The Robustness of LWPP and WPP, with an Application to Graph Reconstruction

Edith Hemaspaandra
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623, USA

Lane A. Hemaspaandra*
Department of Computer Science
University of Rochester
Rochester, NY 14627, USA

Holger Spakowski†
Department of Mathematics and Applied Mathematics
University of Cape Town
Rondebosch 7701, South Africa

Osamu Watanabe
Dept. of Mathematical and Computing Sciences
Tokyo Institute of Technology
Tokyo 152-8552, Japan

November 3, 2017; revised July 6, 2018

Abstract

We show that the counting class LWPP [FFK94] remains unchanged even if one allows a polynomial number of gap values rather than one. On the other hand, we show that it is impossible to improve this from polynomially many gap values to a superpolynomial number of gap values by relativizable proof techniques.

The first of these results implies that the Legitimate Deck Problem (from the study of graph reconstruction) is in LWPP (and thus low for PP, i.e., $\mathrm{PP}^{\text{Legitimate Deck}} = \mathrm{PP}$) if the weakened version of the Reconstruction Conjecture holds in which the number of nonisomorphic preimages is assumed merely to be polynomially bounded. This strengthens the 1992 result of Köbler, Schöning, and Torán [KST92] that the Legitimate Deck Problem is in LWPP if the Reconstruction Conjecture holds, and provides strengthened evidence that the Legitimate Deck Problem is not NP-hard.

We additionally show on the one hand that our main LWPP robustness result also holds for WPP, and also holds even when one allows both the rejection- and acceptance- gap-value targets to simultaneously be polynomial-sized lists; yet on the other hand, we show that for the #P-based analog of LWPP the behavior much differs in that, in some relativized worlds, even two target values already yield a richer class than one value does. Despite that nonrobustness result for a #P-based class, we show that the #P-based "exact counting" class $\mathrm{C}_{=}\mathrm{P}$ remains unchanged even if one allows a polynomial number of target values for the number of accepting paths of the machine.

*This work was done in part while on a sabbatical stay at ETH Zürich's Department of Computer Science, generously supported by that department.

†This work was done in part while visiting the University of Rochester.

1 Introduction

Nothing is more natural than wanting to better understand an object by knowing what it can and cannot do. Whether wondering how fast a (rental?) car can go in reverse or wondering if $\mathrm{NP}^{\mathrm{NP}}$ can without loss of generality be assumed to ask at most one question per nondeterministic path (as it indeed can, as is implicit in the quantifier characterization [SM73, Wra77] of $\mathrm{NP}^{\mathrm{NP}}$), we both in life and as theoreticians want to find how robust things are.

We are often particularly happy when a class proves to be quite robust under definitional perturbations. Such robustness on one hand suggests that perhaps there is something broadly natural about the class, and on the other hand such robustness often makes it easier to put the class to use.

This paper shows that the counting classes LWPP and WPP, defined in 1994 in the seminal work of Fenner, Fortnow, and Kurtz [FFK94] on gap-based counting classes, are quite robust. Even though their definitions are in terms of having the gap function (the difference between the number of accepting and rejecting paths of a machine) hit a single target value, we prove (in Section 3) that one can allow a list of up to polynomially many target values without altering the descriptive richness of the class, i.e., without changing the class.

We then apply this to the question of whether the Legitimate Deck Problem is in LWPP.

The Legitimate Deck Problem (a formal definition will be given in Section 4) is the decision problem of determining whether, given a multiset of (unlabeled) graphs, there exists a graph G such that that multiset is precisely (give or take isomorphisms) the multiset of one-node-deleted subgraphs of G (a.k.a. the *deck* of G) [KH94]. *The Reconstruction Conjecture* [Kel42, Ula60]—which in the wake of the resolution of the Four-Color Conjecture was declared by the editorial board of the Journal of Graph Theory to be the foremost open problem in graph theory [Edi77]—states that every graph with three or more nodes is uniquely determined (give or take isomorphisms) by its multiset of one-node-deleted subgraphs. The Legitimate Deck Problem was defined in 1978 by Nash-Williams [NW78], in his paper that framed the algorithmic/complexity issues related to reconstructing graphs—such as telling whether a given deck is legitimate (i.e., is the deck of some graph).

Our application of our LWPP robustness result to the question of whether the Legitimate Deck Problem is in LWPP is the following. The strongest previous evidence of the simplicity of the Legitimate Deck Problem is the 1992 result of Köbler, Schöning, and Torán [KST92] that the Legitimate Deck Problem is in LWPP (and thus is PP-low, i.e., $\mathrm{PP}^{\text{Legitimate Deck}} = \mathrm{PP}$) if the Reconstruction Conjecture (i.e., that each deck whose elements all have at least two nodes has at most *one* preimage, give or take isomorphisms) holds. Using this paper's main robustness result as a tool, Section 4 proves that the Legitimate Deck Problem is in LWPP (and thus is PP-low) if a weakened version of the Reconstruction Conjecture holds, namely, that each deck has at most *a polynomial number* of nonisomorphic preimages.

This weakened version is not known to be equivalent to the Reconstruction Conjecture itself. And so our result for the first time gives a path to proving that the Legitimate Deck Problem is PP-low that does not require one to, on the way, resolve the foremost open problem in graph theory [Edi77].

We started this section by noting that it is natural to want to know both flexibilities and limitations of classes. Our main result is about flexibility: going from one target gap to instead a polynomial number. But are we leaving money on the table? Could we extend our result to slightly superpolynomial numbers of target gaps, or even to exponential numbers of target gaps?

In Section 5 we note that if the robustness of LWPP were to hold up to exponentially many target gaps, then NP would be in LWPP and so would be PP-low (i.e., $\mathrm{PP}^{\mathrm{NP}} = \mathrm{NP}$); yet NP is widely suspected not to be PP-low. We also, by encoding nondeterministic oracle Turing machines by low-degree multivariate polynomials so as to capture the gap functions of those machines, show an oracle relative to which robustness fails for all superpolynomial numbers of target gaps; thus, no extension beyond this paper's polynomial-number-of-target-gaps robustness result for LWPP can be proven by a relativizable proof. And for the #P-based analogue of LWPP, in Section 6 we show that even allowing two target values yields, in some relativized worlds, a richer class than one target value. For the important #P-based counting class $\mathrm{C_{=}P}$, however, we prove in Section 7 that the class does not change even if one allows not just one but instead a polynomial number of target values for the number of accepting paths of the underlying #P function. We also (in the final part of Section 3) extend our main result to show that the simultaneous expansion to polynomial-sized lists of both the acceptance *and rejection* target-gap lists still, for LWPP and WPP, yields the classes LWPP and WPP.

To summarize: In this paper, we prove that LWPP and WPP are robust enough that they remain unchanged when their single target gap is allowed to be expanded to a polynomial-sized list; we apply this new robustness of LWPP to show that the PP-lowness of the Legitimate Deck Problem follows from a weaker hypothesis than was previously known; we show that our polynomial robustness of LWPP is optimal with respect to relativizable proofs; and we prove a number of related results on limitations and extensions.

2 Preliminaries

We first present the definitions of many of the counting classes that we will be speaking of, taking the definitions from the seminal paper of Fenner, Fortnow, and Kurtz [FFK94].

Definition 2.1 ([FFK94]). *1. For each nondeterministic polynomial-time Turing machine N, the function $\mathrm{acc}_N : \Sigma^* \to \mathbb{N}$ is defined such that for every $x \in \Sigma^*$, $\mathrm{acc}_N(x)$ equals the number of accepting computation paths of N on input x.*

2. For each nondeterministic polynomial-time Turing machine N, the function $\mathrm{rej}_N : \Sigma^* \to \mathbb{N}$ is defined such that for every $x \in \Sigma^*$, $\mathrm{rej}_N(x)$ equals the number of rejecting computation paths of N on input x.

3. For each nondeterministic polynomial-time Turing machine N, the function $\mathrm{gap}_N : \Sigma^* \to \mathbb{Z}$ is defined such that for every $x \in \Sigma^*$,

$$\mathrm{gap}_N(x) = \mathrm{acc}_N(x) - \mathrm{rej}_N(x).$$

Definition 2.2 ([FFK94]).

$$\mathrm{GapP} = \{\mathrm{gap}_N \mid N \text{ is a polynomial-time nondeterministic Turing machine}\}.$$

Definition 2.3 ([OH93, FFK94]). SPP *is the class of all sets A such that there exists a* GapP *function g such that for all $x \in \Sigma^*$,*

$$x \in A \Longrightarrow g(x) = 1$$
$$x \notin A \Longrightarrow g(x) = 0.$$

The following class, WPP, is potentially larger than SPP. Instead of the "target value" 1 for the case $x \in A$, we allow a target value $f(x)$, where f may be any polynomial-time computable function whose image does not contain 0. FP denotes the class of polynomial-time computable functions.

Definition 2.4 ([FFK94])**.** WPP *is the class of all sets A such that there exists a* GapP *function g and a function $f \in$ FP that maps from Σ^* to $\mathbb{Z} - \{0\}$ such that for all $x \in \Sigma^*$,*

$$x \in A \Longrightarrow g(x) = f(x)$$
$$x \notin A \Longrightarrow g(x) = 0.$$

The class LWPP is the same as WPP except that the "target function" f may depend on only the *length* of the input.

Definition 2.5 ([FFK94])**.** LWPP *is the class of all sets A such that there exists a* GapP *function g and a function $f \in$ FP that maps from 0^* to $\mathbb{Z} - \{0\}$ such that for all $x \in \Sigma^*$,*

$$x \in A \Longrightarrow g(x) = f(0^{|x|})$$
$$x \notin A \Longrightarrow g(x) = 0.$$

Here, as usual, for any $x \in \Sigma^*$, $|x|$ denotes the length of x, and for any $n \in \mathbb{N}$, 0^n is the string consisting of exactly n zeroes.

We now generalize the definition of LWPP to the case of having the target of the GapP function be, for members of the set, not a single value but a collection of values.

One might expect us to formalize this by simply having the polynomial-time computable "what is the target" function output a *list* of the nonzero-integer targets. That would work fine and be equivalent to what we are about to do, as long as we are dealing with lists having at most a polynomial number of elements. However, to be able to speak of even longer lists—as will be important in our negative results offsetting our main result—we use an indexing approach, as follows.

Definition 2.6. *Let r be any function mapping from $\mathbb{N}$ to $\mathbb{N}$. Then the class r-*LWPP *is the class of all sets A such that there exists a* GapP *function g and a function $f \in$ FP that maps to $\mathbb{Z} - \{0\}$ such that for each $x \in \Sigma^*$,*

$$x \in A \Longrightarrow \text{ there exists } i \in \{1, 2, \ldots, r(|x|)\} \text{ such that } g(x) = f(\langle 0^{|x|}, i \rangle)$$
$$x \notin A \Longrightarrow g(x) = 0.$$

The following class, *Poly*-LWPP, will be central to this paper: Our main result is that this class in fact equals LWPP. The "$+\,c$" in Definition 2.7 may seem strange at first. But without it we would have a boundary-case pathology at $n = 0$, namely, the class could not contain any set that contains the empty string.[1]

[1]The "$+\,c$" in Definition 2.7 also, on its surface, would seem to make a difference at length 1, by allowing lists of size greater than one; however, one could work around that issue. In contrast, the exclusion of the empty string would not be avoidable if our class of polynomials were to be a class—such as n^c—such that all of its members evaluate to 0 at $n = 0$. In any case, our use of $n^c + c$ avoids any special worries at lengths 0 and 1. And since for every polynomial p there is a c such that $(\forall n \in \mathbb{N})[p(n) \leq n^c + c]$, using polynomials just of the form $n^c + c$ is in fact not a restriction.

