Quantum and Classical Complexity Classes: Separations, Collapses, and Closure Properties*

Holger Spakowski†
Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany
spakowski@cs.uni-duesseldorf.de

Mayur Thakur‡
Department of Computer Science
University of Missouri–Rolla
Rolla, MO 65409, USA
thakurk@umr.edu

Rahul Tripathi
Department of Computer Science
University of Rochester
Rochester, NY 14627, USA
rahult@cs.rochester.edu

July 05, 2003; rerevised October 25, 2004

Abstract

We study the complexity of quantum complexity classes such as EQP, BQP, and NQP (quantum analogs of P, BPP, and NP, respectively) using classical complexity classes such as ZPP, WPP, and C=P. The contributions of this paper are threefold. First, via oracle constructions, we show that no relativizable proof technique can improve the best known classical upper bound for BQP (BQP ⊆ AWPP [FR99]) to BQP ⊆ WPP and the best known classical lower bound for EQP (P ⊆ EQP) to ZPP ⊆ EQP. Second, we prove that there are oracles A and B such that, relative to A, coRP is immune to NQP and relative to B, BQP is immune to $P^{C=P}$. Extending a result of de Graaf and Valiant [dGV02], we construct a relativized world where EQP is immune to Mod$_p$P. Third, motivated by the fact that counting classes (e.g., LWPP, AWPP, etc.) are the best known classical upper bounds on quantum complexity classes, we study properties of these counting classes. We prove that WPP is closed under polynomial-time truth-table reductions, while we construct an oracle relative to which WPP is not closed under polynomial-time Turing reductions. The latter result implies that proving the equality of the similar appearing classes LWPP and WPP would require nonrelativizable proof techniques. We also prove that both AWPP and APP are closed under $\leq_P$ reductions. We use closure properties of WPP and AWPP to prove interesting consequences, in terms of the complexity of the polynomial-hierarchy, of the following hypotheses: NQP ⊆ BQP and EQP = NQP.

---

* A preliminary version of this paper was presented at the 23rd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS ’03). Research supported in part by grant NSF-INT-9815095/DAAD-315-PFF-gü-ab, a grant from the DAAD, and by DFG project RO 1202/9-1.

†Work done in part while visiting the University of Rochester.

‡Work done in part while the author was affiliated with the University of Rochester.
1 Introduction

Quantum complexity classes such as EQP, BQP [BV97] (quantum analogs, respectively, of P and BPP [Gil77]), and NQP [ADH97] (quantum analog of NP) are defined using quantum Turing machines [BV97], the quantum analog of classical Turing machines. EQP is the class of languages $L$ accepted by a quantum Turing machine $M$ running in polynomial time such that, for each $x \in \Sigma^*$, if $x \in L$, then the probability that $M(x)$ accepts is 1, and if $x \notin L$, then the probability that $M(x)$ accepts is 0. BQP is the class of languages $L$ accepted by a quantum Turing machine $M$ running in polynomial time such that, for each $x \in \Sigma^*$, if $x \in L$, then the probability that $M(x)$ accepts is at least $\frac{2}{3}$, and if $x \notin L$, then the probability that $M(x)$ accepts is at most $\frac{1}{3}$. NQP is the class of languages $L$ accepted by a quantum Turing machine $M$ running in polynomial time such that, for each $x \in \Sigma^*$, $x \in L$ if and only if the probability that $M(x)$ accepts is nonzero.

Quantum complexity classes represent the computational power of quantum computers. Some fundamental computational problems—for example, factoring, discrete logarithm [Sho97], Pell’s equation, and the principal ideal problem [Hal02]—are not believed to be in BQP, and yet have been shown to be in BQP. One of the key issues in quantum complexity theory is studying the relationship between classical and quantum complexity classes. The inclusion relationships of BQP with some natural classical complexity classes are known. Bernstein and Vazirani [BV97] showed that BPP $\subseteq$ BQP $\subseteq$ P$^\#$, Adleman, DeMarrais, and Huang [ADH97] improved that to BQP $\subseteq$ PP. Fortnow and Rogers [FR99] showed that the investigation of counting classes can give us insights into the classical complexity of quantum complexity classes. In particular, they studied the complexity of BQP using gap-definable counting classes [FFK94]. (See section 2 for definitions of complexity classes not defined in this section.) Loosely speaking, gap-definable counting classes capture the power of computing via counting the gap (i.e., difference) between the number of accepting and rejecting paths in a nondeterministic polynomial-time Turing machine. Fortnow and Rogers proved that BQP $\subseteq$ AWPP, where AWPP is a gap-definable counting class. Since AWPP $\subseteq$ PP, this gives a better upper bound on BQP than that of Adleman, DeMarrais, and Huang. Thus, the best known lower and upper bounds for BQP in terms of classical complexity classes are, respectively, BQP and AWPP: BPP $\subseteq$ BQP $\subseteq$ AWPP $\subseteq$ PP. Similarly the best known classical lower and upper bounds for EQP are, respectively, P and LWPP: P $\subseteq$ EQP $\subseteq$ LWPP $\subseteq$ AWPP $\subseteq$ PP. The quantum complexity class NQP coincides with coC$_{=}P$ [FGHP98, YY99].

In light of these connections between quantum and counting complexity classes, it is natural to ask if there are counting (or other classical) complexity classes that give better lower (or upper) bounds for BQP. More formally, is there a counting class $C$ that lies nontrivially between BPP and BQP? Is there a counting class $D$ that lies nontrivially between BQP and AWPP? Similar questions can be asked about EQP. Unfortunately, these questions are often difficult and out of reach of relativizable proof techniques. Green and Pruim [GP01] constructed an oracle relative to which EQP $\nsubseteq$ P$^NP$ and thus they showed that proving EQP $\subseteq$ P$^NP$ is outside the scope of relativizable proof techniques. Furthermore, for each prime $p$ and integer $k \geq 1$, de Graaf and Valiant [dGV02] constructed an oracle relative to which EQP $\nsubseteq$ Mod$_pP$.

In this paper, we use counting classes to study the relativized complexity of EQP, BQP, and NQP. In particular, we study the classes EQP, BQP, and NQP by separating counting classes relative to an oracle. We construct oracles $A$ and $B$ such that ZPP$^A \nsubseteq$ WPP$^A$ and RP$^B \nsubseteq$ C$=\mbox{P}^B$. It follows immediately from known inclusions that ZPP$^A \nsubseteq$ EQP$^A$, BQP$^A \nsubseteq$ WPP$^A$, and RP$^B \nsubseteq$ NQP$^B$.

Note that WPP $\subseteq$ AWPP, P $\subseteq$ ZPP $\subseteq$ BPP $\subseteq$ BQP, and EQP $\subseteq$ LWPP $\subseteq$ WPP $\subseteq$ C$=\mbox{P}$. In fact, WPP is the largest known natural gap-definable subclass of AWPP and ZPP is the smallest known natural probabilistic complexity class that contains P. Thus, even though NQP (=$\mbox{coC}_\mbox{=}\mbox{P}$) contains RP, and hence contains ZPP ($=\mbox{RP} \cap \mbox{coRP}$), in every relativized world, using relativizable proof
techniques it is impossible to show that NQP contains all sets in coRP.

The oracle separations of counting classes mentioned above, for example, $\text{RP}^B \not\subset \text{C=P}^B$, leaves open the possibility that each set in $\text{RP}^B$ can be approximated by a set in $\text{C=P}^B$ in the following sense: For each infinite set $L \in \text{RP}^B$, there exists an infinite subset $L' \subseteq L$ such that $L' \in \text{C=P}^B$. A strong (or immunity) separation of $\text{RP}^B$ from $\text{C=P}^B$ will preclude this possibility. Strong oracle separations have been studied in many different settings, e.g., for the polynomial hierarchy [Ko90, Bru92], for the boolean hierarchy over $\text{RP}$ [BJY90], and for counting classes [Rot99]. We prove strong oracle separations between counting classes, and from these we get strong oracle separations involving quantum and counting complexity classes. For example, we prove that there are oracles $A$ and $A'$ such that $\text{RP}^A$ is $\text{C-P}^A$-immune and $\text{BPP}^{A'}$ is $\text{P}^{C-P}^{A'}$-immune. That implies that $\text{coRP}^A$ is $\text{NQP}^A$-immune and $\text{BQP}^{A'}$ is $\text{P}^{C-P}^{A'}$-immune. For each prime $p$ and integer $k \geq 1$, we construct an oracle relative to which $\text{EQP}$ is $\text{Mod}_{p^k} \text{P}$-immune. This extends an oracle separation of $\text{EQP}$ from $\text{Mod}_{p^k} \text{P}$ by de Graaf and Valiant [dGV02].

Results by Fortnow and Rogers [FR99], de Graaf and Valiant [dGV02], and those of this paper show the connection between quantum and counting complexity classes. This motivates the investigation of reductions closure properties of these counting classes. Fenner, Fortnow, and Kurtz [FFK94] showed that the counting classes $\text{SPP}$ and $\text{LWPP}$ are closed under polynomial-time Turing reductions. (In fact, they proved that $\text{SPP}^{\text{SPP}} = \text{SPP}$ and $\text{SPP}^{\text{LWPP}} = \text{LWPP}$.) They asked whether the same holds for $\text{WPP}$. We give a partial answer to their question. We prove that $\text{WPP}$ is closed under polynomial-time truth-table reductions, and show that this cannot be improved to closure under polynomial-time Turing reductions using relativizable proof techniques: There is an oracle $A$ such that $\text{P}^{\text{WPP}^A} \not\subset \text{WPP}^{A'}$. Thus, it follows that relative to oracle $A$, $\text{WPP}$ does not strictly contain $\text{LWPP}$. For the counting classes $\text{AWPP}$ and $\text{APP}$, we prove a stronger closure property, namely that both $\text{AWPP}$ and $\text{APP}$ are closed under $\leq^P_U$ (unambiguous nondeterministic polynomial-time Turing) reductions.

Vyalii [Vya03] proved using Toda's theorem [Tod91] that $\text{QMA}$, the class of languages such that a “yes” answer can be verified by a 1-round quantum interactive proof, is unlikely to contain $\text{PP}$, since if it does then $\text{PP}$ contains $\text{PH}$. That paper implicitly contains the observation that Toda's theorem [Tod91] implies $\text{PH} \subset \text{UP}^{C-P}$. Using this result and the reduction closure results mentioned above, we prove consequences of the “$\text{NQP} \subset \text{BQP}$” and “$\text{NQP} \subset \text{EQP}^{P}$” hypotheses. Note that these hypotheses are quantum counterparts of the “$\text{NP} \subset \text{BPP}$” and “$\text{NP} \subset \text{P}$” hypotheses. Zachos [Zac88] proved that if $\text{NP} \subset \text{BPP}$, then $\text{PH} \subset \text{BPP}$. We prove that if $\text{NQP} \subset \text{BQP}$, then $\text{PH} \subset \text{AWPP}$, and if $\text{NQP} \subset \text{EQP}$, then $\text{PH} \subset \text{EQP}$ and so, as an immediate consequence, $\text{PH} \subset \text{LWPP}$. Since it is unlikely that the complexity of $\text{PH}$ is restricted to classes such as $\text{AWPP}$ and $\text{LWPP}$, and since $\text{AWPP}$ and $\text{LWPP}$ are low for $\text{PP}$, our results suggest that these hypotheses are less likely to be true.

The paper is organized as follows. Section 2 contains the definitions and notations. In Section 3, we prove relativized separations between counting classes, which in turn lead to relativized separations between quantum and counting classes. In Section 4, we prove our immunity separation results. In Section 5, we prove the closure and collapse results. Section 6 contains some open problems.

\footnote{Note that $\text{EQP} \subset \text{NQP}$ follows trivially from the definitions of these classes.}
2 Preliminaries

Let \( \mathbb{C}, \mathbb{N}, \mathbb{Q}, \) and \( \mathbb{Z} \) denote the set of complex numbers, the set of nonnegative integers, the set of rational numbers, and the set of integers, respectively. Our alphabet is \( \Sigma = \{0, 1\} \). \( \Sigma^* \) denotes the set of all finite length strings over the alphabet \( \Sigma \) and for every \( n \in \mathbb{N} \), \( \Sigma^n \) denotes the set of all strings of length \( n \) in \( \Sigma^* \). For any \( n \in \mathbb{N} \) and any \( x \in \Sigma^* \), \( x \Sigma^n = \{xw \mid w \in \Sigma^n\} \). For any \( x \in \Sigma^* \), \( |x| \) denotes the length of the string \( x \), while \( |x|_0 \) and \( |x|_1 \) denote, respectively, the number of 0’s and the number of 1’s in \( x \). For every set \( L \subseteq \Sigma^* \), \( \|L\| \) denotes the cardinality of \( L \) and let \( \chi_L \) denote the characteristic function of \( L \), i.e., for each \( x \in L \), \( \chi_L(x) = 1 \) and for each \( x \notin L \), \( \chi_L(x) = 0 \). For any \( x \in \Sigma^* \), the integer \( \text{num}(x) \) corresponding to string \( x \) is defined as the value of the binary number \( 1x \). Let \( \langle \ldots \rangle \) denote a standard, fixed, easily computable, and invertible pairing function.

For general complexity-theoretic background and for the definition of complexity classes such as P, NP, FP etc., we refer the reader to the handbook [HO02]. NPTM stands for “nondeterministic polynomial-time Turing machine” and DPTM stands for “deterministic polynomial-time Turing machine.” For a complexity class \( \mathcal{C} \), \( \text{co}\mathcal{C} \) is defined by \( \text{co}\mathcal{C} = \{L \mid \overline{L} \in \mathcal{C}\} \). Throughout this paper, for any (nondeterministic or deterministic or quantum) Turing machine \( N \) and for any \( x \in \Sigma^* \), we use \( N(x) \) as a shorthand for “the computation of \( N \) on input \( x \).”

A computation path \( \rho' \) in an NPTM is a string in \( \Sigma^* \) representing the sequence of nondeterministic guesses. An augmented computation path \( \rho \) in an oracle NPTM is a string \( \langle \rho', \sigma \rangle \) such that \( \sigma \) represents the sequence of answers of the oracle to queries along the path \( \rho' \). We say that an augmented computation path \( \rho = \langle \rho', \sigma \rangle \) occurs in the computation of \( N^A(x) \) if \( \rho' \) is a computation path in \( N^A(x) \) and \( \sigma \) is consistent with the memberships of all strings queried along \( \rho' \) in \( N^A(x) \). Given an augmented path \( \rho = \langle \rho', \sigma \rangle \), let \( \text{naked}(\rho) \) denote \( \rho' \) and let \( \text{ans}(\rho) \) denote \( \sigma \).

In this paper, we will use “path” to mean both “unaugmented path” and “augmented path” (which sense is being used will be clear from the context).

Given NPTM \( N \), \( A \subseteq \Sigma^* \), and \( x \in \Sigma^* \), let \( \text{PATH}(N^A, x) \) denote the set of augmented computation paths that occur in \( N^A(x) \). Let \( \text{ACCEPT}(N^A, x) \) denote the set of paths in \( \text{PATH}(N^A, x) \) that accept. Given NPTM \( N \), \( x \in \Sigma^* \), and augmented path \( \rho \in \Sigma^* \), we let \( \text{sign}(N, x, \rho) = +1 \) if \( \text{naked}(\rho) \) with query answers \( \text{ans}(\rho) \) is an accepting path in \( N(x) \), \( \text{sign}(N, x, \rho) = -1 \) if \( \text{naked}(\rho) \) with query answers \( \text{ans}(\rho) \) is a rejecting path in \( N(x) \), and \( \text{sign}(N, x, \rho) = 0 \), otherwise (i.e., if \( \rho \) is not a valid augmented path in \( N(x) \)). Note that the “sign” value does not depend on the oracle because the oracle answers are already included in the augmented path \( \rho \).

Next, we define the notions of counting in nondeterministic Turing machines. Given an NPTM \( N \) and a string \( x \) in \( \Sigma^* \), we use \( \#\text{acc}_N(x) \) (\( \#\text{rej}_N(x) \)) to denote the number of accepting (respectively, rejecting) paths of \( N \) on \( x \). If \( N \) is an oracle NPTM and \( A \) is a set, then we use \( \#\text{acc}_{N,A}(x) \) (\( \#\text{rej}_{N,A}(x) \)) to denote the number of accepting (respectively, rejecting) paths of \( N \) on \( x \) with oracle \( A \).

\( \#P \) [Val76] is the class of functions \( f \) such that there exists an NPTM \( N \) such that, for each \( x \in \Sigma^* \), \( f(x) = \#\text{acc}_N(x) \). We now define a function \( \text{gap}_N \) that represents the “gap” between the number of accepting and the number of rejecting paths of \( N \).

**Definition 2.1** [FFK94] If \( N \) is an NPTM, define the function \( \text{gap}_N : \Sigma^* \to \mathbb{Z} \) as follows: For all \( x \in \Sigma^* \), \( \text{gap}_N(x) := \#\text{acc}_N(x) - \#\text{rej}_N(x) \). If \( N \) is an oracle NPTM, then for every set \( A \), define the function \( \text{gap}_{N,A} : \Sigma^* \to \mathbb{Z} \) as follows: For all \( x \in \Sigma^* \), \( \text{gap}_{N,A}(x) := \#\text{acc}_{N,A}(x) - \#\text{rej}_{N,A}(x) \).

\( \text{GapP} \) is the class of functions \( f \) such that there exists an NPTM \( N \) such that \( f = \text{gap}_N \). We define the gap-definable counting classes [FFK94] that will be used in the rest of the paper.

**Definition 2.2** 1. [Sim75,Gil77] \( \text{PP} = \{L \mid \exists g \in \text{GapP}(\forall x \in \Sigma^*)(x \in L \iff g(x) > 0)\} \).
2. \([\text{Sim75, Wag86}]\) \(C_\text{P} = \{L \mid (\exists g \in \text{GapP})(\forall x \in \Sigma^*)[x \in L \iff g(x) = 0]\}\).

