Electronic Colloquium on Computational Complexity, Report No. 318 (2026)

SEPARATIONS WITH IMMUNITY RELATIVE TO A RANDOM ORACLE IN COMPUTATIONAL COMPLEXITY

GABRIEL ISTRATE

Abstract. Rossman, Servedio, and Tan [41] proved that the polynomial hierarchy is infinite relative to a random oracle. This paper strengthens this celebrated result by considering *strong separations*, witnessed by immune languages.

The main result of the paper shows that with respect to a random oracle the levels of the polynomial hierarchy separate with immunity. Specifically, with probability one over a random oracle A, for every $k \geq 1$ there is a language in $\Sigma_k^{\mathsf{P},A}$ that is immune to $\Pi_k^{\mathsf{P},A}$, and, symmetrically, a language in $\Pi_k^{\mathsf{P},A}$ that is immune to $\Sigma_k^{\mathsf{P},A}$. We thus extend classical results due to Bennett and Gill [7] and Vereshchagin [46, 47].

We accomplish this by developing a "rare-or-wrong" approach to strong separations. We highlight the flexibility of the approach by strengthening other separations with respect to a random oracle from the complexity-theoretic literature.

Contents

1. Introduction 2
2. Formal Statement of Our Results 3
3. Highlights of Our Approach 5
3.1. The rare-or-wrong inequalities. 6
3.2. The spacing of input lengths. 8
4. Preliminaries 9
4.1. Notions from Computational Complexity. 9
4.2. Notions from Probability Theory. 11
4.3. Notions from Cryptography. 11
4.4. Hardness Results. 12
5. Proofs of Our Other Results 13
5.1. Proof of Theorem 2. 14
5.2. Proof of Theorem 4. 17
5.3. Proof of Theorem 6. 19
5.4. Proof of Theorem 7. 24
6. Proof of the Main Theorem 30
6.1. Polynomial time computability of the intermediate fan-ins. 30
6.2. The reparameterized Sipser functions. 32
6.3. Useful properties of the reparameterized Sipser functions. 33
6.4. Hardness result for family $G_{d,m}$. 35
6.5. Deriving the Main Theorem from the Hardness Lemma. 37
7. Conclusions and Open Problems 40
8. Acknowledgments 41
References 42

2020 *Mathematics Subject Classification.* 68Q15; Secondary 68Q17, 03D15.
Key words and phrases. random oracles, immune sets, computational complexity.

ISSN 1433-8092

1. Introduction

Complexity theory is (or at least promised to be[1]) a theory of distinctions: between what can and cannot be efficiently computed, between different kinds of computational resources, and between apparently similar models whose computational capabilities may turn out to differ in rather significant ways[2]. Yet not all distinctions between complexity classes are equally informative. To prove that a class $\mathcal{D}$ is not contained in a class $\mathcal{C}$ it is enough to exhibit a single language in $\mathcal{D} \setminus \mathcal{C}$. A stronger kind of separation asks for a witness that has no infinite "easy core" in $\mathcal{C}$ at all. This is the point of view embodied by *immunity*.

The notion itself migrated to Complexity Theory from (the older field of) Recursion Theory[3]: Post's construction of simple sets [39] indicated that incomputability could be studied not only through the undecidability of a set as a whole, but also through the absence of large recursively enumerable subsets. The subsequent theory of immune and biimmune sets, including the study of their degrees [32], made immunity a standard structural tool of recursion theory. When imported into computational complexity (see e.g. Homer and Maass [30]), the idea of immunity acquired a resource-bounded interpretation: a language L is $\mathcal{C}$-immune if it is infinite but contains no infinite sublanguage in $\mathcal{C}$. Thus a language being $\mathcal{C}$-immune witnesses considerably more than nonmembership in $\mathcal{C}$; it says that every infinite fragment of the witness is "$\mathcal{C}$-hard".

Should immunity be viewed simply as a technically stronger form of class separation? There is (some) evidence that the answer to the previous question is negative, and that the existence of immune languages within complexity classes is, indeed, significant. For instance, P-bi-immunity is prevalent (at least in E) [36], is equivalent to a form of almost-everywhere polynomial-time hardness [5], and implies strong distributional hardness [13, 18]. More strikingly, Pavan and Selman [38] showed that if, for some $\epsilon > 0$, NP contains a $\mathsf{DTIME}(2^{n^\epsilon})$-biimmune language, then natural notions of NP-completeness separate: there exists a language that is 2-Turing complete for NP but not 1-truth-table complete[4]. In another paper [37] they showed that the existence of P-biimmune sets has implications for average-case complexity. These results suggest that immunity is not just a convenient strengthening of non-membership: it captures a form of intractability with potentially important[5] structural consequences.

Strong separations by immune sets were investigated for relativized polynomial hierarchies by Ko [34] and Bruschi [8]. Bennett and Gill's random-oracle work [7], dating to what one might call *the classical period of Complexity Theory*, already contained an immunity theorem at the first level; Vereshchagin later obtained the more symmetric conclusion that, relative to a random oracle, NP and coNP contain sets immune to the opposite class [46, 47]. These results suggest a natural question: *how far can one push the classical setting of strong separation into the theory of random oracles?*

A customary inspection of the literature reveals that probability-one separations with immunity are substantially less common in the literature than ordinary random-oracle separations, which have achieved a respectable degree of success (e.g. [3, 7, 9, 10, 15, 19, 27, 28, 41, 46]). Beyond the classical results discussed above, not that many people seemed to care[6]. A notable exception is the work of Hemaspaandra and Zimand, who introduced stronger notions of immunity and proved, in particular, that relative to a random oracle NP^A and $\oplus\mathrm{P}^A$ contain P^A-balanced-immune sets [28]. By contrast,

[1] *How well* Complexity Theory has fulfilled this promise is another discussion; one that we prefer, however, not to undertake in this paper.

[2] Yes, I know, it sounds like a "stock LLM" introduction. I still like it though, especially after my changes and "humanization": I was writing philosophical, *compare-and-contrast* introductions before the LLM era, e.g. [31].

[3] I prefer the old, historical name of this area.

[4] Hitchcock, Pavan and Vinodchandran later obtained a related separation under the weaker hypothesis of partial biimmunity [29].

[5] But, sadly, somewhat forgotten/ignored.

[6] At least by writing subsequent papers on the topic.

much of the subsequent literature on immunity and relativization concerns specially constructed oracles, or generic oracles, rather than probability-one statements; see, for instance, [8, 43]. Thus the extensive theory of random-oracle separations has, to a significant extent, developed without a parallel theory of random-oracle separations witnessed by immune languages.

The most striking case is, perhaps, that of separating the relativized polynomial hierarchy: here separations with respect to a random oracle were achieved by the seminal result of Rossman, Servedio and Tan [41] (henceforth abbreviated as RST; see also [27]), while for separations with immunity there has been essentially no progress since Bennett-Gill and Vereshchagin.

The goal of this paper is to propose an approach that can often lift separations with respect to a random oracle to separations with immunity. A major success of this approach is that *we strengthen the results of Bennett-Gill and Vereshchagin (and the random oracle consequence of Rossman, Servedio and Tan) by separating arbitrary levels of the polynomial hierarchy with immunity.* Our approach is more general than this single result: we demonstrate this through a number of other applications (see the next section for specific details).

The rest of the paper is organized as follows: Section 2 formally states the results we prove. Section 3 discusses the common ideas behind our approach. Section 4 collects the complexity-theoretic, probabilistic, cryptographic, and lower-bound preliminaries. Section 5 proves all claimed results except the main one (the separation of the polynomial hierarchy with immunity with respect to a random oracle), beginning with the applications in which the method is most transparent. The proof of the main result is given in Section 6, and has to clear up several hurdles:

- We note that the bottom/intermediate fan-ins of the Sipser formulas from [41] are polynomial-time computable; on the other hand the top fan-in is not obviously so. Deriving the random-oracle separation from the circuit lower bound follows the framework of [25] (Chapter 7), which requires the hard formulas to be *uniformly constructible*; for the balanced Sipser functions of [41, 27, 42] this is not explicitly addressed in the literature, but is quite important to ensure that the test languages we construct in the proof of Theorem 1 are members of the required complexity classes.
- We offer a safe way to circumvent this issue by proposing a reparameterized version of Sipser formulas, and deriving a hardness result for these formulas from the corresponding hardness result in [41].
- Since the reparameterized formulas will have in general a different number of variables from the original one, to derive such a result we need to resort to a *coupling* of the variables in the two formulas.
- Once such a hardness result is available for the reparameterized class, the proof proceeds along avenues similar to those for the other results.

Finally, Section 7 discusses some limitations of our approach and subsequent open problems.

2. Formal Statement of Our Results

Our main theorem improves upon the seminal results of Bennett and Gill [7] and Vereshchagin [46, 47], as well as the (single oracle) corresponding strong separations of Bruschi [8] and Ko [34], separating the levels of the polynomial hierarchy (with respect to a random oracle) *with immunity:*

Theorem 1. *With probability one over the random oracle A, simultaneously for every integer $k \geq 1$:*

(a) there exists an infinite language $L_k^A \in \Sigma_k^{\mathsf{P},A}$ that is $\Pi_k^{\mathsf{P},A}$-immune;
(b) there exists an infinite language $R_k^A \in \Pi_k^{\mathsf{P},A}$ that is $\Sigma_k^{\mathsf{P},A}$-immune.

Relatedly, Cai [9] (see also Babai [4] for a simpler proof) had separated PH from PSPACE with respect to a random oracle. This separation also holds with immunity:

Theorem 2. *With probability one over a random oracle A, there exists a language $L^A \in \oplus\mathsf{P}^A \subseteq \mathsf{PSPACE}^A$ that is PH^A-immune. Consequently, with probability 1*

$$\mathsf{PSPACE}^A \text{ contains a } \mathsf{PH}^A\text{-immune language.} \tag{1}$$

Remark 3. Regan and Royer [40, Corollary 26] observed that Cai's earlier result, together with their theorem, implies that with probability one $\mathsf{PH}^A \subsetneq \oplus\mathsf{P}^A$. The previous theorem strengthens this separation to separation with immunity.

On the other hand, Bennett and Gill [7] had claimed that with probability 1 PSPACE^A contains a language not in PP^A. However, Beigel [6] observed that their proof was not correct. The claim was finally established by Aspnes, Beigel, Furst, and Rudich [3]. We improve over this last result as follows:

Theorem 4. *With probability 1 over a random oracle A, $\oplus\mathsf{P}^A$ contains a PP^A-immune language, hence with probability 1*

$$\mathsf{PSPACE}^A \text{ contains a } \mathsf{PP}^A\text{-immune language.} \tag{2}$$

Remark 5. Rothe has presented (Corollary 4.3 in [43]) an oracle relative to which $\oplus\mathsf{P}$ contains a language immune to both relativized PH and PP, and has asked (in the closing lines of [43]) whether any of the oracle constructions in his paper extend to immunity with respect to a random oracle.

The (common) language we use to prove Theorems 2 and 4 yields a positive answer to Rothe's question.