Definition 2.7.

$$\textit{Poly}\text{-LWPP} = \bigcup_{c \in \mathbb{N}^+} (n^c + c)\text{-LWPP}.$$

It is easy to see that 1-LWPP = LWPP, and that, of course, more flexibility as to targets never removes sets from the class, i.e., speaking loosely for the moment as to notation (and the log case will not be defined or used again in this paper, but it is clear from context here what we mean by it; the exponential case will be defined only in Section 5), 1-LWPP $\subseteq$ 2-LWPP $\subseteq$ 3-LWPP $\subseteq \cdots \subseteq$ *Log*-LWPP $\subseteq$ *Poly*-LWPP $\subseteq$ *Exp*-LWPP. As mentioned above, in this paper we will prove that the first five of these "$\subseteq$"s are in fact all equalities. We will also prove that the sixth "$\subseteq$" cannot be an equality unless NP is PP-low.

We now show that for every function r, r-LWPP is contained in the co-class of the well-known counting class $\mathrm{C_{=}P}$.

Definition 2.8 ([Sim75, Wag86])**.** $\mathrm{C_{=}P}$ *is the class of all sets A such that there is a nondeterministic polynomial-time Turing machine N and a function $f \in \mathrm{FP}$ such that for each $x \in \Sigma^*$,*

$$x \in A \iff \mathrm{acc}_N(x) = f(x).$$

More convenient for us is the following characterization of $\mathrm{C_{=}P}$ using GapP functions.

Theorem 2.9 ([FFK94])**.** *For each $A \subseteq \Sigma^*$, $A \in \mathrm{C_{=}P}$ if and only if there exists a function $g \in \mathrm{GapP}$ such that for all $x \in \Sigma^*$,*

$$x \in A \iff g(x) = 0.$$

We thus certainly have the following, which holds simply by taking the GapP function g required in Theorem 2.9 to be the same as the function g in Definition 2.6.

Theorem 2.10. *For each function $r : \mathbb{N} \to \mathbb{N}$, r-LWPP $\subseteq \mathrm{coC_{=}P}$.*

3 Main Result: LWPP Stays the Same If for Accepted Inputs We Allow Polynomially Many Gap Values Instead of One

We now state our main result: LWPP altered to allow even a polynomial number of target gap values is still LWPP (i.e., with just one target gap value).

Theorem 3.1. *Poly*-LWPP = LWPP.

For the proof, we need the following closure properties shown by Fenner, Fortnow, and Kurtz [FFK94].[2] For function classes $\mathcal{F}_1$ and $\mathcal{F}_2$, $\mathcal{F}_1 \circ \mathcal{F}_2 = \{f_1 \circ f_2 \mid f_1 \in \mathcal{F}_1 \wedge f_2 \in \mathcal{F}_2\}$, where $\circ$ denotes composition.

Closure Property 3.2 ([FFK94])**.** $\mathrm{GapP} \circ \mathrm{FP} = \mathrm{GapP}$ *and* $\mathrm{FP} \subseteq \mathrm{GapP}$.

[2] We mention in passing that, regarding Closure Property 3.4, if the polynomial q were allowed to have coefficients that are uncomputable—or that are extremely expensive to compute prefixes of the values of—real numbers, that claimed closure property might fail; we here are, as is typical in such settings, tacitly assuming that the polynomials have rational coefficients.

Closure Property 3.3 ([FFK94]). *If* $g \in$ GapP *then* $-g \in$ GapP.

Closure Property 3.4 ([FFK94]). *If* $g \in$ GapP *and* q *is a polynomial, then the function*

$$h(x) = \prod_{0 \le i \le q(|x|)} g(\langle x, i \rangle)$$

is in GapP.

Closure Property 3.5 ([FFK94]). GapP *is closed under addition, subtraction, and multiplication.*

Proof of Theorem 3.1. As mentioned previously, it is easy to see that LWPP $\subseteq$ *Poly*-LWPP.

To show *Poly*-LWPP $\subseteq$ LWPP, let A be a set in *Poly*-LWPP defined by $g \in$ GapP, $f \in$ FP, and polynomial $r(n) = n^c + c$ according to Definitions 2.6 and 2.7.

Let h_1 be a function such that for all $x \in \Sigma^*$ and $i \in \mathbb{N}^+$,

$$h_1(\langle x, i \rangle) = f(\langle 0^{|x|}, i \rangle) - g(x).$$

We have $h_1 \in$ GapP since $f \in$ FP $\subseteq$ GapP, $g \in$ GapP, and GapP is closed under subtraction [FFK94]. We define h_2 such that for all $x \in \Sigma^*$,

$$h_2(x) = \prod_{1 \le i \le r(|x|)} h_1(\langle x, i \rangle).$$

By Closure Property 3.4, $h_2 \in$ GapP. Note that for all $x \in \Sigma^*$,

$$h_2(x) = \prod_{1 \le i \le r(|x|)} \left(f(\langle 0^{|x|}, i \rangle) - g(x) \right).$$

It follows that for every $x \in \Sigma^*$,

$$h_2(x) = \begin{cases} 0 & \text{if there exists } i \in \{1, 2, \ldots, r(|x|)\} \text{ such that } g(x) = f(\langle 0^{|x|}, i \rangle) \\ \prod_{1 \le i \le r(|x|)} f(\langle 0^{|x|}, i \rangle) & \text{if } g(x) = 0. \end{cases} \tag{1}$$

Now we define the function $\widehat{g}$ such that for all $x \in \Sigma^*$,

$$\widehat{g}(x) = h_2(x) - \prod_{1 \le i \le r(|x|)} f(\langle 0^{|x|}, i \rangle).$$

Using the closure properties, it is easy to see that $\widehat{g} \in$ GapP.

Note that by Eqn. (1), we have that for all $x \in \Sigma^*$,

$$\widehat{g}(x) = \begin{cases} -\prod_{1 \le i \le r(|x|)} f(\langle 0^{|x|}, i \rangle) & \text{if there exists } i \in \{1, 2, \ldots, r(|x|)\} \text{ such that } g(x) = f(\langle 0^{|x|}, i \rangle) \\ 0 & \text{if } g(x) = 0. \end{cases} \tag{2}$$

Let $\widehat{f}$ be a function such that for all $\ell \in \mathbb{N}$,

$$\widehat{f}(0^\ell) = - \prod_{1 \le i \le r(\ell)} f(\langle 0^\ell, i \rangle).$$

It is easy to see that $\widehat{f} \in \text{FP}$. Keeping in mind Eqn. (2), it follows from the above that for every $x \in \Sigma^*$,

$$\widehat{g}(x) = \begin{cases} \widehat{f}(0^{|x|}) & \text{if } x \in A \\ 0 & \text{otherwise.} \end{cases}$$

Since $\widehat{g} \in \text{GapP}$ and $\widehat{f} \in \text{FP}$, this implies that $A \in \text{LWPP}$. □

A theorem analogous to Theorem 3.1 also holds for the corresponding class that allows the target values to depend on the actual input instead on only the length of the input.

Definition 3.6.

$$\textit{Poly}\text{-WPP} = \bigcup_{c \in \mathbb{N}^+} (n^c + c)\text{-WPP}.$$

Theorem 3.7. *Poly*-WPP = WPP.

The proof is almost exactly the same as the proof of Theorem 3.1, simply taking into account the fact that for WPP the "gap" function can vary even among inputs of the same length.

We remark that the robustness results stated in Theorems 3.1 and 3.7 do not seem to in any obvious way follow as corollaries to the closure of LWPP under polynomial-time Turing reductions [FFK94] or of WPP under polynomial-time truth-table reductions [STT05].

Let us now see whether we can extend Theorems 3.1 and 3.7. Suppose we not only allow the target-gap set for acceptance to have polynomially many values, but in addition allow the target-gap set for rejection to have polynomially many values. (In contrast, both LWPP and r-LWPP allow rejection only when the gap's value is 0). Is LWPP so robust that even *that* class is no larger than LWPP? We will now build on our main result to give the answer "yes" to that question, thus extending our main result to this more symmetric case for LWPP and for WPP.

To this end, let us define (r_A, r_R)-LWPP as follows. (One of course can in the clear, analogous way define a similarly loosened version of WPP, (r_A, r_R)-WPP.)

Definition 3.8. *Let r_A and r_R be any functions mapping from $\mathbb{N}$ to $\mathbb{N}$. Then (r_A, r_R)-*LWPP *is the class of all sets B such that there exists a* GapP *function g and functions—each mapping to $\mathbb{Z}$—$f_A \in \text{FP}$ and $f_R \in \text{FP}$ such that both of the following hold:*

1. *For each $j \in \mathbb{N}$, $A_j \cap R_j = \emptyset$, where*

$$A_j = \{n \mid (\exists i \in \{1, 2, \ldots, r_A(j)\})[f_A(\langle 0^j, i\rangle) = n]\}$$

and

$$R_j = \{n \mid (\exists i \in \{1, 2, \ldots, r_R(j)\})[f_R(\langle 0^j, i\rangle) = n]\}.$$

2. *For each $x \in \Sigma^*$,*

$$x \in B \Longrightarrow g(x) \in A_{|x|}$$
$$x \notin B \Longrightarrow g(x) \in R_{|x|}.$$

It is not hard to see, via shifting gaps with dummy paths, that the class $(r, 1)$-LWPP equals r-LWPP. However, what can be shown more generally about the classes (r_A, r_R)-LWPP? For example, for which functions $r_{1,A}$, $r_{1,R}$, $r_{2,A}$, and $r_{2,A}$ does it hold that $(r_{1,A}, r_{1,R})$-LWPP $\subseteq$

$(r_{2,A}, r_{2,R})$-LWPP? (There are some literature notions that, in different ways, have at least a somewhat similar flavor to our notion, namely, defining classes by being more flexible regarding acceptance types. In particular, the counting classes CP_S of Cai et al. [CGH$^+$89] and the "C-class" framework of Bovet, Crescenzi, and Silvestri [BCS92] have such a flavor. But in contrast with those, our classes here are ones whose definitions are centered on the notion of gaps.) We do not here undertake that general study, but instead resolve what seems the most compelling question, namely, we prove that $(Poly, Poly)$-LWPP = LWPP.

Definition 3.9. $(Poly, Poly)$-LWPP $= \bigcup_{c \in \mathbb{N}^+} (n^c + c, n^c + c)$-LWPP.

Theorem 3.10. $(Poly, Poly)$-LWPP = $Poly$-LWPP.

The proof of Theorem 3.10 can be found in the appendix. Together with Theorem 3.1, we get the following corollary.