3. \([\text{CH90, Her90, BG92}]\) For each \(k \geq 2\), \(\text{Mod}_k \text{P} = \{L \mid (\exists g \in \text{GapP})(\forall x \in \Sigma^*)[x \in L \iff g(x) \neq 0 \pmod k]\}\). Equivalently, for each \(k \geq 2\), \(\text{Mod}_k \text{P} = \{L \mid (\exists f \in \#\text{P})(\forall x \in \Sigma^*)[x \in L \iff f(x) \neq 0 \pmod k]\}\).

4. \([\text{PZ83, GP86}]\) \(\oplus \text{P} = \text{Mod}_2 \text{P}\).

5. \([\text{OH93, FFK94}]\) \(\text{SPP} = \{L \mid (\exists g \in \text{GapP})(\forall x \in \Sigma^*)[(x \in L \implies g(x) = 1) \land (x \notin L \implies g(x) = 0)]\}\).

6. \([\text{FFK94}]\) \(\text{LWPP} = \{L \mid (\exists g \in \text{GapP})(\exists h \in \text{FP} : 0 \notin \text{range}(h))(\forall x \in \Sigma^*)[(x \in L \implies g(x) = h(0^{\lfloor |x| \rfloor})) \land (x \notin L \implies g(x) = 0)]\}\).

7. \([\text{FFK94}]\) \(\text{WPP} = \{L \mid (\exists g \in \text{GapP})(\exists h \in \text{FP} : 0 \notin \text{range}(h))(\forall x \in \Sigma^*)[(x \in L \implies g(x) = h(x)) \land (x \notin L \implies g(x) = 0)]\}\).

The counting classes \(\text{AWPP} [\text{FFKL03}]\) and \(\text{APP} [\text{Li93}]\) were defined to study the sets that are low for \(\text{PP}\). The original definition of these classes included the amplification property.

**Definition 2.3** \([\text{FFKL03}]\) A language \(L \subseteq \Sigma^*\) is in \(\text{AWPP}\) if for every polynomial \(r\), there exist a function \(g \in \text{GapP}\) and a polynomial \(p\) such that, for all \(x \in \Sigma^*\),

\[
x \in L \implies 1 - 2^{-r(|x|)} \leq \frac{g(x)}{2^p(|x|)} \leq 1, \text{ and}
\]

\[
x \notin L \implies 0 \leq \frac{g(x)}{2^p(|x|)} \leq 2^{-r(|x|)}.
\]

**Definition 2.4** \([\text{Li93}]\) A language \(L \subseteq \Sigma^*\) is in \(\text{APP}\) if for every polynomial \(r\), there exist \(g, h \in \text{GapP}\) such that, for all \(n \in \mathbb{N}\) and \(x\) with \(n \geq |x|\), \(h(0^n) > 0\) and

\[
x \in L \implies 1 - 2^{-r(n)} \leq \frac{g(x, 0^n)}{h(0^n)} \leq 1, \text{ and}
\]

\[
x \notin L \implies 0 \leq \frac{g(x, 0^n)}{h(0^n)} \leq 2^{-r(n)}.
\]

Fenner \([\text{Fen03}]\) simplified the definition of these classes and showed that \(\text{AWPP} \subseteq \text{APP}\).

**Theorem 2.5** \([\text{Fen03}]\)

1. A language \(L \subseteq \Sigma^*\) is in \(\text{AWPP}\) if and only if there exist a function \(g \in \text{GapP}\) and a polynomial \(p\) such that, for all \(x \in \Sigma^*\),

\[
x \in L \implies \frac{3}{4} \leq \frac{g(x)}{2^p(|x|)} < 1, \text{ and}
\]

\[
x \notin L \implies 0 \leq \frac{g(x)}{2^p(|x|)} \leq 1/4.
\]

2. A language \(L \subseteq \Sigma^*\) is in \(\text{APP}\) if and only if there exist \(g, h \in \text{GapP}\), \(h > 0\), such that, for all \(x \in \Sigma^*\),

\[
x \in L \implies \frac{3}{4} \leq \frac{g(x)}{h(0^n)} \leq 1, \text{ and}
\]

\[
x \notin L \implies 0 \leq \frac{g(x)}{h(0^n)} \leq 1/4.
\]
We also define certain classes that we will relate to the gap-definable classes mentioned in Definitions 2.2, 2.3, and 2.4 in this paper.

Definition 2.6 [Val76] A language \( L \subseteq \Sigma^* \) is in UP if there is an NPTM \( N \) such that, for all \( x \in \Sigma^* \), \( x \in L \implies \#acc_N(x) = 1 \) and \( x \notin L \implies \#acc_N(x) = 0 \).

Definition 2.7 [All86,AR88] A language \( L \subseteq \Sigma^* \) is in FewP if there is an NPTM \( N \) and a polynomial \( p \) such that, for all \( x \in \Sigma^* \), \( x \in L \implies 1 \leq \#acc_N(x) \leq p(|x|) \) and \( x \notin L \implies \#acc_N(x) = 0 \).

Definition 2.8 [Gil77] A language \( L \subseteq \Sigma^* \) is in RP if there is a polynomial-time predicate \( R \), a polynomial \( p \), and \( 0 < \epsilon \leq 1 \) such that, for all \( x \in \Sigma^* \),

\[
\begin{align*}
x \in L & \implies \|\{y \mid |y| \leq p(|x|) \land R(x, y)\}\| \geq \frac{(1 + \epsilon)}{2} \cdot 2^{p(|x|)}, \text{ and} \\
x \notin L & \implies \|\{y \mid |y| \leq p(|x|) \land R(x, y)\}\| = 0.
\end{align*}
\]

Definition 2.9 [Gil77] ZPP = RP \( \cap \) coRP.

Definition 2.10 [Gil77] A language \( L \subseteq \Sigma^* \) is in BPP if there is a polynomial-time predicate \( R \), a polynomial \( p \), and \( 0 < \epsilon \leq 1 \) such that, for all \( x \in \Sigma^* \),

\[
\begin{align*}
x \in L & \implies \|\{y \mid |y| \leq p(|x|) \land R(x, y)\}\| \geq \frac{(1 + \epsilon)}{2} \cdot 2^{p(|x|)}, \text{ and} \\
x \notin L & \implies \|\{y \mid |y| \leq p(|x|) \land R(x, y)\}\| \leq \frac{(1 - \epsilon)}{2} \cdot 2^{p(|x|)}.
\end{align*}
\]

For background information on quantum complexity theory and for the definition of quantum Turing machine, we recommend [NC00,Gru99]. We define the quantum complexity classes that will be used in this paper.

Definition 2.11 [BV97,ADH97] EQP (BQP, NQP) is the class of all languages \( L \subseteq \Sigma^* \) such that there is a polynomial-time quantum Turing machine \( M \) such that, for each \( x \in \Sigma^* \), \( x \in L \implies Pr[M(x) \text{ accepts}] = 1 \) (respectively, \( \geq 2/3, \neq 0 \)) and \( x \notin L \implies Pr[M(x) \text{ accepts}] = 0 \) (respectively, \( \leq 1/3, = 0 \)).

The following proposition gives known inclusion relationships among the classes defined above.

Proposition 2.12 [FFK94,Gil77,KSTT92,FFKL03,Fen03,FR99,FGHP98,YY99] The following inclusion relations hold relative to all oracles.

1. \( P \subseteq ZPP \subseteq RP \subseteq BPP \subseteq AWPP \).
2. \( P \subseteq UP \subseteq \text{FewP} \subseteq \text{SPP} \subseteq \text{LWPP} \subseteq \text{WPP} \subseteq \text{C}_mP \subseteq \text{PP} \).
3. \( ZPP \subseteq \text{coRP} \subseteq \text{coNP} \subseteq \text{C}_mP \).
4. \( \text{WPP} \subseteq \text{AWPP} \subseteq \text{APP} \subseteq \text{PP} \).
5. \( P \subseteq \text{EQP} \subseteq \text{LWPP} \subseteq \text{WPP} \subseteq \text{coC}_mP \).
6. \( \text{EQP} \subseteq \text{BQP} \subseteq \text{AWPP} \).
7. \( \text{SPP} \subseteq \oplus P \).
8. \( \text{FewP} \subseteq \text{NP} \subseteq \text{coC}_mP = \text{NQP} \).
9. \( \text{BPP} \subseteq \text{BQP} \).


Next we define the reductions used in this paper. We say that \( A \leq^p_B \) (\( A \) polynomial-time Turing reduces to \( B \)) if there exists an oracle DPTM \( M \) such that \( L(M_B) = A \). We say that \( A \leq^t_B \) (\( A \) polynomial-time truth-table reduces to \( B \)) if there exists a DPTM \( M \) and a polynomial-time computable function \( f \) such that, for each \( x \in \Sigma^* \), there exists an integer \( m \) such that

1. \( f(x) = \langle q_1, q_2, \ldots, q_m \rangle \), and

2. \( M(\langle x, \chi_B(q_1), \chi_B(q_2), \ldots, \chi_B(q_m) \rangle) \) accepts if and only if \( x \in A \).

### 3 Separating Quantum Classes Using Counting Class Separations

One way to study the power of quantum complexity classes is to search for well-known complexity classes that provide a good lower bound for these quantum classes. The best known lower bound for EQP is P. In fact, EQP is not known to contain even a single problem that is not already known to be in P. Bennett et al. [BBBV97] showed that relative to a random oracle, NP is not contained in EQP with probability one, and relative to a permutation oracle chosen uniformly at random, NP \( \cap \coNP \) is not contained in EQP with probability one. Thus it is interesting to ask the following questions. Are there natural classes between P and NP \( \cap \coNP \) that are contained in EQP? Are there natural classes between P and NP \( \cap \coNP \) that are not contained in EQP in some relativized world? We prove that the latter is true by showing that there is a relativized world where ZPP is not contained in EQP. In fact, we prove a slightly stronger statement. We prove, as the next theorem, that there is an oracle relative to which ZPP is not contained in WPP, a superclass of EQP [FR99]. It is interesting to note that there is an oracle, due to Fortnow [For99], relative to which SPP, a subclass of WPP, strictly contains an infinite polynomial hierarchy. In contrast, our oracle provides a completely different picture of WPP in a relativized world: A world in which WPP sets are not powerful enough to capture a seemingly small subclass, namely ZPP, of NP.

**On the Proof Technique:** The oracle constructions in Theorems 3.1 and 3.13 use a “gap analog” of the counting technique used by Torán [Tor91] in the relativized separation of counting classes from each other, for example in the construction of an oracle \( A \) such that NP\(^A\) \( \not\subseteq \) P\(^A\). Many of the powerful state-of-the-art oracle construction techniques have the following flavor in proof steps: (1) abstract the oracle construction problem in terms of some combinatorial object, such as a boolean circuit over some base set of gates or a polynomial over some ring, satisfying certain properties, and (2) prove existence of an appropriate oracle segment based on the limitations of the combinatorial object satisfying those properties. Because of this transition in proof from machines and oracles to pure combinatorial objects with implicit dependence on behavior of machines, these oracle construction techniques are more combinatorial than complexity-theoretic in nature. In contrast, the proof techniques by Torán [Tor91] and Beigel [Bei91], which we use in Theorems 3.1, 3.13, and 4.4, do not purely abstract the problem from machines and oracles to combinatorial objects, and depend heavily on the behavior of machines with different oracles. From the point of view of traditional and perfectionist complexity theorists, the proof techniques by Torán [Tor91] and Beigel [Bei91] might be more appealing, because of being more complexity-theoretic in nature, than the other known combinatorial proof techniques for oracle constructions.

For the above reason, we choose to use the “gap analog” of the proof technique by Torán [Tor91] and Beigel [Bei91] in Theorems 3.1, 3.13, and 4.4. In a subsequent paper [ST04b,ST04a], the first and the third author of this paper used the combinatorial polynomial degree bound technique to prove an extension of Theorem 3.1. They showed that there is a relativized world where ZPP \( \not\subseteq \) WPP\(^{WPP}\).
and claimed that their proof can be easily extended to construct a relativized world where ZPP is not contained in any level $k$, where $k \geq 1$, of the WPP hierarchy formed by composing WPP with itself up to $k$ levels. It is illuminating to compare the power of the two vastly different proof techniques for the oracle construction in Theorem 3.1.

**Theorem 3.1** $(\exists \mathcal{A}) \left[ \text{ZPP}^A \not\subseteq \text{WPP}^A \right]$.

**Proof** For every set $B$, let $L_B = \{0^n \mid (\exists w \in \Sigma^n)(0w \in B)\}$. Define predicates “I-Promise” and “II-Promise” as follows.

$$
\begin{align*}
\text{I-Promise}(B, n, k) &\equiv |B \cap 0\Sigma^n| > k \text{ and } |B \cap 1\Sigma^n| = 0, \text{ and} \\
\text{II-Promise}(B, n, k) &\equiv |B \cap 0\Sigma^n| = 0 \text{ and } |B \cap 1\Sigma^n| > k.
\end{align*}
$$

We say that a set $B$ satisfies the promise at length $n$ with threshold $k$ if [I-Promise$(B, n, k) \lor \text{II-Promise}(B, n, k)$] is true. Clearly, if $B$ satisfies the promise at each length $n \geq 1$ with threshold $2^{n-1}$, then $L_B \in \text{ZPP}^B$. We show that there is an oracle $\mathcal{A}$ such that $\mathcal{A}$ satisfies the promise at each length $n \geq 1$ with threshold $2^{n-1}$ and $L_\mathcal{A} \not\subseteq \text{WPP}^\mathcal{A}$.

Let $(N_s, M_s, p_s)_{s \geq 1}$ be an enumeration of all triples such that the first component of the triple is a nondeterministic polynomial-time oracle Turing machine, the second component is a deterministic polynomial-time oracle transducer, the third component is a polynomial, and the running time of both $N_s$ and $M_s$ is bounded by the polynomial $p_s$ regardless of the oracle.

The oracle is constructed in stages. Before the start of stage 1, let $\mathcal{A} := \bigcup_{n \in \mathbb{N}} 0\Sigma^n$. In stage $s \geq 1$, the membership in $\mathcal{A}$ of strings of length $n_s + 1$ is changed. Finally at the end of every stage, $\mathcal{A}$ is assigned as the oracle.

**Stage $s$, where $s > 0$:** Let $n_s$ be the smallest integer such that the following hold: $n_s > n_{s-1}$, $2^{n_s} > 4p_s(n_s) + 2$, and no machine considered in previous stages ever queries a string of length greater than $n_s$. Let $\mathcal{A} := \mathcal{A} - \Sigma^{n_s+1}$. In stage $s$, we diagonalize against the triple $(N_s, M_s, p_s)$. Note that the FP function $h$ used in the definition of WPP (see Definition 2.2(7)) is non-zero for all $x \in \Sigma^*$. Therefore, without loss of generality we may assume that $M_s$ never outputs zero with any oracle that satisfies the promise at length $n_s$ with threshold $2^{n_s-1}$.

Let $T_0$ ($T_1$) denote the set of queries of length $n_s + 1$ asked by $M^\mathcal{A}_s(0^{n_s})$ whose first bit is zero (respectively, one). Let the value computed by $M^\mathcal{A}_s(0^{n_s})$ be denoted by $\text{val}$. As explained above, we may assume that $\text{val} \neq 0$.

$(\star)$ Choose a set $B$ such that $B \subseteq \overline{\bigcup_{n \geq 0} T_0} \cap T_1 \cap \Sigma^{n_s+1}$ and one of the following conditions is satisfied:

$$
\begin{align*}
(3.a) & \quad \text{I-Promise}(B, n_s, 2^{n_s-1}) \quad \text{and} \quad \text{gap}_{N^\mathcal{A}_s \cup B}(0^{n_s}) \neq \text{val}, \quad \text{or} \\
(3.b) & \quad \text{II-Promise}(B, n_s, 2^{n_s-1}) \quad \text{and} \quad \text{gap}_{N^\mathcal{A}_s \cup B}(0^{n_s}) \neq 0.
\end{align*}
$$

Let $\mathcal{A} := \mathcal{A} \cup B$ and move to stage $s + 1$.

**End of Stage**

Obviously, the construction guarantees that $L_\mathcal{A} \in \text{ZPP}^\mathcal{A}$ but $L_\mathcal{A} \not\subseteq \text{WPP}^\mathcal{A}$. The feasibility of the construction follows from the following claim.

**Claim 1** For each $s \geq 1$, there exists a set $B$ satisfying $(\star)$.

**Proof of Claim 1.** In order to prove the above claim, we first describe a combinatorial tool similar to the one used by Torán [Tor91] to separate NP from $\mathbb{C} = \mathbb{P}$. Torán used his $Q$-notation to count
the number of accepting paths that have certain restrictions on the queries asked along them. Our $Q$-notation represents the gap between the number of accepting and rejecting paths that have similar restrictions on the queries asked along them.

Recall from the definition of “sign” given in Section 2 that $\text{sign}(N, x, \rho)$ is 0, +1, or −1, if, respectively, $\rho$ is not a valid augmented path in $N(x)$, $N(x)$ on path naked($\rho$) with query answers $\text{ans}(\rho)$ accepts, or $N(x)$ on path naked($\rho$) with query answers $\text{ans}(\rho)$ rejects.

**Definition 3.2** For every set $E \subseteq \Sigma^{n+1}$ and for each $a_1, a_2, \ldots, a_k, b_1, b_2, \ldots, b_l \in \Sigma^*$, let $Q_{a_1, a_2, \ldots, a_k, b_1, b_2, \ldots, b_l}^E = \sum_{\rho \in R} \text{sign}(N, \rho)$, where $R = \{ \rho : \rho \in \text{PATH}(N^A \cup \mathcal{E}, 0^{n*}) \text{ and } N^A \cup \mathcal{E}(0^{n*}) \text{ on path naked}(\rho) \text{ queries each string in } \{a_1, a_2, \ldots, a_k\} \text{ and } N^A \cup \mathcal{E}(0^{n*}) \text{ on path naked}(\rho) \text{ does not query any string in } \{b_1, b_2, \ldots, b_l\}\}$.