Our next application concerns the failure of the random oracle hypothesis [15]. To establish this result, Chang et al. showed that with probability 1 over a random oracle A there is a language in $\mathsf{coNP}^A \setminus \mathsf{IP}^A$. The corresponding improvement is:

Theorem 6. *With probability one over a random oracle A, there is a language $L^A \in \mathsf{coNP}^A$ that is IP^A-immune. Hence with probability 1*

$$\mathsf{PSPACE}^A \text{ contains an } \mathsf{IP}^A\text{-immune language.} \tag{3}$$

Finally, separating P^A and $\mathsf{NP}^A \cap \mathsf{coNP}^A$ with respect to a random oracle had long been an open problem: Bennett and Gill [7] proved such a result relative to a *random permutation* (see also [45]). Recently Ghosal et al. [19] obtained a conditional separation under the additional hypothesis $\mathrm{UP} \not\subseteq \mathrm{RP}$. We cannot extend this result to separations with immunity; instead, we employ a stronger hypothesis. Specifically, we prove:

Theorem 7. *Assume that a uniform one-way permutation exists in the sense of Definition 31. Then, with probability one over the random oracle A, there exist two disjoint languages $L_0^A, L_1^A \in \mathrm{UP}^A \cap \mathrm{coUP}^A$ such that*

$$L_0^A \cup L_1^A = T_{\geq 2} := \{1^{2^{j!}} : j \geq 2\} \in \mathsf{P} \tag{4}$$

and both L_0^A and L_1^A are P^A-immune. Consequently, with probability 1 there is a language in $\mathrm{UP}^A \cap \mathrm{coUP}^A$ (hence in $\mathsf{NP}^A \cap \mathsf{coNP}^A$) that is P^A-immune.

The previous series of results may have given the impression that every separation result under a random oracle extends (perhaps at the cost of stronger conditional hypotheses) to separation with immunity. This is *not* true: Cai [10] had separated the Boolean Hierarchy [12] (see below) with respect to a random oracle.

Definition 8. Let A be an oracle. The levels of the Boolean hierarchy relative to A are defined recursively as follows: $\mathsf{BH}_0^A = \mathsf{P}^A$, $\mathsf{BH}_1^A = \mathsf{NP}^A$, and, for every $k \geq 1$,

$$\mathsf{BH}_{k+1}^A = \begin{cases} \{L_1 \setminus L_2 : L_1 \in \mathsf{BH}_k^A,\ L_2 \in \mathsf{NP}^A\}, & \text{if } k \text{ is odd,} \\ \{L_1 \cup L_2 : L_1 \in \mathsf{BH}_k^A,\ L_2 \in \mathsf{NP}^A\}, & \text{if } k \text{ is even.} \end{cases} \tag{5}$$

Equivalently,

$$\mathsf{BH}_{2i}^A = \{L_1 \cap L_2 : L_1 \in \mathsf{BH}_{2i-1}^A,\ L_2 \in \mathsf{coNP}^A\}, \tag{6}$$

and

$$\mathsf{BH}_{2i+1}^A = \{L_1 \cup L_2 : L_1 \in \mathsf{BH}_{2i}^A,\ L_2 \in \mathsf{NP}^A\}. \tag{7}$$

The *Boolean hierarchy relative to* A is $\mathsf{BH}^A = \bigcup_{k\geq 1} \mathsf{BH}_k^A$. For the empty oracle we simply write BH_k and BH. In particular, $\mathsf{BH}_1^A = \mathsf{NP}^A$ and

$$\mathsf{BH}_2^A = \{L_1 \cap L_2 : L_1 \in \mathsf{NP}^A,\ L_2 \in \mathsf{coNP}^A\}. \tag{8}$$

The latter class is conventionally denoted by DP^A.

Cai's level-by-level random-oracle separation [10] cannot be strengthened to level-by-level immunity beyond the base level:

Observation 9. *The following statements hold:*

(a) With probability one over a random oracle A, there is a language in DP^A that is NP^A-immune.
(b) For every oracle A and every $k \geq 2$, no infinite language $L \in \mathsf{BH}^A$ is BH_k^A-immune.

Proof. (a). By Vereshchagin's result [46], with probability 1 there is a language in $\mathsf{coNP}^A \subseteq DP^A$ that is NP^A-immune.

(b). Part (b) is the relativized form of [11, Theorem 4.3]. Alternatively, it follows immediately from their normal-form Theorem 2.1.1: every $L \in \mathrm{BH}^A$ is a finite union of DP^A languages, so if L is infinite, at least one of these languages is an infinite DP^A subset of L.

□

3. Highlights of Our Approach

Our proofs exploit existing lower bounds on boolean functions, together with three ingredients:

- **The main ingredient is a *rare-or-wrong* principle.** Fix a machine from the class against which immunity is to be proved, and consider its attempts to recognize an infinite subset of the language being constructed. Intuitively, at each relevant input length, either the machine accepts only with sufficiently small conditional probability (the "rare" case), or, if it accepts often enough, the remaining randomness gives it a substantial probability of accepting an input outside the target language (the "wrong" case). Thus an infinite correct subset would require the machine to avoid both alternatives indefinitely.
- *The probabilistic mechanism excluding this possibility* is **a conditional, or Lévy-type, version of the Borel-Cantelli lemma.** We expose the random oracle through a filtration $(\mathcal{F}_s)_{s\geq 0}$ (see Section 4 below) so that at stage s the past is contained in $\mathcal{F}_{s-1}$ while a sufficiently large block of fresh oracle bits remains unexposed. Conditional on the past, we estimate the probabilities of acceptance and error: if the sum of the former is finite, the machine accepts at only finitely many relevant stages, whereas if it diverges, the rare-or-wrong estimate forces the conditional probabilities of errors to have divergent sum, and Borel-Cantelli then makes errors occur infinitely often. The mechanism that couples (globally) the good and bad events at different lengths is the summability of the difference terms between good and bad events in the "rare-or-wrong" lemma(s).
- Finally, **an appropriately *spaced set* of relevant input lengths** separates the two roles played by the oracle. The computation at an earlier stage is completed before the fresh randomness needed at a later stage is exposed. This spacing prevents dependencies created by oracle queries from contaminating the next probabilistic experiment and thereby allows us to separate the complexity-theoretic part of the argument (which provides a hardness or

unpredictability estimate at a single stage) from the probabilistic part, which converts these local estimates into an almost-sure global immunity statement.

There are, of course, some problem-specific technical complications. They will be presented when giving each specific proof. For now we want to give some details on some of our main technical points:

3.1. **The rare-or-wrong inequalities.** Immunity is one-sided: to preclude immunity, an algorithm need not approximate the target on most inputs; it may accept a very sparse collection of positive inputs. The following elementary inequality extracts exactly the one-sided information needed to guarantee immunity under a random oracle:

Lemma 10. *Let $H, C : \{0,1\}^N \to \{0,1\}$, and let P be a probability on $\{0,1\}^N$. Suppose that, for some $\delta, \varepsilon \geq 0$,*

$$\left| \mathbf{Pr}_P(H = 1) - \frac{1}{2} \right| \leq \delta, \tag{9}$$

$$\mathbf{Pr}_P(C = H) \leq \frac{1}{2} + \varepsilon. \tag{10}$$

Set

$$p = \mathbf{Pr}_P(C = 1), \qquad q = \mathbf{Pr}_P(C = 1 \text{ and } H = 0).$$

Then

$$q \geq \frac{p}{2} - \frac{\delta + \varepsilon}{2}. \tag{11}$$

Proof. Let $\mu = \mathbf{Pr}_P(H = 1)$. Since

$$\mathbf{Pr}_P(C = 1, H = 1) = p - q \quad \text{and} \quad \mathbf{Pr}_P(C = 0, H = 0) = 1 - \mu - q,$$

we obtain

$$\mathbf{Pr}_P(C = H) = 1 - \mu + p - 2q.$$

Therefore

$$q = \frac{1 - \mu + p - \mathbf{Pr}_P(C = H)}{2}.$$

Now $1 - \mu \geq \frac{1}{2} - \delta$ by (9), while $\mathbf{Pr}_P(C = H) \leq \frac{1}{2} + \varepsilon$ by (10). Substitution proves (11). □

Remark 11. The name "rare-or-wrong" is intuitively motivated (in this case) by the following observation: for an arbitrary $\rho > 0$ either $p \leq \rho$ (the "good event" is rare) or $q \geq \frac{\rho - \delta - \epsilon}{2}$ (the "bad event" is not rare).

Lemma 10 will be useful in several of our results. However, when diagonalizing against probabilistic algorithms we have not one but *two* thresholds to take into consideration. In such cases the following "two-threshold rare-or-wrong" lemma may prove more appropriate:

Lemma 12. *Let Ω be a nonempty finite set, equipped with the uniform probability measure, let*

$$H : \Omega \to \{0,1\}, \qquad v : \Omega \to [0,1],$$

and fix two thresholds $0 \leq b < a \leq 1$.

Define

$$E = \{x \in \Omega : H(x) = 1,\ v(x) > a\},$$

and

$$B = \{y \in \Omega : H(y) = 0,\ v(y) > b\}.$$

Suppose that there is a relation $R \subseteq E \times B$ and integer constants $d, D > 0$ such that:

(1) For every $x \in E$ there are at least d indices $y \in B$ such that $(x, y) \in R$;

(2) For every $y \in B$ there are at most D indices $x \in E$ with $(x, y) \in R$.

Then

$$\mathbf{Pr}(B) \geq \frac{d}{D}\Pr(E). \tag{12}$$

Proof. Double count the pairs in R.

By the first hypothesis, every $x \in E$ is incident to at least d pairs in R. Hence $|R| \geq d\,|E|$.

By the second hypothesis, every $y \in B$ is incident to at most D pairs in R. Hence $|R| \leq D\,|B|$. Therefore $d\,|E| \leq D\,|B|$, and thus

$$|B| \geq \frac{d}{D}|E|. \tag{13}$$

Since Ω carries the uniform probability measure,

$$\mathbf{Pr}(E) = \frac{|E|}{|\Omega|}, \qquad \mathbf{Pr}(B) = \frac{|B|}{|\Omega|}. \tag{14}$$

Hence the desired inequality (12) follows. □

Remark 13. In this case the "rare-or-wrong" name is motivated by the following inequality: For every $\rho > 0$

$$\mathbf{Pr}(E) \leq \rho \text{ ("rare event"), or } \mathbf{Pr}(B) > \frac{d}{D}\rho \text{ ("wrong event.")} \tag{15}$$

Remark 14. The purpose of this lemma is to handle a randomized acceptance probability $v : \Omega \to [0,1]$, rather than a single bit. Intuitively:

- $H(x) = 1$ will mean that x belongs to the target language.
- $v(x)$ will be the probability that the probabilistic algorithm M we diagonalize against (in our specific application it will be an interactive-proof verifier) accepts x.
- a, b are the thresholds for acceptance/rejection of this machine.
- The "good event" E is the fact that x is in the language and M accepts x with probability $> a$.
- The "bad event" B is the fact that x is not in the language but M accepts x with probability $> b$.
- The "rare-or-wrong" inequality is (15).

