^{\mathcal{A}}, q_j)$ and so $L(N_i^{\mathcal{A}}, q_j)$ cannot be $\leqslant_T^p$-hard for $\mathrm{UP}^{\mathcal{A}} \cap \mathrm{coUP}^{\mathcal{A}}$.

It is clear that if each AWPP pair (N_i, q_j) fulfills one of these requirements, then $\mathrm{AWPP}^{\mathcal{A}}$ has no $\leqslant_T^p$-hard set for $\mathrm{UP}^{\mathcal{A}} \cap \mathrm{coUP}^{\mathcal{A}}$. This completes the proof of Theorem 4.6. □

Proof of Claim 3. We prove only the existence of a string $z_0 \in 0\Sigma^n$ satisfying the conditions of the claim; the existence of a string $z_1 \in 1\Sigma^n$, as promised in the claim, can be proved similarly. For any string β_e ($1 \leqslant e \leqslant \ell$), let

$$C(\beta_e) = \{\alpha \in 0\Sigma^n \mid \beta_e \in L(N_i^{\mathcal{A} \cup \{\alpha\}}, q_j) \Leftrightarrow \beta_e \notin L(N_i^{\mathcal{A}}, q_j)\}.$$

Apply Lemma 4.5 with $\mathcal{O} := \mathcal{A}$ and $x := \beta_e$. Since $C(\beta_e)$ satisfies the conditions of the lemma, we obtain $\|C(\beta_e)\| \leqslant 6 \cdot r(r(n))^2$.

Because $2^n > 6 \cdot r(n) \cdot r(r(n))^2 \geqslant 6 \cdot \ell \cdot r(r(n))^2$, we can find a string $z_0 \in 0\Sigma^n \setminus (C(\beta_1) \cup C(\beta_2) \cup \cdots \cup C(\beta_\ell))$, which satisfies the conditions of the claim. □

Corollary 4.7. *There is an oracle $\mathcal{A}$ such that for every complexity class $\mathcal{C} \in$* {UP ∩ coUP, UP, FewP, Few, SPP, LWPP, WPP, AWPP}*, $\mathcal{C}^{\mathcal{A}}$ has no $\leqslant_T^{p,\mathcal{A}}$-complete set.*

We next construct in Theorem 4.8 a relativized world where AWPP has no polynomial-time Turing hard set for ZPP. We essentially use an extension of the ideas used in the proof of Theorem 4.6 for proving this result.

Theorem 4.8. $(\exists\mathcal{A})$ [$\mathrm{AWPP}^{\mathcal{A}}$ *has no* $\leqslant_T^{p,\mathcal{A}}$*-hard set for* $\mathrm{ZPP}^{\mathcal{A}}$].

Proof. The proof is similar to the one of Theorem 4.6. Let (N_i, q_j, M_k) and the test language $L_{\langle i,j\rangle}(B)$ be defined as in the proof of Theorem 4.6. For each $B \subseteq \Sigma^*$ and $n, \xi \in \mathbb{N}$, we define predicates "Zeros" and "Ones" as follows.

$$\mathrm{Zeros}(B,n,\xi) \equiv B \subseteq 0\Sigma^n \quad \text{and} \quad \|B\| > \xi,$$
$$\mathrm{Ones}(B,n,\xi) \equiv B \subseteq 1\Sigma^n \quad \text{and} \quad \|B\| > \xi.$$

Since AWPP is closed under join operation in every relativized world, by Lemma 4.4 it suffices to construct an oracle $\mathcal{A}$ such that $\mathrm{AWPP}^{\mathcal{A}}$ has no $\leqslant_T^p$-hard set for $\mathrm{ZPP}^{\mathcal{A}}$. The oracle $\mathcal{A}$ is constructed in stages. Initially, $\mathcal{A} := 0\Sigma^*$. In stage $\langle i,j,k\rangle$, we diagonalize against tuple (N_i, q_j, M_k) and modify oracle $\mathcal{A}$ at some length. The details are as follows.

Stage $\langle i, j, k\rangle$: Let $r(\cdot)$ be a polynomial that bounds the running time of both N_i and M_k. Choose an integer n satisfying the following requirements: (a) n is a power of the $\langle i,j\rangle$th prime number, (b) $2^{n-1} > 6 \cdot r(n) \cdot r(r(n))^2$, (c) n is large enough so that n satisfies any promises made in the previous stages and no string of length greater than or equal to n is queried in any of the previous stages, and (d) n is larger than the value of n in the previous stage. Let $\mathcal{A} := \mathcal{A} - \Sigma^{n+1}$.

Consider $M_k(0^n)$ with oracle $L(N_i^{\mathcal{A}}, q_j)$. Let $\beta_1, \beta_2, \ldots, \beta_\ell$, where $0 \leqslant \ell \leqslant r(n)$, be the sequence of queries asked by $M_k(0^n)$ to the oracle $L(N_i^{\mathcal{A}}, q_j)$.

If there exists a set $B \subseteq \Sigma^{n+1}$ such that $(N_i^{\mathcal{A}\cup B}, q_j)$ is not a valid $\mathrm{AWPP}^{\mathcal{A}\cup B}$ pair, then set $\mathcal{A} := \mathcal{A} \cup B$. Move to the next stage with the promise to choose the value of n in the next stage to be larger than $r(|w|)$, where w is an arbitrary input string that makes $\mathrm{AWPP}^{\mathcal{A}}$ pair $(N_i^{\mathcal{A}}, q_j)$ invalid.

Otherwise, proceed with the following claim.

Claim 4. *There exist sets B_0 and B_1 such that* (a) $\mathrm{Zeros}(B_0, n, 2^{n-1})$ *and* $\mathrm{Ones}(B_1, n, 2^{n-1})$ *are true, and* (b) B_γ $(\gamma \in \{0,1\})$ *can be added to $\mathcal{A}$ without changing the answers of the* $\mathrm{AWPP}^{\mathcal{A}}$ *pair $(N_i^{\mathcal{A}}, q_j)$ to the queries $\beta_1, \beta_2, \ldots, \beta_\ell$, and hence without changing the acceptance behavior of $M_k(0^n)$.*

Let us assume that the claim is true. If $M_k(0^n)$ with oracle $L(N_i^{\mathcal{A}}, q_j)$ accepts, then set $\mathcal{A} := \mathcal{A} \cup B_1$. If $M_k(0^n)$ with oracle $L(N_i^{\mathcal{A}}, q_j)$ rejects, then set $\mathcal{A} := \mathcal{A} \cup B_0$. Move to the next stage.