Corollary 3.11. $(Poly, Poly)$-LWPP = LWPP.

An analogous statement can also be shown for the analogously defined class $(Poly, Poly)$-WPP:

Theorem 3.12. $(Poly, Poly)$-WPP = WPP.

4 Applying the Main Result to Graph Reconstruction

Definition 4.1. *Let $\langle G_1, G_2, \ldots, G_n \rangle$ be a sequence of graphs and $G = (V, E)$ a graph with $V = \{1, 2, \ldots, n\}$. Suppose that there is a permutation $\pi \in S_n$ such that for each $k \in \{1, 2, \ldots, n\}$, the graph $G_{\pi(k)}$ is isomorphic to the graph $(V - \{k\}, E - \{\{k, \ell\} : \ell \in V\})$ obtained by deleting vertex k from G. Then $\langle G_1, G_2, \ldots, G_n \rangle$ is called a* deck *of G and G is called a* preimage *of the sequence $\langle G_1, G_2, \ldots, G_n \rangle$.*

A sequence of graphs $\langle G_1, G_2, \ldots, G_n \rangle$ is called a legitimate deck *if there exists a graph G that is a preimage of $\langle G_1, G_2, \ldots, G_n \rangle$.*

The *Reconstruction Conjecture* (see, e.g., the surveys [BH77, Man88, Bon91] and the book [LS03]) says that each legitimate deck consisting of graphs with at least two vertices has exactly one preimage up to isomorphism. This conjecture is a very prominent conjecture in graph theory—as mentioned in Section 1 it is perhaps the most important conjecture in that area—and has been studied for many decades.

Nash-Williams [NW78], Mansfield [Man82], Kratsch and Hemachandra [KH94], and Hemaspaandra et al. [HHRT07] introduced various decision problems related to the Reconstruction Conjecture, as part of a stream of work studying the algorithmic and complexity issues of reconstruction. We here are interested mainly in the Legitimate Deck Problem, which is defined as the following decision problem.

Legitimate Deck (a.k.a. the Legitimate Deck Problem) [Man82]: Given a sequence of graphs $\langle G_1, G_2, \ldots, G_n \rangle$, is $\langle G_1, G_2, \ldots, G_n \rangle$ a legitimate deck?

Mansfield [Man82] showed that the Graph Isomorphism Problem, GI (given two graphs G_1 and G_2, are they isomorphic?), is polynomial-time many-one reducible to the Legitimate Deck Problem. However, to this day it remains open whether there is a polynomial-time many-one reduction from the Legitimate Deck Problem to GI.

So how hard is the Legitimate Deck Problem? It is easy to see that the Legitimate Deck Problem is in NP. In the following, we will see that there is some evidence that the Legitimate Deck Problem is not NP-hard, and we will improve that evidence.

Let us define the following function problem.

Preimage Counting [KH91, KH94]: Given a sequence of graphs $\langle G_1, G_2, \ldots, G_n \rangle$, compute the number $\mathrm{PCount}(\langle G_1, G_2, \ldots, G_n \rangle)$ of all nonisomorphic preimages for the sequence $\langle G_1, G_2, \ldots, G_n \rangle$.

Köbler, Schöning, and Torán [KST92] showed the following theorem.

Theorem 4.2 ([KST92])**.** *There is a function $h : 0^* \to \mathbb{N} - \{0\}$ in* FP *such that the function that maps every sequence $\langle G_1, G_2, \ldots, G_n \rangle$ to $h(0^n)$ times the number of nonisomorphic preimages, i.e.,*

$$\langle G_1, G_2, \ldots, G_n \rangle \mapsto h(0^n) \cdot \mathrm{PCount}(\langle G_1, G_2, \ldots, G_n \rangle)$$

is in GapP.

Theorem 4.2 has the following corollary.

Corollary 4.3 ([KST92])**.** *If the Reconstruction Conjecture holds, then the Legitimate Deck Problem is in* LWPP*.*

Since all LWPP sets are PP-low, that immediately gives some evidence that the Legitimate Deck Problem is not NP-hard.

Corollary 4.4. *If the Reconstruction Conjecture holds, then the Legitimate Deck Problem is not* NP*-hard (or even* NP*-Turing-hard) unless* NP *is* PP*-low.*

Unfortunately, we do not know whether the Reconstruction Conjecture holds. However, perhaps we can prove the membership of the Legitimate Deck Problem in LWPP under a *weaker assumption* than the Reconstruction Conjecture, for instance, what if the number of nonisomorphic preimages is not always 0 or 1 (as holds under the Reconstruction Conjecture), but rather is merely relatively small, e.g., some constant or some polynomial in the number of vertices (as must hold for each graph class having bounded minimum degree)? We will now use the results of the previous section to prove that this indeed is the case.

Conjecture 4.5 (q-Reconstruction Conjecture)**.** *For each legitimate deck there exist at most $q(n)$ nonisomorphic preimages, where n is the number of graphs in the deck.*

For any function r, we now define a complexity class r-$\widehat{\mathrm{LWPP}}$. This class may not seem very natural, but we will see that it is very-well suited to helping us classify the problem Legitimate Deck. In some sense, it is a tool that we will use in our proof, and then will discard by noting that it in fact turns out to be a disguised version of r-LWPP.

Definition 4.6. *Let r be any function mapping from $\mathbb{N}$ to $\mathbb{N}$. Then the class r-$\widehat{\mathrm{LWPP}}$ is the class of all sets A such that there exists a* GapP *function g, and a function $f \in$ FP that maps from 0^* to $\mathbb{Z} - \{0\}$, such that for each $x \in \Sigma^*$,*

$$x \in A \Longrightarrow \text{there exists } i \in \{1, 2, \ldots, r(|x|)\} \text{ such that } g(x) = i \cdot f(0^{|x|})$$
$$x \notin A \Longrightarrow g(x) = 0.$$

Theorem 4.7. *Let q be any nondecreasing function from $\mathbb{N}$ to $\mathbb{N}$. Then the following holds.*
If the q-Reconstruction Conjecture holds, then the Legitimate Deck Problem is in q-$\widehat{\text{LWPP}}$.

Proof. Suppose that the q-Reconstruction Conjecture holds. Let $\langle G_1, G_2, \ldots, G_n\rangle$ be an input to the Legitimate Deck Problem. By our assumption, we have that

$$\langle G_1, G_2, \ldots, G_n\rangle \in \text{Legitimate Deck} \Longrightarrow \text{PCount}(\langle G_1, G_2, \ldots, G_n\rangle) \in \{1, 2, \ldots, q(n)\}, \text{and}$$
$$\langle G_1, G_2, \ldots, G_n\rangle \notin \text{Legitimate Deck} \Longrightarrow \text{PCount}(\langle G_1, G_2, \ldots, G_n\rangle) = 0.$$

Let $h \in \text{FP}$ be the function discussed in Theorem 4.2. Then the function g defined such that for every sequence $\langle G_1, G_2, \ldots, G_n\rangle$,

$$g(\langle G_1, G_2, \ldots, G_n\rangle) = h(0^n) \cdot \text{PCount}(\langle G_1, G_2, \ldots, G_n\rangle)$$

is in GapP. It follows that

$$\langle G_1, G_2, \ldots, G_n\rangle \in \text{Legitimate Deck} \Longrightarrow g(\langle G_1, G_2, \ldots, G_n\rangle) = h(0^n) \cdot i \text{ for some } i \in \{1, 2, \ldots, q(n)\}, \text{ and}$$
$$\langle G_1, G_2, \ldots, G_n\rangle \notin \text{Legitimate Deck} \Longrightarrow g(\langle G_1, G_2, \ldots, G_n\rangle) = 0.$$

This almost directly implies that Legitimate Deck $\in q$-$\widehat{\text{LWPP}}$. However, note that to satisfy Definition 4.6, function h must be a function *that depends only on the length of the input* $\langle G_1, G_2, \ldots, G_n\rangle$. The problem here is that (depending on how exactly we decide to encode the graphs $G_1, G_2, \ldots, G_n$ in the input $\langle G_1, G_2, \ldots, G_n\rangle$) even the value n (the number of graphs in the deck) may depend on the actual input $\langle G_1, G_2, \ldots, G_n\rangle$ and not only on the length of the input $\langle G_1, G_2, \ldots, G_n\rangle$.

Fortunately, there is a way to get around this problem. From the length of the input, we get at least an upper bound for n since we can certainly assume that

$$n \leq |\langle G_1, G_2, \ldots, G_n\rangle|.$$

Define a function $\widehat{h}$ such that for each $m \in \mathbb{N}$, $\widehat{h}(0^m)$ is the product of all "h-values" *up to length* m. That is, define function $\widehat{h}$ such that for all $m \in \mathbb{N}$,

$$\widehat{h}(0^m) = \prod_{0 \leq i \leq m} h(0^i).$$

Define function h' such that for all inputs $\langle G_1, G_2, \ldots, G_n\rangle$,

$$h'(\langle G_1, G_2, \ldots, G_n\rangle) = \prod_{0 \leq i \leq |\langle G_1, G_2, \ldots, G_n\rangle| \,\wedge\, i \neq n} h(0^i).$$

Now we can see that for all inputs $\langle G_1, G_2, \ldots, G_n\rangle$,

$$g(\langle G_1, G_2, \ldots, G_n\rangle) \cdot h'(\langle G_1, G_2, \ldots, G_n\rangle) = \begin{cases} 0 & \text{if } \langle G_1, G_2, \ldots, G_n\rangle \notin \text{Legitimate Deck} \\ i \cdot \widehat{h}(0^{|\langle G_1, G_2, \ldots, G_n\rangle|}) \text{ for some } i \in \{1, 2, \ldots, q(n)\} & \text{if } \langle G_1, G_2, \ldots, G_n\rangle \in \text{Legitimate Deck.} \end{cases} \quad (3)$$

Note that $\widehat{h}, h' \in$ FP and $g \in$ GapP. By Closure Properties 3.2 and 3.5, the function

$$x \mapsto g(x) \cdot h'(x)$$

is in GapP. Finally, note that since q is nondecreasing and $n \leq |\langle G_1, G_2, \ldots, G_n \rangle|$, we have that $q(n) \leq q(|\langle G_1, G_2, \ldots, G_n \rangle|)$ in Eqn. (3). Thus by Definition 4.6—with the GapP function g there being our $g(x) \cdot h'(x)$ and the function f there being our $\widehat{h}$—we have that Legitimate Deck $\in$ q-$\widehat{\text{LWPP}}$. □

We want to show that if there exists a polynomial q such that the q-Reconstruction Conjecture holds, then Legitimate Deck is in LWPP. To this end, we need the following inclusion.