A third "rare or wrong" principle is appropriate in cases when the sample space carries an involution that flips the value of a bit-signature H:

Lemma 15. *Let Ω be a nonempty finite set equipped with the uniform probability measure, and let $\Phi\colon \Omega \to \Omega$ be an* involution*, i.e. $\Phi \circ \Phi = \mathrm{id}$. Assume that $H\colon \Omega \to \{0,1\}$ satisfies*

$$H(\Phi(\omega)) = 1 - H(\omega) \qquad \textit{for every } \omega \in \Omega. \tag{16}$$

Let $E, T \subseteq \Omega$ be such that T is Φ-invariant (i.e. $\Phi(T) = T$), and such that[7]

$$E \triangle \Phi(E) \;\subseteq\; T. \tag{17}$$

For $b \in \{0,1\}$ put $B_b = E \cap \{H \neq b\}$. Then

$$|B_b| \;\geq\; \tfrac{1}{2}\big(|E| - |T|\big), \qquad \textit{equivalently} \qquad \mathbf{Pr}(B_b) \;\geq\; \tfrac{1}{2}\big(\mathbf{Pr}(E) - \mathbf{Pr}(T)\big). \tag{18}$$

Proof. Put $E_0 = E \setminus T$.

Claim 16. E_0 is Φ-invariant.

Proof. Let $\omega \in E_0$. Since $\Phi(T) = T$ and $\omega \notin T$, also $\Phi(\omega) \notin T$. Moreover $\omega \in E$ and $\omega \notin T$, so by (17) $\omega \notin E \triangle \Phi(E)$, whence $\omega \in \Phi(E)$, say $\omega = \Phi(e)$ with $e \in E$. As Φ is an involution, $\Phi(\omega) = e \in E$. Therefore $\Phi(\omega) \in E_0$, and applying the same argument to $\Phi(\omega)$ gives $\Phi(E_0) = E_0$. □

[7]Intuitively, outside the exceptional set T, membership in E is unchanged by the involution Φ.

By (16), Φ maps $E_0 \cap \{H = b\}$ into $E_0 \cap \{H \neq b\}$ and vice versa; being an involution, it is a bijection between these two sets. Hence

$$\big|E_0 \cap \{H = b\}\big| = \big|E_0 \cap \{H \neq b\}\big|. \tag{19}$$

The two sets partition E_0, so each has exactly $\frac{1}{2}|E_0|$ elements.

Since $E_0 \cap \{H \neq b\} \subseteq B_b$,

$$|B_b| \;\geq\; \big|E_0 \cap \{H \neq b\}\big| \;=\; \tfrac{1}{2}|E_0| \;\geq\; \tfrac{1}{2}\big(|E| - |T|\big).$$

Dividing by $|\Omega|$ gives the probabilistic form of (18). □

Remark 17. As before, the name is explained by a disjunctive reformulation: in the setting of Lemma 15, for every $\rho > 0$ either

$$\mathbf{Pr}(E) \leq \mathbf{Pr}(T) + \rho \qquad \text{or} \qquad \mathbf{Pr}(B_b) > \tfrac{\rho}{2}$$

That is, intuitively, either the basic event E is rare, or a random sample ω from E is often "wrong" (i.e. $H(\omega)$ differs from a target value $b \in \{0,1\}$). It is, however, the inequality (18) that we apply.

3.2. **The spacing of input lengths.** A standard random-oracle construction assigns a hard target bit to every input length, schematically

$$L^A_{\mathrm{std}} = \left\{1^n : F_n(A \restriction \{0,1\}^n) = 1\right\}. \tag{20}$$

For ordinary separations, this strategy is often enough to force a disagreement, hence separation. Immunity requires more: whenever a machine continues to accept designated inputs with nonsummable conditional probability, one must force a false positive.

On input 1^n, a polynomial-time oracle machine may query strings whose lengths are as large as $q(n)$ for some polynomial q. Revealing enough of the oracle to determine its behavior at length n can therefore reveal the target blocks for many larger input lengths.

We avoid this problem by using only the lengths

$$t_j = 2^{j!}, \qquad j \geq 1. \tag{21}$$

They satisfy

$$t_{j+1} = t_j^{j+1}. \tag{22}$$

The target languages constructed below will be subsets of $T = \{1^{t_j} : j \geq 1\}$. This spacing has two properties:

Lemma 18. *The unary language T is decidable in deterministic polynomial time.*

Proof. On input 1^n, first test whether n is a power of two. If not, reject. If $n = 2^r$, compute $1!, 2!, 3!, \ldots$ until the first value that is at least r, and accept exactly when one of these values equals r. All integers involved have $O(\log n)$ bits, and even an implementation that scans the unary input runs in time polynomial in n. □

Lemma 19. *For every polynomial q, there is j_0 such that for all $j \geq j_0$*

$$q(t_j) < t_{j+1} \tag{23}$$

Proof. Choose $c \geq 1$ and n_0 such that $q(n) \leq n^c$ for all $n \geq n_0$. Choose j_1 such that $t_j \geq n_0$ for $j \geq j_1$. If $j \geq \max\{c, j_1\}$, then $q(t_j) \leq t_j^c < t_j^{j+1} = t_{j+1}$. □

The inequality $q(t_j) < t_{j+1}$ enables the two-step exposure approach used later in the proofs: first reveal every relevant oracle coordinate except the current block, and then reveal the current block.

4. Preliminaries

4.1. **Notions from Computational Complexity.** We will assume familiarity with standard notions in complexity theory as presented e.g. in [2], including finite alphabets, (oracle) Turing machines, languages, complexity classes such as $\mathsf{P}, \mathsf{NP}, \Sigma_k^P, \Pi_k^P, \Delta_k^P, \mathsf{PH}, \mathsf{PSPACE}, \mathsf{IP}, \mathsf{PP}, \oplus\mathsf{P}$. For an oracle A, the corresponding relativized classes are denoted by using A as a superscript (e.g. $\Sigma_k^{\mathsf{P},A}$, $\Pi_k^{\mathsf{P},A}$, and $\Delta_k^{\mathsf{P},A}$).

We fix the following acceptance convention for PP: A nondeterministic polynomial-time oracle machine M *accepts x relative to A* if *at least as many computation paths of $M^A(x)$ accept as reject*; $L(M^A)$ is the set of such x, and PP^A is the class of languages of this form. The convention is harmless: if M instead uses the strict rule, that more paths accept than reject, replace M by the machine that first makes one nondeterministic binary choice, on one branch simulating M and on the other branch rejecting outright after a single step. This adds one rejecting path, so ties under the new rule correspond to strict majorities under the old one, and the two machines accept the same language relative to every oracle, since the added branch makes no oracle query.

We will use notation $\mathrm{bin}_\ell(k)$ to denote the ℓ-bit binary representation of an integer k, for $0 \le k < 2^\ell$.

Definition 20. Let $\Omega = \{0,1\}^{\{0,1\}^*}$ be the space of all Boolean oracles. We equip Ω with the product probability measure under which the coordinates $A(z)$, $z \in \{0,1\}^*$, are independent unbiased bits. A statement holds *with respect to a random oracle* or *almost surely* if it holds for a set of oracles of measure one.

Definition 21. Let $\mathcal{C}$ be a class of languages. A language L is *$\mathcal{C}$-immune* if

(i) L is infinite, and
(ii) every language $S \in \mathcal{C}$ satisfying $S \subseteq L$ is finite.

A language L is *$\mathcal{C}$-biimmune* if both L and its complement are $\mathcal{C}$-immune.

To prove Theorem 6 we need *interactive proof systems* [23, 24, 35, 44]. We review in the sequel only the specific concepts we will need:

Definition 22. A *(clocked probabilistic oracle) verifier* is a probabilistic oracle Turing machine V together with an explicit polynomial clock. On an input of length n, the clock bounds the running time, the number of oracle queries, every oracle-query length, every message length, and the total number of rounds by a polynomial in n.

A *prover* is an arbitrary, computationally unbounded party interacting with V; a *prover strategy* P is a function mapping each partial transcript to the next prover message, and we allow P's choice to depend on the input x and on the oracle A.

The following approach to defining acceptance of a language agrees with the approach in [15] (see e.g. Section 3.1):

Definition 23. Fix a verifier V such as in Definition 22. For an oracle A and input x, define

$$\mathrm{val}_V(A,x) = \max_P \mathbf{Pr}[V^A \text{ accepts } x \text{ after interacting with } P], \tag{24}$$

where the probability is over the private randomness of V and the maximum is over all prover strategies.

Observation 24. *The maximum from equation (24) exists (see* [15]*, from where this convention is imported).*

Definition 25. A verifier V^A is *an interactive proof verifier for a language* S if, for all words x, it holds that

$$x \in S \Longrightarrow \mathrm{val}_V(A,x) > \frac{2}{3}, \tag{25}$$

$$x \notin S \Longrightarrow \mathrm{val}_V(A,x) < \frac{1}{3}. \tag{26}$$

The constants $\frac{2}{3}$ and $\frac{1}{3}$ are a normalization: for any oracle A, a verifier with completeness c and soundness s satisfying $c - s \geq 1/p(n)$ for a polynomial p can be amplified, by polynomially many independent repetitions against the same oracle followed by an acceptance threshold strictly between s and c, for example $(c+s)/2$, into a clocked verifier meeting (25) and (26), and recognizing the same language. We may therefore restrict attention throughout to verifiers of this form.

Definition 26. A *polynomial threshold function* is a real function $f : \{0,1\}^N \to \{0,1\}$ having the form $f(x) = sign(q(x))$ where $q(\cdot)$ is a real polynomial and *sign* is the function

$$sign(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{otherwise.} \end{cases}$$

Finally, we assume knowledge of the approach of Rossman, Servedio and Tan (for a readable introduction see [42]).

For integers $d, m \geq 2$, a depth-d bottom fan-in m *Sipser formula* is a read-once monotone formula

$$\mathrm{Sipser}_{d,m} : \{0,1\}^{N_{RST}(d,m)} \longrightarrow \{0,1\} \tag{27}$$

consisting of d alternating layers of unbounded-fan-in AND and OR gates. The bottom gates are AND gates of fan-in m. All gates at a given layer have the same type and the same fan-in, consecutive layers have opposite types, and every input variable labels exactly one leaf. The fan-ins are chosen as follows (indexed from the root):

$$w_0, \quad \underbrace{w, \ldots, w}_{d-2 \text{ intermediate levels}}, m. \tag{28}$$

We next review the parameters chosen in [41] for the function $Sipser_{d,m}$[8]. Indeed, put

$$\rho = 2^{-m}, \qquad q = \sqrt{\rho} = 2^{-m/2}. \tag{29}$$

The intermediate fan-in, as chosen in [41], is

$$w = w(m) = \left\lfloor \frac{m2^m}{\log_2(e)} \right\rfloor = \lfloor m2^m \ln 2 \rfloor . \tag{30}$$

RST define

$$\lambda = \frac{(\log_2 w)^{3/2}}{w^{5/4}}, \tag{31}$$

and recursively define

$$t_{d-1} = \frac{\rho - \lambda}{q}, \tag{32}$$

$$t_{r-1} = \frac{(1-t_r)^{qw} - \lambda}{q}, \qquad r = d-1, d-2, \ldots, 2. \tag{33}$$

Let

$$D_m = -q \ln(1 - t_1), \qquad R_m = \frac{\ln 2}{D_m}. \tag{34}$$

[8]We refer to the full version of [41]. More precisely, the definitions of ρ and w below are respectively equations (2) and (3) of [41, Section 6, p. 15], while the definitions of q and λ are given in equation (7) of [41, Section 7.1, p. 16]. The journal version [27] uses slightly different values of the parameters.