Thus, for any strings $w_1, w_2, \ldots, w_k, w'_1, w'_2, \ldots, w'_l$, each accepting path of $N^A \cup \mathcal{E}(0^{n*})$ that queries all of $w_1, w_2, \ldots, w_k$ and none of $w'_1, w'_2, \ldots, w'_l$ has a contribution of +1 in $Q_{w_1, w_2, \ldots, w_k, w'_1, w'_2, \ldots, w'_l}^E$, and each rejecting path of $N^A \cup \mathcal{E}(0^{n*})$ that queries all of $w_1, w_2, \ldots, w_k$ and none of $w'_1, w'_2, \ldots, w'_l$ has a contribution of −1 in $Q_{w_1, w_2, \ldots, w_k, w'_1, w'_2, \ldots, w'_l}^E$.

The following lemma, which we state without a proof, is a “gap analog” of Lemma 5.2 by Torán [Tor91]. It says that the “$Q$” value corresponding to paths querying $w_1, \ldots, w_k$ can be split into 2 parts: (a) contribution from paths querying $w_1, \ldots, w_k$ that also query $w_{k+1}$ and (b) contribution from paths querying $w_1, \ldots, w_k$ that do not query $w_{k+1}$.

**Lemma 3.3** For every set $E \subseteq \Sigma^{n+1}$ and for all $w_1, \ldots, w_{k+1} \in \Sigma^{n+1}$, the following equality holds:

$$Q_{w_1, \ldots, w_k}^E = Q_{w_1, \ldots, w_k, w_{k+1}}^E + Q_{w_1, \ldots, w_k}^E(w_{k+1}).$$

The following definition is similar to the one by Torán [Tor91] with the only exception that the $Q$-terms have the “gap” meaning as in Definition 3.2.

**Definition 3.4** For any $E, D \subseteq \Sigma^{n+1}$ with $E \cap D = \emptyset$,

$$J_D^E = \frac{|D|!}{|E|!} \sum_{i=0}^{n} (-1)^i \sum_{A \subseteq D, |A| = i} Q_{D \cup A}^E.$$  

Intuitively, $E \subseteq \Sigma^{n+1}$ represents an oracle segment and $D \subseteq \Sigma^{n+1}$ represents a set of query strings, which is disjoint from $E$, in the expression $J_D^E$. The expression $J_D^E$ is an arithmetic sum of “$Q$” values, where the “$Q$” values correspond to oracles $E \cup A$, for all $A \subseteq D$, and correspond to computation paths with a fixed query set $D$. For example, for a set $D = \{w_1, w_2, w_3\} and an arbitrary set $E \subseteq \Sigma^{n+1} - \{w_1, w_2, w_3\}$.

$$J_D^E = Q_D^E - (Q_D^{E \cup \{w_1\}} + Q_D^{E \cup \{w_2\}} + Q_D^{E \cup \{w_3\}} + (Q_D^{E \cup \{w_1, w_2\}} + Q_D^{E \cup \{w_1, w_3\}} + Q_D^{E \cup \{w_2, w_3\}} - Q_D^{E \cup \{w_1, w_2, w_3\}}.$$  

For notational convenience, we use the following shorthands: For every $E \subseteq \Sigma^{n+1}$, let $Q^E = \text{gap}_{N^A \cup \mathcal{E}}(0^{n*})$ and $J_0^E = Q^E$.

**Lemma 3.5** [See Tor91, Lemma 5.3.] For every sequence of words $w_1, \ldots, w_k, w_{k+1}$ in $\Sigma^{n+1}$ and for every set $E \subseteq \Sigma^{n+1}$ with $E \cap \{w_1, \ldots, w_k, w_{k+1}\} = \emptyset$, the following holds:

$$J_{E \cup \{w_{k+1}\}}^{w_1, \ldots, w_k} = J_{w_1, \ldots, w_k}^E - J_{w_1, \ldots, w_k, w_{k+1}}^E.$$
Proof All properties of $Q$ and $J$ used in the proof of Lemma 5.3 in [Tor91] carry over to our gap versions of $Q$ and $J$. Hence, Torán’s proof works also for our lemma. (Lemma 3.5)

Suppose that in stage $s$ no set satisfying (⋆) exists. Then, for every set $B$ such that $B \subseteq \overline{T}_0 \cap \overline{T}_1 \cap \Sigma_{\nu_s+1}$,

\[ (3.c) \quad \text{I-Promise}(B, \nu_s, 2^{\nu_s-1}) \implies \text{gap}_{\Sigma_{\nu_s}^2} \left( 0^{\nu_s} \right) = \text{val}, \quad \text{and} \]

\[ (3.d) \quad \text{II-Promise}(B, \nu_s, 2^{\nu_s-1}) \implies \text{gap}_{\Sigma_{\nu_s}^2} \left( 0^{\nu_s} \right) = 0. \]

**Lemma 3.6** For each $s \geq 1$ and every $t \in \{0, \ldots, 2^{\nu_s-1}\}$, if for every set $B$ such that $B \subseteq \overline{T}_0 \cap \overline{T}_1 \cap \Sigma_{\nu_s+1}$, it holds that

\[ \text{I-Promise}(B, \nu_s, t) \implies \text{gap}_{\Sigma_{\nu_s}^2} \left( 0^{\nu_s} \right) = \text{val}, \]

then the following hold:

1. For every nonempty sequence of words $0w_1, \ldots, 0w_k$ such that $\{0w_1, \ldots, 0w_k\} \subseteq \overline{T}_0 \cap \Sigma_{\nu_s}$ and for every set $R$ that satisfies $R \subseteq \overline{T}_0 \cap \Sigma_{\nu_s}$, I-Promise($R, \nu_s, t$), and $\{0w_1, \ldots, 0w_k\} \cap R = \emptyset$, it holds that $J^R_{0w_1, \ldots, 0w_k} = 0$.

2. For every set $C_t$ such that $C_t \subseteq \overline{T}_0 \cap \Sigma_{\nu_s}$, $||C_t|| = t$, and $\{0w_1, \ldots, 0w_k\} \cap C_t = \emptyset$, it holds that $J_{0w_1, \ldots, 0w_k}^{C_t} = \text{gap}_{\Sigma_{\nu_s}^2 \cup C_t} \left( 0^{\nu_s} \right) - \text{val}$.

**Proof** By hypothesis and by using the definition of $J$, for every set $R$ such that $R \subseteq \overline{T}_0 \cap \Sigma_{\nu_s}$ and I-Promise($R, \nu_s, t$) is true, the following holds: $J^R = Q^R = \text{val}$. By Lemma 3.5,

\[ J^R_{0w_1, \ldots, 0w_k} = J^R_{0w_1, \ldots, 0w_k} - J^R_{0w_1, \ldots, 0w_k}. \]

We now prove (1) by induction on $k$.

For $k = 1$, $J^R_{0w_1} = J^R - J^R_{\emptyset} = \text{val} - \text{val} = 0$.

For $k > 1$, $J^R_{0w_1, \ldots, 0w_k} = J^R_{0w_1, \ldots, 0w_k} - J^R_{0w_1, \ldots, 0w_{k-1}}$, where by induction hypothesis both terms are 0.

We prove (2) also by induction on $k$.

For $k = 1$,

\[ J_{0w_1}^{C_t} = Q_{0w_1}^{C_t} - Q_{0w_1}^{C_t \cup \{0w_1\}} = Q_{0w_1}^{C_t} - Q_{0w_1}^{C_t} = Q_{0w_1}^{C_t \cup \{0w_1\}} - Q_{0w_1}^{C_t \cup \{0w_1\}} = Q_{0w_1}^{C_t} - Q_{0w_1}^{C_t \cup \{0w_1\}}, \]

where the third equality follows from the fact that, by Definition 3.2, $Q_{0w_1}^{C_t} = Q_{0w_1}^{C_t \cup \{0w_1\}}$. Thus,

\[ J_{0w_1}^{C_t} = \text{gap}_{\Sigma_{\nu_s}^2 \cup C_t} \left( 0^{\nu_s} \right) - \text{val}, \]

using the definition of $C_t$ and using the fact that I-Promise($C_t \cup \{0w_1\}, \nu_s, t$) is true. For $k > 1$, by Lemma 3.5,

\[ J_{0w_1, \ldots, 0w_k}^{C_t} = J_{0w_1, \ldots, 0w_k}^{C_t} - J_{0w_1, \ldots, 0w_k}^{C_t}. \]

By (1),

\[ J_{0w_1, \ldots, 0w_k}^{C_t \cup \{0w_k\}} = 0. \]

Also, by the induction hypothesis,

\[ J_{0w_1, \ldots, 0w_k}^{C_t} = \text{gap}_{\Sigma_{\nu_s}^2 \cup C_t} \left( 0^{\nu_s} \right) - \text{val}. \]
Thus,

\[ J_{0w_1,\ldots,0w_k}^{C_t} = \text{gap}_{N^*_{A^*C_t}}(0^{n^*}) - \text{val}. \]

(\text{Lemma } 3.6)

**Lemma 3.7** For each \( s \geq 1 \) and every \( t \in \{0, \ldots, 2^{n_s-1}\} \), if for every set \( B \) such that \( B \subseteq T_0 \cap T_1 \cap \Sigma^{n_0+1} \), it holds that

\[ \text{I-Promise}(B, n_s, t) \implies \text{gap}_{N^*_{A^*B}}(0^{n^*}) = \text{val}, \]

then for every set \( C_t \) such that \( C_t \subseteq T_0 \cap 0 \Sigma^{n_0} \) and \( ||C_t|| = t \), it holds that \( \text{gap}_{N^*_{A^*C_t}}(0^{n^*}) = \text{val} \).

**Proof** Let \( C_t \) be an arbitrary set such that \( C_t \subseteq T_0 \cap 0 \Sigma^{n_0} \) and \( ||C_t|| = t \). Suppose the hypothesis of the lemma holds but \( \text{gap}_{N^*_{A^*C_t}}(0^{n^*}) \neq \text{val} \). Then, by Lemma 3.6(2), for any nonempty sequence of words \( 0w_1, \ldots, 0w_k \), where \( \{0w_1, \ldots, 0w_k\} \subseteq T_t \cap T_0 \cap 0 \Sigma^{n_0} \), it holds that \( J_{0w_1,\ldots,0w_k}^{C_t} \neq 0 \). Let \( 0w_1, \ldots, 0w_{p_s(n_s)}+1 \) be a sequence of words such that \( \{0w_1, \ldots, 0w_{p_s(n_s)}+1\} \subseteq T_t \cap T_0 \cap 0 \Sigma^{n_0} \). (Such a sequence of words exists since \( 2^{n_s} > 4p_s(n_s) + 2 \).) Since the running time of \( \text{N}^{(1)}_{s}(0^{n^*}) \) is bounded by \( p_s(n_s) \), \( \text{N}^{(1)}_{s}(0^{n^*}) \) can make at most \( p_s(n_s) \) queries to the oracle on every computation path. Therefore, \( J_{0w_1,\ldots,0w_{p_s(n_s)}+1}^{C_t} = 0 \). This follows from the fact that \( J_{0w_1,\ldots,0w_{p_s(n_s)}+1}^{C_t} \) is the sum of sign values associated with computation paths of \( \text{N}^{(1)}_{s}(0^{n^*}) \) in which all the words \( 0w_1, \ldots, 0w_{p_s(n_s)}+1 \) are queried. Thus we obtain a contradiction. So, \( \text{gap}_{N^*_{A^*C_t}}(0^{n^*}) = \text{val} \). 

(\text{Lemma } 3.7)

If in stage \( s \) no set satisfying \((*)\) exists, then \((3.c)\) is true. Thus, from Lemma 3.7, for \( t = 2^{n_s-1} \) and for any set \( C_t \) such that \( C_t \subseteq T_0 \cap 0 \Sigma^{n_0} \) and \( ||C_t|| = t \), it holds that \( \text{gap}_{N^*_{A^*C_t}}(0^{n^*}) = \text{val} \). Thus, under the assumption that in stage \( s \) no set satisfying \((*)\) exists, the following holds.

For every set \( B \) such that \( B \subseteq T_0 \cap T_1 \cap \Sigma^{n_s+1} \),

\[ \text{I-Promise}(B, n_s, 2^{n_s}-1) \implies \text{gap}_{N^*_{A^*B}}(0^{n^*}) = \text{val}, \]

(3.e)

\[ \text{II-Promise}(B, n_s, 2^{n_s}-1) \implies \text{gap}_{N^*_{A^*B}}(0^{n^*}) = 0. \]

(3.f)

Note that \((3.e)\) is a logically stronger statement than \((3.c)\). We now argue that the above chain of arguments can be pushed further to show that, if in stage \( s \) no set satisfying \((*)\) exists, then \( \text{gap}_{N^*_{A^*}}(0^{n^*}) = \text{val} \).

**Lemma 3.8** For each \( s \geq 1 \), if in stage \( s \) no set satisfying \((*)\) exists, then \( \text{gap}_{N^*_{A^*}}(0^{n^*}) = \text{val} \).

**Proof** Assume that in stage \( s \) no set satisfying \((*)\) exists. For any \( t' \in \mathbb{Z} \), let \( P(t') \) be the following statement:

\[ P(t') \equiv (\forall B \subseteq T_0 \cap T_1 \cap \Sigma^{n+s+1}) \left[ \text{I-Promise}(B, n_s, t') \implies \text{gap}_{N^*_{A^*B}}(0^{n^*}) = \text{val} \right]. \]

We prove using induction that, for every \( t' \in \{-1, \ldots, 2^{n_s}-1\} \), \( P(t') \) holds. The base cases (when \( t' = 2^{n_s-1} \) or \( t' = 2^{n_s-1} - 1 \)) are already shown to be true. By induction hypothesis, for \( t' = k \), where \( k \geq 0 \),

\[ P(k) \equiv (\forall B \subseteq T_0 \cap T_1 \cap \Sigma^{n+s+1}) \left[ \text{I-Promise}(B, n_s, k) \implies \text{gap}_{N^*_{A^*B}}(0^{n^*}) = \text{val} \right]. \]

is true. Now applying Lemma 3.7 for \( t = k \), we have that

\[ (\forall C \subseteq T_0 \cap 0 \Sigma^{n_s}) ||C|| = k \implies \text{gap}_{N^*_{A^*C}}(0^{n^*}) = \text{val}. \]
Thus, certainly $(\forall C \subseteq \overline{T_0} \cap \overline{T_1} \cap 0^\Sigma^{n_s}, ||C|| = k \Rightarrow \text{gap}_{\text{II-Promise}}(0^{n_s}) = \text{val})$. Combining this with the induction hypothesis, we have that

$$P(k-1) \equiv (\forall B \subseteq \overline{T_0} \cap \overline{T_1} \cap \Sigma^{n_s+1}) [\text{I-Promise}(B, n_s, k-1) \Rightarrow \text{gap}_{\text{II-Promise}}(0^{n_s}) = \text{val}]$$

is also true. Thus, by induction, for all $t' \in \{-1, \ldots, 2^{n_s}-1\}$, $P(t')$ is true. Since $P(-1)$ and $\text{I-Promise}(\emptyset, n_s, -1)$ are true, it follows that $\text{gap}_{\text{II-Promise}}(0^{n_s}) = \text{val}$. \hfill (Lemma 3.8)

We now show that if in stage $s$ no set satisfying (*) exists, then, using (3.d), a contradiction can be achieved from Lemma 3.8; this will imply that in stage $s$ a set $B$ satisfying (*) always exists. We first prove a lemma analogous to Lemma 3.6.

**Lemma 3.9** For each $s \geq 1$ and every $t \in \{0, \ldots, 2^{n_s}-1\}$, if for every set $B$ such that $B \subseteq \overline{T_0} \cap \overline{T_1} \cap \Sigma^{n_s+1}$, it holds that $\text{II-Promise}(B, n_s, t) \Rightarrow \text{gap}_{\text{II-Promise}}(0^{n_s}) = 0$,

then the following hold:

1. For every nonempty sequence of words $1w_1, 1w_2, \ldots, 1w_k$ such that $\{1w_1, \ldots, 1w_k\} \subseteq \overline{T_1} \cap 1^\Sigma^{n_s}$ and for every set $R$ that satisfies $R \subseteq \overline{T_1} \cap 1^\Sigma^{n_s}$, II-Promise$(R, n_s, t)$, and $\{1w_1, \ldots, 1w_k\} \cap R = \emptyset$, it holds that $J_{1w_1, \ldots, 1w_k}^R = 0$.

2. For every set $D_t$ such that $D_t \subseteq \overline{T_1} \cap 1^\Sigma^{n_s}$, $||D_t|| = t$, and $\{1w_1, \ldots, 1w_k\} \cap D_t = \emptyset$, it holds that $J_{1w_1, \ldots, 1w_k}^{D_t} = \text{gap}_{\text{II-Promise}}(0^{n_s})$.

**Proof** We prove (1) by induction on $k$. By hypothesis and by the definition of $J$, for every set $R$ such that $R \subseteq \overline{T_1} \cap 1^\Sigma^{n_s}$ and II-Promise$(R, n_s, t)$ is true, the following holds: $J_R = Q_R = 0$. By Lemma 3.5,

$$J_{1w_1, \ldots, 1w_k}^R = J_{1w_1, \ldots, 1w_k}^R - J_{1w_1, \ldots, 1w_k}^R.$$

For $k = 1$, $J_{1w_1}^R = J_R - J_{1w_1}^R = Q_R - Q_{1w_1}^R = 0 - 0 = 0$.

For $k > 1$, $J_{1w_1, \ldots, 1w_k}^R = J_{1w_1, \ldots, 1w_k}^R - J_{1w_1, \ldots, 1w_{k-1}}^R$, where by induction hypothesis both terms are 0.