Theorem 4.8. *For each function $r : \mathbb{N} \to \mathbb{N}$, r-$\widehat{\text{LWPP}} \subseteq r$-LWPP.*

Proof. Let A be a set in r-$\widehat{\text{LWPP}}$ via $f_1 \in$ FP and $g \in$ GapP. Let $f_2 \in$ FP be the function defined such that for every $n, i \in \mathbb{N}^+$, $f_2(\langle 0^n, i \rangle) = i \cdot f_1(0^n)$. Then A is in r-LWPP via $f_2 \in$ FP and $g \in$ GapP. □

Corollary 4.9. *If the q-Reconstruction Conjecture holds for some polynomial q, then the Legitimate Deck Problem is in* LWPP*.*

Proof. Suppose the q-Reconstruction Conjecture holds for nondecreasing polynomial q. (If q is not nondecreasing but the q-Reconstruction Conjecture holds, then obviously we can replace q with a nondecreasing polynomial q' that on each input n is greater than or equal to $q(n)$ and the q'-Reconstruction Conjecture will hold. So we may w.l.o.g. take it that q is nondecreasing.) By Theorems 4.7 and 4.8, Legitimate Deck $\in$ q-$\widehat{\text{LWPP}} \subseteq q$-LWPP and hence Legitimate Deck $\in$ *Poly*-LWPP. With Theorem 3.1, it follows that Legitimate Deck $\in$ LWPP. □

This gives us our new, more flexible—though still conditional—evidence that the Legitimate Deck Problem is not NP-hard.

Corollary 4.10. *If the q-Reconstruction Conjecture holds for some polynomial q, then the Legitimate Deck Problem is not* NP*-hard (or even* NP*-Turing-hard) unless* NP *is* PP*-low.*

Definition 4.11 ([KH94])**.** *Let $\mathcal{H}$ be any class of graphs. Then the* Legitimate Deck Problem restricted to $\mathcal{H}$ *consists of all sequences of graphs $\langle G_1, G_2, \ldots, G_n \rangle$ such that $\langle G_1, G_2, \ldots, G_n \rangle$ is a legitimate deck and for each $i \in \{1, 2, \ldots, n\}$, G_i is in $\mathcal{H}$.*

Note that the above definition is so flexible that it allows even the case where the preimage(s) may not be in $\mathcal{H}$.

Theorem 4.12. *Let $\mathcal{H}$ be any P-recognizable class of graphs such that decks consisting only of graphs in $\mathcal{H}$ have a number of nonisomorphic preimages that is bounded polynomially in the number of graphs in the deck. Then the Legitimate Deck Problem restricted to $\mathcal{H}$ is in* LWPP*.*

Proof. Let $\mathcal{H}$ be any P-recognizable class of graphs and q a nondecreasing polynomial such that decks consisting only of graphs in $\mathcal{H}$ have a number of nonisomorphic preimages that is bounded by $q(n)$, where n is the number of graphs in the deck.

First, we show that the Legitimate Deck Problem restricted to $\mathcal{H}$ is in q-$\widehat{\text{LWPP}}$. Let $\langle G_1, G_2, \ldots, G_n \rangle$ be an input to the Legitimate Deck Problem. Check if for every $i \in \{1, 2, \ldots, n\}$,

G_i is in $\mathcal{H}$. If this is not the case then reject in the sense of q-$\widehat{\text{LWPP}}$, i.e., produce a gap of zero. Otherwise, the deck $\langle G_1, G_2, \ldots, G_n\rangle$ has at most $q(n)$ preimages, i.e., $\text{PCount}(\langle G_1, G_2, \ldots, G_n\rangle \leq q(n)$. Proceed as in the proof of Theorem 4.7.

Since by Theorems 4.8 and 3.1, q-$\widehat{\text{LWPP}} \subseteq q$-LWPP $\subseteq$ LWPP, it follows that the Legitimate Deck Problem restricted to $\mathcal{H}$ is in LWPP. □

As usual, for each graph G, $\delta(G)$ denotes the degree of a minimum-degree vertex of G.

Theorem 4.13. *For each $k \in \mathbb{N}^+$, let*

$$\mathcal{H}_k = \{G \,|\, G \text{ is a graph such that } \delta(G) \leq k\}.$$

Then the Legitimate Deck Problem restricted to $\mathcal{H}_k$ is in LWPP*.*

Proof. In the proof of their Theorem 6.1, Kratsch and Hemaspaandra [KH94] showed that for each class of graphs with bounded minimum degree, the number of nonisomorphic preimages is polynomially bounded. Now the theorem follows from Theorem 4.12. □

It is interesting to note that for each of the $\mathcal{H}_k$ classes $\text{GI}_{\mathcal{H}_k} \equiv_m^p$ GI trivially holds (for example, via adding to each of the two graphs being tested for isomorphism an isolated node), notwithstanding the fact that intersection with $\mathcal{H}_k$ pulls the Legitimate Deck Problem's complexity into LWPP.

In two of this section's corollaries we used the fact that all LWPP sets are PP-low. We mention that since all LWPP set are also $\text{C}_=\text{P}$-low [KST92, FFK94], the altered versions of Corollaries 4.4 and 4.10 in which the conclusion is changed from "unless NP is PP-low" to "unless NP is $\text{C}_=\text{P}$-low" both hold, respectively due to Köbler, Schöning, and Torán [KST92] and the present paper.

5 Optimality of the Main Result

It is easy to see that the proof of Theorem 3.1 breaks down if *Poly*-LWPP is replaced by the analogous class where the size of the set of allowed gap values can be larger than polynomial in the input length. This does not necessarily imply that the corresponding theorem does not hold. However, in this section, we establish that relativizable proof techniques are not sufficient to improve Theorem 3.1 from *Poly*-LWPP to r-LWPP for any function r that is not polynomially bounded. That is, we show that our main result is optimal with respect to what can be proven by relativizable proof techniques.

But first, let us briefly consider the "more extreme" case of the class *Exp*-LWPP, where the number of allowed gap values can be an exponential function.

Definition 5.1.

$$\textit{Exp}\text{-LWPP} = \bigcup_{c \in \mathbb{N}^+} 2^{n^c+c}\text{-LWPP}.$$

We show that the whole class NP is contained in in *Exp*-LWPP. This gives strong evidence that *Exp*-LWPP $\not\subseteq$ LWPP because LWPP is known to be low for PP [FFK94], but NP is widely believed not to be low for PP.

Theorem 5.2. $\text{coC}_=\text{P} \subseteq$ *Exp*-LWPP *(and hence—since* $\text{coC}_=\text{P} \supseteq$ *Exp*-LWPP *is immediate from the definitions—*$\text{coC}_=\text{P} =$ *Exp*-LWPP*).*

Since NP $\subseteq$ coC$_=$P and all LWPP sets are PP-low, we obtain the following corollaries.

Corollary 5.3. NP $\subseteq$ *Exp*-LWPP.

Corollary 5.4. *If Exp*-LWPP = LWPP *then* NP $\subseteq$ LWPP *and* $\mathrm{PP}^{\mathrm{NP}} = \mathrm{PP}$ *(and, indeed,* coC$_=$P $\subseteq$ LWPP *and* $\mathrm{PP}^{\mathrm{coC_=P}} = \mathrm{PP}$*).*

All of the above, except the PP-lowness claims, analogously hold for *Exp*-WPP/WPP, since r-LWPP is a subset of r-WPP. So for example we have the following.

Corollary 5.5. *If Exp*-WPP = WPP *then* NP $\subseteq$ WPP *(and, indeed,* coC$_=$P $\subseteq$ WPP*).*

Proof of Theorem 5.2. Let $A \in$ coC$_=$P. By Theorem 2.9, there exists a function $g \in$ GapP such that for every $x \in \Sigma^*$,

$$x \in A \Longrightarrow g(x) \neq 0, \text{ and}$$
$$x \notin A \Longrightarrow g(x) = 0.$$

Let p be a polynomial such that for each $x \in \Sigma^*$, $-2^{p(|x|)} \leq g(x) \leq 2^{p(|x|)}$. Let $f \in$ FP be the function such that, for every $n \in \mathbb{N}^+$ and every $i \in \mathbb{N}^+$,

$$f(\langle 0^n, i\rangle) = \begin{cases} i/2 & \text{if } i \text{ is even} \\ -(i+1)/2 & \text{otherwise.} \end{cases}$$

We can now see that according to Definition 2.6, $A \in 2^{p(n)+1}$-LWPP and hence $A \in$ *Exp*-LWPP. □

Theorem 5.6. *Let r be any function from $\mathbb{N}$ to $\mathbb{N}$ such that for every $c \in \mathbb{N}$, $r \notin \mathcal{O}(n^c)$. Then there exists an oracle $\mathcal{O}$ such that r-*LWPP$^{\mathcal{O}} \not\subseteq$ LWPP$^{\mathcal{O}}$*.*

To prove Theorem 5.6, we will encode nondeterministic oracle Turing machines by low-degree multivariate polynomials. This technique is apparently folklore and has been used, for example, by de Graaf and Valiant [dGV02] to construct a relativized world where the quantum complexity class EQP is not contained in the modularity-based complexity class MOD$_{p^k}$P. The general technique of replacing oracle machines by simpler combinatorial objects such as circuits, decision trees, or polynomials and then using properties of such combinatorial objects to show the existence of a desired oracle dates to the seminal work of Furst, Saxe, and Sipser [FSS84], who made the connection between circuit lower bounds and the relativization of the polynomial hierarchy—a connection that has led to the resolution of many previously long-open relativized questions, such as the achievement of an oracle making the polynomial hierarchy infinite [Yao85, Hås87] and of oracles making the polynomial hierarchy extend exactly k levels [Ko89].

As indicated above, in Definition 5.7 and Proposition 5.8 below, we will encode nondeterministic oracle Turing machines by low-degree multivariate polynomials, in order to obtain polynomials that compute the gap of polynomial-time nondeterministic oracle machines.

Definition 5.7 (Folklore; see also [STT05])**.** *Let $N^{(\cdot)}(x)$ be a nondeterministic polynomial-time oracle Turing machine with running time $t(.)$ and input $x \in \Sigma^*$. Let $x_1, x_2, \ldots, x_m$ be an enumeration of all strings in Σ^* up to length $t(|x|)$.*

A polynomial encoding of $N^{(\cdot)}(x)$ *is a multilinear polynomial $p \in \mathbb{Z}[y_1, y_2, \ldots, y_m]$ defined as follows: Call a computation path ρ* valid *if ρ is a computation path of $N^D(x)$ for some oracle*

$D \subseteq \Sigma^*$. Let $x_{i_1}, x_{i_2}, \ldots, x_{i_\ell}$ be the distinct queries along a valid computation path ρ. Create a monomial $\mathrm{mono}(\rho)$ that is the product of terms z_{i_k}, $k \in \{1, 2, \ldots, \ell\}$, where $z_{i_k} = y_{i_k}$ if x_{i_k} is answered "yes" and $z_{i_k} = (1 - y_{i_k})$ if x_{i_k} is answered "no" along ρ. Define

$$p(y_1, y_2, \ldots, y_m) = \sum_{\rho:\rho \text{ is valid}} \mathrm{sign}(\rho) \cdot \mathrm{mono}(\rho).$$

Here, $\mathrm{sign}(\rho) = 1$ if ρ is an accepting path and $\mathrm{sign}(\rho) = -1$ if ρ is a rejecting path.

The next proposition states that the multilinear polynomial p has low total degree, and contains all the necessary information about $N^{(\cdot)}(x)$ to yield the value $\mathrm{gap}_{N^B}(x)$ for every oracle $B \subseteq \Sigma^*$.