End of Stage

The correctness of the construction is as in the proof of Theorem 4.6. This completes the proof of Theorem 4.8. □

Proof of Claim 4. We prove only the existence of a set B_0 satisfying the conditions of the claim; a similar proof for the existence of a set B_1, as promised in the claim, can be given. The proof is by iteration of the idea in the proof of Claim 3. First apply Lemma 4.5 with $\mathcal{O} := \mathcal{A}$ to claim the existence of a string $z_0 \in 0\Sigma^n$ that can be added to $\mathcal{A}$ without changing the answers of $(N_i^{\mathcal{A}}, q_j)$ to the queries $\beta_1, \beta_2, \ldots, \beta_\ell$. Next apply Lemma 4.5 with $\mathcal{O} := \mathcal{A} \cup \{z_0\}$ to claim the existence of a string $z_0' \in 0\Sigma^n$ that can be added to $\mathcal{A} \cup \{z_0\}$ without changing the answers of $(N_i^{\mathcal{A}\cup\{z_0\}}, q_j)$, and hence of $(N_i^{\mathcal{A}}, q_j)$, to the queries $\beta_1, \beta_2, \ldots, \beta_\ell$. Because $2^{n-1} > 6 \cdot r(n) \cdot r(r(n))^2 \geqslant 6 \cdot \ell \cdot r(r(n))^2$, we can add 2^{n-1} strings to $\mathcal{A}$, one after the other in this manner, always without changing the answers of $(N_i^{\mathcal{A}}, q_j)$ to the queries $\beta_1, \beta_2, \ldots, \beta_\ell$. □

Corollary 4.9. [19,24,26] *There is an oracle $\mathcal{A}$ such that for every class $\mathcal{C} \in$ {ZPP, RP, coRP, BPP, BQP}, $\mathcal{C}^{\mathcal{A}}$ has no $\leqslant_T^{p,\mathcal{A}}$-complete set.*

Note. An alternative proof of Theorems 4.6 and 4.8 can be obtained using a lemma by Vereshchagin [43,44] on proving whether a complexity class has a polynomial-time Turing hard set for another complexity class. Fortnow and Rogers [19] used this lemma to prove that BQP has no polynomial-time Turing hard set for BPP in some relativized world.

5. Relativized noninclusion

Beigel [9] constructed an oracle relative to which $\mathrm{P}^{\mathrm{NP}} \not\subseteq \mathrm{PP}$. As a consequence, there is a relativized world in which NP is not low for PP. However, in contrast to NP, it is not clear whether $\mathrm{NP} \cap \mathrm{coNP}$ is not low for PP in some relativized world. Spakowski, Thakur, and Tripathi [37] showed that there is an oracle relative to which ZPP is not contained in WPP, a class known to be low for PP. Thus it follows that relative to the same oracle, $\mathrm{NP} \cap \mathrm{coNP} \not\subseteq \mathrm{WPP}$. In Theorem 5.2, we extend this result and show that there is a relativized world in which $\mathrm{NP} \cap \mathrm{coNP} \not\subseteq \mathrm{AWPP}$, where AWPP is a class known to be low for PP. This supports our belief that $\mathrm{NP} \cap \mathrm{coNP}$ might not be low for PP in a suitable relativized world.

We use the following lemma by Ehlich and Zeller [14] and Rivlin and Cheney [32] to lower bound the degree of univariate polynomials that satisfy certain constraints. This is a standard technique (see, e.g., [7,9,29]).

Lemma 5.1. [14,32] *Let $p \in \mathbb{R}[y]$ be a univariate polynomial with the following properties*:

(1) *for every integer ℓ with $0 \leqslant \ell \leqslant N$, $b_1 \leqslant p(\ell) \leqslant b_2$, and*
(2) *for some real $0 \leqslant z \leqslant N$, the derivative of p satisfies $|p'(z)| \geqslant c$.*

Then $\deg(p) \geqslant \sqrt{cN/(c + b_2 - b_1)}$.

Theorem 5.2. $(\exists \mathcal{A})$ $[\mathrm{NP}^{\mathcal{A}} \cap \mathrm{coNP}^{\mathcal{A}} \not\subseteq \mathrm{AWPP}^{\mathcal{A}}]$.

Proof. Let (N_i, q_j) be an enumeration of pairs, where N_i is a nondeterministic polynomial-time oracle Turing machine and q_j is a polynomial. The test language $L(B)$ is defined by

$$L(B) = \{0^n \mid \|B \cap 0\Sigma^n\| \neq 0\}.$$

We will construct an oracle $\mathcal{A}$ in stages such that for each $n \in \mathbb{N}^+$, either $\emptyset \subset \mathcal{A}^{=n+1} \subseteq 0\Sigma^n$ or $\emptyset \subset \mathcal{A}^{=n+1} \subseteq 1\Sigma^n$ holds. This ensures that $L(\mathcal{A})$ is in $\mathrm{NP}^{\mathcal{A}} \cap \mathrm{coNP}^{\mathcal{A}}$. Initially, $\mathcal{A} := 0\Sigma^*$. In stage $\langle i, j\rangle$, we diagonalize against pair (N_i, q_j) and modify $\mathcal{A}$ at some length. We now give a description of stage $\langle i, j\rangle$.

Stage $\langle i, j\rangle$: Let $r(\cdot)$ be a polynomial that bounds the running time of N_i. Choose n large enough so that (a) $2^n > 7 \cdot r(n)^2$, (b) no machine considered in the previous stages queries a string of length n or more, and (c) n is larger than the value of n in the previous stage. Let $\mathcal{A} := \mathcal{A} - \Sigma^{n+1}$.

If there exists a nonempty set $B \subseteq 0\Sigma^n$ or $B \subseteq 1\Sigma^n$ such that $\mathrm{gap}_{N_i^{\mathcal{A}\cup B}}(0^n)/2^{q_j(n)} \notin [0, 1/3] \cup [2/3, 1]$, then set $\mathcal{A} := \mathcal{A} \cup B$ and move to the next stage.

Otherwise, the following claim applies.

Claim 5. *There exists a nonempty set $B \subseteq \Sigma^{n+1}$ such that the following holds*:

$$B \subseteq 0\Sigma^n \quad \textit{and} \quad \mathrm{gap}_{N_i^{\mathcal{A}\cup B}}\big(0^n\big)/2^{q_j(n)} \in [0, 1/3], \quad \textit{or}$$

$$B \subseteq 1\Sigma^n \quad \textit{and} \quad \mathrm{gap}_{N_i^{\mathcal{A}\cup B}}\big(0^n\big)/2^{q_j(n)} \in [2/3, 1].$$

Let us assume that the claim is true. Take such a set B. Set $\mathcal{A} := \mathcal{A} \cup B$. Move to the next stage.

End of Stage

Clearly, $L(\mathcal{A}) \in \mathrm{NP}^{\mathcal{A}} \cap \mathrm{coNP}^{\mathcal{A}}$ and one of the following is true for each AWPP pair (N_i, q_j).

(1) $(N_i^{\mathcal{A}}, q_j)$ violates the promise of a valid $\mathrm{AWPP}^{\mathcal{A}}$ pair, or
(2) $(N_i^{\mathcal{A}}, q_j)$ is a valid $\mathrm{AWPP}^{\mathcal{A}}$ pair, but there exists a length n such that

$$0^n \in L(\mathcal{A}) \quad \Leftrightarrow \quad 0^n \notin L\big(N_i^{\mathcal{A}}, q_j\big).$$

Thus it follows that $L(\mathcal{A}) \in \mathrm{NP}^{\mathcal{A}} \cap \mathrm{coNP}^{\mathcal{A}}$ but $L(\mathcal{A}) \notin \mathrm{AWPP}^{\mathcal{A}}$. This completes the proof of Theorem 5.2. □

Proof of Claim 5. Assume to the contrary that no set $B \subseteq \Sigma^{n+1}$ satisfies the conditions of the claim. Then the following holds:

$$\emptyset \subset B \subseteq 0\Sigma^n \quad \Rightarrow \quad \mathrm{gap}_{N_i^{\mathcal{A}\cup B}}\big(0^n\big)/2^{q_j(n)} \in [2/3, 1], \quad \text{and} \tag{3}$$

$$\emptyset \subset B \subseteq 1\Sigma^n \quad \Rightarrow \quad \mathrm{gap}_{N_i^{\mathcal{A}\cup B}}\big(0^n\big)/2^{q_j(n)} \in [0, 1/3]. \tag{4}$$

We will show that Statement (3) implies

$$\mathrm{gap}_{N_i^{\mathcal{A}}}\big(0^n\big)/2^{q_j(n)} \geqslant 3/5. \tag{5}$$

By an analogous proof, it can be shown that Statement (4) implies $\mathrm{gap}_{N_i^{\mathcal{A}}}(0^n)/2^{q_j(n)} \leqslant 2/5$, which gives a contradiction with Statement (5).