The top fan-in is chosen in [41] as

$$w_0 = \min\left\{s \in \mathbb{N} : (1-t_1)^{qs} \le \frac{1}{2}\right\} = \lceil R_m \rceil \tag{35}$$

The number of variables of the boolean function $Sipser_{d,m}$ is thus

$$N = N_{RST}(d,m) = w_0 \cdot w^{d-2} \cdot m. \tag{36}$$

For fixed $d \ge 2$, the RST choice of fan-ins satisfies

$$\log_2 N_{RST}(d,m) = \Theta_d(m). \tag{37}$$

In particular, there are constants $c_d, C_d > 0$ such that, for all sufficiently large m,

$$2^{c_d \cdot m} \le N_{RST}(d,m) \le 2^{C_d \cdot m}. \tag{38}$$

4.2. **Notions from Probability Theory.** We assume familiarity with introductory probability theory, including probability spaces, σ-fields, the classical Borel-Cantelli lemmas, at the level of, say, [17]. To make the paper self-contained, we recall the following:

Lemma 27 (Second Borel-Cantelli lemma)**.** *Let $(D_n)_{n\ge1}$ be a sequence of mutually independent events in a probability space. If*

$$\sum_{n=1}^{\infty} \mathbf{Pr}(D_n) = \infty, \tag{39}$$

then

$$\mathbf{Pr}(D_n \text{ infinitely often}) = 1. \tag{40}$$

We will also need a standard (e.g. [14], Chapter 5, pp. 95–102) conditional variant of Borel-Cantelli:

Definition 28. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. A *filtration* is an increasing sequence of sub-σ-fields

$$\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}. \tag{41}$$

A sequence of events $(E_n)_{n\ge1}$ is said to be *adapted* to the filtration $(\mathcal{F}_n)_{n\ge0}$ if $E_n \in \mathcal{F}_n$ for every $n \ge 1$. In this case, the conditional probability $\mathbf{Pr}(E_n \mid \mathcal{F}_{n-1})$ can be interpreted as the probability that E_n occurs given all the information revealed before stage n.

Lemma 29 (Lévy's extension of the Borel-Cantelli lemmas)**.** *Let $(\mathcal{H}_n)_{n\ge0}$ be a filtration, and let $D_n \in \mathcal{H}_n$ for every $n \ge 1$. Then*

$$\mathbf{Pr}\left(\{D_n \text{ i.o.}\} \triangle \left\{\sum_{n=1}^{\infty} \mathbf{Pr}(D_n \mid \mathcal{H}_{n-1}) = \infty\right\}\right) = 0,$$

where $\triangle$ denotes symmetric difference. In particular, almost surely, if the conditional-probability sum is finite, only finitely many D_n occur; if it is infinite, infinitely many occur.

4.3. **Notions from Cryptography.** We use some notions from cryptography as well [20, 33], specifically one-way permutations. There are several inequivalent such notions in the literature; see e.g. Goldreich's discussion [21]. We therefore state explicitly the specific type we need.

We first recall the following

Definition 30. A function $\nu : \mathbb{N} \to \mathbb{R}_{\ge0}$ is *negligible* (e.g. [33]) if, for every polynomial p, there exists m_0 such that

$$\nu(m) < \frac{1}{p(m)}$$

for every $m \ge m_0$.

Definition 31. A *uniform one-way permutation* is a deterministic polynomial-time algorithm F such that, for every $m \geq 1$, $f_m(x) := F(1^m, x)$ is a permutation $f_m : \{0,1\}^m \longrightarrow \{0,1\}^m$ and for every probabilistic polynomial-time algorithm $I(\cdot,\cdot;\cdot)$ there is a negligible function ν_I such that

$$\mathbf{Pr}_{x \leftarrow U_m, r}\left[f_m(I(1^m, f_m(x); r)) = f_m(x)\right] \leq \nu_I(m). \tag{42}$$

Here U_m denotes the uniform distribution on $\{0,1\}^m$ and r denotes the internal randomness of I. To allow for the possibility that I fails (denoted $\perp$), define $f_m(\perp) = \perp$.

Since f_m is a permutation, $f_m(x)$ is uniform when x is uniform. Thus (42) is equivalently the requirement

$$\mathbf{Pr}_{y \leftarrow U_m, r}\left[f_m(I(1^m, y; r)) = y\right] \leq \nu_I(m). \tag{43}$$

Remark 32. Merely assuming that no deterministic polynomial-time algorithm computes the inverse everywhere seems insufficient for the proof. We need the inversion probability on a uniformly random image to be negligible.

Length-preserving one-to-one one-way functions have been constructed from standard number-theoretic assumptions such as RSA- and discrete-logarithm-type assumptions; see Goldreich, Levin, and Nisan [22]. Because a length-preserving injection from $\{0,1\}^m$ to itself is automatically surjective, such a function is a permutation at each length.

Remark 33. The hypothesis in Definition 31 is at least as strong as the worst-case complexity-theoretic existence of a one-to-one one-way function. It is also at least as strong (and probably stronger) than the hypothesis UP $\not\subseteq$ RP used by Ghosal et al (whose goal was to avoid one-way permutations). The latter follows already from suitable injective one-way functions [19].

4.4. **Hardness Results.** Our strong separation results exploit various hardness results from the literature (some of them tailored to our purpose). We review in the sequel the specific form of (some of) these hardness results.

First, the following correlation bound on parity is due to Håstad (e.g. [26]):

Lemma 34 (Average-case parity lower bound)**.** *For every fixed depth $d \geq 2$, there is a constant $k_d > 0$ such that the following holds. Let $C : \{0,1\}^N \to \{0,1\}$ be an unbounded-fan-in AND/OR/NOT circuit of depth at most d and size at most S. Then*

$$\left|\mathbf{Pr}_X[C(X) = \mathrm{PARITY}_N(X)] - \frac{1}{2}\right| \leq 2^{-k_d N/(\log_2(2S))^{d-1}}. \tag{44}$$

Here X is uniformly distributed on $\{0,1\}^N$.

We use the following consequence of the main voting-polynomial lower bound of [3]:

Lemma 35. *Let p be a real polynomial of degree at most d in N Boolean variables, and let $g(x) = sign[p(x)]$. If* PARITY_N *denotes parity on $\{0,1\}^N$, then*

$$\mathbf{Pr}_{x \in \{0,1\}^N}\left[g(x) \neq \mathrm{PARITY}_N(x)\right] \geq 2^{-N} \sum_{i=0}^{\lfloor (N-d-1)/2 \rfloor} \binom{N}{i}. \tag{45}$$

Consequently there is an absolute constant $C > 0$ such that, whenever $d < N$,

$$\mathbf{Pr}[g(x) \neq \mathrm{PARITY}_N(x)] \geq \frac{1}{2} - C\frac{d+1}{\sqrt{N}}. \tag{46}$$

Proof. The first assertion is the parity specialization of the ABFR voting-polynomial lower bound; parity has weak degree N. For the second assertion, compare the sum in (45) with the lower half of

the binomial distribution. Moving the cutoff from the center by $O(d+1)$ removes at most $O(d+1)$ central binomial coefficients. Since

$$2^{-N}\binom{N}{\lfloor N/2\rfloor} = O(N^{-1/2}),$$

(46) follows. □

Finally, we use the following hardness result from [41] (Theorem 7 from the full version of [41], reindexed by the bottom-width parameter m):

Definition 36. A Boolean circuit is in *normal form* if it is a layered alternating AND/OR circuit (in the sense of [41]) whose leaves are literals or the constants $0, 1$; in particular, NOT gates occur only on input literals. The root gate determines the alternation pattern. The size of a circuit is its number of gates.

Every circuit can be converted into normal form without increasing its depth, blowing up its size from S to $S^{O_d(1)} + O_d(N)$, by pushing negations to the inputs, merging consecutive gates of the same type and inserting unary dummy gates.

Remark 37. Constant leaves can be eliminated: Repeatedly replace a gate having an absorbing constant input (0 for AND, 1 for OR) by that constant, delete neutral constant inputs (1 for AND, 0 for OR), and treat a gate left without inputs as the corresponding constant. This only deletes gates, preserves layering, does not increase depth, and preserves the root type unless the root itself becomes constant. In the latter case, replace the circuit by a depth-2 constant gadget with the required root type: $\mathrm{OR}(\mathrm{AND}(x), \mathrm{AND}(\neg x)) \equiv 1$, $\mathrm{AND}(\mathrm{OR}(x, \neg x)) \equiv 1$, $\mathrm{AND}(\mathrm{OR}(x), \mathrm{OR}(\neg x)) \equiv 0$, $\mathrm{OR}(\mathrm{AND}(x, \neg x)) \equiv 0$. Hence every normal-form circuit C of depth at most $d \geq 2$ is equivalent to a constant-free normal-form circuit with the same root type, depth at most d, and size at most $\max(\mathrm{size}(C), 3)$. This is used in Corollary 39 below to extend Proposition 38 to normal-form circuits with constant leaves.

Proposition 38 ([41], Theorem 7)**.** *For every fixed $d \geq 2$ there is $b_d > 0$ such that, for all sufficiently large m, every constant-free normal-form circuit C of depth exactly d, with root gate opposite to that of $\mathrm{Sipser}_{d,m}$ and $\mathrm{size}(C) \leq 2^{N^{1/(6(d-1))}}$, satisfies*

$$\mathbf{Pr}_{X\in\{0,1\}^N}\left[C(X) = \mathrm{Sipser}_{d,m}(X)\right] \leq \tfrac{1}{2} + N^{-b_d}.$$

Corollary 39. *Let $a_d := 1/(7(d-1))$. For all sufficiently large m, the conclusion of Proposition 38 holds for every normal-form circuit C (constant leaves allowed) of depth at most d, with root gate opposite to that of $\mathrm{Sipser}_{d,m}$ and $\mathrm{size}(C) \leq 2^{N^{a_d}}$.*

Proof. By Remark 37, C is equivalent to a constant-free normal-form circuit C' with the same root type, depth at most d, and $\mathrm{size}(C') \leq \max(\mathrm{size}(C), 3)$. Pad C' to depth exactly d without changing its root type: add two unary gates above the root (a new root of the same type over a unary gate of the opposite type) as often as needed, and, if the parity of the remaining depth difference is odd, insert one unary gate of the opposite type between a deepest gate and one of its literal inputs. This adds at most d gates. Since $2^{N^{a_d}} + d + 3 \leq 2^{N^{1/(6(d-1))}}$ for all sufficiently large m, Proposition 38 applies to the padded circuit. □

The proof of Theorem 1 below will employ this bound to derive a specially-tailored hardness lemma. We point to Lemma 74 below for details.

5. Proofs of Our Other Results

In this section we prove the results stated in Section 2, except for the main theorem, which gets a separate section. We do not follow the order of their presentation. Instead, we begin with the case

in which the method from Section 3 appears in its simplest form, and then treat the cases requiring additional ingredients.

5.1. **Proof of Theorem 2.** We begin with Theorem 2. This proof gives the cleanest illustration of the rare-or-wrong argument and of the delayed exposure of the random oracle.