Proposition 5.8 (Folklore; see also [STT05])**.** *The polynomial $p(y_1, y_2, \ldots, y_m)$ defined in Definition 5.7 has the following properties:*

1. *$\deg(p) \leq t(|x|)$, and*
2. *for each $B \subseteq \Sigma^*$, $p(\chi_B(x_1), \chi_B(x_2), \ldots, \chi_B(x_m)) = \mathrm{gap}_{N^B}(x)$.*

Lemma 5.9 ([STT05])**.** *Let $N, p \in \mathbb{N}$ be such that p is a prime and $p \leq N/2$. Let $s \in \mathbb{Z}[y_1, y_2, \ldots, y_N]$ be a multilinear polynomial with total degree $\deg(s) < p$. If for some $val \in \mathbb{Z}$, it holds that*

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

then $p \mid val$.

Lemma 5.10 states a variant of the prime number theorem. We will need it in our oracle construction to get a lower bound for the number of primes in a given set of natural numbers (see set S defined below).

Lemma 5.10 ([RS62])**.** *For every $n \geq 17$, the number of primes less than or equal to n, $\pi(n)$, satisfies*

$$\pi(n) > n/\ln n.$$

Proof of Theorem 5.6. First, we need a test language.

For every set $B \subseteq \Sigma^*$, define L_B as

$$L_B = \{0^n \mid \|B^{=n}\| > 0\},$$

where $B^{=n}$ denotes $B \cap \Sigma^n$. If B satisfies the condition that for every length n it holds that $\|B^{=n}\| \leq r(n)$, then $L_B \in r$-LWPP. (To see this, note that the function $d(0^n) = \|B^{=n}\|$ is a $\#\mathrm{P}^B$ function, where #P is Valiant's [Val79] class of functions that count the number of *accepting* paths of nondeterministic polynomial-time Turing machines. But $\#\mathrm{P} \subseteq \mathrm{GapP}$ [FFK94], and that fact itself relativizes, so $d(0^n) \in \mathrm{GapP}^B$. Since our test language should reject when $d(0^n) = 0$ and should accept when $1 \leq d(0^n) \leq r(n)$, and by our "if B satisfies" those are the only possibilities, we have $L_B \in r\text{-LWPP}^B$.)

We will construct an oracle B such that for each n, $\|B^{=n}\| \leq r(n)$ and $L_B \notin \mathrm{LWPP}^B$. Let $(N_j, M_j, p_j)_{j \geq 1}$ be an enumeration of all triples such that N_j is a nondeterministic polynomial-time

oracle Turing machine, M_j is a deterministic polynomial-time oracle Turing machine computing a function, and p_j is a monotonically increasing polynomial such that the running time of both N_j and M_j is bounded by p_j regardless of the oracle. We construct the oracle B in stages. In stage j, we decide the membership in B of strings of length n_j and extend the initial segment B_{j-1} of B to B_j. Initially, we set $B_0 = \emptyset$.

Stage j, where $j \geq 1$: Let n_j be large enough that: (a) $n_j > p_{j-1}(n_{j-1})$ (to ensure that the previous stages are not affected), (b) $r(n_j) \geq p_j(n_j)^4$, (c) $(2^{n_j} - p_j(n_j))/2 \geq p_j(n_j)^4$, and (d) $p_j(n_j)^3 - p_j(n_j) \geq p_j(n_j)^2$. Such an n_j exists because p_j is a monotonically increasing polynomial and for each $c \in \mathbb{N}$, $r \notin \mathcal{O}(n^c)$. We diagonalize against nondeterministic polynomial-time oracle Turing machine N_j and deterministic polynomial-time oracle Turing machine M_j. That is, we make sure that L_B is not decided according to the definition of LWPP by GapP function g computed by N_j together with FP function f computed by M_j (see Definition 2.5). Let val be the value computed by $M_j^{B_{j-1}}(0^{n_j})$. Because of the condition $0 \notin \text{range}(f)$ in the definition of LWPP, we can assume that $val \neq 0$. (If $val = 0$ then we can right away go to stage $j+1$.)

Let

$$T = \{w \in \Sigma^{n_j} \mid M_j^{B_{j-1}}(0^{n_j}) \text{ queries } w\}.$$

Note that $\|T\| \leq p_j(n_j)$ since the computation time of $M_j^{B_{j-1}}(0^{n_j})$ is bounded by $p_j(n_j)$. In the following, we will never add any string from T to the oracle. This ensures that the value val computed by $M_j^{B_{j-1}}(0^{n_j})$ is never changed when we replace oracle B_{j-1} by B_j.

$(*)$ We choose a set $C \subseteq \Sigma^{n_j} - T$ such that

- $\|C\| \in \{1, 2, \ldots, r(n_j)\}$ and $\text{gap}_{N_j^{B_{j-1} \cup C}}(0^{n_j}) \neq val$, or
- $\|C\| = 0$ and $\text{gap}_{N_j^{B_{j-1} \cup C}}(0^{n_j}) \neq 0$.

Let $B_j = B_{j-1} \cup C$.

End of Stage j.

This construction guarantees that for each n, $\|B^{=n}\| \leq r(n)$ and $L_B \notin \text{LWPP}^B$. Thus our proof is complete if we can show that it is always possible to find a set C satisfying $(*)$. We state and prove that as the following claim and its proof.

Claim 5.11. *For each $j \geq 1$, there exists a set C satisfying $(*)$.*

Proof of Claim 5.11. Suppose that in stage j no set C satisfying $(*)$ exists. Then for every $C \subseteq \Sigma^{n_j} - T$, the following holds:

$$\|C\| \in \{1, 2, \ldots, r(n_j)\} \Longrightarrow \text{gap}_{N_j^{B_{j-1} \cup C}}(0^{n_j}) = val, \text{ and} \tag{4}$$

$$\|C\| = 0 \Longrightarrow \text{gap}_{N_j^{B_{j-1} \cup C}}(0^{n_j}) = 0. \tag{5}$$

Let $s' \in \mathbb{Z}[y_1, y_2, \ldots, y_m]$ be the polynomial encoding of $N_j^{(\cdot)}(0^{n_j})$ as in Definition 5.7. W.l.o.g. assume that $x_1, x_2, \ldots, x_N$ enumerate the strings in $\Sigma^{n_j} - T$, and $x_{N+1}, x_{N+2}, \ldots, x_m$ enumerate the remaining strings of length at most $p_j(n_j)$. By Proposition 5.8, polynomial $s'(y_1, y_2, \ldots, y_m)$ satisfies the following:

1. For all $C \subseteq \Sigma^{n_j} - T$, it holds that

$$s'(\chi_C(x_1), \chi_C(x_2), \ldots, \chi_C(x_N), \chi_{B_{j-1}}(x_{N+1}), \ldots, \chi_{B_{j-1}}(x_m)) = \mathrm{gap}_{N_j^{B_{j-1} \cup C}}(0^{n_j}), \text{ and}$$

2. $\deg(s') \leq p_j(n_j)$.

Note that the sets B_{j-1} and $\Sigma^{n_j} - T$ are disjoint. The values $\chi_{B_{j-1}}(x_{N+1}), \ldots, \chi_{B_{j-1}}(x_m)$ do not depend on C. We fix these values and obtain the following polynomial $s(y_1, y_2, \ldots, y_N)$:

$$s(y_1, y_2, \ldots, y_N) = s'(y_1, y_2, \ldots, y_N, \chi_{B_{j-1}}(x_{N+1}), \ldots, \chi_{B_{j-1}}(x_m)).$$

Hence the polynomial $s(y_1, y_2, \ldots, y_N)$ satisfies the following:

1. For all $C \subseteq \Sigma^{n_j} - T$, it holds that

$$s(\chi_C(x_1), \chi_C(x_2), \ldots, \chi_C(x_N)) = \mathrm{gap}_{N_j^{B_{j-1} \cup C}}(0^{n_j}), \text{ and} \tag{6}$$

2. $\deg(s) \leq \deg(s') \leq p_j(n_j)$.

Statements (5) and (4) imply that

- $s(0, 0, \ldots, 0) = 0$, and
- for all $z_1, z_2, \ldots, z_N \in \{0, 1\}$ such that $\sum_{1 \leq i \leq N} z_i \in \{1, 2, \ldots, r(n_j)\}$, we have $s(z_1, z_2, \ldots, z_N) = val$.

By Lemma 5.9, for every prime k in

$$S = \{\deg(s) + 1, \deg(s) + 2, \ldots, \min(N/2, r(n_j))\},$$

it holds that $k \mid val$.

To obtain a lower bound for val, we determine a lower bound for the number of primes in S. First, note that at the beginning of stage j, we have taken n_j large enough such that $N/2 \geq (2^{n_j} - p_j(n_j))/2 \geq p_j(n_j)^4$ and $r(n_j) \geq p_j(n_j)^4$, and thus $\min(N/2, r(n_j)) \geq p_j(n_j)^4$. In light of Lemma 5.10, we hence obtain

$$\pi(\min(N/2, r(n_j))) \geq \frac{p_j(n_j)^4}{\ln(p_j(n_j)^4)} \geq p_j(n_j)^3.$$

Further, we obviously have

$$\pi(\deg(s)) \leq \deg(s) \leq p_j(n_j).$$

Hence the number of primes in S is at least $p_j(n_j)^3 - p_j(n_j)$, which is (by the choice of n_j at the beginning of stage j) greater than or equal to $p_j(n_j)^2$. Since each prime in S is greater than or equal to 2 and $val \neq 0$, we have that the absolute value of val is at least $2^{p_j(n_j)^2}$. Yet we also must have that the absolute value of val is less than or equal to $2^{p_j(n_j)}$, since the running time of $M_j^{(\cdot)}(0^{n_j})$ is bounded by $p_j(n_j)$ regardless of the oracle. This is a contradiction.

So for each $j \geq 1$, B_{j-1} can always be extended in stage j as required. This finishes the proof of Claim 5.11. □

And since Claim 5.11 was all that remained in our proof of Theorem 5.6, that theorem itself is now proven. □

Is the oracle constructed in the proof of Theorem 5.6 necessarily recursive? It might not be, since Theorem 5.6 put no complexity or computability restrictions on the function r. However, aside from that our construction is clearly effective; so if r is a computable function, then the oracle our construction builds is certainly recursive.

6 LWPP$^+$

Theorem 3.1 of Section 3 established a robustness property of LWPP, namely, that *Poly*-LWPP = LWPP. That is, having one target value for acceptance and having a list of target values for acceptance yield the same class of languages, in the content of LWPP, which, recall, is defined in terms of the values of GapP functions. That robustness result is itself robust in the sense that it holds both in the real world and, it is easy to see, in every relativized world.

On the other hand, this equivalence for LWPP, in terms of descriptive richness, between one target value and a polynomial number of values, may not hold even for quite similar counting-class situations. In particular, in this section we prove that in some relativized worlds, for the analog of LWPP defined in terms of #P rather than GapP functions, having even two target values for acceptance yields a richer class of languages than having one value. So it is not the case that single targets and lists of targets inherently function identically as to descriptive richness for counting classes.