Suppose that $g =_{\mathrm{df}} \mathrm{gap}_{N_i^{\mathcal{A}}}(0^n)/2^{q_j(n)} < 3/5$. Let $s' \in \mathbb{Z}[y_1, y_2, \ldots, y_{2^n}]$ be the polynomial encoding of $N_i^{\mathcal{A}}(0^n)$ w.r.t. $0\Sigma^n$. Define $s \in \mathbb{R}[y_1, y_2, \ldots, y_{2^n}]$ as follows:

$$s(y_1, y_2, \ldots, y_{2^n}) = \frac{1}{2^{q_j(n)}} \cdot s'(y_1, y_2, \ldots, y_{2^n}).$$

It is easy to verify that $s(y_1, y_2, \ldots, y_{2^n})$ satisfies the following properties:

- For each $y_1, y_2, \ldots, y_{2^n} \in \{0, 1\}$ such that $\sum_{\ell=1}^{2^n} y_\ell \geqslant 1$, $s(y_1, y_2, \ldots, y_{2^n}) \in [2/3, 1]$.
- $s(0, 0, \ldots, 0) = g < 3/5$.
- $\deg(s) \leqslant r(n)$.

We follow closely the proof of Nisan and Szegedy [29, Lemma 3.5]. Let $\tilde{s}$ be the univariate polynomial giving the symmetrization of s. Polynomial $\tilde{s}$ satisfies the following properties:

(1) $\deg(\tilde{s}) \leqslant \deg(s) \leqslant r(n)$.
(2) For every integer ℓ with $0 \leqslant \ell \leqslant 2^n$, $g \leqslant \tilde{s}(\ell) \leqslant 1$.
(3) $\tilde{s}(0) = g$.
(4) $\tilde{s}(1) \geqslant 2/3$.

Properties (3) and (4) together imply that for some real $0 \leqslant z \leqslant 1$, the derivative $\tilde{s}'(z) \geqslant 2/3 - g$. We can now apply Lemma 5.1 and obtain

$$\deg(\tilde{s}) \geqslant \sqrt{\frac{(2/3 - g) \cdot 2^n}{(2/3 - g) + 1 - g}} = \sqrt{\frac{2^n}{1 + \frac{1-g}{2/3-g}}} \geqslant \frac{2^{n/2}}{\sqrt{7}},$$

which contradicts the property (1) of $\tilde{s}$.

Analogously (using the polynomial encoding of $N_i^{\mathcal{A}}(0^n)$ w.r.t. $1\Sigma^n$) it can be shown that Statement (4) implies $\mathrm{gap}_{N_i^{\mathcal{A}}}(0^n)/2^{q_j(n)} \leqslant 2/5$, which gives the desired contradiction. This completes the proof of Claim 5. □

Certain classes are known to be not very powerful in some relativized worlds, however their composition with themselves are found to be more powerful classes in every relativized world. For instance, Spakowski, Thakur, and Tripathi [37] showed the existence of a relativized world in which RP is immune to $\mathrm{C_{=}P}$. But $\mathrm{C_{=}P^{C_{=}P}}$ is known to contain the polynomial hierarchy in every relativized world. In fact, in every relativized world, $\mathrm{UP^{C_{=}P}}$ and $\mathrm{ZPP^{C_{=}P}}$, which are subclasses of $\mathrm{C_{=}P^{C_{=}P}}$, contain the polynomial hierarchy. Using Torán's [42] combinatorial technique, Spakowski, Thakur, and Tripathi [37] constructed an oracle relative to which $\mathrm{ZPP} \nsubseteq \mathrm{WPP}$. Corollary 3.8 shows that there is a relativized world where WPP is not self-low, and so we cannot conclude directly from their result that ZPP is not contained in $\mathrm{WPP^{WPP}}$ relative to an oracle. Therefore, we are interested in whether or not WPP shows a similar behavior as its superclass $\mathrm{C_{=}P}$, i.e. whether $\mathrm{WPP^{WPP}}$ is as big a class as to contain the polynomial hierarchy in every relativized world. Theorem 5.8 shows that this is not the case by stating a relativized world in which ZPP is not contained in $\mathrm{WPP^{WPP}}$. For the proof, we will need Lemmas 5.4, 5.5, 5.6, and 5.7. Below, we state the idea of the proof.

Proof Idea: The proof of Theorem 5.8 is in two steps and the idea is as follows. Let $(N_{i_1}, M_{j_1}, N_{i_2}, M_{j_2})$ be a tuple of machines at some stage of oracle construction, where we treat (N_{i_1}, M_{j_1}) as a base WPP pair and treat (N_{i_2}, M_{j_2}) as a WPP pair acting as an oracle to (N_{i_1}, M_{j_1}). In the first step, we express the dependency on an oracle segment of the acceptance behavior of WPP pair (N_{i_2}, M_{j_2}) on any input w by a low degree multilinear polynomial p_w with variables corresponding to the strings of the oracle segment. This step is identified in Lemma 5.5. In the second step, we express the acceptance behavior of WPP pair (N_{i_1}, M_{j_1}) on input 0^n with access to the oracle defined by the $\mathrm{WPP}^{(\cdot)}$ pair $(N_{i_2}^{(\cdot)}, M_{j_2}^{(\cdot)})$ by a low degree multilinear polynomial in which variables are substituted by low degree polynomials obtained from the first step. We identify this step in Lemma 5.6. Since the composition of low degree polynomials is a low degree polynomial, we finally obtain a low degree polynomial that satisfies certain conditions. Using Lemma 5.7, we obtain the desired result.

Definition 5.3. For any nondeterministic oracle Turing machine N, deterministic oracle transducer M, $A \subseteq \Sigma^*$, and $w \in \Sigma^*$, we say that $\mathrm{Valid}(N^A, M^A, w)$ is true if it holds that $M^A(w) \neq 0$ and $\mathrm{gap}_{N^A}(w) \in \{0, M^A(w)\}$.

Lemma 5.4. *Let M be a deterministic oracle transducer with running time $t(\cdot)$ and let $w \in \Sigma^*$. Let $x_1, x_2, \ldots, x_m$ be the lexicographic enumeration of all strings up to length $t(|w|)$. There is a multilinear polynomial $p \in \mathbb{Q}[y_1, y_2, \ldots, y_m]$ having the following properties*:

(1) *for every $A \subseteq \Sigma^*$ such that $M^A(w) \neq 0$, $p(\chi_A(x_1), \chi_A(x_2), \ldots, \chi_A(x_m)) = 1/M^A(w)$, and*
(2) $\deg(p) \leqslant t(|w|)$.