We prove (2) also by induction on $k$.

For $k = 1$,

$$J_{1w_1}^{D_t} = Q_{1w_1}^{D_t} - Q_{1w_1}^{D_t \cup \{1w_1\}} = Q_{1w_1}^{D_t} - Q_{1w_1}^{D_t \cup \{1w_1\}} + Q_{1w_1}^{D_t \cup \{1w_1\}} = Q_{1w_1}^{D_t} - Q_{1w_1}^{D_t \cup \{1w_1\}},$$

where the third equality follows from the fact that, by Definition 3.2, $Q_{1w_1}^{D_t} = Q_{1w_1}^{D_t \cup \{1w_1\}}$. Thus,

$$J_{1w_1}^{D_t} = \text{gap}_{\text{II-Promise}}(0^{n_s}) - 0 = \text{gap}_{\text{II-Promise}}(0^{n_s}),$$

using the definition of $D_t$ and using the fact that II-Promise$(D_t \cup \{1w_1\}, n_s, t)$ is true. For $k > 1$, by Lemma 3.5,

$$J_{1w_1, \ldots, 1w_k}^{D_t} = J_{1w_1, \ldots, 1w_k}^{D_t} - J_{1w_1, \ldots, 1w_{k-1}}^{D_t \cup \{1w_k\}}.$$

By (1),

$$J_{1w_1, \ldots, 1w_k}^{D_t \cup \{1w_k\}} = 0.$$

12
Also, by the induction hypothesis,
\[ J_{1 \ldots w_k}^{D_t} = \text{gap}_{N \cup D_i} (0^{n^*}) \text{.} \]

Thus,
\[ J_{1 \ldots w_k}^{D_t} = \text{gap}_{N \cup D_i} (0^{n^*}) \text{.} \]

\[ \text{(Lemma 3.9)} \]

The next two lemmas, Lemma 3.10 and Lemma 3.11, are analogous to Lemma 3.7 and Lemma 3.8, respectively, with minor differences in their statement. So, we omit their proof in the paper.

**Lemma 3.10** For each \( s \geq 1 \) and every \( t \in \{0, \ldots, 2^{n^*} - 1\} \), if for every set \( B \) such that \( B \subseteq T_0 \cap T_1 \cap 1 \Sigma^* \), it holds that
\[ \text{II-Promise}(B, n_s, t) \implies \text{gap}_{N \cup B} (0^{n^*}) = 0 \text{,} \]
then for every set \( D_t \) such that \( D_t \subseteq T_1 \cap 1 \Sigma^* \) and \( ||D_t|| = t \), it holds that \( \text{gap}_{N \cup D_t} (0^{n^*}) = 0 \).

**Proof** Omitted, since the proof is similar to that of Lemma 3.7.

If in stage \( s \) no set satisfying \((\ast)\) exists, then \((3.d)\) is true. Thus, from Lemma 3.10, for \( t = 2^{n^*} - 1 \) and for any set \( D_t \) such that \( D_t \subseteq T_1 \cap 1 \Sigma^* \) and \( ||D_t|| = t \), it holds that \( \text{gap}_{N \cup D_t} (0^{n^*}) = 0 \). Thus, under the assumption that in stage \( s \) no set satisfying \((\ast)\) exists, the following holds.

For every set \( B \) such that \( B \subseteq T_0 \cap T_1 \cap 1 \Sigma^* \),
\[ \text{I-Promise}(B, n_s, 2^{n^*} - 1) \implies \text{gap}_{N \cup B} (0^{n^*}) = \text{val} \text{, and} \]
\[ \text{II-Promise}(B, n_s, 2^{n^*} - 1 - 1) \implies \text{gap}_{N \cup B} (0^{n^*}) = 0 \text{.} \]

The above chain of arguments can be pushed further to show that, if in stage \( s \) no set satisfying \((\ast)\) exists, then \( \text{gap}_{N \cup A} (0^{n^*}) = 0 \).

**Lemma 3.11** For each \( s \geq 1 \), if in stage \( s \) no set satisfying \((\ast)\) exists, then \( \text{gap}_{N \cup A} (0^{n^*}) = 0 \).

**Proof** Omitted, since the proof is similar to that of Lemma 3.8.

Since \( \text{val} \neq 0 \), Lemma 3.8 and Lemma 3.11 imply a contradiction. Thus, for each \( s \geq 1 \), a set \( B \) satisfying \((\ast)\) can always be found in stage \( s \).

**Corollary 3.12** There exists an oracle \( A \subseteq \Sigma^* \) such that, for each \( C \in \{\text{ZPP, RP, BPP, NP, BQP, } \cap \text{coC}_{=P}, \text{AWPP, APP}\} \) and for each \( D \in \{\text{UP, FewP, SPP, EQP, LWPP, WPP}\} \), \( C^A \not\subseteq D^A \).

**Proof** Let \( A \) be the oracle constructed in Theorem 3.1. Then, \( C^A \not\subseteq D^A \) follows from the oracle separation of ZPP from WPP in Theorem 3.1, and from ZPP\(^A \subseteq C^A \) and \( D^A \subseteq \text{WPP}^A \) (see Proposition 2.12).

Corollary 3.12 shows that proving that error-free quantum polynomial-time (EQP) algorithms exist for all problems in ZPP will require nonrelativizable techniques. Corollary 3.12 also shows that, using relativizable techniques, we cannot lower the best known classical upper bound for BQP from AWPP to even WPP, the largest known natural gap-definable subclass of AWPP. Note that WPP
can be considered as a weak subclass of AWPP in the following sense: There is an oracle relative to which UP ∩ coUP is not low for WPP [ST04b,ST04a], whereas SPP, a superclass of UP, is low for AWPP in every relativized world. Thus, Corollary 3.12 shows that BQP might not be as weak a subclass of AWPP as WPP in some relativized world.

Theorem 3.1 provides an intuition on the relativized complexity of EQP and BQP. But, what about the relativized complexity of NQP? It is equally worth investigating the relativized complexity of C=P for this question, since NQP equals coC=P in every relativized world [FGHP98,YY99]. Theorem 3.13 makes a step in this direction and shows that there is a relativized world where coRP is not contained in NQP.

Tarui [Tar91] used a lower bound technique in decision trees for a certain AC0 function to show that BPP is not contained in PC=P in some relativized world. Green [Gre93] used circuit lower bound techniques to obtain the same result. In contrast with BPP, RP is contained in PC=P in every relativized world. In Theorem 3.13, we use the “gap analog” of the proof technique by Torán [Tor91] to construct an oracle relative to which RP is not contained in C=P. This result is optimal in the sense that the largest known natural subclass of RP, namely ZPP, is contained in C=P in every relativized world. This oracle separation of RP from C=P is also a strengthening of the oracle separation of NP from C=P by Torán [Tor91].

**Theorem 3.13** \(\exists A [\text{RP}^A \not\subseteq \text{C}^A].\)

**Proof** The oracle construction is similar to that in Theorem 3.1. For every set \(B\), let \(L_B = \emptyset \{0^n \mid (3w \in \Sigma^n) [w \in B]\}\). Define predicates “I-Promise” and “II-Promise” as follows.

\[
\begin{align*}
\text{I-Promise}(B, n, k) & \equiv |B \cap \Sigma^n| > k, \text{ and} \\
\text{II-Promise}(B, n, k) & \equiv |B \cap \Sigma^n| = 0.
\end{align*}
\]

We say that a set \(B\) satisfies the promise at length \(n\) with threshold \(k\) if \([\text{I-Promise}(B, n, k) \lor \text{II-Promise}(B, n, k)]\) is true. We show that there is an oracle \(A\) such that \(A\) satisfies the promise at each length \(n \geq 1\) with threshold \(2^{n-1}\) and \(L_A \not\subseteq \text{C}^A\).

Let \((N_s, p_s)_{s \geq 1}\) be an enumeration of all pairs such that the first component is a nondeterministic polynomial-time oracle Turing machine, the second component is a polynomial, and the running time of \(N_s\) is bounded by the polynomial \(p_s\) regardless of the oracle. We now describe the stages in the construction of the oracle.

**Stage 0:** Let \(A_0 := \emptyset\) and let \(n_0 := 0\).

**Stage s, where \(s > 0\):** Let \(n_s\) be the smallest integer such that the following hold: \(n_s > n_{s-1}\), \(2^{n_s} > 2p_s(n_s) + 2\), and \(n_s\) is large enough so that the constructions in previous stages are not affected. In stage \(s\), we diagonalize against the pair \((N_s, p_s)\).

(\(\star\)) Choose a set \(B \subseteq \Sigma^{n_s}\) such that one of the following conditions are satisfied:

\[(3i)\] \quad \text{I-Promise}(B, n_s, 2^{n_{s-1}}) \quad \text{and} \quad \text{gap} \, N_{s-1}^{A_{s-1}}\upsilon B \, (0^n) \neq 0, \text{ or} \]

\[(3j)\] \quad \text{II-Promise}(B, n_s, 2^{n_{s-1}}) \quad \text{and} \quad \text{gap} \, N_{s-1}^{A_{s-1}}\upsilon B \, (0^n) = 0.\]

Let \(A_s := A_{s-1} \cup B\) and move to stage \(s + 1\).

**End of Stage**
Let \( \mathcal{A} := \lim_{s \to \infty} \mathcal{A}_s \). Obviously, the construction guarantees that \( L_\mathcal{A} \not\in \mathsf{C} = \mathsf{P}^\mathcal{A} \). The feasibility of the construction follows from the following claim.

**Claim 2** For each \( s \geq 1 \), there exists an oracle extension \( B \) satisfying (**) for every set \( B \subseteq \Sigma^{n_s} \).

**Proof of Claim 2.** Our combinatorial tool is the same as the one we used in proving Claim 1. The \( Q \) and \( J \) notations used here are as in Definitions 3.2 and 3.4, respectively. (The only minor changes are that \( E \subseteq \Sigma^{n_{s+1}} \) is replaced by \( E \subseteq \Sigma^{n_s} \) and \( \mathcal{A} \) is replaced by \( \mathcal{A}_{s-1} \) in the new definition.) Suppose that in stage \( s \) no set satisfying (**) exists. Then, for every set \( B \subseteq \Sigma^{n_s} \),

\[
(3.1) \quad \text{I-Promise}(B, n_s, 2^{n_s-1}) \implies \text{gap}_{N^{A_{s-1} \cup B}}(0^{n_s}) = 0, \text{ and }
\]

(3.1) \( \quad \text{II-Promise}(B, n_s, 2^{n_s-1}) \implies \text{gap}_{N^{A_{s-1} \cup B}}(0^{n_s}) \neq 0. \)

Lemmas 3.14, 3.15, and 3.16 can be proved in a way similar to Lemmas 3.6, 3.7, and 3.8, respectively, are proved. So, we omit their proofs in the paper.

**Lemma 3.14** For each \( s \geq 1 \) and every \( t \in \{0, \ldots, 2^{n_s-1}\} \), if for every set \( B \subseteq \Sigma^{n_s} \), it holds that

\[
\text{I-Promise}(B, n_s, t) \implies \text{gap}_{N^{A_{s-1} \cup B}}(0^{n_s}) = 0,
\]

then the following hold:

1. For every nonempty sequence of words \( w_1, \ldots, w_k \), where \( \{w_1, \ldots, w_k\} \subseteq \Sigma^{n_s} \) and for every set \( R \subseteq \Sigma^{n_s} \), I-Promise\( (R, n_s, t) \) is true and \( \{w_1, \ldots, w_k\} \cap R = \emptyset \), it holds that \( J^R_{w_1, \ldots, w_k} = 0 \).

2. For every set \( C_t \) such that \( C_t \subseteq \Sigma^{n_s} \), \( |C_t| = t \), and \( \{w_1, \ldots, w_k\} \cap C_t = \emptyset \), it holds that \( J^{C_t}_{w_1, \ldots, w_k} = \text{gap}_{N^{A_{s-1} \cup C_t}}(0^{n_s}) \).

**Lemma 3.15** For each \( s \geq 1 \) and every \( t \in \{0, \ldots, 2^{n_s-1}\} \), if for every set \( B \subseteq \Sigma^{n_s} \), it holds that

\[
\text{I-Promise}(B, n_s, t) \implies \text{gap}_{N^{A_{s-1} \cup B}}(0^{n_s}) = 0,
\]

then for every set \( C_t \) such that \( C_t \subseteq \Sigma^{n_s} \) and \( |C_t| = t \), it holds that \( \text{gap}_{N^{A_{s-1} \cup C_t}}(0^{n_s}) = 0 \).

If in stage \( s \) no set satisfying (**) exists, then (3.k) is true. Thus, from Lemma 3.15, for \( t = 2^{n_s-1} \) and for any set \( C_t \) such that \( C_t \subseteq \Sigma^{n_s} \) and \( |C_t| = t \), it holds that \( \text{gap}_{N^{A_{s-1} \cup C_t}}(0^{n_s}) = 0 \). Thus, under the assumption that in stage \( s \) no set satisfying (**) exists, the following holds.

For every set \( B \subseteq \Sigma^{n_s} \),

\[
(3.m) \quad \text{I-Promise}(B, n_s, 2^{n_s-1} - 1) \implies \text{gap}_{N^{A_{s-1} \cup B}}(0^{n_s}) = 0, \text{ and }
\]

\[
(3.n) \quad \text{II-Promise}(B, n_s, 2^{n_s-1}) \implies \text{gap}_{N^{A_{s-1} \cup B}}(0^{n_s}) \neq 0. \)

Note that (3.m) is a logically stronger statement than (3.k). As in the proof of Claim 1, the above chain of arguments can be pushed further to show that, if in stage \( s \) no set satisfying (**) exists, then \( \text{gap}_{N^{A_{s-1}}}(0^{n_s}) = 0 \).

**Lemma 3.16** For each \( s \geq 1 \), if in stage \( s \) no set satisfying (**) exists, then \( \text{gap}_{N^{A_{s-1}}}(0^{n_s}) = 0 \).
Lemma 3.16 and (31) imply a contradiction. Thus, for each \( s \geq 1 \), partial oracle \( \mathcal{A}_{i-1} \) can always be extended in stage \( s \).

(Claim 2 and Theorem 3.13)

The following corollary follows immediately from Theorem 3.13 and the inclusions stated in Proposition 2.12.

**Corollary 3.17** There exists an oracle \( \mathcal{A} \subseteq \Sigma^* \) such that, for each \( \mathcal{C} \in \{ \text{coRP}, \text{BPP}, \text{coNP}, \text{BQP}, \text{AWPP}, \text{APP} \} \), \( \mathcal{C}^\mathcal{A} \not\subseteq \text{NQP}^{\mathcal{A}} \).

Later in Section 4, we will prove that Theorem 3.13 can be strengthened to a relativized immunity separation of \( \text{RP} \) from \( \text{C=P} \) (see Theorem 4.9).

## 4 Immunity Separations of Quantum Classes from Counting Classes

A (simple) separation of a class \( \mathcal{C} \) from another class \( \mathcal{D} \) shows the existence of sets \( A \) in \( \mathcal{C} \) that do not belong to \( \mathcal{D} \). However, it does not preclude the possibility that every such set \( A \) has a close approximation (in some sense) by some set in \( \mathcal{D} \). For instance, it is possible that \( \mathcal{C} \) separates from \( \mathcal{D} \), yet every infinite set in \( \mathcal{C} \) has an infinite subset (approximation) in \( \mathcal{D} \). An immunity separation of a class \( \mathcal{C} \) from a class \( \mathcal{D} \) is stronger than a simple separation, since the immunity separation also shows the existence of sets \( S \) in \( \mathcal{C} \) that cannot even be approximated (in the sense mentioned above) by any infinite set in \( \mathcal{D} \). The sets \( S \) are said to be \( \mathcal{D} \)-immune.

A relativized separation of one class from another does not necessarily imply a relativized immunity separation of the classes. We give an example regarding the boolean hierarchy. Cai et al. [CGH**88] showed that there is an oracle relative to which the levels of the boolean hierarchy separate completely. Since every set in the boolean hierarchy can be expressed, using one of the normal form representations of the set, as a finite union of sets in \( \text{DP} \), the second level of the boolean hierarchy, there is no relativized world where some set in the boolean hierarchy is \( \text{DP} \)-immune. Another example is as follows: Though \( \text{P} \) separates from \( \text{NP} \) in some relativized world [BGS75], it is known that relative to a generic oracle, \( \text{NP} \) has no infinite \( \text{P} \)-immune set [Bl87].

Immunity (strong) separations have been used to study the relativized complexity of classes in different settings. Ko [Ko90] showed that there is a relativized world, where for every \( k \geq 1 \), \( \Sigma_k^p \) contains a \( \Sigma_{k-1}^p \)-immune set. Bruschi [Bru92] extended this result of Ko and showed that there exist oracles that witness the following: For every \( k \geq 1 \), the existence of a \( \Delta_k^p \)-immune set in \( \Sigma_k^p \), the existence of a set \( L \) in \( \Sigma_k^p \) such that \( L \) is \( \Sigma_{k-1}^p \)-immune, the existence of a \( \Delta_k^p \)-immune set in a relativized polynomial hierarchy for which \( \Sigma_k^p = \Pi_k^p \neq \Delta_k^p \), and for every \( k > 1 \), the existence of a \( \Sigma_{k-1}^p \)-immune set in a relativized polynomial hierarchy for which \( \Sigma_k^p = \Delta_k^p \). Bruschi, Joseph, and Young [BJY90] strongly separated the boolean hierarchy over \( \text{RP} \), and Rothe [Rot99] used immunity separations to study the relativized complexity of counting classes. Rothe [Rot99], in particular, showed that there are oracles \( A_1, A_2 \), and \( A_3 \) such that \( \text{C=P}^{A_1} \) contains a \( \text{BPP}^{\oplus \text{P}^{A_1}} \)-immune set, \( \text{NP}^{A_2} \) contains a \( \text{C=P}^{A_2} \)-immune set, and \( \oplus \text{P}^{A_3} \) contains a \( \text{PP}^{\text{PH}^{A_3}} \)-immune set.