For each $j \geq 1$, let $V_j = \{0,1\}^{t_j}$. The sets V_j are pairwise disjoint blocks of oracle coordinates. Define

$$Y_j^A = \bigoplus_{y \in V_j} A(y), \tag{47}$$

and let

$$L^A = \left\{ 1^{t_j} : Y_j^A = 1 \right\}. \tag{48}$$

Thus L^A is supported on the sparse unary set T introduced in Section 3.

Lemma 40. *For every oracle A,*

$$L^A \in \oplus \mathsf{P}^A \subseteq \mathsf{PSPACE}^A.$$

Proof. On input x, first verify deterministically that $x = 1^n$ and that $1^n \in T$. If either test fails, reject. Otherwise, use exactly n binary nondeterministic choices to guess $y \in \{0,1\}^n$, query $A(y)$, and accept that computation path exactly when $A(y) = 1$. There is one computation path for each $y \in \{0,1\}^n$. Hence the parity of the number of accepting paths is

$$\bigoplus_{y \in \{0,1\}^n} A(y),$$

which is the membership bit in (48). Therefore $L^A \in \oplus \mathsf{P}^A$.

□

Lemma 41. *With probability one over a random oracle A, the language L^A is infinite.*

Proof. For every j, the random variable Y_j^A is unbiased, since it is the parity of a nonempty family of independent unbiased oracle bits. Moreover, $Y_1^A, Y_2^A, \ldots$ are mutually independent because the blocks $V_1, V_2, \ldots$ are disjoint. Consequently,

$$\mathbf{Pr}_A(Y_j^A = 1) = \frac{1}{2}$$

for every j. The second Borel–Cantelli lemma therefore gives

$$\mathbf{Pr}_A(Y_j^A = 1 \text{ for infinitely many } j) = 1.$$

By (48), this is precisely the assertion that L^A is infinite almost surely. □

Lemma 42 (Summability of the parity-correlation errors)**.** *Fix constants $c \geq 1$, $d \geq 2$, and $\kappa > 0$. Then*

$$\sum_{j \geq 1} 2^{-\kappa \cdot 2^{t_j}/(1+t_j^c)^{d-1}} < \infty.$$

Proof. Set $K = c(d-1)$. Since $t_j \geq 1$, $(1+t_j^c)^{d-1} \leq 2^{d-1} t_j^K$. Thus, with $\kappa' = \kappa/2^{d-1} > 0$, $2^{-\kappa \cdot 2^{t_j}/(1+t_j^c)^{d-1}} \leq 2^{-\kappa' \cdot 2^{t_j}/t_j^K}$. The quantity $2^m/m^K$ eventually dominates m/κ'. Hence, for all sufficiently large j,

$$\kappa' \cdot \frac{2^{t_j}}{t_j^K} \geq t_j, \tag{49}$$

and the j-th summand is at most $2^{-t_j} \leq 2^{-j}$. The tail is therefore dominated by a geometric series. □

Let $(M_e)_{e\geq 1}$ enumerate all syntactically clocked alternating oracle machines having a constant number of alternations. For every oracle A, the languages $L(M_e^A)$ range over PH^A. It remains to prove the following machine-wise assertion.

Lemma 43. *For every $e \geq 1$,*

$$\mathbf{Pr}_A\left[L(M_e^A) \not\subseteq L^A \text{ or } L(M_e^A) \text{ is finite}\right] = 1. \tag{50}$$

Proof. Fix e and write $M = M_e$. Since M belongs to a fixed level of the polynomial hierarchy, there are constants $d \geq 2$ and $c \geq 1$ such that, on every input x of length n, the acceptance predicate of $M^A(x)$ can be represented by an unbounded-fan-in Boolean circuit C_x in the relevant oracle variables satisfying

$$\operatorname{depth}(C_x) \leq d, \qquad \operatorname{size}(C_x) \leq 2^{n^c}. \tag{51}$$

There is also a polynomial q such that every oracle variable occurring in C_x corresponds to a query string of length at most $q(n)$.

One way to obtain this representation is to start from a quantified characterization of the language decided by M. Expand each polynomially bounded existential quantifier into an unbounded-fan-in OR gate and each universal quantifier into an unbounded-fan-in AND gate. The remaining deterministic polynomial-time oracle computation can be expanded into its oracle decision tree and expressed as a DNF or CNF. Because M is fixed, this adds only a constant number of layers and produces a circuit of size $2^{n^{O(1)}}$; increasing c if necessary gives (51).

By the spacing property of (t_j) established in Section 3, there is j_0 such that

$$q(t_j) < t_{j+1} \tag{52}$$

for every $j \geq j_0$. The finitely many indices below j_0 do not affect summability or any infinitely-often event, so all subsequent definitions and arguments are restricted to $j \geq j_0$.

Define

$$\mathcal{F}_j = \sigma\big(A(z) : |z| < t_{j+1}\big), \tag{53}$$

$$\mathcal{G}_j = \sigma\big(A(z) : |z| < t_{j+1},\ z \notin V_j\big). \tag{54}$$

Then

$$\mathcal{F}_{j-1} \subseteq \mathcal{G}_j \subseteq \mathcal{F}_j. \tag{55}$$

Let

$$E_j = \{M^A(1^{t_j}) \text{ accepts}\}. \tag{56}$$

By (52), every query made by M on 1^{t_j} has length less than t_{j+1}, and hence $E_j \in \mathcal{F}_j$.

Condition on an assignment ω to the oracle coordinates exposed by $\mathcal{G}_j$. The only still-unexposed coordinates of length less than t_{j+1} are those in V_j. Therefore $A \restriction V_j$ remains uniform on $\{0,1\}^{N_j}$, where

$$N_j = |V_j| = 2^{t_j}. \tag{57}$$

Under this conditioning, the acceptance predicate of $M^A(1^{t_j})$ becomes a fixed circuit $C_{j,\omega} : \{0,1\}^{V_j} \longrightarrow \{0,1\}$ of depth at most d and size at most $2^{t_j^c}$. Restricting the other oracle variables cannot increase either parameter. The target bit Y_j^A remains the parity of all N_j fresh variables.

Lemma 34, applied on every atom of $\mathcal{G}_j$, gives

$$a_j := \Pr(\mathbf{1}_{E_j} = Y_j^A \mid \mathcal{G}_j) \leq \frac{1}{2} + \eta_j, \tag{58}$$

where, after decreasing the positive constant k_d in the exponent, if necessary, to a lower value κ_d we may take

$$\eta_j = 2^{-\kappa_d 2^{t_j}/(1+t_j^c)^{d-1}} \tag{59}$$

for a constant $\kappa_d > 0$ depending only on d.

Lemma 42 shows that

$$\sum_{j \geq j_0} \eta_j < \infty. \tag{60}$$

Define the $\mathcal{G}_j$-measurable conditional probabilities

$$p_j = \Pr(E_j \mid \mathcal{G}_j), \tag{61}$$

$$b_j = \Pr(E_j \cap \{Y_j^A = 0\} \mid \mathcal{G}_j). \tag{62}$$

Since $\mathcal{G}_j$ is generated by finitely many oracle coordinates, we may work atom by atom. For each assignment ω exposed by $\mathcal{G}_j$, consider the conditional probability space obtained by fixing that assignment. In Lemma 10, take $H = Y_j^A$, $C = \mathbf{1}_{E_j}$, $\delta = 0$, $\varepsilon = \eta_j$. The parity bit is exactly balanced in this conditional space, so

$$\left|\mathbf{Pr}(H = 1) - \frac{1}{2}\right| = 0 \leq \eta_j, \tag{63}$$

and (58) gives $\Pr(C = H) \leq \frac{1}{2} + \eta_j$. Lemma 10 therefore yields, on every atom of $\mathcal{G}_j$,

$$b_j \geq \frac{1}{2} p_j - \frac{1}{2}\eta_j. \tag{64}$$

Let

$$B_j = E_j \cap \{Y_j^A = 0\}. \tag{65}$$

The event B_j is a false positive: M accepts 1^{t_j} although $1^{t_j} \notin L^A$. Since both E_j and Y_j^A are $\mathcal{F}_j$-measurable, $B_j \in \mathcal{F}_j$, and

$$\mathbf{Pr}(B_j \mid \mathcal{G}_j) = b_j. \tag{66}$$

We now apply Lemma 29 to an explicit interleaving of the two sigma-fields. Define a filtration $(\mathcal{H}_n)$ by

$$\mathcal{H}_{2j-1} = \mathcal{G}_j, \qquad \mathcal{H}_{2j} = \mathcal{F}_j \qquad (j \geq j_0), \tag{67}$$

and fill the finitely many earlier terms with $\mathcal{F}_{j_0-1}$. Set all corresponding earlier events to $\varnothing$. The inclusions in (55) show that this is indeed a filtration. For the acceptance events, set

$$D_{2j-1}^E = \varnothing, \qquad D_{2j}^E = E_j. \tag{68}$$

Then

$$\Pr(D_{2j}^E \mid \mathcal{H}_{2j-1}) = p_j, \tag{69}$$

so Lemma 29 gives

$$E_j \text{ occurs infinitely often} \iff \sum_{j \geq j_0} p_j = \infty \tag{70}$$

almost surely. Applying the same construction with $D_{2j}^B = B_j$ and $D_{2j-1}^B = \varnothing$ gives

$$B_j \text{ occurs infinitely often} \iff \sum_{j \geq j_0} b_j = \infty \tag{71}$$

almost surely.

Work now on the probability-one event on which both equivalences hold.

- If $\sum_j p_j < \infty$, then (70) implies that M accepts only finitely many strings of the form 1^{t_j}. Since L^A contains no strings outside this support, the implication
$$L(M^A) \subseteq L^A \quad \Longrightarrow \quad L(M^A) \text{ is finite}$$
follows.
- If instead $\sum_j p_j = \infty$, then (64) and Lemma 42 imply

$$\sum_j b_j \geq \frac{1}{2}\sum_j p_j - \frac{1}{2}\sum_j \eta_j = \infty. \tag{72}$$

By (71), the false-positive events B_j occur infinitely often. In particular, M accepts a string outside L^A, and hence
$$L(M^A) \not\subseteq L^A.$$

The two cases prove (50). □

There are countably many machines in the enumeration. Intersecting the probability-one events from Lemma 43 over all e, and then intersecting with the probability-one event from Lemma 41, shows that, almost surely, L^A is infinite and has no infinite PH^A subset. Thus L^A is PH^A-immune almost surely. Lemma 40 now completes the proof of Theorem 2.

5.2. **Proof of Theorem 4.** The intuition behind the proof of Theorem 4 is that a PP machine's decision on the fresh oracle block is a low-degree polynomial threshold function; by Lemma 35, such a function cannot predict parity, so it must either accept too rarely to define an infinite sublanguage or make false-positive errors infinitely often. The Lévy extension of the Borel-Cantelli lemma converts this intuition (via an application of Lemma 10) into a proof of immunity.

Indeed, fix an enumeration $(M_e)_{e\geq 1}$ of polynomial-time oracle nondeterministic machines. We use the same language L^A as in the previous proof.