Proof. For every potential computation path ρ of $M^{(\cdot)}$ on input w, i.e. computation path ρ of M^A on input w for some arbitrary oracle A, create $\mathrm{mono}(\rho)$ as in Definition 2.5 with $\mathcal{O} := \emptyset$ and $\mathcal{T} := (\Sigma^*)^{\leqslant t(|w|)}$. Let $val(\rho)$ be the value output by M on path ρ. Define

$$p(y_1, y_2, \ldots, y_m) = \sum_{\text{path } \rho:\ val(\rho) \neq 0} \frac{\mathrm{mono}(\rho)}{val(\rho)}. \qquad \square$$

Lemma 5.5. *Let N be a nondeterministic oracle Turing machine, M be a deterministic oracle transducer, both running in time $t(\cdot)$, and let $w \in \Sigma^*$. Let $x_1, x_2, \ldots, x_m$ be the lexicographic enumeration of all strings up to length $t(|w|)$. There is a multilinear polynomial $p_w \in \mathbb{Q}[y_1, y_2, \ldots, y_m]$ having the following properties*:

(1) *For every $A \subseteq \Sigma^*$ such that* $\mathrm{Valid}(N^A, M^A, w)$ *is true, it holds that*

$$p_w\big(\chi_A(x_1), \chi_A(x_2), \ldots, \chi_A(x_m)\big) = \begin{cases} 1 & \textit{if } \mathrm{gap}_{N^A(w)} = M^A(w), \textit{ and} \\ 0 & \textit{if } \mathrm{gap}_{N^A(w)} = 0. \end{cases}$$

(2) $\deg(p_w) \leqslant 2t(|w|)$.

Proof. Let p_1 be a polynomial representing $\mathrm{gap}_{N^A}(w)$ as in Definition 2.5 with $\mathcal{O} := \emptyset$ and $\mathcal{T} := (\Sigma^*)^{\leqslant t(|w|)}$. Let p_2 be a polynomial representing $1/M^A(w)$ as in Lemma 5.4. Then we get the required polynomial p_w by setting $p_w = p_1 \cdot p_2$. Clearly, $\deg(p) \leqslant 2t(|w|)$. □

Lemma 5.6. *Let N_1, N_2 be nondeterministic oracle Turing machines, M_1, M_2 be deterministic oracle transducers, all with running time $t(\cdot)$, and let $w \in \Sigma^*$. Let $x_1, x_2, \ldots, x_m$ be the lexicographic enumeration of all strings up to length $t(t(|w|))$. There is a multilinear polynomial $p \in \mathbb{Q}[y_1, y_2, \ldots, y_m]$ of total degree $\leqslant 4t(|w|) \cdot t(t(|w|))$ having the following property: For every $A \subseteq \Sigma^*$ satisfying*

(1) $\mathrm{Valid}(N_2^A, M_2^A, v)$ *is true for every $v \in \Sigma^*$, and*
(2) $\mathrm{Valid}(N_1^{L(N_2^A, M_2^A)}, M_1^{L(N_2^A, M_2^A)}, w)$ *is true,*

it holds that

$$p\big(\chi_A(x_1), \chi_A(x_2), \ldots, \chi_A(x_m)\big) = \begin{cases} 1 & \textit{if}\ \mathrm{gap}_{N_1^{L(N_2^A, M_2^A)}}(w) = M_1^{L(N_2^A, M_2^A)}(w), \\ 0 & \textit{if}\ \mathrm{gap}_{N_1^{L(N_2^A, M_2^A)}}(w) = 0. \end{cases}$$

Proof. Apply Lemma 5.5 to get the polynomials $p_{x_1}, p_{x_2}, \ldots, p_{x_m}$ that encode the computations of (N_2, M_2) on inputs $x_1, x_2, \ldots, x_m$, respectively. The total degree of each of these polynomials is $\leqslant 2t(t(|w|))$. Apply Lemma 5.5 to get the polynomial p_w that encodes the computation of the base machine (N_1, M_1) on input w. Clearly, $\deg(p_w) \leqslant 2t(|w|)$.

To get the desired polynomial $p(y_1, y_2, \ldots, y_m)$, take $p_w(y_1, y_2, \ldots, y_m)$ and substitute every variable y_i by the corresponding polynomial p_{x_i}. Clearly, $\deg(p) \leqslant 4t(|w|) \cdot t(t(|w|))$. □

In the proof of Theorem 5.8, we use the following lemma by Tarui [39], which states that if a multilinear polynomial is zero on a certain large collection of inputs over a boolean domain, then the polynomial itself is a zero polynomial.

Lemma 5.7. [39] *Let $\mathcal{R}$ be a ring. Let s be a multilinear polynomial in $\mathcal{R}[y_1, y_2, \ldots, y_N]$ of total degree at most d and let i be a nonnegative integer such that $i + d \leqslant N$ and $s(y_1, y_2, \ldots, y_N) = 0$ for each $y_1, y_2, \ldots, y_N \in \{0, 1\}$ satisfying $i \leqslant \sum_{j=1}^{N} y_j \leqslant i + d$. Then $s \equiv 0$.*

Theorem 5.8. $(\exists \mathcal{A})\ [\mathrm{ZPP}^{\mathcal{A}} \not\subseteq \mathrm{WPP}^{\mathrm{WPP}^{\mathcal{A}}}]$.

Proof. Let the predicates "Zeros" and "Ones" be defined as in the proof of Theorem 4.8. The test language L_B is defined by

$$L_B = \big\{0^n \mid \big\|B \cap 0\Sigma^n\big\| \neq 0\big\}.$$

We will construct an oracle $\mathcal{A}$ such that for each $n \geqslant 1$, either $\mathrm{Zeros}(\mathcal{A}^{=n+1}, n, 2^{n-1})$ is true or $\mathrm{Ones}(\mathcal{A}^{=n+1}, n, 2^{n-1})$ is true. This will guarantee that $L_{\mathcal{A}}$ is in $\mathrm{ZPP}^{\mathcal{A}}$. Let

$(N_{i_1}, M_{j_1}, N_{i_2}, M_{j_2})$ be an enumeration of tuples where N_{i_1} and N_{i_2} are nondeterministic polynomial-time oracle Turing machines, and M_{j_1} and M_{j_2} are deterministic polynomial-time oracle transducers. Initially, $\mathcal{A} := 0\Sigma^*$. In stage $\langle i_1, j_1, i_2, j_2\rangle$, we diagonalize against $(N_{i_1}, M_{j_1}, N_{i_2}, M_{j_2})$, treating (N_{i_1}, M_{j_1}) as a base WPP pair and treating (N_{i_2}, M_{j_2}) as a WPP pair acting as an oracle to (N_{i_1}, M_{j_1}), and modify oracle $\mathcal{A}$ at some length. The details are as follows.

Stage $\langle i_1, j_1, i_2, j_2\rangle$: Let $r(\cdot)$ be a polynomial that bounds the running time of each of N_{i_1}, M_{j_1}, N_{i_2}, and M_{j_2}. Choose n large enough such that the previous stages are not affected, $2^n > 8r(n) \cdot r(r(n))$, and n is larger than the value of n in the previous stage. Let $\mathcal{A} := \mathcal{A} - \Sigma^{n+1}$. Perform the following three steps.