Before we state and prove our results, we formally define the notions of immunity and strong separations.

**Definition 4.1** Let \( \mathcal{C} \) be a class of languages. An infinite language \( L \) is called \( \mathcal{C} \)-immune if (\( \forall L' \in \mathcal{C} \)) \( ||L'|| = \infty \Rightarrow L' \cap L \neq \emptyset \). A class \( \mathcal{D} \) is \( \mathcal{C} \)-immune (or, is immune to \( \mathcal{C} \)) if there is a set \( L \) in \( \mathcal{D} \) that is \( \mathcal{C} \)-immune.
Definition 4.2 Given relativizable classes $C_1$ and $C_2$, an oracle $E$ strongly separates $C_2^E$ from $C_1^E$ if $C_2^E$ is $C_1^E$-immune.

M. de Graaf and P. Valiant [dGV02] proved that, for any prime $p$ and integer $k \geq 1$, there exists an oracle $A'$ such that $EQP^{A'} \not\subseteq \text{Mod}_{p^k}P^{A'}$. In Theorem 4.4, we strengthen this result by proving that there is a relativized world where every infinite set in $EQP$ cannot even be approximated by some set in $\text{Mod}_{p^k}P$, i.e., $EQP$ strongly separates from $\text{Mod}_{p^k}P$. To prove that the test language we use is in (relativized) $EQP$, we make use of the observation by Boyer et al. [BBHT98] that quantum database searching can be done in polynomial time with certainty if the number of solutions is exactly one fourth of the total search-space. We also need the following theorem by Lucas [Luc78] to determine $(^n_k) \bmod p$ for any prime $p$ and integers $n, k \geq 1$.

Theorem 4.3 [Luc78] For any prime $p$ and integers $n, k \geq 1$, $(^n_k) \bmod p = \prod_{i=0}^{n} \left(\frac{\binom{n}{k}}{\binom{i}{k}}\right) \bmod p$, where for any $m \in \mathbb{N}$, $(m)_p^i$ is the coefficient of $p^i$ in the base-$p$ expansion of $m$ and $r$ is the largest integer such that, either $(n)_p^r \neq 0$ or $(k)_p^r \neq 0$. We take here the standard convention that for any $n, k \in \mathbb{N}$, $(^n_k) = 0$ if $n < k$.

Theorem 4.4 For every prime $p$ and integer $k \geq 1$, there exist an oracle $A$ and an infinite set $L_A$ such that $L_A \in EQP^{A}$ and $L_A$ is $\text{Mod}_{p^k}P^{A}$-immune.

Proof Since for each prime $p$ and integer $k \geq 1$, $\text{Mod}_{p^k}P = \text{Mod}_{p}P$ [BG02] relative to all oracles, we may henceforth assume that $k = 1$. For each $n \in \mathbb{N}$, let $f_n = 2(1+3p)^{p^{n+1}}$, $g_n = (p^2 + 5p - 2)p^n/2$, and $ext(n) = \lfloor \log_2 f_n \rfloor$. For each $D \subseteq \Sigma^*$ and $n \in \mathbb{N}$, let $D^{ext(n)}$ be defined as: $D^{ext(n)} = D \cap \{ x \in \Sigma^{ext(n)} \mid \text{rank}(x) \leq f_n \}$, where rank$(x)$ is the number of strings of length $|x|$ that are lexicographically less than or equal to $x$. For each $D \subseteq \Sigma^*$, define $L_D$ as follows:

$$L_D = \{0^n \mid |D^{ext(n)}| = (1/4) * f_n \}.$$

We say that an oracle $D \subseteq \Sigma^*$ is relevant, if for each $n \in \mathbb{N}$, $|D^{ext(n)}| \in \{f_n/4, 3f_n/4\}$. Boyer et al. [BBHT98] observed that Grover’s [Gro96] algorithm can be used to design a quantum polynomial-time algorithm that decides with certainty whether the number of solutions in a database is exactly one fourth or three fourths of the total search space. Using their result, it is easy to see that for any relevant oracle $D$, $L_D \in EQP^D$.

We construct a relevant oracle $A$ such that $L_A$ is $\text{Mod}_{p}P^{A}$-immune. The oracle $A$ we construct will have the following form: $A = \bigcup_{n \in \mathbb{N}} (Z_n \cup W_n)$, where for every $n \in \mathbb{N}$, $Z_n = \{ x \in \Sigma^{ext(n)} \mid g_n < \text{rank}(x) \leq g_n + f_n/4 \}$ and $W_n = \{ x \in \Sigma^{ext(n)} \mid g_n + f_n/4 < \text{rank}(x) \leq f_n \}$. Note that since $A$ is relevant, this means that, for any $n$, the cardinality of $W_n$ is either exactly 0 or exactly $f_n/2$.

The set $A$ is constructed in stages. Let $A_0 = \bigcup_{n \in \mathbb{N}} Z_n$. In stage $s, s > 0$, we construct a set $B$ consisting of strings of length $ext(n_s)$, for sufficiently large $n_s$, from the set $\{ x \in \Sigma^{ext(n_s)} \mid g_n + f_n/4 < \text{rank}(x) \leq f_n \}$. At the end of each stage, we set $A_s := A_{s-1} \cup B$. Let $A = \lim_{s \to \infty} A_s$.

Let $N_1, N_2, \ldots$ be an enumeration of all oracle NPTMs such that (a) for every $i \geq 1$, the running time of $N_i$ is bounded by nondecreasing polynomial $p_i$ independent of the oracle, and (b) there are an infinite number of indices $j$ such that, for each $X \subseteq \Sigma^*$ and each $x \in \Sigma^*$, $\#acc_{N_j}(x) \equiv 0 \pmod{p_i}$. In order to construct a set $L_A$ that is $\text{Mod}_{p^2}P^{A}$-immune, we will use a set $T$ of conditions (requirements) that are to be met during the construction of the oracle. We will update $T$ during the construction in the following way. At each stage $s$, we add to $T$ the condition corresponding to machine $N_s$, and delete from $T$ a condition if during stage $s$ the condition is met. Thus, at the beginning of each stage of the construction, $T$ will contain all the conditions that have yet to be satisfied. For each
$i \in \mathbb{N}$, define condition $T_i$ as \( \text{“} L(N_i^A) \cap \overline{L_A} \neq \emptyset \text{”} \). Note that if condition $T_i$ is satisfied, then $L(N_i^A)$ is not a subset of $L_A$; thus $L(N_i^A)$ is definitely not an infinite subset of $L_A$.

**Claim 3** For each $T = \{T_{t_1}, T_{t_2}, \ldots, T_{t_k}\} \subseteq \mathbb{N}$, where $t_1, t_2, \ldots, t_k$ are indices of oracle NPTMs $N_{t_1}, N_{t_2}, \ldots, N_{t_k}$, respectively, and for each $W \subseteq \Sigma^*$, there exists an oracle NPTM $\tilde{N}_T$ such that, $\#\text{acc}_{\tilde{N}_T}^W(x) \neq 0 \pmod{p}$ if and only if there exists $T_{i_j} \in T$ such that $\#\text{acc}_{N_{i_j}}^W(x) \neq 0 \pmod{p}$.

Moreover, the running time of $\tilde{N}_T^W$ on every $x$ is bounded by $p^3 \sum_{i=1}^k p_i(|x|)$.

**Proof** The claim follows from the fact that, for any prime $p$, Mod$_p$P is closed under union [BG92].

**Notation:** For any set $T = \{T_{t_1}, T_{t_2}, \ldots, T_{t_k}\}$ of conditions, let $\text{Index}(T)$ denote the set $\{t_1, t_2, \ldots, t_k\}$ of indices of machines.

**Stage 0:** Set $A_0 := \bigcup_{n \in \mathbb{N}} Z_n$. $T = \emptyset$.

**Stage s, where $s \geq 1$:** Add condition $T_s$ to $T$. Let $T = \{T_{t_1}, T_{t_2}, \ldots, T_{t_k}\}$. Choose $n_s$ large enough such that previous stages are not affected and $p^{n_s} > p^3 \sum_{i=1}^k p_i(n_s)$. We call $B$ allowable if $B$ contains exactly 0 or exactly $f_{n_s}/2$ strings from the set $\{x \in \Sigma^{\text{ext}(n_s)} \mid g_{n_s} + f_{n_s}/4 < \text{rank}(x) \leq f_{n_s}\}$. We will choose an allowable set $B$ such that

\[B = \emptyset \iff \#\text{acc}_{N_{A_{s-1}} \cup B}^0(0^{n_s}) \equiv 0 \pmod{p}\]

is satisfied. (We prove in Claim 6 that such an allowable set always exists.) Let $B_s$ be the allowable set chosen. If $B_s$ is nonempty, then there exists some $t_j \in \text{Index}(T)$ such that $\#\text{acc}_{N_{A_{s-1}} \cup B}^0(0^{n_s}) \neq 0 \pmod{p}$. Let $t_j$ be the smallest such element. Remove $T_{t_j}$ from $T$. Set $A_s = A_{s-1} \cup B_s$. Go to Stage $s + 1$.

**End of Stage.**

Note that, as stated earlier, $A = \lim_{s \to \infty} A_s$.

**Claim 4** $L_A$ is an infinite set.

**Proof** Since for each $s$, $0^{n_s} \in L_A$ if and only if $B_s$ is empty, it suffices to prove that there are an infinite number of stages $s$ such that $B_s$ is empty. Let us assume that the claim is false. Then, there exists a stage $m$ such that, for each stage $s > m$, $B_s$ is nonempty. Thus, for each $s > m$, $\#\text{acc}_{N_{A_{s-1}} \cup B_s}^0(0^{n_s}) \neq 0 \pmod{p}$, and by construction, some element is removed from $T$. Also, note that the enumeration of NPTMs has an infinite number of indices $j$ such that $N_j$ robustly rejecting (i.e., for all oracles) and for all strings has a number of accepting paths divisible by $p$. Let such machines $N_j$ be called robustly rejecting. Since a condition $T_i$ can be removed from $T$ only if $N_i$ with some oracle on some string has a number of accepting paths not divisible by $p$, a condition corresponding to a robustly rejecting machine $N_j$ can never be removed from $T$. Recall that in every stage $s$, we add a condition $T_s$ corresponding to machine $N_s$ in $T$. By assumption, for every stage $s > m$, one of the conditions corresponding to some machine $N_s$, where $N_s$ is not robustly rejecting, is removed from $T$. Thus in every stage $s > m$, if $N_s$ is robustly rejecting, then the number of conditions corresponding to machines that are not robustly rejecting decreases by one and if $N_s$ is not robustly rejecting, then the number of conditions corresponding to machines that are not robustly rejecting remains the same at the end of stage $s$. Since there are infinitely many robustly rejecting machines in our enumeration of oracle NPTMs, there will exist an integer $q > m$ such that after stage $q$,
Index($T$) will consist of indices of robustly rejecting machines. Thus, at stage $q + 1$, for all allowable $B$, #acc$_{\hat{N}^A_T \cup B} (0^{n + 1}) \equiv 0 \pmod p$. Thus, $B_{q+1}$ will be empty, contradicting our assumption.

Claim 5 For each $i$, $N^A_i$ does not accept (in the Mod$_p$P sense) an infinite subset of $L_A$.

**Proof** For each $i$, either condition $T_i$ is satisfied at some stage $s$ or condition $T_i$ is never satisfied. Consider the case that $T_i$ is satisfied at stage $s$. Then, $L(N^A_i) \cap L_A \neq \emptyset$. Thus, $N^A_i$ does not accept a subset of $L_A$. Now consider the case that $T_i$ is never satisfied. We will now prove that $L(N^A_i) \cap L_A$ is finite. Since $T_i$ is added to $T$ at stage $i$, it suffices to prove that, for each $k \geq i$, if $0^{n_k} \in L_A$ then $0^{n_k} \notin L(N^A_i)$. Assume that $0^{n_k} \in L_A$. Then, by construction, #acc$_{\hat{N}^A_T} (0^{n_k}) \equiv 0 \pmod p$. Thus, by Claim 3, $N^A_i (0^{n_k})$ has a number of accepting paths divisible by $p$ and $0^{n_k} \notin L(N^A_i)$.

Claims 4 and 5 together imply that if the construction is feasible, then $L_A$ is Mod$_p$P$^A$-immune. It remains to show that the construction is feasible. That is, we need to show that at each stage $s \geq 1$, $B_s$ can be chosen.

Claim 6 For each $s \geq 1$, there exists an allowable set $B$ such that the following holds:

$$B = \emptyset \iff \#\text{acc}_{\hat{N}^A_T \cup B} (0^{n_s}) \equiv 0 \pmod p.$$

**Proof** The application of the combinatorial “double counting” technique of “reversing the order of summation” in the construction of oracles was introduced by Beigel [Bei91]. This technique can be described as follows. Let $N$ be an oracle NPTM. Let $B$ be a finite set of oracles and let $x \in \Sigma^*$. Then the sum of the number of accepting paths in $N^B(x)$ for each $B \in B$ can be computed in two ways. One way is to sum the total number of accepting paths for each $B \in B$. The other way is to sum, for each path $\rho$, the number of oracles $B \in B$ that make $N^B(x)$ accept along $\rho$. Formally, in our setting, we will use the following equality to prove the claim:

$$(4a) \quad \sum_B \#\text{acc}_{\hat{N}^A_T \cup B} (0^{n_s}) = \sum_\rho |\{B : \rho \in \text{ACCEPT}(\hat{N}^A_T \cup B, 0^{n_s})\}|.$$

If the claim is false, then the sum of the number of accepting paths of $\hat{N}^A_T \cup B (0^{n_s})$ over all allowable $B$, i.e., the left hand side of the equality, is not divisible by $p$. Let $\rho$ be an arbitrary augmented path in $\hat{N}^T (0^{n_s})$. In what follows, we say that a set $B$ supports an augmented computation path $\rho$ of $\hat{N}^T (0^{n_s})$ if $\rho \in \text{ACCEPT}(\hat{N}^A_T \cup B, 0^{n_s})$. We will show that the number of allowable sets $B$ supporting path $\rho$ of $\hat{N}^T (0^{n_s})$ is a multiple of $p$ in each case. According to Eq. (4a), this is a contradiction.

Let $\{q_1, \ldots, q_\ell\}$ be the set of distinct queries from $\{x \in \Sigma^{\text{ext}(n_s)} | g_{n_s} + f_{n_s} / 4 < \text{rank}(x) \leq f_{n_s}\}$ asked on path $\rho$ of $\hat{N}^T (0^{n_s})$. Let $a_i = 0$ if and only if query $q_i$ is negatively answered, and $a_i = 1$ if and only if query $q_i$ is positively answered on path $\rho$. For each $x_1, x_2, \ldots, x_m \in \Sigma^*$ such that $x_1, x_2, \ldots, x_m$ are pairwise distinct and for each $b_1, b_2, \ldots, b_m \in \{0, 1\}$, denote by $S(x_1, b_1, x_2, b_2, \ldots, x_m, b_m)$ the set of allowable sets $B$ such that $x_i \in \hat{B} \iff b_i = 1$. If path $\rho$ queries any string from $\{x \in \Sigma^{\text{ext}(n_s)} | \text{rank}(x) \leq g_{n_s}\}$ positively or any string from $Z_{n_s}$ negatively, then there is no allowable set $B$ supporting path $\rho$ of $\hat{N}^T (0^{n_s})$. Otherwise, $S(q_1, a_1, q_2, a_2, \ldots, q_\ell, a_\ell)$ is the set of allowable sets $B$ supporting path $\rho$ of $\hat{N}^T (0^{n_s})$. Let $q_{\ell+1}, q_{\ell+2}, \ldots, q_{p+\ell} \in \{x \in \Sigma^{\text{ext}(n_s)} | g_{n_s} + f_{n_s} / 4 < \text{rank}(x) \leq f_{n_s}\}$ be arbitrary strings such that $q_1, \ldots, q_{p+\ell}$ are pairwise distinct. Then

$$||S(q_1, a_1, q_2, a_2, \ldots, q_\ell, a_\ell)|| = \sum_{a_{\ell+1}=0}^{1} \sum_{a_{\ell+2}=0}^{1} \cdots \sum_{a_{p+\ell}=0}^{1} ||S(q_1, a_1, q_2, a_2, \ldots, q_{p+\ell}, a_{p+\ell})||.$$
To prove that \(|S(q_1, a_1, q_2, a_2, \ldots, q_{|q|}, a_{|q|})| \equiv 0 \pmod{p}\), we show that for each \(a_{i+1}, a_{i+2}, \ldots, a_{p^{|a|}}\), \(|S(q_1, a_1, q_2, a_2, \ldots, q_{|q|}, a_{p^{|a|}})| \equiv 0 \pmod{p}\). Consider arbitrary \(a_{i+1}, a_{i+2}, \ldots, a_{p^{|a|}}\). Let \(C\) be an arbitrary allowable set from \(S(q_1, a_1, q_2, a_2, \ldots, q_{|q|}, a_{p^{|a|}})\).

**Case 1:** \(a_1 = 0, \ldots, a_{p^{|a|}} = 0\).

Then \(q_1, \ldots, q_{p^{|a|}}\) cannot be in \(C\). There are exactly \((3/4) * f_{n_1} - g_{n_1} - p^{|a|} = 3p^{n_1} + (p-1)p^{n_1+1}\), strings remaining that potentially may be included in \(C\). To be allowable, \(C\) must either be empty or must contain exactly \(f_{n_1}/2 = 3p^{n_1} + p^{n_1+1}\), strings. Therefore, by Lucas’s theorem, the number of choices modulo \(p\) is, \(\left[1 + \left(\frac{3p^{n_1} + (p-1)p^{n_1+1}}{3p^{n_1} + p^{n_1+1}}\right)\right] \mod p = 0\).