Lemma 44. *For every $e \geq 1$*

$$\mathbf{Pr}_A\Big[L(M_e^A) \subseteq L^A \text{ and } L(M_e^A) \text{ is infinite}\Big] = 0. \tag{73}$$

Proof. There is a polynomial q such that, on inputs of length n, every computation path runs for at most $q(n)$ steps and hence asks at most $q(n)$ oracle queries, all of length at most $q(n)$. Without loss of generality we can choose q so that it is monotonically increasing and $q(n) \geq n$.

By the spacing property there exists $j_0 \geq 1$ such that for $j \geq j_0$ we have $q(t_j) < t_{j+1}$. Put

$$X_j = \{M_e^A(1^{t_j}) \text{ accepts in the } \mathsf{PP} \text{ sense}\} \tag{74}$$

and let

$$Y_j = \{1^{t_j} \in L^A\}, \qquad B_j = X_j \cap Y_j^c. \tag{75}$$

Thus B_j is the event that M_e makes a false positive with respect to L^A at the j-th special length.

Let $V_j = \{0,1\}^{t_j}$ be the j-th block of oracle coordinates and write

$$Z_j = (A(z) : z \in V_j) \in \{0,1\}^{N_j}, \qquad N_j = 2^{t_j}, \tag{76}$$

so that Y_j is the event that the parity of Z_j equals 1. Define the two σ-fields

$$\mathcal{F}_j = \sigma\big(A(z) : |z| < t_{j+1}\big), \tag{77}$$

$$\mathcal{G}_j = \sigma\big(A(z) : |z| < t_{j+1},\ z \notin V_j\big). \tag{78}$$

Since V_j consists of strings of length exactly t_j, while $\mathcal{F}_{j-1}$ exposes only strings of length $< t_j$, we have

$$\mathcal{F}_{j-1} \subseteq \mathcal{G}_j \subseteq \mathcal{F}_j \subseteq \mathcal{G}_{j+1} \subseteq \cdots, \tag{79}$$

an increasing interleaved filtration. By the spacing property every oracle query made by M_e on 1^{t_j} has length at most $q(t_j) < t_{j+1}$; hence $X_j \in \mathcal{F}_j$, and $Y_j \in \mathcal{F}_j$ because Y_j is determined by Z_j. Consequently $B_j = X_j \cap Y_j^c \in \mathcal{F}_j$ as well.

The point of using $\mathcal{G}_j$ rather than $\sigma(A(z) : |z| \le q(t_j))$ is the following. Fix an atom ω of $\mathcal{G}_j$, that is, an assignment to all oracle coordinates of length $< t_{j+1}$ outside V_j. Every coordinate that $M_e^A(1^{t_j})$ can query is then either frozen by ω or belongs to V_j: there are no further unexposed coordinates that could affect the computation. Moreover, conditionally on ω the vector Z_j is uniformly distributed on $\{0,1\}^{N_j}$. We may therefore work atom by atom, and on each atom the acceptance predicate is a function of the fresh block alone. We record the standard representation of that function.

We need a standard representation lemma:

Lemma 45. *Fix $j \ge j_0$ and an atom ω of $\mathcal{G}_j$. Viewed as a function of the fresh block Z_j, the predicate X_j is the sign of a real multilinear polynomial $p_{j,\omega}(Z_j)$ of degree at most $d_j = q(t_j)$.*

Proof. Consider a computation path of M_e on 1^{t_j}. Once its internal nondeterministic choices are fixed, the condition that this path is followed can be expressed as a product of literals corresponding to the answers to its adaptive oracle queries.

Since at most $q(t_j)$ oracle queries occur on a path, the resulting path-indicator polynomial has degree at most $q(t_j)$. (A path may repeat a query; the corresponding literal then occurs with a power, which may be reduced using $z^2 = z$ on $\{0,1\}$ for each of the N_j potential variables without changing the function or raising the degree.)

Sum these indicators over accepting paths, and subtract the corresponding sum over rejecting paths. The resulting gap polynomial has degree at most $q(t_j)$: summing/subtracting over the exponentially many paths does not increase the degree: the gap polynomial has degree at most $q(t_j)$ regardless of how many paths there are.

On the other hand the *sign* of the gap polynomial determines whether the accepting-path count is at least the rejecting-path count. □

Fix an atom ω of $\mathcal{G}_j$. By Lemma 45, X_j is represented there by the sign of a polynomial of degree at most $d_j = q(t_j)$ in the N_j fresh bits, while Y_j is the parity of those same bits. Lemma 35 therefore gives

$$\Pr(X_j \triangle Y_j \mid \mathcal{G}_j) \;\ge\; \frac{1}{2} - \varepsilon_j, \qquad \varepsilon_j = C\,\frac{q(t_j)+1}{2^{t_j/2}}, \tag{80}$$

on every atom of $\mathcal{G}_j$. Applying Lemma 10 on that atom with $H = \mathbf{1}_{Y_j}$, $C = \mathbf{1}_{X_j}$, $\delta = 0$ (the parity of a nonempty block of fresh unbiased bits is exactly balanced) and $\varepsilon = \varepsilon_j$, we obtain

$$\Pr(B_j \mid \mathcal{G}_j) \;\ge\; \frac{\Pr(X_j \mid \mathcal{G}_j) - \varepsilon_j}{2}. \tag{81}$$

Since the atom was arbitrary and $\mathcal{G}_j$ is generated by finitely many oracle coordinates, (81) holds everywhere. Ignoring the finitely many indices $j \le j_0$, put

$$Q_j = \Pr(X_j \mid \mathcal{G}_j), \qquad R_j = \Pr(B_j \mid \mathcal{G}_j), \tag{82}$$

so that (81) reads $R_j \;\ge\; (Q_j - \varepsilon_j)/2$.

Lemma 46. *We have:*

$$\sum_j \varepsilon_j < \infty, \tag{83}$$

Proof. Since q is a polynomial, for all sufficiently large j, $C(q(t_j)+1) \le 2^{t_j/4}$, so $\epsilon_j \le 2^{-t_j/4} \le 2^{-j/4}$. Hence $\sum_j \varepsilon_j < \infty$. □

As in the proof of Theorem 2, we feed (79) to Lévy's extension of the Borel-Cantelli lemmas. Define a filtration $(\mathcal{H}_n)$ by

$$\mathcal{H}_{2j-1} = \mathcal{G}_j, \qquad \mathcal{H}_{2j} = \mathcal{F}_j \qquad (j \geq j_0), \tag{84}$$

filling the finitely many earlier terms with $\mathcal{F}_{j_0-1}$ and the corresponding events with $\varnothing$; by (79) this is indeed a filtration. Setting $D^X_{2j-1} = \varnothing$, $D^X_{2j} = X_j$ gives $\Pr(D^X_{2j} \mid \mathcal{H}_{2j-1}) = Q_j$, and likewise $D^B_{2j-1} = \varnothing$, $D^B_{2j} = B_j$ gives $\Pr(D^B_{2j} \mid \mathcal{H}_{2j-1}) = R_j$. Lemma 29 therefore yields, almost surely,

$$X_j \text{ i.o.} \iff \sum_j Q_j = \infty, \qquad B_j \text{ i.o.} \iff \sum_j R_j = \infty, \tag{85}$$

and we may complete the proof by the same two-case analysis as in the proof of Theorem 2:

- Case 1: $\sum_j Q_j < \infty$.
  Then Lemma 29, applied to X_j, implies that almost surely X_j occurs only finitely often. Thus M_e^A accepts only finitely many of the candidate strings 1^{t_j}.
- Case 2: $\sum_j Q_j = \infty$.
  Since $R_j \geq (Q_j - \varepsilon_j)/2$, while $\sum_j Q_j = \infty$ and $\sum_j \varepsilon_j < \infty$ by Lemma 46 it follows that $\sum_j R_j = \infty$. Applying Lemma 29 to B_j shows that B_j occurs infinitely often almost surely. In particular, $L(M_e^A)$ cannot be a subset of L^A.

□

There are only countably many polynomial-time oracle machines (including all polynomial clocks), so the union of the exceptional null sets in (73) is null. Therefore, with probability one, no infinite PP^A language is a subset of L^A, in other words (we already proved that L^A is almost surely infinite) L^A is PP^A-immune.

Since (we showed already that) $L^A \in \oplus\mathsf{P}^A \subseteq \mathsf{PSPACE}^A$, the theorem follows.

5.3. **Proof of Theorem 6.** The construction of the proof partitions a portion of the oracle at each designated length into *exponentially many disjoint segments*, and declares the corresponding unary string to belong to L^A exactly when none of these segments is completely filled with ones; this makes L^A a coNP^A language and leaves it infinite almost surely. Now fix an interactive-proof verifier V. If V accepts a positive instance with probability greater than $2/3$, then, since every computation examines only polynomially many of the exponentially many segments, most segments are queried with probability less than $1/3$. Completing any such segment produces a negative instance that V still accepts with probability greater than $1/3$.

Double counting these modifications as in Lemma 12 allows converting this dichotomy (via the conditional Borel-Cantelli lemma) into the conclusion of IP^A-immunity.

Indeed, fix an even integer $n \geq 4$ and define

$$s(n) = \frac{n}{2}, \qquad M(n) = 2^{s(n)}. \tag{86}$$

Order $\{0,1\}^n$ lexicographically, equivalently by interpreting n-bit strings as integers in $\{0, \ldots, 2^n - 1\}$. For $i \in \{0, \ldots, M(n) - 1\}$, define the segment

$$S_i(n) = \{\mathrm{bin}_n(i\, s(n) + r) : 0 \leq r < s(n)\}\,. \tag{87}$$

Since

$$M(n)s(n) = 2^{n/2}\frac{n}{2} \leq 2^n \tag{88}$$

for all sufficiently large n, the segments are well defined, pairwise disjoint, and contained in $\{0,1\}^n$.

Let

$$W(n) = \bigcup_{i=0}^{M(n)-1} S_i(n). \tag{89}$$

The segment addressing is uniform: given n, i, and r, the string $\mathrm{bin}_n(i\,s(n)+r)$ is computable in time polynomial in n.

For an oracle A, define

$$L^A = \left\{ 1^{t_j} : j \geq 2, \forall i < M(t_j),\ S_i(t_j) \not\subseteq A \right\}. \tag{90}$$

Lemma 47. *For every oracle A,*

$$L^A \in \mathsf{coNP}^A. \tag{91}$$

Proof. We prove instead that $\overline{L^A}$, the complement of the language, is in NP^A.

Indeed, on an input x, first test whether $x = 1^n$ and whether $1^n \in T_{\geq 2}$. If not, accept, since all off-support strings lie in $\overline{L^A}$.

Suppose $n = t_j$. A nondeterministic machine guesses an index $i < M(n) = 2^{n/2}$. The index has $n/2$ bits. It then queries A on all strings in $S_i(n)$ and accepts if every answer is 1. Thus the machine accepts exactly when some segment is fully contained in A, i.e. exactly when $1^n \notin L^A$. □

Lemma 48. *We have*

$$\lim_{1^n \in T, n \to \infty} \mathbf{Pr}_A[1^n \in L^A] = e^{-1}. \tag{92}$$

Proof. For a fixed support length $1^n \in T$, a segment $S_i(n)$ is completely contained in a random oracle with probability

$$2^{-s(n)} = \frac{1}{M(n)}. \tag{93}$$

Since the segments are disjoint, these events are mutually independent over i. Hence

$$\mathbf{Pr}_A[1^n \in L^A] = \left(1 - \frac{1}{M(n)}\right)^{M(n)} \longrightarrow e^{-1} \tag{94}$$

as $n \to \infty$. □

Lemma 49. *With probability one, L^A is infinite.*

Proof. Same proof approach, based on the second Borel-Cantelli lemma, as for Lemma 41. □

Fix a clocked verifier V. Let $q(n)$ be a polynomial that simultaneously bounds:

(i) the number of oracle queries made by V on inputs of length n, and
(ii) the length of every oracle query made by V on inputs of length n.