The analog of WPP defined using #P functions rather than GapP functions already exists in the literature, namely it is the class known as F$_=$P that was recently introduced by Cox and Pay [CP18]. Similarly, we here define, and denote as LWPP$^+$, the analog of LWPP except defined using #P functions rather than GapP functions. Clearly, LWPP$^+ \subseteq$ LWPP, and it would be natural to guess that the containment is strict, although obviously proving that strictness is a much stronger result than proving P $\neq$ PSPACE and so seems beyond current techniques.

We now define LWPP$^+$ and then we prove as Theorem 6.3 that, unlike LWPP, there are relativized worlds where size-two target sets yields languages that cannot be obtained via any size-one target set.

Definition 6.1. LWPP$^+$ *is the class of all sets A such that there exists a function $g \in$ #P and a function $f \in$ FP that maps from 0^* to $\mathbb{N}^+$ such that for all $x \in \Sigma^*$,*

$$x \in A \Longrightarrow g(x) = f(0^{|x|})$$
$$x \notin A \Longrightarrow g(x) = 0.$$

Definition 6.2. Two-LWPP$^+$ *is the class of all sets A such that there exists a function $g \in$ #P and functions $f_1, f_2 \in$ FP that map from 0^* to $\mathbb{N}^+$ such that for all $x \in \Sigma^*$,*

$$x \in A \Longrightarrow g(x) \in \{f_1(0^{|x|}), f_2(0^{|x|})\}$$
$$x \notin A \Longrightarrow g(x) = 0.$$

Theorem 6.3. *There exists an oracle $\mathcal{O}$ such that* $(\text{LWPP}^+)^{\mathcal{O}} \subsetneq (\text{Two-LWPP}^+)^{\mathcal{O}}$.

We will soon prove this theorem, but we first state and prove a corollary, and will give some groundwork for the theorem's proof, and also pointers to some related work.

Corollary 6.4. *There exists an oracle $\mathcal{O}$ such that* $\mathrm{LWPP}^{\mathcal{O}} \not\subseteq (\mathrm{LWPP}^{+})^{\mathcal{O}}$.

Proof. It follows from the definitions that, for each oracle $\mathcal{Q}$, $(\text{Two-LWPP}^{+})^{\mathcal{Q}} \subseteq \textit{Poly}\text{-}\mathrm{LWPP}^{\mathcal{Q}}$. Since, as mentioned above, Theorem 3.1 clearly relativizes, we have that for each oracle $\mathcal{Q}$, $(\text{Two-LWPP}^{+})^{\mathcal{Q}} \subseteq \mathrm{LWPP}^{\mathcal{Q}}$. Thus, for the oracle $\mathcal{O}$ of Theorem 6.3, $\mathrm{LWPP}^{\mathcal{O}} \not\subseteq (\mathrm{LWPP}^{+})^{\mathcal{O}}$. □

Let us give the definition of the complexity class that is now known as $\mathrm{UP}_{\leq 2}$. This class was introduced by Beigel [Bei89] (in which the class was denoted $\mathrm{U_2P}$).

Definition 6.5. $\mathrm{UP}_{\leq 2}$ *is the class of all sets A such that there exists a function $g \in \#\mathrm{P}$ such that for all $x \in \Sigma^*$,*

$$x \in A \Longrightarrow g(x) \in \{1, 2\}$$
$$x \notin A \Longrightarrow g(x) = 0.$$

Just as Valiant's class UP [Val76] ("unambiguous NP") is the restriction of NP to allowing at most one accepting path, Beigel's $\mathrm{UP}_{\leq 2}$ is the restriction of NP to allowing at most two accepting paths. Analogously to the fact that 1-to-1 complexity-theoretic one-way functions exist if and only if $\mathrm{P} \neq \mathrm{UP}$ [GS88, Ko85], it holds that 2-to-1 complexity-theoretic one-way functions exist if and only if $\mathrm{P} \neq \mathrm{UP}_{\leq 2}$ [HZ93].

Note that $\mathrm{UP}_{\leq 2} \subseteq \text{Two-LWPP}^{+}$. To prove Theorem 6.3, we will prove the stronger result that there is an oracle relative to which even $\mathrm{UP}_{\leq 2}$ is not contained in contained in LWPP^{+}.

Theorem 6.6. *There exists an oracle $\mathcal{O}$ such that* $\mathrm{UP}_{\leq 2}^{\mathcal{O}} \not\subseteq (\mathrm{LWPP}^{+})^{\mathcal{O}}$.

Proof. First, we need a test language. For every set $B \subseteq \Sigma^*$, define L_B as

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

If B satisfies the condition that for every length n it holds that $\|B^{=n}\| \leq 2$, then $L_B \in \mathrm{UP}_{\leq 2}^{B}$.

In this proof, we are going to centrally use a slightly unusual notion of what is a computation path of a nondeterministic polynomial-time oracle Turing machine on a given oracle and a given input. Our notion of a computation path of such a machine relative to some oracle on some input will give not just the nondeterministic choices of the path, but will also contain a set of strings that ideally should capture exactly the set of queries that are asked and answered Yes along the path, and a set of strings that ideally should capture exactly the set of queries that are asked and answered No along the path. However, this is made a bit trickier as to formalizing it by the fact that we'll be dealing with various hypothetical oracles in our proof, and so sometimes things will fail to be a computation path of a machine relative to an oracle due to the action of the machine on the given input under the given oracle not being in harmony with nondeterministic bits or sets that are specifying the computation path. So our basic notion of a path will not enforce much on the two sets; rather, only when asserting that the path is in harmony with a machine under some oracle on a certain input will we require that the "natural" things hold.

Definition 6.7. *1. A* computation path *is a triple: a binary string, a set of strings, and second set of strings that is disjoint from the first set of strings.*

2. (This is implicit in the above part of the definition, but is here stated explicitly for clarity, as this will be tacitly but importantly drawn on in our proof.) Two computation paths ρ_1 and ρ_2 are equal *if and only if their first components are identical, and their second components are identical, and their third components are identical.*

3. *A computation path ρ is said to be* a computation path of a nondeterministic polynomial-time oracle Turing machine N with oracle $\mathcal{O}$ on input x *if ρ consists of a binary string $\rho_{choices}$, a set of strings $Q^+(\rho)$, and a set of strings $Q^-(\rho)$, that satisfy all of the following conditions.*

   (a) *$Q^+(\rho) \subseteq \mathcal{O}$ and $Q^-(\rho) \subseteq \overline{\mathcal{O}}$.*[3]

   (b) *When one runs machine N with oracle $Q^+(\rho)$ on input x, there is a path r in that machine's computation tree such that all the following hold:*

      i. *the sequence of nondeterministic choices in r is precisely $\rho_{choices}$ (i.e., the path makes precisely $|\rho_{choices}|$ (binary) choices and $\rho_{choices}$ lists those choices in the order they are made); and*

      ii. *the set of strings queried along path r is precisely $Q^+(\rho) \cup Q^-(\rho)$.*

4. *If in the above we after 3(b)i and 3(b)ii add new third condition that states "the path r accepts," then that is our definition of computation path ρ being* an accepting computation path of nondeterministic polynomial-time oracle Turing machine N with oracle $\mathcal{O}$ on input x.

Given an oracle $\mathcal{O}$, a computation path ρ may or may not be a computation path of a nondeterministic Turing machine N relative to $\mathcal{O}$ on input x. For example, if what the computation path specifies is inconsistent with the machine's actions on that input and oracle, then it would not be a computation path of that machine with that oracle and that input. As other examples, if relative to the machine's action on that oracle and that input the $\rho_{choices}$ of a computation path has too few or too many bits to be precise yield a complete, actual path (in the traditional sense of the word, not the more encumbered one we are using here) through the machine's action, or if one of the two "Q" sets contains a string that is not queried by the machine when it is run on x with oracle $\mathcal{O}$ making nondeterministic choices $\rho_{choices}$, then the computation path would not be a computation path of that machine with that oracle and that input.

Let $(N_j, M_j, p_j)_{j\geq 1}$ be an enumeration of all triples such that N_j is a nondeterministic polynomial-time oracle Turing machine, M_j is a deterministic polynomial-time oracle Turing machine computing a function, and p_j is a monotonically increasing polynomial such that the running time of both N_j and M_j is bounded by p_j regardless of the oracle. We construct the oracle B in stages. In stage j, we decide the membership in B of strings of length n_j and extend the initial segment B_{j-1} of B to B_j. Initially, we set $B_0 = \emptyset$.

Stage j, where $j \geq 1$: Let n_j be large enough that: (a) $n_j > p_{j-1}(n_{j-1})$ (to ensure that the previous stages are not affected), and (b) $2^{n_j} - p_j(n_j) > 6p_j(n_j) + 1$.

We diagonalize against nondeterministic polynomial-time oracle Turing machine N_j and deterministic polynomial-time oracle Turing machine M_j. That is, we make sure that L_B is not decided according to the definition of LWPP$^+$ by #P function g computed by N_j together with FP function f computed by M_j (see Definition 6.1). Let *val* be the value computed by $M_j^{B_{j-1}}(0^{n_j})$. Because of the condition $0 \notin \text{range}(f)$ in the definition of LWPP$^+$, we can assume that $val \neq 0$. (If $val = 0$ then we can right away go to stage $j+1$.)

Let

$$T = \{w \in \Sigma^{n_j} \mid M_j^{B_{j-1}}(0^{n_j}) \text{ queries } w\}.$$

[3]Note that this implies that $Q^+(\rho)$ and $Q^-(\rho)$ are disjoint.

Note that $\|T\| \leq p_j(n_j)$ since the computation time of $M_j^{B_{j-1}}(0^{n_j})$ is bounded by $p_j(n_j)$. In the following, we will never add any string from T to the oracle. This ensures that the value val computed by $M_j^{B_{j-1}}(0^{n_j})$ is never changed when we replace oracle B_{j-1} by B_j.

$(**)$ We (and we will soon prove that such a set must exist) choose a set $C \subseteq \Sigma^{n_j} - T$ such that

- $\|C\| \in \{1, 2\}$ and $\mathrm{acc}_{N_j^{B_{j-1} \cup C}}(0^{n_j}) \neq val$, or
- $\|C\| = 0$ and $\mathrm{acc}_{N_j^{B_{j-1} \cup C}}(0^{n_j}) \neq 0$.

Let $B_j = B_{j-1} \cup C$.

End of Stage j.

This construction guarantees that for each n, $\|B^{=n}\| \leq 2$ and thus $L_B \notin (\mathrm{LWPP}^+)^B$. Thus our proof is complete if we can show that it is always possible to find a set C satisfying $(**)$. We now state and prove that as Claim 6.8.