**Case 2:** Exactly \(j\) of the \(a_i\)'s are equal to \(1\) (\(1 \leq j \leq p^{|a|}\)).

In this case, \(C\) may not be the empty set. The membership in \(C\) of \(p^{|a|}\) strings, namely \(q_1, \ldots, q_{p^{|a|}}\), is fixed. From the remaining \((3/4) * f_{n_1} - g_{n_1} - p^{|a|}\) possible strings that may be included in some allowable set, we may take an arbitrary subset of strings such that we have altogether exactly \(f_{n_1}/2\) strings in \(C\). (Note that exactly \(j\) strings are already fixed to be in \(C\).) Hence we have \(\frac{(3p^{n_1} + (p-1)p^{n_1+1})}{(p-2)p^{n_1+1}}\) different choices for \(C\). (Here, we use \(\binom{n}{j} = \binom{\alpha}{n_1}\).) Therefore by Lucas’s theorem, the number of choices modulo \(p\) is \(\left(\binom{3}{0} \cdot \binom{p-1}{2} \cdot \binom{j}{j}\right) \mod p = 0\), since \(j \geq 1\).

\(\triangleright \) (Claim 6)

**Corollary 4.5** There exists an oracle \(A\) such that, for each \(C \in \{\text{EQP, BQP, LWPP, WPP, AWPP, APP, C=P, PP}\}\) and for each \(D \in \{\text{UP, FewP, SPP}\}\), \(C^A\) is immune to \(D^A\).

**Proof** This follows from Theorem 4.4 and the inclusions stated in Proposition 2.12.

Theorem 3.13 separates \(\text{RP}\) from \(\text{C=P}\) and as a corollary we get a separation of \(\text{coRP}\) from \(\text{NQP}\). In Theorem 4.9, we prove that, relative to an oracle, \(\text{RP}\) strongly separates from \(\text{C=P}\), which in turn implies that \(\text{coRP}\) strongly separates from \(\text{NQP}\). In the proof of Theorem 4.9, we will use a sufficient condition, stated in Theorem 4.8, due to Boes, Crescenzi, and Silvestri [BCS92], for lifting simple separations between complexity classes to immunity separations.

**Definition 4.6** [BCS92] A function \(\sigma : \Sigma^* \rightarrow \Sigma^*\) is polynomially bit-computable if there exist two polynomial-time transducers \(R : \Sigma^* \times (N - \{0\}) \rightarrow \Sigma\) and \(\ell : \Sigma^* \rightarrow (N - \{0\})\) such that, for any \(x \in \Sigma^*\),

\[
\sigma(x) = R(x, 1)R(x, 2) \ldots R(x, \ell(x)).
\]

**Definition 4.7** [BCS92] Let \((A, B)\) be a pair of languages. The \(C\)-class \(C(A, B)\) is the set of all languages \(L\) for which there exists a polynomially bit computable function \(\sigma\) such that

\[
\sigma(L) \subseteq \sigma(A) \text{ and } \sigma(\overline{L}) \subseteq \overline{B}.
\]

For any class \(D\), \(D\) is \(C\)-class representable if there exist \(A, B \subseteq \Sigma^*\) such that \(D = C(A, B)\).

**Theorem 4.8** [BCS92] There exists an oracle \(E\) which strongly separates \(C^E(A_2, B_2)\) from \(C^E(A_1, B_1)\) if the following conditions hold:

1. There exists an oracle \(H\) such that \(C^H(A_2, B_2) \nsubseteq C^H(A_1, B_1)\),
2. For all \(K\), \(C^K(A_1, B_1)\) admits a \(\leq_{\text{m}}^P K\)-complete language.
3. For all \(K\), \(C^K(A_1, B_1)\) is closed with respect to the union of languages.
Using Theorem 4.8, Theorem 3.13 can be strengthened as follows.

**Theorem 4.9** There exists an oracle $A$ such that $\text{RP}^A$ contains a $\text{C}_=\text{P}^A$-immune set.

**Proof** Both the classes $\text{RP}$ and $\text{C}_=\text{P}$ are $C$-class representable because $\text{RP} = C(A_1, B_1)$ where $A_1 = \{z \mid z \in \Sigma^* \land |z|_1 > \frac{2}{3}|z|\}$ and $B_1 = \{0\}^*$, and $\text{C}_=\text{P} = C(A_2, B_2)$ where $A_2 = \{z \mid z \in \Sigma^* \land |z|_0 = |z|_1\}$ and $B_2 = \{z \mid z \in \Sigma^* \land |z|_0 \neq |z|_1\}$. From Theorem 3.13, there exists an oracle $H$ such that $\text{RP}^H \not\subseteq \text{C}_=\text{P}^H$. Also, in every relativized world, $\text{C}_=\text{P}$ admits a many-one complete language and is closed with respect to the union of languages. Thus, the relativized strong separation follows from Theorem 4.8.

Tarui [Tar91] and Green [Gre93] independently showed that $\text{BPP}$ separates from $\text{P}^{\text{C}-\text{P}}$ in some relativized world. In Theorem 4.12 we extend the oracle separation of $\text{BPP}$ from $\text{P}^{\text{C}-\text{P}}$ to a strong separation result. From this it follows that, relative to an oracle, $\text{BQP}$ strongly separates even from $\text{P}^{\text{C}-\text{P}}$. To prove this result, we use Theorem 4.11, which states that, if a class $D$ is $C$-class representable, then the Turing closure of $D$ is also $C$-class representable.

**Theorem 4.10** [Tar91, Gre93] There exists an oracle $A$ such that $\text{BPP}^A \not\subseteq \text{P}^{\text{C}-\text{P}}^A$.

**Theorem 4.11** [BCS95] For any pair of languages $(A, B)$, there exists a pair of languages $(A', B')$ such that, for all sets $X \subseteq \Sigma^*$,

$$p^{c_X}(A, B) = c^{c_X}(A', B')$$

**Theorem 4.12** There exists an oracle $A$ such that for every complexity class $C \in \{\text{BPP}, \text{BQP}, \Sigma_2^P \cap \Pi_2^P, \text{AWPP}, \text{APP}, \text{PP}\}$, $C^A$ contains a $\text{P}^{\text{C}-\text{P}}$-immune set.

**Proof** We show the existence of an oracle $A$ such that $\text{BPP}^A$ is immune to $\text{P}^{\text{C}-\text{P}}^A$. The statement holds for the other classes because they contain $\text{BPP}$ in every relativized world. $\text{BPP}$ is $C$-class representable because $\text{BPP} = C(A_1, B_1)$, where $A_1 = \{z \mid z \in \Sigma^* \land |z|_1 \leq \frac{2}{3}|z|\}$ and $B_1 = \{z \mid z \in \Sigma^* \land |z|_0 \leq \frac{2}{3}|z|\}$. Since $\text{C}_=\text{P}$ is $C$-class representable, it follows from Theorem 4.11 that $\text{P}^{\text{C}-\text{P}}$ is also $C$-class representable. It is known that in every relativized world, $\text{P}^{\text{C}-\text{P}}$ admits a complete language and is closed under union. Thus from Theorem 4.10 and Theorem 4.8, it follows that $\text{BPP}$ contains a $\text{P}^{\text{C}-\text{P}}$-immune set in some relativized world.

Since it is not clear whether $\text{EQP}$ is $C$-class representable, Theorem 4.8 cannot be applied to strengthen the oracle separation of $\text{EQP}$ from $\text{Mod}_p \text{P}$ [dGV02] to a relativized immunity separation. However, Theorem 4.4 shows that $\text{EQP}$ separates from $\text{Mod}_p \text{P}$ with immunity in a certain relativized world. Theorem 4.8 is also not useful for strengthening the oracle separation of Theorem 3.1, since there are oracles relative to which $\text{WPP}$ has no complete sets [ST04b, ST04a].

## 5 Closure and Collapse Results

In this section, we further study properties of these counting classes and use them to prove consequences of the following hypotheses about quantum classes: $\text{NQP} \subseteq \text{BQP}$ and $\text{NQP} \subseteq \text{EQP}$. Note that these hypotheses are the quantum analogs of the “$\text{NP} \subseteq \text{BPP}$” and “$\text{NP} \subseteq \text{P}$” hypotheses. We show that these hypotheses involving quantum classes have interesting consequences for the polynomial hierarchy.

Zachos [Zac88] proved that $\text{NP} \not\subseteq \text{BPP}$ unless the polynomial hierarchy is contained in $\text{BPP}$, and thus it is unlikely that $\text{NP} \subseteq \text{BPP}$. In this section, we prove in Corollary 5.5 a similar consequence.
for \( \text{NQP} \subseteq \text{BQP} \): \( \text{NQP} \subseteq \text{BQP} \implies \text{PH} \subseteq \text{AWPP} \). For the proof of that implication, we show a new reduction closure property of AWPP and make use of an observation—PH is contained in \( \text{UP}^{\text{C-L-P}} \) in every relativized world—that follows from Toda’s theorem [Toda91]. This observation is implicit in a paper by Vyalyi [Vyay03], who showed that if QMA, the quantum analog of the Merlin-Arthur class MA, equals PP, then the polynomial hierarchy is contained in PP. For completeness, we sketch a proof of this observation by Vyalyi [Vyay03].

**Theorem 5.1** [Vyay03] \( \text{PH} \subseteq \text{UP}^{\text{C-L-P}} \).

**Proof** Let \( L \) be an arbitrary language in the polynomial hierarchy. From Toda’s [Toda91] theorem, \( \text{PH} \subseteq \text{P}^{\text{#F}[1]} \), there exists an oracle DPTM \( M \) and a \#P function \( f \) such that \( L = L(M^f) \) and for all \( x \in \Sigma^* \), \( M(x) \) asks only a single query to oracle \( f \). For every \( x \in \Sigma^* \), let \( q_x \) be the query asked by \( M(x) \) to \( f \). Since \( M \) is a polynomial-time machine and \( f \) is a \#P oracle, we can assume that there is a polynomial \( p(.) \) such that, for all \( x \in \Sigma^* \), \( 0 \leq f(q_x) \leq 2^{p(|x|)} \). Define \( A = \{ (y,k) \mid f(y) = k \} \). It is easy to see that \( A \in \text{C=P} \). (The function \( g((y,k)) = f(y) - k \) is in GapP since \( f \in \#P \subseteq \text{GapP} \), and we know that GapP is closed under intersection. Apply the definition of \( \text{C=P} \).) Consider an oracle NPTM \( N \) such that \( N^A(x) \) is defined as follows.

1. Imitate the behavior of \( M(x) \) until the query \( q_x \) to the \#P-oracle \( f \) is generated.
2. Guess an integer \( j \) between 0 and \( 2^{p(|x|)} \) and query to \( A \) whether \((q_x,j) \in A \).
3. If the oracle answer is “yes,” then go on with the simulation of \( M(x) \) with \( j \) taken as oracle answer. Accept \( x \) if and only if \( M(x) \) accepts.
4. Otherwise, i.e., if the oracle answer is “no,” then reject.

It is easy to verify that \( L(N^A) \) is in \( \text{UP}^A \) and \( L(N^A) = L \). It follows that \( \text{PH} \subseteq \text{UP}^{\text{C-L-P}} \). \( \square \)

Li [Li93] studied closure properties of AWPP and APP, and showed that these classes are closed under intersection, union, complementation, and join. Li [Li93] also showed that APP is closed under polynomial-time Turing reductions and used this result to prove that APP is low for PP. In Theorem 5.4 we show that both AWPP and APP are closed under stronger \( \leq_{\text{UP}} \) (unambiguous nondeterministic polynomial-time Turing) reductions. From this closure property of AWPP and Theorem 5.1, we conclude that if \( \text{NQP} \subseteq \text{BQP} \) then \( \text{PH} \subseteq \text{AWPP} \).

**Definition 5.2** We say that \( A \leq_{\text{UP}} B \) if there exists an oracle NPTM \( N \) such that the following hold:

1. \( L(N^B) = A \), and
2. \( (\forall x \in \Sigma^*)[\#\text{acc}_{N^B}(x) \leq 1] \).

**Definition 5.3** For any \( C \), \( \text{UP}^C = \{ L \mid (\exists A \in C)[L \leq_{\text{UP}} A] \} \).

**Theorem 5.4** (a) \( \text{UP}^{\text{AWPP}} \subseteq \text{AWPP} \). (b) \( \text{UP}^{\text{APP}} \subseteq \text{APP} \).

**Proof** We will prove (a) only, since the proof for (b) is similar. The proof of (a) is also similar to the proof of the result that APP is closed under polynomial-time Turing reductions by Li [Li93]. Let \( L \in \text{UP}^{\text{AWPP}} \) via an oracle NPTM \( N \) and an oracle \( B \in \text{AWPP} \). Let \( r(.) \) be a polynomial (which we will fix later) and let the corresponding GapP function \( g_B \) and the polynomial \( p(.) \) define \( B \) as in Definition 2.3. Without loss of generality, we assume that there exist polynomials \( t(.) \), \( q(.) \), and \( s(.) \) such that for each \( A \subseteq \Sigma^* \), \( x \in \Sigma^* \), and augmented path \( p = (p',\sigma) \) of \( N^A(x) \), \( |p| \leq t(|x|) \), \( |\sigma| = q(|x|) \), and the length of any query along \( p \) is at most \( s(|x|) \). We define an NPTM \( N' \) as follows. On input \( x \in \Sigma^* \), \( N'(x) \) guesses an augmented computation path \( \rho_N = (\rho_N',\sigma) \) of \( N(x) \) and
simulates $N(x)$ on $\rho'_N$ with query answers from $\sigma$. If $N(x)$ on $\rho'_N$ with answers $\sigma_1, \ldots, \sigma_{q(|x|)}$ to the queries rejects, then $N'(x)$ generates a gap of 0 by branching into two paths, rejecting on the one and accepting on the other, and if $N(x)$ on $\rho_N$ with answers $\sigma_1, \ldots, \sigma_{q(|x|)}$ to the queries accepts, then $N'(x)$ generates a gap of $h((x, \rho_N))$, where $h$ is defined for each $x, \rho_N \in \Sigma^*$ as follows. (Note that in the definition of $h$ below, $z_1, \ldots, z_{q(|x|)}$ are the queries that $N(x)$ asks of its oracle along augmented path $\rho_N$.) Let $h((x, \rho_N)) = \prod_{i=1}^{q(|x|)} h_i$, where, for each $1 \leq i \leq q(|x|)$, $h_i$ is defined as follows:

$$h_i = \begin{cases} g_B(z_i) - 2^{(|x|)} - g_B(z_i) & \text{if } \sigma_i = 1, \\
2^{(|x|)} - g_B(z_i) & \text{if } \sigma_i = 0. \end{cases}$$

Clearly, $h \in \text{GapP}$. Let $p'(\cdot), r'(\cdot)$ be polynomials such that, for all $n \in \mathbb{N}$, $p'(n) = p(s(n))$ and $r'(n) = r(s(n))$. It is easy to see that, for each $1 \leq i \leq q(|x|)$,

$$\sigma_i = \chi_B(z_i) \implies (1 - 2^{-r'(|x|)})2^{r'(|x|)} \leq h_i \leq 2^{-r'(|x|)},$$

and

$$\sigma_i \neq \chi_B(z_i) \implies 0 \leq h_i \leq 2^{-r'(|x|)}2^{r'(|x|)}.$$ 

If $x \in L$, then there is a unique augmented accepting path $\rho_{acc} = \langle \rho'_N, \sigma_{acc} \rangle$ with query answers $\sigma_{acc}$ consistent with $B$ in $N(x)$. Clearly, $(1 - 2^{-r'(|x|)})g_B(x)/2^{r'(|x|)} \leq h((x, \rho_{acc})) \leq 2^{r'(|x|)}q(|x|)$. However, if $x \not\in L$, then for each augmented path $\rho$ of $N(x)$, $0 \leq h((x, \rho)) \leq 2^{-r'(|x|)}2^{r'(|x|)}q(|x|)$.