Fix a sufficiently large support length n and abbreviate

$$s = s(n) = \frac{n}{2}, \qquad M = M(n) = 2^s, \qquad W = W(n). \tag{95}$$

Fix an arbitrary assignment ω to all oracle coordinates relevant to the computation except the coordinates in W. For an assignment $\beta \in \{0,1\}^W$, let $A_{\omega,\beta}$ denote the resulting oracle on the finite query cone, and define

$$v_\omega(\beta) = \mathrm{val}_V(A_{\omega,\beta}, 1^n). \tag{96}$$

Call β *still unfilling* if no segment $S_i(n)$ is entirely filled with 1s. Call β *uniquely filling* if exactly one segment is entirely filled with 1s. Define

$$\mathcal{E}_\omega = \left\{ \beta : \beta \text{ is still unfilling and } v_\omega(\beta) > \frac{2}{3} \right\}, \tag{97}$$

$$\mathcal{B}_\omega = \left\{ \gamma : \gamma \text{ is uniquely filling and } v_\omega(\gamma) > \frac{1}{3} \right\}. \tag{98}$$

Intuitively, we will exploit the fact that the verifier can afford to watch only polynomially many of the $2^{n/2}$ segments; filling an unwatched one changes membership without changing the verifier's

behavior, and the uniqueness of the filled segment is what lets each resulting counterexample be traced back to few sources.

To be able to implement the double-counting principle underlying Lemma 12, we need a relation, together with two degree guarantees. The following result provides them:

Definition 50. Fix $\beta \in \mathcal{E}_\omega$. Let P_β be an optimal prover strategy for $V^{A_{\omega,\beta}}$ on input 1^n. For each segment $S_i(n)$, define

$$\theta_i(\beta) = \mathbf{Pr}[\text{during the interaction with } P_\beta, V^{A_{\omega,\beta}}(1^n) \text{ queries at least one string in } S_i(n)]. \tag{99}$$

Also, given an index i, let $\beta^{(i)}$ be obtained from β by setting every bit of $S_i(n)$ equal to 1 and leaving all other coordinates unchanged.

Lemma 51 (Segment-completion relation)**.** *For every sufficiently large n and every outside assignment ω, define*

$$R_\omega = \{(\beta,\gamma) \in E_\omega \times \{0,1\}^W \ : \ \gamma = \beta^{(i)} \text{ for some } i < M \text{ with } \theta_i(\beta) < \tfrac{1}{3}\}. \tag{100}$$

Then $R_\omega \subseteq E_\omega \times B_\omega$, and

- *every $\beta \in E_\omega$ has at least $M - 3q(n)$ distinct R_ω-neighbours in B_ω;*
- *every $\gamma \in B_\omega$ has at most $M-1$ distinct R_ω-preimages in E_ω.*

Proof. Because the segments are disjoint and every computation path makes at most $q(n)$ oracle queries, the number of distinct segments touched on any computation path is at most $q(n)$. Taking expectations,

$$\sum_{i=0}^{M-1} \theta_i(\beta) \le q(n). \tag{101}$$

Consequently, at most $3q(n)$ indices can satisfy $\theta_i(\beta) \ge 1/3$. Thus at least $M - 3q(n)$ indices satisfy $\theta_i(\beta) < \frac{1}{3}$.

Since β is still unfilling, no segment is full under β. Because the segments are disjoint and only $S_i(n)$ is modified, $\beta^{(i)}$ is uniquely filled, with $S_i(n)$ as its unique full segment.

Now run against $A_{\omega,\beta^{(i)}}$ the same prover strategy P_β. Couple the two verifier computations by using the same random coin string. Until the verifier queries a string in $S_i(n)$, the oracle answers are identical and hence the two transcripts coincide. Therefore the acceptance probability can decrease by at most the probability that a query enters $S_i(n)$:

$$\mathrm{val}_V(A_{\omega,\beta^{(i)}}, 1^n) \ge v_\omega(\beta) - \theta_i(\beta). \tag{102}$$

Here the left side is the optimal value for the modified oracle, so it is at least the acceptance probability obtained by reusing P_β.

Since $\beta \in \mathcal{E}_\omega$ and i satisfies $\theta_i(\beta) < \frac{1}{3}$, we have

$$v_\omega(\beta) > \frac{2}{3} \tag{103}$$

Equation (102) therefore gives

$$\mathrm{val}_V(A_{\omega,\beta^{(i)}}, 1^n) > \frac{1}{3}. \tag{104}$$

Hence $\beta^{(i)} \in \mathcal{B}_\omega$.

The argument above shows that every $\beta \in E_\omega$ has at least $M - 3q(n)$ distinct R_ω-neighbors in B_ω. On the other hand, fix $\gamma \in B_\omega$, and let $S_i(n)$ be its unique full segment. If $\gamma = \beta^{(i)}$ for some $\beta \in E_\omega$, then β agrees with γ outside $S_i(n)$, whereas inside $S_i(n)$ it can be any proper subset. Hence γ has at most $2^s - 1 = M - 1$ preimages under R_ω.

□

Lemma 52 (One-sided segment-completion lemma)**.** *For every sufficiently large n and every outside assignment ω,*

$$\frac{|B_\omega|}{2^{|W|}} \;\geq\; c_V(n)\,\frac{|E_\omega|}{2^{|W|}}, \qquad c_V(n) = \frac{M - 3q(n)}{M-1}, \tag{105}$$

and $c_V(n) \to 1$, so $c_V(n) \geq \frac{1}{2}$ for all sufficiently large n.

Proof. For $\beta \in \mathcal{E}_\omega$, let

$$\deg^+_{R_\omega}(\beta) = \big|\{\gamma \in \mathcal{B}_\omega : (\beta,\gamma) \in R_\omega\}\big|,$$

and, for $\gamma \in \mathcal{B}_\omega$, let

$$\deg^-_{R_\omega}(\gamma) = \big|\{\beta \in \mathcal{E}_\omega : (\beta,\gamma) \in R_\omega\}\big|.$$

Double-counting the pairs in R_ω gives

$$|R_\omega| = \sum_{\beta\in\mathcal{E}_\omega} \deg^+_{R_\omega}(\beta) = \sum_{\gamma\in\mathcal{B}_\omega} \deg^-_{R_\omega}(\gamma).$$

By the first degree bound in Lemma 51, every $\beta \in \mathcal{E}_\omega$ has at least $M - 3q(n)$ distinct R_ω-neighbours. Hence

$$|R_\omega| \geq \big(M - 3q(n)\big)|\mathcal{E}_\omega|.$$

By the second degree bound in that lemma, every $\gamma \in \mathcal{B}_\omega$ has at most $M-1$ R_ω-preimages. Therefore

$$|R_\omega| \leq (M-1)|\mathcal{B}_\omega|.$$

Combining these inequalities yields

$$|\mathcal{B}_\omega| \geq \frac{M - 3q(n)}{M-1}\,|\mathcal{E}_\omega| = c_V(n)|\mathcal{E}_\omega|.$$

Dividing by the total number $2^{|W|}$ of assignments to the coordinates in W gives

$$\frac{|\mathcal{B}_\omega|}{2^{|W|}} \geq c_V(n)\frac{|\mathcal{E}_\omega|}{2^{|W|}},$$

as claimed.

Finally, since $M = 2^{n/2}$ and q is a polynomial,

$$c_V(n) = \frac{M-3q(n)}{M-1} = 1 - \frac{3q(n)-1}{M-1} \longrightarrow 1.$$

In particular, $c_V(n) \geq 1/2$ for all sufficiently large n. □

By Lemma 19, after discarding finitely many support indices we may assume $q(t_j) < t_{j+1}$ for every remaining j.

Define

$$\mathcal{F}_j = \sigma\big(A(z) : |z| < t_{j+1}\big). \tag{106}$$

Let $W_j = W(t_j)$ be the union of the current segments, and define

$$\mathcal{G}_j = \sigma\big(A(z) : |z| < t_{j+1},\ z \notin W_j\big). \tag{107}$$

Because $W_j \subseteq \{0,1\}^{t_j}$,

$$\mathcal{F}_{j-1} \subseteq \mathcal{G}_j \subseteq \mathcal{F}_j. \tag{108}$$

Moreover, $\mathcal{F}_j \subseteq \mathcal{G}_{j+1}$, because $\mathcal{F}_j$ contains only strings of length strictly less than t_{j+1}, whereas $W_{j+1} \subseteq \{0,1\}^{t_{j+1}}$.

Thus

$$\mathcal{F}_{j-1} \subseteq \mathcal{G}_j \subseteq \mathcal{F}_j \subseteq \mathcal{G}_{j+1} \subseteq \cdots \tag{109}$$

is an increasing interleaved filtration.

Conditioned on $\mathcal{G}_j$, the coordinates in W_j remain independent unbiased bits.

Define the events

$$E_j = \left\{ 1^{t_j} \in L^A \text{ and } \operatorname{val}_V(A, 1^{t_j}) > \frac{2}{3} \right\}, \tag{110}$$

$$B_j = \left\{ \begin{array}{l} \text{exactly one segment } S_i(t_j) \text{ is contained in } A, \\ \operatorname{val}_V(A, 1^{t_j}) > \frac{1}{3} \end{array} \right\}. \tag{111}$$

By Lemma 19, for $j \geq j_0$ the value $\operatorname{val}_V(A, 1^{t_j})$ is determined by oracle coordinates of length strictly less than t_{j+1}, so $E_j, B_j \in \mathcal{F}_j$.

Set

$$P_j = \mathbf{Pr}(E_j \mid \mathcal{G}_j), \qquad Q_j = \mathbf{Pr}(B_j \mid \mathcal{G}_j). \tag{112}$$

Lemma 53 (Conditional one-sided estimate)**.** *For all sufficiently large j,*

$$Q_j \geq c_V(t_j) P_j \tag{113}$$

almost surely. In particular, eventually

$$Q_j \geq \frac{1}{2} P_j. \tag{114}$$

Proof. Condition on an arbitrary atom of $\mathcal{G}_j$. All relevant oracle bits outside W_j are then fixed, while the bits of W_j remain uniformly and independently distributed.