(1) Look for a set $B \subseteq \Sigma^{n+1}$ such that either $\mathrm{Zeros}(B, n, 2^{n-1})$ is true or $\mathrm{Ones}(B, n, 2^{n-1})$ is true, and the following holds: There is a string $w \in \Sigma^*$ such that $\mathrm{Valid}(N_{i_2}^{\mathcal{A}\cup B}, M_{j_2}^{\mathcal{A}\cup B}, w)$ is not true. If such a set B exists, then set $\mathcal{A} := \mathcal{A} \cup B$ and move to the next stage. Otherwise, go to step (2).
(2) Look for a set $B \subseteq \Sigma^{n+1}$ such that either $\mathrm{Zeros}(B, n, 2^{n-1})$ is true or $\mathrm{Ones}(B, n, 2^{n-1})$ is true, and the following holds: There is a string $w \in \Sigma^*$ such that $\mathrm{Valid}(N_{i_1}^{L(N_{i_2}^{\mathcal{A}\cup B}, M_{j_2}^{\mathcal{A}\cup B})}, M_{j_1}^{L(N_{i_2}^{\mathcal{A}\cup B}, M_{j_2}^{\mathcal{A}\cup B})}, w)$ is not true. If such a set B exists, then set $\mathcal{A} := \mathcal{A} \cup B$ and move to the next stage. Otherwise, go to step (3).
(3) Choose a set $B \subseteq \Sigma^{n+1}$ such that one of the following holds:

$$\mathrm{Zeros}\big(B, n, 2^{n-1}\big) \quad \text{and} \quad \mathrm{gap}_{N_{i_1}^{L(N_{i_2}^{\mathcal{A}\cup B}, M_{j_2}^{\mathcal{A}\cup B})}}\big(0^n\big) = 0, \quad \text{or}$$

$$\mathrm{Ones}\big(B, n, 2^{n-1}\big) \quad \text{and} \quad \mathrm{gap}_{N_{i_1}^{L(N_{i_2}^{\mathcal{A}\cup B}, M_{j_2}^{\mathcal{A}\cup B})}}\big(0^n\big) = M_{j_1}^{L(N_{i_2}^{\mathcal{A}\cup B}, M_{j_2}^{\mathcal{A}\cup B})}\big(0^n\big).$$

We will show in Claim 6 that if step (3) is reached then there is always a set $B \subseteq \Sigma^{n+1}$ satisfying the conditions of step (3). Set $\mathcal{A} := \mathcal{A} \cup B$ and move to the next stage. It is clear that such a set B suffices to successfully finish stage $\langle i_1, j_1, i_2, j_2\rangle$.

End of Stage

Claim 6. *In each stage $\langle i_1, j_1, i_2, j_2\rangle$, if step* (3) *is reached, then there is a set B satisfying the conditions of step* (3).

Proof. Assume to the contrary that no such set B exists. Let $p \in \mathbb{Q}[y_1, y_2, \ldots, y_m]$ be the polynomial that encodes the computation of the WPP pair (N_{i_1}, M_{j_1}) on input 0^n with oracle $L(N_{i_2}^{(\cdot)}, M_{j_2}^{(\cdot)})$ as given by Lemma 5.6. We know that for every $B \subseteq \Sigma^{n+1}$ such that $\mathrm{Zeros}(B, n, 2^{n-1})$ or $\mathrm{Ones}(B, n, 2^{n-1})$ is true, the set $A = \mathcal{A} \cup B$ satisfies the hypothesis of Lemma 5.6. Hence

$$\mathrm{Zeros}\big(B, n, 2^{n-1}\big) \quad \Rightarrow \quad p\big(\chi_{\mathcal{A}\cup B}(x_1), \chi_{\mathcal{A}\cup B}(x_2), \ldots, \chi_{\mathcal{A}\cup B}(x_m)\big) = 1, \tag{6}$$

$$\mathrm{Ones}\big(B, n, 2^{n-1}\big) \quad \Rightarrow \quad p\big(\chi_{\mathcal{A}\cup B}(x_1), \chi_{\mathcal{A}\cup B}(x_2), \ldots, \chi_{\mathcal{A}\cup B}(x_m)\big) = 0. \tag{7}$$

W.l.o.g. assume that $x_1, x_2, \ldots, x_{2^n}$ enumerate the strings in $0\Sigma^n$, and that $x_{2^n+1}, x_{2^n+2}, \ldots, x_{2^{n+1}}$ enumerate the strings in $1\Sigma^n$. Statement (6) implies that for every $z_1, z_2, \ldots, z_{2^n}$ satisfying $\sum_{i=1}^{2^n} z_i > 2^{n-1}$,

$$p\bigl(z_1, z_2, \ldots, z_{2^n}, \underbrace{0, 0, \ldots 0}_{2^n}, \chi_{\mathcal{A}\cup B}(x_{2^{n+1}+1}), \ldots, \chi_{\mathcal{A}\cup B}(x_m)\bigr) - 1 = 0, \tag{8}$$

and Statement (7) implies that for every $z_1, z_2, \ldots, z_{2^n}$ satisfying $\sum_{i=1}^{2^n} z_i > 2^{n-1}$,

$$p\bigl(\underbrace{0, 0, \ldots 0}_{2^n}, z_1, z_2, \ldots, z_{2^n}, \chi_{\mathcal{A}\cup B}(x_{2^{n+1}+1}), \ldots, \chi_{\mathcal{A}\cup B}(x_m)\bigr) = 0. \tag{9}$$

Since $\deg(p) \leqslant 4r(n) \cdot r(r(n)) < 2^{n-1}$, we can apply Lemma 5.7 to Eqs. (8) and (9). We obtain $p(0, 0, \ldots, 0, \chi_{\mathcal{A}\cup B}(x_{2^{n+1}+1}), \ldots, \chi_{\mathcal{A}\cup B}(x_m)) - 1 = 0$, and $p(0, 0, \ldots, 0, \chi_{\mathcal{A}\cup B}(x_{2^{n+1}+1}), \ldots, \chi_{\mathcal{A}\cup B}(x_m)) = 0$, respectively. A contradiction. This completes the proofs of Claim 6 and Theorem 5.8. □

For any $k \in \mathbb{N}^+$, let WPP^k denote the kth level of WPP hierarchy formed by composing WPP with itself up to k levels. The proof of Theorem 5.8 can be easily extended to show the following general result: $(\forall k \in \mathbb{N}^+)$ $(\exists \mathcal{A})$ $[\mathrm{ZPP}^{\mathcal{A}} \nsubseteq \mathrm{WPP}^{k,\mathcal{A}}]$.

6. Extensions to other classes

In this section, we demonstrate the technique of using degree lower bound of polynomials in constructing relativized worlds for classes defined by probabilistic oracle Turing machines. Hemaspaandra, Jain, and Vereshchagin [26] showed that relative to an oracle, IP ∩ coIP has no polynomial-time Turing hard sets for ZPP. We extend their result in Theorem 6.3 by constructing an oracle world where MIP ∩ coMIP has no polynomial-time Turing hard sets for ZPP. In the proof, we use the characterization of MIP in terms of oracle proof systems as given by Fortnow, Rompel, and Sipser [20]. Note that in the real world (i.e. relative to ∅ as an oracle) $\mathrm{MIP}^{\emptyset} \cap \mathrm{coMIP}^{\emptyset} = \mathrm{NEXP} \cap \mathrm{coNEXP}$ and so, $\mathrm{MIP}^{\emptyset} \cap \mathrm{coMIP}^{\emptyset}$ contains polynomial-time Turing hard sets for $\mathrm{ZPP}^{\emptyset} = \mathrm{ZPP}$. It follows that Theorem 6.3 does not hold in the real world.