Claim 6.8. *For each $j \geq 1$, there exists a set C satisfying $(**)$.*

Proof of Claim 6.8. Suppose that in stage j no set C satisfying $(**)$ exists. Then for every $C \subseteq \Sigma^{n_j} - T$, the following holds:

$$\begin{aligned} \|C\| \in \{1, 2\} &\Longrightarrow \mathrm{acc}_{N_j^{B_{j-1} \cup C}}(0^{n_j}) = val, \text{ and} \\ \|C\| = 0 &\Longrightarrow \mathrm{acc}_{N_j^{B_{j-1} \cup C}}(0^{n_j}) = 0. \end{aligned} \tag{7}$$

It follows that (i) $N_j^{B_{j-1}}(0^{n_j})$ has no accepting computation paths, and (ii) for each string $\alpha \in \Sigma^{n_j} - T$, $N_j^{B_{j-1} \cup \{\alpha\}}(0^{n_j})$ has exactly val distinct accepting computation paths $\rho_1(\alpha), \rho_2(\alpha), \ldots, \rho_{val}(\alpha)$. For each $\alpha \in \Sigma^{n_j} - T$, let A_α be the set of accepting computation paths of $N_j^{B_{j-1} \cup \{\alpha\}}(0^{n_j})$. Then for each $\alpha \in \Sigma^{n_j} - T$, we have that $\|A_\alpha\| = val$.

Since $N_j^{B_{j-1}}(0^{n_j})$ has no accepting computation paths, we claim that for each α_1 and α_2, $\alpha_1 \neq \alpha_2$, in $\Sigma^{n_j} - T$, it holds that $A_{\alpha_1} \cap A_{\alpha_2} = \emptyset$. Why? Let $\alpha_1, \alpha_2 \in \Sigma^{n_j} - T$ be distinct and let ρ be any computation path in A_{α_1}. By definition, ρ is an accepting computation path of $N_j^{B_{j-1} \cup \{\alpha_1\}}(0^{n_j})$. Suppose $\alpha_1 \notin Q^+(\rho)$. Note that in light of that supposition we either have that (a) $\alpha_1 \in Q^-(\rho)$ and α_1 is queried by $N_j^{B_{j-1} \cup \{\alpha_1\}}(0^{n_j})$, or (b) $\alpha_1 \notin Q^+(\rho) \cup Q^-(\rho)$ and α_1 is not queried by $N_j^{B_{j-1} \cup \{\alpha_1\}}(0^{n_j})$. In each of those two cases, however, it is clear that ρ will be an accepting computation path of $N_j^{B_{j-1}}(0^{n_j})$, which is a contradiction because by our assumption, $N_j^{B_{j-1}}(0^{n_j})$ has no accepting computation paths. Hence we have shown that $\alpha_1 \in Q^+(\rho)$. But this implies that ρ is not a computation path of $N_j^{B_{j-1} \cup \{\alpha_2\}}(0^{n_j})$ (since $\alpha_1 \in Q^+(\rho)$ yet $\alpha_1 \notin B_{j-1} \cup \{\alpha_2\}$), and therefore $\rho \notin A_{\alpha_2}$.

Now we define for each $\alpha \in \Sigma^{n_j} - T$

$$\begin{aligned} \mathrm{conflicting}(\alpha) = \{\beta \in \Sigma^{n_j} - T \mid &\text{ there are at least } \lfloor val/3 \rfloor + 1 \text{ computation paths in } A_\alpha \\ &\text{that are not computation paths of } N_j^{B_{j-1} \cup \{\alpha, \beta\}}(0^{n_j})\}. \end{aligned} \tag{8}$$

We will show that for each $\alpha \in \Sigma^{n_j} - T$,

$$\|\text{conflicting}(\alpha)\| \leq 3p_j(n_j).$$

Fix some string $\alpha \in \Sigma^{n_j} - T$. To get a contradiction, suppose that $\|\text{conflicting}(\alpha)\| > 3p_j(n_j)$. Let β be some string in conflicting(α). Since adding β to $B_{j-1} \cup \{\alpha\}$ causes at least $\lfloor val/3 \rfloor + 1$ computation paths in A_α to disappear from $N_j^{B_{j-1} \cup \{\alpha, \beta\}}(0^{n_j})$, this means that

$$\|\{\rho \in A_\alpha \,|\, \beta \in Q^-(\rho)\}\| \geq \lfloor val/3 \rfloor + 1.$$

Then

$$\sum_{\beta \in \text{conflicting}(\alpha)} \|\{\rho \in A_\alpha \,|\, \beta \in Q^-(\rho)\}\| > 3p_j(n_j)(\lfloor val/3 \rfloor + 1) \geq p_j(n_j) \cdot val,$$

and thus

$$\sum_{\beta \in \Sigma^{n_j} - T} \|\{\rho \in A_\alpha \,|\, \beta \in Q^-(\rho)\}\| > p_j(n_j) \cdot val.$$

Hence it follows that

$$\sum_{\rho \in A_\alpha} \|Q^-(\rho)\| > p_j(n_j) \cdot val. \tag{9}$$

On the other hand, because each computation path ρ in A_α is of length at most $p_j(n_j)$, for each $\rho \in A_\alpha$,

$$\|Q^-(\rho)\| \leq p_j(n_j).$$

Since $\|A_\alpha\| = val$, it follows that

$$\sum_{\rho \in A_\alpha} \|Q^-(\rho)\| \leq p_j(n_j) \cdot val,$$

which contradicts Eqn. (9). Hence we have shown that for each $\alpha \in \Sigma^{n_j} - T$,

$$\|\text{conflicting}(\alpha)\| \leq 3p_j(n_j). \tag{10}$$

Because we have chosen n_j large enough to ensure that $\|\Sigma^{n_j} - T\| > 6p_j(n_j) + 1$, Eqn. (10) implies that there exist distinct $\gamma_1, \gamma_2 \in \Sigma^{n_j} - T$ such that $\gamma_1 \notin \text{conflicting}(\gamma_2)$ and $\gamma_2 \notin \text{conflicting}(\gamma_1)$. Now consider

$$N_j^{B_{j-1} \cup \{\gamma_1, \gamma_2\}}(0^{n_j}).$$

Since $\gamma_2 \notin \text{conflicting}(\gamma_1)$, there are at most $\lfloor val/3 \rfloor$ computation paths in A_{γ_1} that are not computation paths of $N_j^{B_{j-1} \cup \{\gamma_1, \gamma_2\}}(0^{n_j})$. Further, because $\|A_{\gamma_1}\| = val$, it follows that there are at least $val - \lfloor val/3 \rfloor$ computation paths in A_{γ_1} that *are* computation paths of $N_j^{B_{j-1} \cup \{\gamma_1, \gamma_2\}}(0^{n_j})$.

The same reasoning applies when we swap the roles of γ_1 and γ_2: Since $\gamma_1 \notin \text{conflicting}(\gamma_2)$, there are at most $\lfloor val/3 \rfloor$ computation paths in A_{γ_2} that are not computation paths of $N_j^{B_{j-1} \cup \{\gamma_1, \gamma_2\}}(0^{n_j})$. Further, because $\|A_{\gamma_2}\| = val$, it follows that there are at least $val - \lfloor val/3 \rfloor$ computation paths in A_{γ_2} that *are* computation paths of $N_j^{B_{j-1} \cup \{\gamma_1, \gamma_2\}}(0^{n_j})$.

For the reasons given earlier in the proof, A_{γ_1} and A_{γ_2} are disjoint sets. Hence there are at least $2(val - \lfloor val/3 \rfloor)$ (distinct!) computation paths in $A_{\gamma_1} \cup A_{\gamma_2}$ that are computation paths

of $N_j^{B_{j-1}\cup\{\gamma_1,\gamma_2\}}(0^{n_j})$. Also, all computation paths in A_{γ_1} and A_{γ_2} are accepting. Note that for any two distinct computation paths $\rho_1, \rho_2 \in A_{\gamma_1} \cup A_{\gamma_1}$ of $N_j^{B_{j-1}\cup\{\gamma_1,\gamma_2\}}(0^{n_j})$, the corresponding nondeterministic choices $\rho_{1_{choices}}$ and $\rho_{2_{choices}}$ must be different. It follows that $N_j^{B_{j-1}\cup\{\gamma_1,\gamma_2\}}(0^{n_j})$ has at least $2(val - \lfloor val/3 \rfloor) \geq 2val - \frac{2}{3}val = \frac{4}{3}val$ accepting computation paths. Since $val \in \mathbb{N}^+$, this contradicts Eqn. (7) if we set $C = \{\gamma_1, \gamma_2\}$. This completes the proof of Claim 6.8. □

Since Claim 6.8 was all that remained in our proof of Theorem 6.6, that theorem itself is now proven. □

7 On Multiple-Target C$_=$P

We show that the class C$_=$P (see Definition 2.8) is robust in the sense that even if one changes the number of target values of acceptance-path cardinality from 1 to a polynomial, the class remains unchanged.

Definition 7.1. *Let r be any function mapping from $\mathbb{N}$ to $\mathbb{N}$. Then the class r-C$_=$P is the class of all sets A such that there exists a nondeterministic polynomial-time Turing machine N and a function $f \in$ FP that maps to $\mathbb{Z}$ such that for each $x \in \Sigma^*$,*

$$x \in A \iff \textit{there exists } i \in \{1, 2, \ldots, r(|x|)\} \textit{ such that } \mathrm{acc}_N(x) = f(\langle x, i\rangle).$$

Definition 7.2.

$$\textit{Poly}\text{-C}_=\text{P} = \bigcup_{c\in\mathbb{N}^+} (n^c + c)\text{-C}_=\text{P}.$$

Theorem 7.3. *Poly*-C$_=$P = C$_=$P.

Proof. It is easy to see that C$_=$P $\subseteq$ *Poly*-C$_=$P.

To show *Poly*-C$_=$P $\subseteq$ C$_=$P, let A be a set in *Poly*-C$_=$P defined by nondeterministic polynomial-time Turing machine N, $f \in$ FP, and polynomial $r(n) = n^c + c$ according to Definitions 7.1 and 7.2.

Let h_1 be a function such that for all $x \in \Sigma^*$ and $i \in \mathbb{N}^+$,

$$h_1(\langle x, i\rangle) = f(\langle x, i\rangle) - \mathrm{acc}_N(x).$$

We have $h_1 \in$ GapP since $f \in$ FP $\subseteq$ GapP, $\mathrm{acc}_N \in \#\text{P} \subseteq$ GapP, and GapP is closed under subtraction [FFK94]. We define h_2 such that for all $x \in \Sigma^*$,

$$h_2(x) = \prod_{1\leq i\leq r(|x|)} h_1(\langle x, i\rangle).$$

By Closure Property 3.4, $h_2 \in$ GapP. Note that for all $x \in \Sigma^*$,

$$h_2(x) = \prod_{1\leq i\leq r(|x|)} \left(f(\langle x, i\rangle) - \mathrm{acc}_N(x)\right).$$

From the above equality, for each $x \in \Sigma^*$ we have that $h_2(x) = 0$ if and only if $(\exists i \in \{1, 2, \ldots, r(|x|)\})[\mathrm{acc}_N(x) = f(\langle x, i\rangle)]$. But the *Poly*-C$_=$P structures defining A specify that for each $x \in \Sigma^*$ we have $x \in A$ if and only if $(\exists i \in \{1, 2, \ldots, r(|x|)\})[\mathrm{acc}_N(x) = f(\langle x, i\rangle)]$. So $x \in A$ if and only if the GapP function $h_2(x)$ equals 0, and so by Theorem 2.9 it follows that $A \in$ C$_=$P. □

8 Conclusions and Open Questions

In this paper, we proved that LWPP and WPP are robust enough that they remain unchanged when their single target gap is allowed to be expanded to polynomial-sized lists. We then applied this new robustness of LWPP to show that the PP-lowness of the Legitimate Deck Problem follows from a weaker hypothesis than was previously known. In doing so, we provided enhanced evidence that the Legitimate Deck Problem is not NP-hard or even NP-Turing-hard. We also showed: that the polynomial-target robustness of LWPP that we have established is optimal (i.e., cannot be extended to any superpolynomial number of targets) with respect to relativizable proofs; that for the #P-based analogue of the (GapP-based) class LWPP, in some relativized worlds even two targets give more languages than one target; that our robustness of LWPP holds even when one simultaneously expands both the acceptance target-gap set and the rejection target-gap set to be polynomial-sized lists; that our main results also hold for WPP; and that $C_{=}P$ is polynomial-target robust.