Thus,

$$\vspace{0.3cm}$$

(5.a) \hspace{2cm} x \in L \implies (1 - 2^{-r'(|x|)})g_B(x)/2^{r'(|x|)} \leq 1 + (2^{r'(|x|)} - 1)2^{-r'(|x|)},$$

(5.b) \hspace{2cm} x \not\in L \implies 0 \leq \frac{g_B(x)}{2^{r'(|x|)q(|x|)}} \leq (2^{r'(|x|)} - 1)2^{-r'(|x|)}.$$ 

Since for all $x \in \Sigma^*$, $t(|x|) \geq q(|x|)$, we get

$$(1 - 2^{-r'(|x|)})g_B(x)/2^{r'(|x|)} \geq 1 - q(|x|)2^{-r'(|x|)} \geq 1 - 2^{-r'(|x|)}q(|x|) \geq 1 - 2^{-r'(|x|)}t(|x|)).$$ 

Thus, (5.a) and (5.b) can be expressed as

$$\vspace{0.3cm}$$

(5.c) \hspace{2cm} x \in L \implies 1 - 2^{-r''(|x|)} \leq \frac{g_B(x)}{2^{r'(|x|)q(|x|)}} \leq 1 + 2^{-r''(|x|)}t(|x|)),$$

(5.d) \hspace{2cm} x \not\in L \implies 0 \leq \frac{g_B(x)}{2^{r'(|x|)q(|x|)}} \leq 2^{-r''(|x|)}t(|x|)).$$

We choose the polynomial $r''(\cdot)$ large enough so that, for all $x \in \Sigma^*$, $r''(|x|) - t(|x|) = r'''(|x|) > 1$ and $r'''$ is nondecreasing. Thus,

$$\vspace{0.3cm}$$

(5.e) \hspace{2cm} x \in L \implies 1 - 2^{-r'''(|x|)} \leq \frac{g_B(x)}{2^{r'(|x|)q(|x|)}} \leq 1 + 2^{-r'''(|x|)},$$

(5.f) \hspace{2cm} x \not\in L \implies 0 \leq \frac{g_B(x)}{2^{r'(|x|)q(|x|)}} \leq 2^{-r'''(|x|)}.$$ 

Consider a GapP function $G$ such that, for all $x \in \Sigma^*$, $G(x) = (2^{r''(|x|)}g_B(x) + 2^{r''(|x|)q(|x|)})2^{-r''(|x|)}t(|x|) - 1)$. For all $x \in \Sigma^*$, let $r'''(|x|) = r''(|x|) - 1$ and $p''(|x|) = p''(|x|)q(|x|) + 2r''(|x|) - 1$. Then, $\vspace{0.3cm}$

$$\vspace{0.3cm}$$

$$\vspace{0.3cm}$$

$$\begin{align*}
x \in L & \implies 1 - 2^{-r'''(|x|)} \leq \frac{G(x)}{2^{r''(|x|)}} \leq (1 + 2^{-r'''(|x|)})(1 - 2^{-r'''(|x|)}) \leq 1, \\
x \not\in L & \implies 0 \leq \frac{G(x)}{2^{r''(|x|)}} \leq 2^{-r'''(|x|)}.$$

23
By Theorem 2.5(1), $L \in \text{AWPP}$.

**Corollary 5.5** If $\text{NQP} \subseteq \text{BQP}$, then $\text{PH} \subseteq \text{AWPP}$.

**Proof** Suppose that $\text{NQP} \subseteq \text{BQP}$. Since $\text{NQP} = \text{coC}_{=\text{P}} [\text{FGHP98,YY99}]$, $\text{BQP} \subseteq \text{AWPP} [\text{FR99}]$, and BQP is closed under complementation, we get that $\text{C}_{=\text{P}} \subseteq \text{BQP} \subseteq \text{AWPP}$. By Theorem 5.1 and 5.4(a), it follows that $\text{PH} \subseteq \text{UP}^{\text{coC}_{=\text{P}}} \subseteq \text{UP}^{\text{AWPP}} \subseteq \text{AWPP}$.

Since AWPP is low for PP, we also conclude that if $\text{NQP} \subseteq \text{BQP}$, then PH is low for PP. However, the mere PP-lowness of PH can also be derived easily from known results as follows. By Toda and Ogihara [TO92] and Tarui [Tar93], we have that $\text{PH} \subseteq \text{BPP}^{\text{C}_{=\text{P}}}$. Since $\text{coC}_{=\text{P}} = \text{NQP}$ and BPP is low for PP relative to all oracles, we get that $\text{PP}^{\text{PH}} \subseteq \text{P}^{\text{BPP}^{\text{C}_{=\text{P}}}} \subseteq \text{P}^{\text{P}^{\text{NQP}}}$. Hence, under the assumption that $\text{NQP} \subseteq \text{BQP}$, it follows that $\text{PP}^{\text{PH}} \subseteq \text{P}^{\text{BQP}}$, and so by the PP-lowness of BQP, we get that $\text{PP}^{\text{PH}} \subseteq \text{P}^{\text{BQP}} \subseteq \text{PP}$. We emphasize that the interest of Theorem 5.4 lies in deriving the stronger result $\text{NQP} \subseteq \text{BQP} \implies \text{PH} \subseteq \text{AWPP}$ instead of the mere lowness of PH for PP. Assuming that $\text{NQP} \subseteq \text{BQP}$, Corollary 5.5 provides an intuition on the position of the polynomial hierarchy relative to the classes that are low for PP, whereas it is not possible to get the same intuition with the mere lowness result of the polynomial hierarchy.

In the rest of this section, we turn our attention to closure properties of WPP. In Theorem 5.6, we prove that WPP is closed under the polynomial-time truth-table reduction, while its closure under the stronger $\leq_{\text{UP}}$ reduction is contained in $\text{coC}_{=\text{P}}$ (in every relativized world). By Corollary 5.5, $\text{NQP} \subseteq \text{BQP}$ implies $\text{PH} \subseteq \text{AWPP}$. On the other hand, Theorem 5.6(b) along with Theorem 5.1 allows us to show that the stronger hypothesis, namely $\text{NQP} \subseteq \text{EQP}$, implies a stronger conclusion about the polynomial hierarchy, namely $\text{PH} \subseteq \text{EQP}$.

**Theorem 5.6** (a) WPP is closed under polynomial-time truth-table reductions. (b) $\text{UP}^{\text{WPP}} \subseteq \text{coC}_{=\text{P}}$.

**Proof** We will prove (a). The proof for (b) is similar. The proof of (a) uses ideas reminiscent of the proof of Gap$^{\text{P}^{\text{PP}}}$ = Gap$^{\text{P}}$ by Fenner, Fortnow, and Kurtz [FFK94]. Let $L$ be an arbitrary language accepted by an oracle DPTM $M$ with nonadaptive access to an oracle $B \in \text{WPP}$. Then, there is an NPTM $N_B$ and an NP function $f_B$ with $0 \notin \text{range}(f_B)$ such that, for all $x \in \Sigma^*$, if $x \in B$, then $\text{gap}_{N_B}(x) = f_B(x)$, and if $x \notin B$, then $\text{gap}_{N_B}(x) = 0$. Without loss of generality, we may assume that, on any input $x$, $M^A(x)$ makes exactly $p(|x|)$ queries (in parallel, i.e., non-adaptively) where $p$ is some polynomial. We define an NPTM $M'$ as follows. $M'$, on an arbitrary input $x$, simulates $M(x)$ to get a list of queries that $M(x)$ asks of its oracle. $M'$ then nondeterministically guesses the answers to these queries. Let $q_1, q_2, \ldots, q_{p(|x|)}$ be the queries that $M(x)$ asks of its oracle, and let $a_1, a_2, \ldots, a_{p(|x|)}$ be the corresponding guessed answers (for each $i$, $a_i \in \{0, 1\}$) along an arbitrary computation path $\rho$. On path $\rho$, $M'$ does the following. If $M(x)$, with answers $a_1, a_2, \ldots, a_{p(|x|)}$ to its queries rejects, then $M'(x)$ along path $\rho$ generates a gap of 0 by guessing a non-deterministic bit and accepting on the path if and only if the guessed bit is 1. On the other hand, if $M(x)$, with answers $a_1, a_2, \ldots, a_{p(|x|)}$ to its queries accepts, then $M'(x)$ generates a gap of $g((x, \rho))$ extending $\rho$, where $g$ is defined, for each $x$ and $\rho$, as follows. (Note that in the definition of $g$ below, $q_1, q_2, \ldots, q_{p(|x|)}$ are the queries that $M(x)$ asks of its oracle and $a_1, a_2, \ldots, a_{p(|x|)}$ are the answers to these queries guessed on the path $\rho$. $g((x, \rho)) = g(\prod_{i=1}^{p(|x|)} q_i)$, where, for each $1 \leq i \leq p(|x|)$, $g_i$ is defined as follows:

$$g_i := \begin{cases} \text{gap}_{N_B}(q_i), & \text{if } a_i = 1, \\ f_B(q_i) - \text{gap}_{N_B}(q_i), & \text{if } a_i = 0. \end{cases}$$
Since GapP is closed under subtraction and polynomial times iterated multiplication [FFK94], \( g \) is indeed a GapP function. It is easy to see, that for each \( 1 \leq i \leq p(|x|) \),

\[
g_i = \begin{cases} f_B(q_i) & \text{if } a_i = \chi_B(q_i), \\ 0 & \text{otherwise.} \end{cases}
\]

Since there is exactly one path \( \rho \) in \( M'(x) \) that guesses the answers of all the queries correctly, it follows that \( \text{gap}_{M'}(x) = \prod_{i=1}^{p(|x|)} f_B(q_i) \) if \( M(x) \) accepts, and \( \text{gap}_{M'}(x) = 0 \) otherwise. Define \( f' \) as follows. For each \( x \in \Sigma^* \), \( f'(x) = \prod_{i=1}^{p(|x|)} f_B(q_i) \), where \( q_1, q_2, \ldots, q_{p(|x|)} \) are the queries asked by \( M(x) \) of its oracle. Since \( f_B \in \text{FP} \), \( f' \in \text{FP} \). Thus, for all \( x \in \Sigma^* \), if \( x \in L \), then \( \text{gap}_{M'}(x) = f'(x) \), otherwise (i.e., if \( x \notin L \)) \( \text{gap}_{M'}(x) = 0 \). Thus, \( M' \) and \( f' \) witness that \( L \) is in WPP. Therefore, WPP is closed under polynomial-time truth-table reductions.

**Corollary 5.7** If \( \text{NQP} \subseteq \text{EQP} \), then \( \text{PH} \subseteq \text{EQP} \).

**Proof** Suppose that \( \text{NQP} \subseteq \text{EQP} \). Since \( \text{NQP} = \text{coC=P} [\text{FGHP98,YY99}] \), \( \text{EQP} \subseteq \text{LWPP} [\text{FR99}] \), and \( \text{EQP} \) is closed under complementation, we get that \( \text{C=P} \subseteq \text{EQP} \subseteq \text{LWPP} \subseteq \text{WPP} \). By Theorems 5.1 and 5.6(b), and our assumption that \( \text{NQP} \subseteq \text{EQP} \), it follows that \( \text{PH} \subseteq \text{UP} \cap \text{coUP} \subseteq \text{UP} \cap \text{WPP} \subseteq \text{coC=P} = \text{NQP} \subseteq \text{EQP} \).

Theorem 5.6 states that WPP is closed under truth-table reductions. In Theorem 5.9, we will prove that relativizable techniques cannot improve this to polynomial-time Turing reductions. This gives a partial answer to an open question raised by Fenner, Fortnow, and Kurtz [FFK94]. They showed that \( \text{SPP} \) and \( \text{LWPP} \) are closed under polynomial-time Turing reductions, but left open the corresponding problem for \( \text{WPP} \): Is \( \text{WPP} \) closed under polynomial-time Turing reductions? Since \( \text{LWPP} \) is closed under polynomial-time Turing reductions in every relativized world [FFK94], Theorem 5.9 also yields an oracle separating the seemingly similar classes \( \text{LWPP} \) and \( \text{WPP} \).

Recently, Spakovski and Tripathi [ST04b,ST04a] used ideas in the proof of Theorem 5.9 to construct a relativized world where \( \text{UP} \cap \text{coUP} \) is not low for \( \text{LWPP} \) as well as for \( \text{WPP} \). Using this result, they were able to settle an open question of Fenner, Fortnow, and Kurtz [FFK94]: Is \( \text{WPP} \) uniformly gap-definable? They showed that both \( \text{LWPP} \) and \( \text{WPP} \) are not uniformly gap-definable. (See [FFK94] for the definition of uniform and non-uniform gap-definability.) Their result makes both \( \text{LWPP} \) and \( \text{WPP} \) peculiar compared to other known uniformly gap-definable counting classes (such as \( \text{PP} \), \( \text{C=P} \), \( \text{Mod}_k \text{P} \), and \( \text{SPP} \)), since \( \text{LWPP} \) and \( \text{WPP} \) are natural counting classes that are nonuniformly gap-definable but are not uniformly gap-definable. The ideas used in the proof of Theorem 5.9 might have more applications in constructing oracles involving counting classes and, therefore, might be helpful in understanding the relativized complexity of quantum classes such as \( \text{EQP} \), \( \text{BQP} \), and \( \text{NQP} \).

We now prove that, relative to an oracle, \( \text{WPP} \) is not closed under polynomial-time Turing reductions. Before we state and prove Theorem 5.9, we state a lemma that will be needed in the proof of this result.

**Lemma 5.8** [RS62] For every \( n \geq 17 \), the number of primes less than or equal to \( n \), \( \pi(n) \), satisfies

\[
n/\ln n < \pi(n) < 1.25506n/\ln n.
\]

**Theorem 5.9** (\( \exists A \)[\( \text{P}^{\text{WPP}} \) \( \not\subset \) \( \text{WPP}^A \)].

**Proof** For any \( \alpha \in \Sigma^* \), let \( \text{pos}(\alpha) = \text{num}(\alpha) - 2^{|\alpha|} + 1 \) represent the lexicographic position of \( \alpha \) among strings of length \( |\alpha| \). For every set \( A \subseteq \Sigma^* \), \( \alpha \in \Sigma^* \), and \( n \in \mathbb{N} \), we define “Witcount,” “Promise,” and “Boundary” as follows.
Witcount\((A, \alpha) = \| \{ x \in \Sigma^* \mid |x| = |\alpha| \land \alpha x \in A \} \| \),
\text{Promise}(A, n) \equiv (\forall \alpha \in \Sigma^n)[\text{Witcount}(A, \alpha) = 0 \lor \text{Witcount}(A, \alpha) = \text{pos}(\alpha)] \land
(\forall \alpha_1, \alpha_2 \in \Sigma^n)[\text{pos}(\alpha_1) \leq \text{pos}(\alpha_2) \land \text{Witcount}(A, \alpha_2) \neq 0 \Rightarrow \text{Witcount}(A, \alpha_1) \neq 0] , \text{and}
\text{Boundary}(A, n) = \max\{\text{pos}(\alpha) \mid |\alpha| = n \land \text{Witcount}(A, \alpha) \neq 0\}.

For every set \( A \subseteq \Sigma^* \), define \( L_A \) as follows.
\[
L_A = \{0^n \mid \text{Boundary}(A, n) \equiv 1 \pmod{2}\}.
\]

Clearly, if \( A \) satisfies \( \text{Promise}(A, n) \) at each length \( n \), then \( L_A \) is in \( \text{P}^{\text{WPP}^A} \) (using binary search along the strings \( \alpha \) with \( |\alpha| = n \)). We construct an oracle \( A \) such that, for each \( n \), \( \text{Promise}(A, n) \) is true and \( L_A \notin \text{WPP}^A \). Let \( (N_s, M_s, p_s)_{s \geq 1} \) be an enumeration of all triples such that \( N_s \) is a nondeterministic polynomial-time oracle Turing machine, \( M_s \) is a deterministic polynomial-time oracle transducer, \( p_s \) is a polynomial, and the running time of both \( N_s \) and \( M_s \) is bounded by \( p_s \) regardless of the oracle. We need the following technical lemmas, the proofs of which are given later at the end of this proof.

**Lemma 5.10** Let \( N, p \in \mathbb{N} \), where \( 1 < p \leq N/2 \). Let \( s(y_1, y_2, \ldots, y_N) \) be a multilinear polynomial with rational coefficients, where each monomial has exactly \( p-1 \) different variables. Suppose that for some \( \text{val} \in \mathbb{Q} \), it holds that \( s(y_1, y_2, \ldots, y_N) = \text{val} \) for every \( y_1, y_2, \ldots, y_N \in \{0, 1\} \) with \( \sum_{i=1}^{N} y_i = p \). Then each monomial in \( s(y_1, y_2, \ldots, y_N) \) has the same rational coefficient, i.e.,
\[
s(y_1, y_2, \ldots, y_N) = \sum_{1 \leq i_1 < i_2 < \ldots < i_{p-1} \leq N} (\text{val}/p) \cdot y_{i_1} y_{i_2} \cdots y_{i_{p-1}}.
\]

**Lemma 5.11** Let \( N \in \mathbb{N} \) and \( p \) be a prime with \( p \leq N/2 \). Let \( s(y_1, y_2, \ldots, y_N) \) be a multilinear polynomial of total degree \( < p \) with integer coefficients. If for some \( \text{val} \in \mathbb{Z} \), it holds that
\begin{enumerate}
  \item \( s(0, 0, \ldots, 0) = 0 \), and
  \item \( s(y_1, y_2, \ldots, y_N) = \text{val} \), for every \( y_1, y_2, \ldots, y_N \in \{0, 1\} \) with \( \sum_{i=1}^{N} y_i = p \),
\end{enumerate}
then \( p \mid \text{val} \).

The oracle \( A \) is constructed in stages. In stage \( s \), the membership in \( A \) of strings of length \( 2n_s \) is decided and the initial segment \( A_{s-1} \) is extended to \( A_s \). Our choice of \( n_s \) guarantees that the oracle extension in stage \( s \) does not affect the computation in earlier stages. Set \( A_0 := \emptyset \) and \( n_0 := 17 \).

**Stage \( s \), where \( s \geq 1 \):** Let \( n_s \) be large enough so that the previous stages are not affected and \( 2^{n_s} > 4n_s^2p_s(n_s) \). We diagonalize against nondeterministic polynomial-time oracle Turing machine \( N_s \) and deterministic polynomial-time oracle transducer \( M_s \). Let \( \text{val} \) be the value computed by \( M_s^{A_{s-1}}(0^n) \). Without loss of generality, we assume that \( \text{val} \neq 0 \). Let \( T = \{ \alpha \in \Sigma^{2n_s} \mid M_s^{A_{s-1}}(0^n) \} \) queries \( \alpha \} \).

(*** Choose a set \( B, B \subseteq T \cap \Sigma^{2n_s}, \) satisfying \( \text{Promise}(B, n_s) \) such that the following holds:
\begin{align*}
\text{Boundary}(B, n_s) &\equiv 1 \pmod{2} \quad \text{and} \quad \text{gap}_{N_s^{A_{s-1} \cup B}}(0^n) \neq \text{val}, \quad \text{or} \\
\text{Boundary}(B, n_s) &\equiv 0 \pmod{2} \quad \text{and} \quad \text{gap}_{N_s^{A_{s-1} \cup B}}(0^n) \neq 0.
\end{align*}

Let \( A_s := A_{s-1} \cup B \). Clearly, the construction guarantees that \( L_A \notin \text{WPP}^A \). The feasibility of the construction follows from the following claim.

**Claim 7** For each \( s \geq 1 \), there exists an oracle extension \( B \) satisfying (***).
**Proof** Suppose that in stage $s$ no set $B$ satisfying $(\ast \ast \ast)$ exists. Then, for every $B \subseteq T \cap \Sigma^{n_s}$ satisfying Promise$(B, n_s)$, the following hold.