Definition 6.1. [20] We say that a set L has an oracle proof system if there exists a probabilistic polynomial-time oracle Turing machine N such that for all $x \in \Sigma^*$,

$$x \in L \quad \Rightarrow \quad (\exists \mathcal{Q} \subseteq \Sigma^*)\ \bigl[\mathrm{Prob}\bigl[N^{\mathcal{Q}}(x) \text{ accepts}\bigr] \geqslant 1 - 2^{|x|}\bigr] \quad \text{and}$$
$$x \notin L \quad \Rightarrow \quad (\forall \mathcal{Q} \subseteq \Sigma^*)\ \bigl[\mathrm{Prob}\bigl[N^{\mathcal{Q}}(x) \text{ accepts}\bigr] \leqslant 2^{-|x|}\bigr],$$

where the probability is over the random coin tosses done by N.

The next theorem states that the class of sets accepted by multiprover interactive protocols (MIP) is the same as the class of sets that are accepted by oracle proof systems.

Theorem 6.2. [20] *A set L is accepted by an oracle proof system if and only if L is accepted by a multiprover interactive protocol.*

Since the proof of Theorem 6.2 relativizes, it suffices to construct a relativized world where no oracle proof system accepts a set that is polynomial-time Turing hard for ZPP. We construct such a relativized world in the next theorem.

Theorem 6.3. *There exists an oracle $\mathcal{A}$ such that* $\mathrm{MIP}^{\mathcal{A}} \cap \mathrm{coMIP}^{\mathcal{A}}$ *has no* $\leqslant_T^{p,\mathcal{A}}$*-hard set for* $\mathrm{ZPP}^{\mathcal{A}}$.

First we prove the following analog of Lemma 4.5 for probabilistic polynomial-time oracle Turing machines.

Lemma 6.4. *Let $\mathcal{O} \subseteq \Sigma^*$ and let N be a probabilistic polynomial-time oracle Turing machine. Let p be a polynomial that bounds the running time of N. Then for every $x \in \Sigma^*$ with* $\mathrm{Prob}[N^{\mathcal{O}}(x)$ *accepts*$] \geqslant 2/3$,

$$\left\| \left\{\alpha \in \Sigma^* \mid \mathrm{Prob}\left[N^{\mathcal{O}\cup\{\alpha\}}(x) \textit{ accepts}\right] \leqslant 1/3\right\} \right\| \leqslant 4p\left(|x|\right)^2.$$

Proof. Let N' be a nondeterministic polynomial-time oracle Turing machine with time bound p such that for every oracle $\mathcal{A}$ and $x \in \Sigma^*$,

$$\mathrm{Prob}\left[N^{\mathcal{A}}(x) \text{ accepts}\right] = \#\mathrm{acc}_{N'^{\mathcal{A}}}(x)/2^{p(|x|)}.$$

Because #P $\subseteq$ GapP relative to every oracle, there is a nondeterministic oracle Turing machine N'' that is time bounded by p such that for every oracle $\mathcal{A}$ and $x \in \Sigma^*$,

$$\mathrm{Prob}\left[N^{\mathcal{A}}(x) \text{ accepts}\right] = \mathrm{gap}_{N''^{\mathcal{A}}}(x)/2^{p(|x|)}.$$

Let $x \in \Sigma^*$ and define

$$C = \left\{\alpha \in \Sigma^* \;\middle|\; \mathrm{Prob}\left[N^{\mathcal{O}\cup\{\alpha\}}(x) \text{ accepts}\right] \leqslant 1/3\right\}.$$

To get a contradiction, assume that $k =_{\mathrm{df}} \|C\| > 4p(|x|)^2$. Let $s \in \mathbb{Z}[y_1, y_2, \ldots, y_k]$ be the polynomial encoding of $N''^{\mathcal{O}}(x)$ w.r.t. C. From Definition 2.5 it is easy to see that s satisfies the following properties:

(1) For every $y_1, y_2, \ldots, y_k \in \{0, 1\}$, $s(y_1, y_2, \ldots, y_k)/2^{p(|x|)} \in [0, 1]$.
(2) $s(0, 0, \ldots, 0)/2^{p(|x|)} \in [2/3, 1]$.
(3) $s(y_1, y_2, \ldots, y_k)/2^{p(|x|)} \in [0, 1/3]$ for every $y_1, y_2, \ldots, y_k \in \{0, 1\}$ with $\sum_{i=1}^{k} y_i = 1$.
(4) $\deg(s) \leqslant p(|x|)$.

Here we cannot directly apply Lemma 4.2, since s may not approximate any boolean function. This is so because for $y_1, y_2, \ldots, y_k \in \{0, 1\}$ with $\sum_{i=1}^{k} y_i \notin \{0, 1\}$, we know only that $s(y_1, y_2, \ldots, y_k)/2^{p(|x|)} \in [0, 1]$ (s may take, say, value 0.5). But inspection of the proof by Nisan and Szegedy [29] reveals that this is sufficient for the proof to go through. Their proof yields that $\deg(s) \geqslant \sqrt{k/4}$. Therefore $p(|x|) \geqslant \deg(s) \geqslant \sqrt{k/4} = \sqrt{\|C\|/4}$, and hence $\|C\| \leqslant 4p(|x|)^2$. A contradiction. This completes the proof of Lemma 6.4. $\square$

Proof of Theorem 6.3. Let (N_i, N_j, M_k) be an enumeration of tuples where N_i and N_j are probabilistic polynomial-time oracle Turing machines as in Definition 6.1, and M_k is a deterministic polynomial-time oracle Turing machine. Also, for each $B \subseteq \Sigma^*$ and for each $(i, j) \in \mathbb{N}^2$, the test language $L_{\langle i,j\rangle}(B)$ is the same as the one in the proof of Theorem 4.6. If N is a probabilistic polynomial-time oracle Turing machine and $B \subseteq \Sigma^*$, then let

$$L(N^B) =_{\mathrm{df}} \{w \in \Sigma^* \mid (\exists \mathcal{Q} \subseteq \Sigma^*) [\mathrm{Prob}[N^{\mathcal{Q}\oplus B}(w) \text{ accepts}] \geqslant 1 - 2^{-|w|}]\}.$$

We say that N^B fails to be a valid MIP^B machine if and only if there exists $w \in \Sigma^*$ such that

- $(\forall \mathcal{Q} \subseteq \Sigma^*)$ $[\mathrm{Prob}[N^{\mathcal{Q}\oplus B}(w) \text{ accepts}] < 1 - 2^{-|w|}]$, and
- $(\exists \mathcal{Q} \subseteq \Sigma^*)$ $[\mathrm{Prob}[N^{\mathcal{Q}\oplus B}(w) \text{ accepts}] > 2^{-|w|}]$.

In stage $\langle i, j, k\rangle$, we diagonalize against tuple (N_i, N_j, M_k) and modify oracle $\mathcal{A}$ at some length. We will treat $N_i^{\mathcal{A}}$ and $N_j^{\mathcal{A}}$ as machines accepting complementary sets in $\mathrm{MIP}^{\mathcal{A}}$. Initially, $\mathcal{A} := 0\Sigma^*$.

Stage $\langle i, j, k\rangle$: Let $r(\cdot)$ be a polynomial that bounds the running time of each of N_i, N_j and M_k. Choose n large enough so that (a) n is a power of the $\langle i, j\rangle$th prime number, (b) $2^{n-1} > 4 \cdot r(n) \cdot r(r(n))^2$, (c) n satisfies any promises made in the previous stages and no string of length n or more is queried in the previous stages, and (d) n is larger than the value of n in the previous stage. Let $\mathcal{A} := \mathcal{A} \setminus \Sigma^{n+1}$.