Regarding the Reconstruction Conjecture, we proved as a consequence of our results that if there exists a polynomial q such that the q-Reconstruction Conjecture holds, then the Legitimate Deck Problem is both PP-low and $C_{=}P$-low. Since NP is widely believed not to be PP-low or $C_{=}P$-low, this provides strengthened evidence that the Legitimate Deck Problem is not NP-hard or even NP-Turing hard (since otherwise even the "Poly"-Reconstruction Conjecture must fail, yet even the far stronger Reconstruction Conjecture is generally believed to hold).

A natural open problem is whether the Legitimate Deck Problem is Σ_k^p-low [Sch83] for some k, i.e., whether for some k it holds that $(\Sigma_k^p)^{\text{Legitimate Deck}} = \Sigma_k^p$. Proving that that holds—though we mention that this has been a known open issue for more than two decades (see [KH94])—would imply that the Legitimate Deck Problem cannot be NP-complete (even with respect to more flexible reductions such as Turing reductions and strong nondeterministic reductions [Lon82]) unless the polynomial hierarchy collapses.

References

[BCS92] D. Bovet, P. Crescenzi, and R. Silvestri. A uniform approach to define complexity classes. *Theoretical Computer Science*, 104(2):263–283, 1992.

[Bei89] R. Beigel. On the relativized power of additional accepting paths. In *Proceedings of the 4th Structure in Complexity Theory Conference*, pages 216–224. IEEE Computer Society Press, June 1989.

[BH77] J. Bondy and R. Hemminger. Graph reconstruction—a survey. *Journal of Graph Theory*, 1:227–268, 1977.

[Bon91] J. Bondy. A graph reconstructor's manual. In *Surveys in Combinatorics*, London Mathematical Society Lecture Notes Series 66, pages 221–252. Cambridge University Press, 1991.

[CGH+89] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy II: Applications. *SIAM Journal on Computing*, 18(1):95–111, 1989.

[CP18] J. Cox and T. Pay. An overview of some semantic and syntactic complexity classes. Technical Report arXiv:1806.03501 [cs.CC], ArXiv.org, June 2018.

[dGV02] M. de Graaf and P. Valiant. Comparing EQP and $\mathrm{MOD}_{p^k}\mathrm{P}$ using polynomial degree lower bounds. Technical Report quant-ph/0211179, Quantum Physics, 2002.

[Edi77] The Editors (of the Journal of Graph Theory). Editorial note. *Journal of Graph Theory*, 1(3), 1977.

[FFK94] S. Fenner, L. Fortnow, and S. Kurtz. Gap-definable counting classes. *Journal of Computer and System Sciences*, 48(1):116–148, 1994.

[FSS84] M. Furst, J. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17:13–27, 1984.

[GS88] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. *SIAM Journal on Computing*, 17(2):309–335, 1988.

[Hås87] J. Håstad. *Computational Limitations of Small-Depth Circuits*. MIT Press, 1987.

[HHRT07] E. Hemaspaandra, L. Hemaspaandra, S. Radziszowski, and R. Tripathi. Complexity results in graph reconstruction. *Discrete Applied Mathematics*, 155(2):103–118, 2007.

[HZ93] L. Hemaspaandra and M. Zimand. Strong forms of balanced immunity. Technical Report TR-480, Department of Computer Science, University of Rochester, Rochester, NY, December 1993. Revised, May 1994.

[Kel42] P. Kelly. *On Isometric Transformations*. PhD thesis, University of Wisconsin, USA, 1942.

[KH91] D. Kratsch and L. Hemachandra. On the complexity of graph reconstruction. In *Proceedings of the 8th Conference on Fundamentals of Computation Theory*, pages 318–328. Springer-Verlag *Lecture Notes in Computer Science #529*, September 1991.

[KH94] D. Kratsch and L. Hemaspaandra. On the complexity of graph reconstruction. *Mathematical Systems Theory*, 27(3):257–273, 1994.

[Ko85] K. Ko. On some natural complete operators. *Theoretical Computer Science*, 37(1):1–30, 1985.

[Ko89] K. Ko. Relativized polynomial-time hierarchies having exactly k levels. *SIAM Journal on Computing*, 18(2):392–408, 1989.

[KST92] J. Köbler, U. Schöning, and J. Torán. Graph isomorphism is low for PP. *Computational Complexity*, 2:301–330, 1992.

[Lon82] T. Long. Strong nondeterministic polynomial-time reducibilities. *Theoretical Computer Science*, 21:1–25, 1982.

[LS03] J. Lauri and R. Scapellato. *Topics in Graph Automorphisms and Reconstruction*. Cambridge University Press, 2003.

[Man82] A. Mansfield. The relationship between the computational complexities of the legitimate deck and isomorphism problems. *Quart. J. Math. Ser.*, 33(2):345–347, 1982.

[Man88] B. Manvel. Reconstruction of graphs: Progress and prospects. *Congressus Numerantium*, 63:177–187, 1988.

[NW78] C. St. J. A. Nash-Williams. The reconstruction problem. In L. Beineke and R. Wilson, editors, *Selected Topics in Graph Theory*, pages 205–236. Academic Press, 1978.

[OH93] M. Ogiwara and L. Hemachandra. A complexity theory for feasible closure properties. *Journal of Computer and System Sciences*, 46(3):295–325, 1993.

[RS62] J. Rosser and L. Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois Journal of Mathematics*, 6:64–94, 1962.

[Sch83] U. Schöning. A low and a high hierarchy within NP. *Journal of Computer and System Sciences*, 27:14–28, 1983.

[Sim75] J. Simon. *On Some Central Problems in Computational Complexity*. PhD thesis, Cornell University, Ithaca, N.Y., January 1975. Available as Cornell Department of Computer Science Technical Report TR75-224.

[SM73] L. Stockmeyer and A. Meyer. Word problems requiring exponential time. In *Proceedings of the 5th ACM Symposium on Theory of Computing*, pages 1–9. ACM Press, 1973.

[STT05] H. Spakowski, M. Thakur, and R. Tripathi. Quantum and classical complexity classes: Separations, collapses, and closure properties. *Information and Computation*, 200(1):1–34, July 2005.

[Ula60] S. Ulam. *A Collection of Mathematical Problems*. Interscience Publishers, New York, 1960.

[Val76] L. Valiant. The relative complexity of checking and evaluating. *Information Processing Letters*, 5(1):20–23, 1976.

[Val79] L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[Wag86] K. Wagner. The complexity of combinatorial problems with succinct input representations. *Acta Informatica*, 23(3):325–356, 1986.

[Wra77] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3:23–33, 1977.

[Yao85] A. Yao. Separating the polynomial-time hierarchy by oracles. In *Proceedings of the 26th IEEE Symposium on Foundations of Computer Science*, pages 1–10, 1985.

A Proof of Theorem 3.10

Theorem 3.10. $(Poly, Poly)$-LWPP = $Poly$-LWPP.

Proof. It is easy to see that $Poly$-LWPP $\subseteq$ $(Poly, Poly)$-LWPP.

To show $(Poly, Poly)$-LWPP $\subseteq$ $Poly$-LWPP, let B be a set in $(Poly, Poly)$-LWPP defined by $g \in$ GapP, $f_A, f_R \in$ FP, and polynomials $r_A(n) = n^c + c$ and $r_R(n) = n^c + c$ according to Definitions 3.8 and 3.9.

Let h be a function such that for all $x \in \Sigma^*$ and $j \in \mathbb{N}^+$,

$$h(\langle x, j\rangle) = f_R(\langle 0^{|x|}, j\rangle) - g(x).$$

We have $h \in$ GapP since $f_R \in$ FP $\subseteq$ GapP, $g \in$ GapP, and as noted earlier GapP is closed under subtraction [FFK94]. We define $\widehat{g}$ such that for all $x \in \Sigma^*$,

$$\widehat{g}(x) = \prod_{1 \le j \le r_R(|x|)} h(\langle x, j\rangle).$$

By Closure Property 3.4, $\widehat{g} \in$ GapP. Note that for all $x \in \Sigma^*$,

$$\widehat{g}(x) = \prod_{1 \le j \le r_R(|x|)} \Big(f_R(\langle 0^{|x|}, j\rangle) - g(x) \Big). \tag{11}$$

Let $\widehat{f}$ be a function such that for all $\ell \in \mathbb{N}$ and $i \in \mathbb{N}^+$,

$$\widehat{f}(\langle 0^{\ell}, i\rangle) = \prod_{1 \le j \le r_R(\ell)} \Big(f_R(\langle 0^{\ell}, j\rangle) - f_A(\langle 0^{\ell}, i\rangle) \Big). \tag{12}$$

It is easy to see that $\widehat{f} \in$ FP. It follows from Eqns. (11) and (12) that for every $x \in \Sigma^*$, the following are true.

1. If there exists $j \in \{1, 2, \ldots, r_R(|x|)\}$ such that $g(x) = f_R(\langle 0^{|x|}, j\rangle)$ then $\widehat{g}(x) = 0$.
2. For each $i \in \{1, 2, \ldots, r_A(|x|)\}$, it holds that $g(x) = f_A(\langle 0^{|x|}, i\rangle)$ implies that $\widehat{g}(x) = \widehat{f}(\langle 0^{|x|}, i\rangle)$.

Hence for each $x \in \Sigma^*$,

$$x \notin B \Longrightarrow \widehat{g}(x) = 0,$$
$$x \in B \Longrightarrow \text{ there exists } i \in \{1, 2, \ldots, r_A(|x|)\} \text{ such that } \widehat{g}(x) = \widehat{f}(\langle 0^{|x|}, i\rangle).$$

Note also (see Eqn. (12)) that for all $\ell \in \mathbb{N}$ and $i \in \{1, 2, \ldots, r_A(\ell)\}$, it holds that $\widehat{f}(\langle 0^{\ell}, i\rangle) \neq 0$ because for each $\ell \in \mathbb{N}$, the sets A_ℓ and R_ℓ in Definition 3.8 are disjoint. By Definition 2.7, it thus follows that $B \in Poly$-LWPP. □