(5.g) \[ \text{Boundary}(B, n_s) \equiv 1 \pmod{2} \implies \text{gap}_{N_s^{A_{s-1} \cup B}}(0^{n_s}) = \text{val}, \]

(5.h) \[ \text{Boundary}(B, n_s) \equiv 0 \pmod{2} \implies \text{gap}_{N_s^{A_{s-1} \cup B}}(0^{n_s}) = 0. \]

Let $U = \{ \alpha \in \Sigma^{n_s} \mid \text{pos}(\alpha) \text{ is prime, and } 2^{n_s-2} \leq \text{pos}(\alpha) \leq \frac{3}{2} \cdot 2^{n_s-2} \}$. Fix an arbitrary $\alpha \in U$. Choose a set $C_\alpha \subseteq T \cap \Sigma^{n_s}$ satisfying (a) Promise$(C_\alpha, n_s)$, and (b) Boundary$(C_\alpha, n_s) = \text{pos}(\alpha) - 1$. Such a set $C_\alpha$ always exists because $2^{n_s} - p_s(n_s) > \frac{3}{2} \cdot 2^{n_s-2}$. Statements (5.g) and (5.h) in particular imply that, for all $D_\alpha \subseteq T \cap \alpha \Sigma^{n_s}$, it holds that

(5.i) \[ \text{Witcount}(D_\alpha, \alpha) = 0 \implies \text{gap}_{N_s^{A_{s-1} \cup C_\alpha \cup D_\alpha}}(0^{n_s}) = 0, \]

(5.j) \[ \text{Witcount}(D_\alpha, \alpha) = 0 \implies \text{gap}_{N_s^{A_{s-1} \cup C_\alpha \cup D_\alpha}}(0^{n_s}) = \text{val}. \]

Henceforth, we use $N$ to denote $|T \cap \alpha \Sigma^{n_s}|$. Let $x_1, x_2, \ldots, x_N$ be the lexicographic enumeration of the strings in $T \cap \alpha \Sigma^{n_s}$. We define $s_\alpha$ to be the function $\{0,1\}^N \to \mathbb{Z}$ that has the following property. For all $D_\alpha \subseteq T \cap \alpha \Sigma^{n_s}$,

(5.k) \[ s_\alpha(\chi_{D_\alpha}(x_1), \chi_{D_\alpha}(x_2), \ldots, \chi_{D_\alpha}(x_N)) = \text{gap}_{N_s^{A_{s-1} \cup C_\alpha \cup D_\alpha}}(0^{n_s}). \]

We will show that $s_\alpha$ can be represented by a multilinear polynomial having low total degree. For arbitrary $z_1, z_2, \ldots, z_N \in \{0,1\}$, we call a computation path $\rho$ of $N_s^{(\ast \ast \ast)}(0^{n_s})$ “($z_1, z_2, \ldots, z_N$)-allowable” if, along $\rho$, all queries $q \in A_{s-1} \cup C_\alpha$ have a “yes” answer, all queries $q \notin A_{s-1} \cup C_\alpha \cup (T \cap \alpha \Sigma^{n_s})$ have a “no” answer, all queries $x_i$ with $z_i = 1$ are answered “yes,” and all queries $x_i$ with $z_i = 0$ are answered “no.” Let $z_1, z_2, \ldots, z_N \in \{0,1\}$ and let $\rho$ be a ($z_1, z_2, \ldots, z_N$)-allowable path of $N_s^{(\ast \ast \ast)}(0^{n_s})$. Let $x_i, x_{i+1}, \ldots, x_{i+j}$, where $\ell \leq p_s(n_s) < 2^{n_s-2}/n_s^2 < 2^{n_s-2}$, be the distinct queries to strings in $T \cap \alpha \Sigma^{n_s}$ along $\rho$. Create a monomial $\text{mono}(\rho)$ that is the product of terms $\gamma_k$, $k = 1, 2, \ldots, \ell$, where $\gamma_k = y_{ik}$ if $z_{ik} = 1$, and $\gamma_k = (1 - y_{ik})$ otherwise. Let

\[ s_\alpha'(y_1, y_2, \ldots, y_N) = \sum_{z_1, z_2, \ldots, z_N \in \{0,1\}} \sum_{\rho \in \text{($z_1, z_2, \ldots, z_N$)-allowable}} \text{sign}(N_s, 0^{n_s}, \rho) \cdot \text{mono}(\rho). \]

It is easy to see that the thus constructed multilinear polynomial $s_\alpha'(y_1, y_2, \ldots, y_N)$ coincides with $s_\alpha$ on $\{0,1\}^N$, and has total degree $\leq p_s(n_s) < 2^{n_s-2}/n_s^2 < \text{pos}(\alpha) < N/2$. Statements (5.i) and (5.j) imply that for all $z_1, z_2, \ldots, z_N \in \{0,1\}$ such that $\sum_{k=1}^N z_i = \text{pos}(\alpha)$,

\[ s_\alpha(z_1, z_2, \ldots, z_N) = \text{val and } s_\alpha(0,0,\ldots,0) = 0. \]

It follows from Lemma 5.11 that $\text{pos}(\alpha) \mid \text{val}$. Therefore, for each $\alpha \in U$, $\text{pos}(\alpha) \mid \text{val}$. Hence,

\[ \text{val} \geq \prod_{\alpha \in U} \text{pos}(\alpha) \geq 2|U|| \geq 2^{\pi(\frac{3}{2} \cdot 2^{n_s-2})} \geq 2^{n_s-2}/n_s^2 \geq 2^{p_s(n_s)}, \]

where the fourth inequality follows from Lemma 5.8 and the fifth inequality follows because $2^{n_s} > 4n_s^2p_s(n_s)$. However, $\text{val} \leq 2^{p_s(n_s)}$, because the running time of $M_s^{(\ast)}(0^{n_s})$ is bounded by $p_s(n_s)$ regardless of the oracle. Thus, for each $s \geq 1$, $A_{s-1}$ can always be extended in stage $s$. 

(Claim 7 and Theorem 5.9)
Proof of Lemma 5.10 Assume that the hypothesis of the lemma is true. For each $1 \leq i \leq N$, we identify variable $y_i$ by its index $i$ and identify a monomial $\prod_{j=1}^{N} y_{j}$ by the set of indices $\{i_1, i_2, \ldots, i_k\}$. Let $A$ denote the collection of all subsets of $\{1, 2, \ldots, N\}$ of size $p$ and let $B$ denote the collection of all subsets of $\{1, 2, \ldots, N\}$ of size $p - 1$. W.l.o.g. we assume that the elements of $A$ and $B$ are ordered in an arbitrary but fixed manner. We use $m = \binom{N}{p}$ to denote the size of $A$ and $n = \binom{N}{p-1}$ to denote the size of $B$. Let $A_i$, where $1 \leq i \leq m$, denote the $i$th element of $A$, and $B_j$, where $1 \leq j \leq n$, denote the $j$th element of $B$. For a 2-dimensional matrix $M_{m \times n}$, let $\text{Row}(M,i)$, where $1 \leq i \leq m$, denote the $i$th row of $M$.

The condition $s(y_1, y_2, \ldots, y_N) = \text{val}$ for every $y_1, y_2, \ldots, y_N \in \{0, 1\}$ with $\sum_{i=1}^{N} y_i = p$, as given in the hypothesis, can be expressed in terms of a matrix equation $M_{m \times n}X_{n \times 1} = b_{n \times 1}$. Here $M_{m \times n}$ is a 0-1 matrix whose $(i,j)$ entry, $M[i,j]$, is one if $A_i \supseteq B_j$ and is zero otherwise, $X_{n \times 1}$ is a column vector with the $j$th entry, $X[j]$, is a variable that denotes the coefficient of monomial $B_j$, and $b_{m \times 1}$ is a column vector with each entry $b_i, 1 \leq i \leq m$, equals $\text{val}$. By assigning all coefficients $X[j]$ the value $\text{val}/p$, we obtain clearly a solution for the system of equations. Hence it is sufficient to prove that the solution is unique, i.e., $\text{rank}(M) = n$. We show that it is possible to express each canonical vector $e_i = [0, \ldots, 0, 1, 0, \ldots, 0], 1 \leq i \leq n$, as a linear combination of row vectors in $M$. W.l.o.g. we show that for vector $e_1 = [1, 0, \ldots, 0]$,

Form a matrix $\tilde{M}_{p \times n}$ in the following way. Row $k$, where $1 \leq k \leq p$, of $\tilde{M}$ is the sum of all rows $i$ in $M$ with $||A_i \cap B_j|| = p - k$. Note that there is at least one row $i$ with $||A_i \cap B_j|| = p - k$. This follows from $\{1, 2, \ldots, N\} - B_i| \geq p$, which is true because of the condition $p \leq N/2$.

Claim 8 The matrix $\tilde{M}$ has the following properties. For every row $k$ ($1 \leq k \leq p$),

1. $\tilde{M}[k,j_1] = \tilde{M}[k,j_2]$ whenever $||B_{j_1} \cap B_1|| = ||B_{j_2} \cap B_1||$,
2. $\tilde{M}[k,j] \neq 0$ for all $j$ with $||B_j \cap B_1|| = p - k$,
3. $\tilde{M}[k,j] = 0$ for all $j$ with $||B_j \cap B_1|| > p - k$.

Proof To see (1), note that for fixed $k$, the cardinality of the set $\{A_i \in A \mid ||A_i \cap B_j|| = p - k \land A_i \supseteq B_j\}$ depends only on the number of elements of $B_j \in B$ that are also in $B_1$. Hence for every $j_1$ and $j_2$, with $||B_{j_1} \cap B_1|| = ||B_{j_2} \cap B_1||$, it holds that

$$||\{A_i \in A \mid ||A_i \cap B_1|| = p - k \land A_i \supseteq B_j\}|| = ||\{A_i \in A \mid ||A_i \cap B_1|| = p - k \land A_i \supseteq B_{j_2}\}||.$$

Statement (1) follows immediately. For the proof of (2), we have to verify that for every $k$ and $j$,

$$S = \{A_i \in A \mid ||A_i \cap B_1|| = p - k \land A_i \supseteq B_j\} \neq \emptyset \text{ if } ||B_j \cap B_1|| = p - k.$$

It is easy to see that, if $||B_j \cap B_1|| = p - k$ then $S$ has as element any set $B_j \cup \{g\}$, where $g \notin B_1$. Finally, to show (3), note that for every $k$ and $j$,

$$||\{A_i \in A \mid ||A_i \cap B_1|| = p - k \land A_i \supseteq B_j\}|| = 0 \text{ if } ||B_j \cap B_1|| \geq p - k + 1,$$

since $||B_j \cap B_1|| \geq p - k + 1$ and $A_i \supseteq B_j$ implies that $||A_i \cap B_1|| \geq p - k + 1.$ (Claim 8)

To complete the proof of the lemma, we show that the structure of the matrix $\tilde{M}$ stated in Claim 8 implies that $e_1$ can be expressed as a linear combination of row vectors of $\tilde{M}$, and hence also as linear combination of row vectors of $M$. We construct a matrix $M'_{p \times n}$ from $\tilde{M}_{p \times n}$, which will turn out to have full rank. From Claim 8(1), we know that column $j_1$ and column $j_2$ of $M$ are equal whenever $||B_{j_1} \cap B_1|| = ||B_{j_2} \cap B_1||$. Thus it makes sense to define a matrix $M'$ eliminating all these
duplicate columns from $\widetilde{M}$. We define column $\ell$, where $1 \leq \ell \leq p$, of $M'$ to equal column $j$ of $\widetilde{M}$ for some $j$ with $|B_j \cap B_1| = p - \ell$. Note that column 1 of $M'$ corresponds uniquely to column 1 of $\widetilde{M}$. Claim 8(3) implies that the matrix $M'$ is an upper triangular matrix, and from Claim 8(2), it follows that all diagonal elements in $M'$ are $\neq 0$. Hence $M'$ has full rank. In particular, row vector $[1, 0, \ldots, 0]_{1 \times p}$ can be written as a linear combination of rows in $M'$. Suppose
\[ [1, 0, \ldots, 0]_{1 \times p} = \sum_{k=1}^{p} c_k \cdot \text{Row}(M', k). \]
for $c_1, \ldots, c_k \in \mathbb{Q}$. Then
\[ e_1 = [1, 0, \ldots, 0]_{1 \times n} = \sum_{k=1}^{p} c_k \cdot \text{Row}(\widetilde{M}, k), \]
because column 1 of $\widetilde{M}$ equals column 1 of $M'$, and all other columns of $\widetilde{M}$ equal a column $j$ in $M'$ with $p \geq j > 1$. \hfill (Lemma 5.10)

**Proof of Lemma 5.11** We transform $s(y_1, y_2, \ldots, y_N)$ to a multilinear polynomial $s'(y_1, y_2, \ldots, y_N)$ with the following properties:

(1) All monomials in $s'$ have exactly $p - 1$ different variables.

(2) $s'(y_1, y_2, \ldots, y_N) = s(y_1, y_2, \ldots, y_N) = \text{val} \in \mathbb{Z}$, for all $y_1, y_2, \ldots, y_N \in \{0, 1\}$ with $\sum_{i=1}^{N} y_i = p$.

(3) The coefficients of the monomials in $s'$ have the form $a/b$, where $a, b \in \mathbb{Z}$, and $p \nmid b$.

Since $s(0, 0, \ldots, 0) = 0$, the polynomial $s$ has no monomial with degree 0. Let $t(y_1, y_2, \ldots, y_N) = a \prod_{i \in A} y_i$ be a monomial, where $A \subseteq \{1, 2, \ldots, N\}$ and $1 \leq |A| = \ell < p - 1$. Define the multilinear polynomial $u_{\ell}$ by
\[ u_{\ell}(y_1, y_2, \ldots, y_N) = \sum_{B \subseteq \{1, \ldots, N\} \setminus A, |B| = \ell} \left( \frac{a}{p - \ell} \prod_{i \in B} y_i \right). \]

**Claim 9** For all $y_1, y_2, \ldots, y_N \in \{0, 1\}$ with $\sum_{i=1}^{N} y_i = p$, $u_{\ell}(y_1, y_2, \ldots, y_N) = t(y_1, y_2, \ldots, y_N)$.

**Proof** Let $y_1, y_2, \ldots, y_N \in \{0, 1\}$ be such that $\sum_{i=1}^{N} y_i = p$. Depending on the choice in the selection of $y_1, y_2, \ldots, y_N$, we have two cases.

**Case 1:** $t(y_1, y_2, \ldots, y_N) = 0$

Then, clearly $u_{\ell}(y_1, y_2, \ldots, y_N) = t(y_1, y_2, \ldots, y_N) = 0$.

**Case 2:** $t(y_1, y_2, \ldots, y_N) \neq 0$

Then $y_i = 1$ for all $i \in A$. Let $D = \{ i \mid i \in \{1, 2, \ldots, N\} - A \land y_i = 1 \}$. Clearly, $|D| = p - \ell$. In the sum on the right hand side of Eq. (5.1), only $B$’s with $B \subseteq D$ contribute a value $\neq 0$.

The sets $B$ have always cardinality $p - 1 - \ell$. Hence, there are exactly $(p - \ell) \choose (p - 1 - \ell)$ sets $B$ contributing the value $a/(p - \ell)$ to the sum. \hfill (Claim 9)

Transform polynomial $s(y_1, y_2, \ldots, y_N)$ to polynomial $s'(y_1, y_2, \ldots, y_N)$ by substituting each monomial $t(y_1, y_2, \ldots, y_N)$ in $s(y_1, y_2, \ldots, y_N)$ of degree $< p - 1$ by the corresponding polynomial $u_{\ell}(y_1, y_2, \ldots, y_N)$. Since $\{a/b \mid a, b \in \mathbb{Z} \land b \neq 0 \land p \nmid b\}$ is closed under addition, it follows that the polynomial $s'(y_1, y_2, \ldots, y_N)$ satisfies properties (1), (2), and (3) stated at the beginning of the
proof. Lemma 5.10 implies that all the coefficients of monomials in \( s'(y_1, y_2, \ldots, y_N) \) are equal to \( \text{val}/p \). Thus to match with property (3) of polynomial \( s'(y_1, y_2, \ldots, y_N) \), \( \text{val}/p \) must be an integer. It follows that \( p \mid \text{val} \).

As an immediate corollary of Theorem 5.9 and the fact that LWPP is closed under polynomial-time Turing reductions in all relativized worlds [FFK94], we get the following result.

**Corollary 5.12** (\( \exists A [\text{WPP}^A \not\subseteq \text{LWPP}^A] \)).

## 6 Conclusion and Open Problems

In this paper, we used classical complexity classes to study the complexity of quantum complexity classes EQP, BQP, and NQP. In particular, we used counting classes to prove relativized separations, strong separations, and conditional collapses. We mention several open problems.

Theorem 3.1 shows that, in some relativized world, ZPP is not contained in WPP. Theorem 4.9 shows that, relative to an oracle, RP contains a set that is immune to \( C_m \). It would be interesting if the oracle separation of ZPP from WPP can be strengthened to an immunity separation result. This will also imply, as a corollary, an immunity separation of ZPP from EQP.

Another interesting open issue is whether or not the conditional collapse results in this paper can be improved. We show in Corollary 5.7 that \( \text{NQP} \subseteq \text{EQP} \Rightarrow \text{PH} \subseteq \text{EQP} \). This is an analog of the collapse known in the classical case, namely \( \text{NP} \subseteq \text{P} \Rightarrow \text{PH} \subseteq \text{P} \). Can we similarly prove that \( \text{NQP} \subseteq \text{BQP} \Rightarrow \text{PH} \subseteq \text{BQP} \)? This would be another analog of the collapse known in the classical case [Zac88].

**Acknowledgment** We thank Lane Hemaspaandra for helpful advice and guidance throughout the project.

## References