If there exists a set $B \subseteq \Sigma^{n+1}$ such that $N_i^{\mathcal{A}\cup B}$ or $N_j^{\mathcal{A}\cup B}$ fails to be a valid $\mathrm{MIP}^{\mathcal{A}\cup B}$ machine or if $L(N_i^{\mathcal{A}\cup B}) \neq \overline{L(N_j^{\mathcal{A}\cup B})}$, then perform the following steps. Set $\mathcal{A} := \mathcal{A} \cup B$ and then move to the next stage with the promise to choose the value of n in the next stage to be larger than $r(|w|)$, where w is an arbitrary string such that one of the following is true.

- w makes $N_i^{\mathcal{A}}$ or $N_j^{\mathcal{A}}$ invalid, or
- w satisfies $w \in L(N_i^{\mathcal{A}}) \Leftrightarrow w \in L(N_j^{\mathcal{A}})$.

Note that setting $\mathcal{A}$ in the former step may cause the test language $L_{\langle i,j\rangle}(\mathcal{A})$ not to be in $\mathrm{ZPP}^{\mathcal{A}}$. However, this is not a problem because the purpose of $L_{\langle i,j\rangle}(\mathcal{A})$ is to witness that $(N_i^{\mathcal{A}}, N_j^{\mathcal{A}})$ does not constitute a set in $\mathrm{MIP}^{\mathcal{A}} \cap \mathrm{coMIP}^{\mathcal{A}}$ that is polynomial-time Turing-hard for $\mathrm{ZPP}^{\mathcal{A}}$, which is already accomplished due to the invalidity of $N_i^{\mathcal{A}}$ or $N_j^{\mathcal{A}}$ as an $\mathrm{MIP}^{\mathcal{A}}$ machine, or due to $L(N_i^{\mathcal{A}}) \neq \overline{L(N_j^{\mathcal{A}})}$.

Otherwise, proceed with the following claim.

Claim 7. *For any $B \subseteq \Sigma^{n+1}$, there exists a set $C \subseteq \Sigma^*$ with $\|C\| \leqslant 4 \cdot r(n) \cdot r(r(n))^2$ such that for every $z \in \Sigma^{n+1} \setminus C$, the replacement of B by $B \cup \{z\}$ does not change the acceptance behavior of $M_k(0^n)$ with oracle $L(N_i^{\mathcal{A}\cup B})$.*

Let us assume that the claim is true. Start with $B := \emptyset$. If $M_k(0^n)$ with oracle $L(N_i^{\mathcal{A}})$ accepts, then apply Claim 7 to add, one after the other, new strings from $1\Sigma^n$ to B such that the acceptance behavior of $M_k(0^n)$ with the oracle $L(N_i^{\mathcal{A}\cup B})$ does not change. Keep adding strings from $1\Sigma^n$ to B until B contains more than 2^{n-1} strings. This is feasible because $2^{n-1} > 4 \cdot r(n) \cdot r(r(n))^2 \geqslant \|C\|$.

The case that $M_k(0^n)$ with oracle $L(N_i^{\mathcal{A}})$ rejects is treated analogously by adding strings from $0\Sigma^n$ to B.

Move to the next stage with $\mathcal{A} := \mathcal{A} \cup B$.

End of Stage

The correctness of the construction is as in the proof of Theorem 4.6. This completes the proof of Theorem 6.3. □

Proof of Claim 7. Let $\beta_1, \beta_2, \ldots, \beta_\ell$, where $0 \leqslant \ell \leqslant r(n)$, be the sequence of queries made by $M_k(0^n)$ to the oracle $L(N_i^{\mathcal{A}\cup B})$. Fix any query β_e from this sequence. Note that both $N_i^{\mathcal{A}\cup B}$ and $N_j^{\mathcal{A}\cup B}$ are valid $\mathrm{MIP}^{\mathcal{A}\cup B}$ machines accepting complementary sets. Therefore by Definition 6.1 and the complementarity of $L(N_i^{\mathcal{A}\cup B})$ and $L(N_j^{\mathcal{A}\cup B})$, one of

- $(\exists \mathcal{Q} \subseteq \Sigma^*)$ $[\mathrm{Prob}[N_i^{\mathcal{Q}\oplus(\mathcal{A}\cup B)}(\beta_e)$ accepts$] \geqslant 2/3]$, or
- $(\exists \mathcal{Q} \subseteq \Sigma^*)$ $[\mathrm{Prob}[N_j^{\mathcal{Q}\oplus(\mathcal{A}\cup B)}(\beta_e)$ accepts$] \geqslant 2/3]$

is true. Fix a set $\mathcal{Q} \subseteq \Sigma^*$ and $\gamma \in \{i, j\}$ such that $\mathrm{Prob}[N_\gamma^{\mathcal{Q}\oplus(\mathcal{A}\cup B)}(\beta_e)] \geqslant 2/3$. Let

$$C(\beta_e) = \left\{\alpha \in \Sigma^* \mid \mathrm{Prob}\left[N_\gamma^{\mathcal{Q}\oplus(\mathcal{A}\cup B\cup\{\alpha\})}(\beta_e) \text{ accepts}\right] \leqslant 1/3\right\}.$$

Applying Lemma 6.4 with $\mathcal{O} := 0\mathcal{Q} \cup 1\mathcal{A} \cup 1B$ and $x := \beta_e$, we obtain $\|C(\beta_e)\| \leqslant 4 \cdot r(r(n))^2$.

By Definition 6.1, $\beta_e \in L(N_\gamma^{\mathcal{A}\cup B})$ and for every $\alpha \in \Sigma^{n+1} - C(\beta_e)$, we have $\beta_e \in L(N_\gamma^{\mathcal{A}\cup B\cup\{\alpha\}})$ as well. Let $C =_{\mathrm{df}} C(\beta_1) \cup C(\beta_2) \cup \cdots \cup C(\beta_\ell)$. Clearly, $\|C\| \leqslant 4 \cdot r(n) \cdot r(r(n))^2$. □

Corollary 6.5. *There is an oracle relative to which*

(1) ZPP, RP, coRP, IP ∩ coIP *have no polynomial-time Turing complete sets* [26],
(2) BPP *has no polynomial-time Turing complete sets* ([24] + [2]), *and*
(3) MIP ∩ coMIP *has no polynomial-time Turing complete sets.*

Acknowledgments

We are grateful to Lane Hemaspaandra for his encouragement, advice, and guidance throughout the project. We thank Mayur Thakur for stimulating discussions.

References

[1] N. Alon, R. Beigel, Lower bounds for approximations by low degree polynomials over $\mathbb{Z}_m$, in: Proceedings of the 16th Annual IEEE Conference on Computational Complexity, Chicago, IL, 18–21, June, IEEE Comput. Soc., 2001, pp. 184–187.

[2] K. Ambos-Spies, A note on complete problems for complexity classes, Inform. Process. Lett. 23 (5) (1986) 227–230.

[3] V. Arvind, P. Kurur, Graph isomorphism is in SPP, in: Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science, Los Alamitos, 16–19, November, IEEE Comput. Soc., 2002, pp. 743–750.

[4] V. Arvind, N. Vinodchandran, Solvable black-box group problems are low for PP, Theoret. Comput. Sci. 180 (1–2) (1997) 17–45.

[5] J. Aspnes, R. Beigel, M. Furst, S. Rudich, The expressive power of voting polynomials, Combinatorica 14 (2) (1994) 135–148.

[6] L. Babai, Trading group theory for randomness, in: Proceedings of the 17th ACM Symposium on Theory of Computing, ACM Press, 1985, pp. 421–429.

[7] R. Beals, H. Buhrman, R. Cleve, M. Mosca, R. de Wolf, Quantum lower bounds by polynomials, J. ACM 48 (2001) 778–797.

[8] R. Beigel, The polynomial method in circuit complexity, in: Proceedings of the 8th Structure in Complexity Theory Conference, San Diego, CA, May, IEEE Comput. Soc., 1993, pp. 82–95.

[9] R. Beigel, Perceptrons, PP, and the polynomial hierarchy, Comput. Complexity 4 (4) (1994) 339–349.

[10] R. Beigel, H. Buhrman, L. Fortnow, NP might not be as easy as detecting unique solutions, in: Proceedings of the 30th ACM Symposium on Theory of Computing, ACM Press, 1998, pp. 203–208.

[11] R. Beigel, N. Reingold, D. Spielman, PP is closed under intersection, J. Comput. System Sci. 50 (2) (1995) 191–202.

[12] M. Ben-Or, S. Goldwasser, J. Kilian, A. Wigderson, Multi-prover interactive proofs: How to remove intractability assumptions, in: Proceedings of the 21st ACM Symposium on Theory of Computing, ACM Press, 1988, pp. 113–131.

[13] M. de Graaf, P. Valiant, Comparing EQP and $\mathrm{MOD}_{p^k}\mathrm{P}$ using polynomial degree lower bounds, Technical report, quant-ph/0211179, Quantum Physics, 2002.

[14] H. Ehlich, K. Zeller, Schwankung von Polynomen zwischen Gitterpunkten, Math. Z. 86 (1964) 41–44.

[15] S. Fenner, PP-lowness and a simple definition of AWPP, Theory Comput. Syst. 36 (2) (2003) 199–212.

[16] S. Fenner, L. Fortnow, S. Kurtz, Gap-definable counting classes, J. Comput. System Sci. 48 (1) (1994) 116–148.

[17] S. Fenner, L. Fortnow, S. Kurtz, L. Li, An oracle builder's toolkit, Inform. and Comput. 182 (2) (2003) 95–136.

[18] S. Fenner, L. Fortnow, L. Li, Gap-definability as a closure property, Inform. and Comput. 130 (1) (1996) 1–17.

[19] L. Fortnow, J. Rogers, Complexity limitations on quantum computation, J. Comput. System Sci. 59 (2) (1999) 240–252.

[20] L. Fortnow, J. Rompel, M. Sipser, On the power of multi-prover interactive protocols, Theoret. Comput. Sci. 134 (1994) 545–557.

[21] J. Goldsmith, D. Joseph, Three results on the polynomial isomorphism of complete sets, in: Proceedings of the 27th IEEE Symposium on Foundations of Computer Science, 1986, pp. 390–397.

[22] S. Goldwasser, S. Micali, C. Rackoff, The knowledge complexity of interactive proof systems, SIAM J. Comput. 18 (2) (1989) 186–208.

[23] S. Gupta, Closure properties and witness reduction, J. Comput. System Sci. 50 (3) (1995) 412–432.

[24] J. Hartmanis, L. Hemachandra, Complexity classes without machines: On complete languages for UP, Theoret. Comput. Sci. 58 (1988) 129–142.

[25] J. Hartmanis, L. Hemachandra, One-way functions and the non-isomorphism of NP-complete sets, Theoret. Comput. Sci. 81 (1) (1991) 155–163.

[26] L. Hemaspaandra, S. Jain, N. Vereshchagin, Banishing robust Turing completeness, Internat. J. Found. Comput. Sci. 4 (3) (1993) 245–265.

[27] L. Hemaspaandra, M. Ogihara, The Complexity Theory Companion, Springer-Verlag, 2002.

[28] L. Hemaspaandra, A. Ramachandran, M. Zimand, Worlds to die for, SIGACT News 26 (4) (1995) 5–15.

[29] N. Nisan, M. Szegedy, On the degree of boolean functions as real polynomials, Comput. Complexity 4 (4) (1994) 301–313.

[30] M. Ogiwara, L. Hemachandra, A complexity theory for feasible closure properties, J. Comput. System Sci. 46 (3) (1993) 295–325.

[31] K. Regan, Polynomials and combinatorial definitions of languages, in: L. Hemaspaandra, A. Selman (Eds.), Complexity Theory Retrospective II, Springer-Verlag, 1997, pp. 261–293.

[32] T. Rivlin, E. Cheney, A comparison of uniform approximations on an interval and a finite subset thereof, SIAM J. Numer. Anal. 3 (2) (June 1966) 311–320.

[33] J. Rosser, L. Schoenfeld, Approximate formulas for some functions of prime numbers, Illinois J. Math. 6 (1962) 64–94.

[34] U. Schöning, A low and a high hierarchy within NP, J. Comput. System Sci. 27 (1983) 14–28.

[35] M. Sipser, On relativization and the existence of complete sets, in: Proceedings of the 9th International Colloquium on Automata, Languages, and Programming, in: Lecture Notes in Comput. Sci., vol. 140, Springer-Verlag, 1982, pp. 523–531.

[36] R. Smolensky, Algebraic methods in the theory of lower bounds for boolean circuit complexity, in: Proceedings of the 19th ACM Symposium on Theory of Computing, ACM Press, 1987, pp. 77–82.

[37] H. Spakowski, M. Thakur, R. Tripathi, Quantum and classical complexity classes: Separations, collapses, and closure properties, Inform. and Comput. 200 (1) (2005) 1–34.

[38] H. Spakowski, R. Tripathi, Degree bounds on polynomials and relativization theory, in: Proceedings of the 3rd IFIP International Conference on Theoretical Computer Science, Kluwer, 2004, pp. 105–118.

[39] J. Tarui, Degree complexity of boolean functions and its applications to relativized separations, in: Proceedings of the 6th Annual Conference on Structure in Complexity Theory, SCTC '91, Chicago, IL, June, IEEE Comput. Soc., 1991, p. 285.

[40] S. Toda, PP is as hard as the polynomial-time hierarchy, SIAM J. Comput. 20 (5) (1991) 865–877.

[41] S. Toda, M. Ogiwara, Counting classes are at least as hard as the polynomial-time hierarchy, SIAM J. Comput. 21 (2) (1992) 316–328.

[42] J. Torán, Complexity classes defined by counting quantifiers, J. ACM 38 (3) (1991) 753–774.

[43] N. Vereshchagin, Relativizable and nonrelativizable theorems in the polynomial theory of algorithms, Russian Acad. Sci. Izv. Math. 42 (2) (1994) 261–298.

[44] N. Vereshchagin, Relativizability in complexity theory, in: L.D. Beklemishev, M. Pentus, N. Vereshchagin (Eds.), Provability, Complexity, Grammars, in: Amer. Math. Soc. Transl. Ser. 2, vol. 192, 1999, pp. 87–172.

[45] N. Vinodchandran, Counting complexity of solvable black-box group problems, SIAM J. Comput. 33 (4) (2004) 852–869.