The conditional event E_j is exactly the set $\mathcal{E}_\omega$ of Lemma 52, and the conditional event B_j is exactly $\mathcal{B}_\omega$. The inequality follows pointwise from Lemma 52. □

To apply Lemma 29, introduce the interleaved filtration

$$\mathcal{H}_{2j-2} = \mathcal{F}_{j-1}, \mathcal{H}_{2j-1} = \mathcal{G}_j, \mathcal{H}_{2j} = \mathcal{F}_j, \tag{115}$$

after reindexing from a sufficiently large initial j. For the events E_j, define

$$D^E_{2j-1} = \varnothing, \qquad D^E_{2j} = E_j. \tag{116}$$

Then

$$\mathbf{Pr}(D^E_{2j} \mid \mathcal{H}_{2j-1}) = \mathbf{Pr}(E_j \mid \mathcal{G}_j) = P_j. \tag{117}$$

Conditional Borel-Cantelli therefore gives a probability-one event Ω^E_V on which

$$E_j \text{ i.o.} \iff \sum_j P_j = \infty. \tag{118}$$

Similarly, with

$$D^B_{2j-1} = \varnothing, \qquad D^B_{2j} = B_j, \tag{119}$$

there is a probability-one event Ω^B_V on which

$$B_j \text{ i.o.} \iff \sum_j Q_j = \infty. \tag{120}$$

Let $\Omega_V = \Omega^E_V \cap \Omega^B_V$. Then $\mathbf{Pr}(\Omega_V) = 1$.

Lemma 54. *Fix a clocked probabilistic polynomial-time oracle verifier V. There is a probability-one set of oracles Ω_V such that, for every $A \in \Omega_V$, every language S for which V^A is a valid interactive proof verifier and which satisfies $S \subseteq L^A$ is finite.*

Proof. Fix $A \in \Omega_V$. The quantities $P_j(A)$ and $Q_j(A)$ are now ordinary nonnegative real numbers.

There are two exhaustive cases.

Case 1. Suppose

$$\sum_j P_j(A) < \infty. \tag{121}$$

By (118), only finitely many events $E_j(A)$ occur.

Let S be any language for which V^A is a valid interactive proof verifier and suppose $S \subseteq L^A$.

If $1^{t_j} \in S$, then, because $S \subseteq L^A$, $1^{t_j} \in L^A$. By completeness (25), $\mathrm{val}_V(A, 1^{t_j}) > 2/3$. Hence $E_j(A)$ occurs.

Therefore only finitely many support strings can belong to S. Since L^A contains no off-support strings and $S \subseteq L^A$, the language S is finite.

Case 2. Suppose

$$\sum_j P_j(A) = \infty. \tag{122}$$

By Lemma 53, eventually

$$Q_j(A) \geq \frac{1}{2} P_j(A). \tag{123}$$

Therefore

$$\sum_j Q_j(A) = \infty. \tag{124}$$

Equation (120) implies that $B_j(A)$ occurs infinitely often.

In particular, for some j, exactly one segment at length t_j is fully contained in A, and

$$\mathrm{val}_V(A, 1^{t_j}) > \frac{1}{3}. \tag{125}$$

The existence of a full segment means $1^{t_j} \notin L^A$. If $S \subseteq L^A$, then $1^{t_j} \notin S$. But soundness (26) for the interactive proof V^A would require

$$\mathrm{val}_V(A, 1^{t_j}) < \frac{1}{3}, \tag{126}$$

contradicting (125).

Thus Case 2 is impossible whenever V^A is a valid interactive proof verifier for a sublanguage of L^A.

We conclude that every such sublanguage is finite. □

Enumerate the clocked polynomial-time oracle verifiers; intersect the probability-one sets supplied by Lemma 54 and the probability-one event on which L^A is infinite; then no infinite IP^A sublanguage of L^A exists, while $L^A \in \mathsf{coNP}^A$.

5.4. **Proof of Theorem 7.** The proof of Theorem 7 is best understood as a cryptographic version of the "rare-or-wrong" strategy. It uses a cryptographically hard function to choose, for every test length, a specific oracle query. It then constructs two complementary languages based on the two possible answers to this hidden query, and notes that:

- an unambiguous machine can locate the relevant oracle bit by guessing its unique address; this puts the two resulting languages in relativized $UP \cap coUP$.
- a deterministic polynomial-time machine can only locate that address with negligible probability, because doing so would invert a one-way permutation;
- without querying that address, the machine cannot distinguish whether the target bit is 0 or 1.

- the flipping involution associated with flipping the hidden bit shows that accepting without locating it has a 1/2-type chance of being wrong; conditional Borel-Cantelli turns this local uncertainty (via Lemma 15) into almost-sure immunity.

Indeed, fix $j \geq 2$ and define

$$n_j = t_j, \qquad m_j = \frac{n_j}{4}. \tag{127}$$

Remark 55. Clearly (by the definition of t_j) m_j is an integer for $j \geq 2$.

We define two explicitly computable sets of oracle coordinates of length exactly n_j.
For $1 \leq r \leq m_j$, let

$$u_{j,r} = 00\ \mathrm{bin}_{m_j}(r)\, 0^{\,n_j - m_j - 2}, \tag{128}$$

For $x \in \{0,1\}^{m_j}$, let

$$v_{j,x} = 01\, x\, 0^{\,n_j - m_j - 2}. \tag{129}$$

The two families of words are disjoint, and words corresponding to different j's are different because they have different lengths.

Define the word

$$Y_j(A) = A(u_{j,1})A(u_{j,2}) \cdots A(u_{j,m_j}) \in \{0,1\}^{m_j}. \tag{130}$$

Since the oracle coordinates in (128) are independent fair bits, $Y_j(A)$ is uniformly distributed over $\{0,1\}^{m_j}$. Because f_{m_j} is a permutation, there is a unique string $X_j(A)$ that is the preimage of $Y_j(A)$, formally

$$X_j(A) = f_{m_j}^{-1}(Y_j(A)). \tag{131}$$

Make the following notations: $Z_j(A) = v_{j,X_j(A)}$, and $H_j(A) = A(Z_j(A))$.

For $b \in \{0,1\}$, define

$$L_b^A = \left\{ 1^{t_j}, j \geq 2 : H_j(A) = b \right\}. \tag{132}$$

Clearly, $L_0^A \cup L_1^A = T_{\geq 2}, L_0^A \cap L_1^A = \emptyset$.

Lemma 56. *For every oracle A and every $b \in \{0,1\}$,*

$$L_b^A \in \mathrm{UP}^A \cap \mathrm{coUP}^A. \tag{133}$$

Proof. Consider first membership in L_b^A.

On an input w, deterministically check whether $w = 1^{t_j}$ for some $j \geq 2$. If not, reject. Suppose $w = 1^{t_j}$. Compute $m_j = t_j/4$ and nondeterministically guess a string $x \in \{0,1\}^{m_j}$. Query the m_j oracle coordinates $u_{j,r}$ and reconstruct $Y_j(A)$. Compute $f_{m_j}(x)$ and check whether $f_{m_j}(x) = Y_j(A)$. Since f_{m_j} is a permutation, exactly one guessed string can pass this test, namely $x = X_j(A)$. On this unique surviving branch, query $A(v_{j,x})$ and accept if and only if $A(v_{j,x}) = b$.

Thus there is either exactly one accepting branch or no accepting branch. Hence $L_b^A \in \mathrm{UP}^A$.

For the complement, if w is not in the unary support $T_{\geq 2}$, accept deterministically. If $w = 1^{t_j}$, perform the same unique-witness procedure and accept on the unique valid preimage branch if and only if $A(v_{j,x}) = 1 - b$. So $L_b^A \in \mathrm{coUP}^A$. □

Lemma 57. *For every $j \geq 2$ and $b \in \{0,1\}$, $\mathbf{Pr}_A[H_j = b] = \frac{1}{2}$. Moreover, the random variables H_j, $j \geq 2$, are independent.*

Proof. Condition on all oracle coordinates $u_{j,r}$. Then Y_j and hence X_j are fixed. The coordinate v_{j,X_j} belongs to the disjoint v-block and remains an unbiased oracle bit. Therefore $\Pr[H_j = b \mid Y_j] = \frac{1}{2}$. Averaging over Y_j proves the first claim.

For different indices j, the entire collections of oracle coordinates used to define H_j have different lengths and are therefore disjoint. Independence follows from the product structure of the random oracle. □

Lemma 58. *For each $b \in \{0,1\}$,*

$$\mathbf{Pr}_A[L_b^A \text{ is infinite}] = 1. \tag{134}$$

Indeed, both L_0^A and L_1^A are infinite simultaneously with probability one.

Proof. By Lemma 57 and the second Borel-Cantelli lemma.

□

Fix a deterministic oracle machine M and an index j. Let

$$E_j = \{M^A(1^{t_j}) \text{ accepts}\}. \tag{135}$$

For a fixed $b \in \{0,1\}$, define the false-positive event

$$B_{j,b} = E_j \cap \{H_j \neq b\}. \tag{136}$$

Finally, define

$$T_j = \left\{M^A(1^{t_j}) \text{ queries the distinguished coordinate } Z_j(A)\right\}. \tag{137}$$

The event T_j is the exceptional case. If it occurs, the computation has explicitly discovered an address containing the unique preimage X_j. If it does not occur, acceptance cannot depend on the target bit H_j. It is useful to formalize the last observation without informally speaking about "conditioning on all bits except a randomly selected coordinate."

Define $\Phi_j : \Omega \longrightarrow \Omega$ by flipping the distinguished coordinate. Formally:

$$(\Phi_j A)(z) = \begin{cases} 1 - A(z), & z = Z_j(A), \\ A(z), & z \neq Z_j(A). \end{cases} \tag{138}$$

Since $Z_j(A)$ depends only on the u-coordinates, which are not changed by Φ_j, $Z_j(\Phi_j A) = Z_j(A)$. Hence $\Phi_j^2 = \mathrm{id}$. Moreover, Φ_j is measure-preserving: after conditioning on the u-block, it simply flips one fixed unbiased coordinate of the independent v-block.

If $A \notin T_j$ then the computations relative to A and $\Phi_j A$ receive identical answers on any queries, and therefore have the same acceptance outcome. Hence $E_j \Delta \Phi_j(E_j) \subseteq T_j$.

The same coupling shows that T_j itself is Φ_j-invariant. If the distinguished coordinate is never queried relative to A, all answers agree in the two computations, so the same transcript is followed and the coordinate is not queried relative to $\Phi_j A$ either. If it is queried, consider the first such query: before it the machine has received no answer from the distinguished coordinate, so the two transcripts agree up to that point and the first distinguished query occurs at the same moment in both. Hence

$$T_j(A) = T_j(\Phi_j A).$$

Lemma 59. *Fix $j \geq 2$ and let $\mathcal{F}$ be a σ-field generated by finitely many oracle coordinates, all of length $< t_{j+1}$, that includes every coordinate on which E_j, T_j and H_j depend, and let*

$$\mathcal{G} = \{\, C \in \mathcal{F} : \Phi_j^{-1}(C) = C \,\}.$$

Then:

- *(i). Φ_j permutes the atoms of $\mathcal{F}$, and all atoms of $\mathcal{F}$ have the same positive probability;*
- *(ii) Every Φ_j-orbit of atoms of $\mathcal{F}$ has exactly two elements, and the atoms of $\mathcal{G}$ are precisely the sets $C \cup \Phi_j(C)$, where C ranges over the atoms of $\mathcal{F}$.*
- *(iii). T_j is a union of atoms of $\mathcal{G}$; in particular $T_j \in \mathcal{G}$.*

Proof. (i) The distinguished coordinate $Z_j(A)$ has length $t_j < t_{j+1}$ and depends only on the u-coordinates $u_{j,1}, \dots, u_{j,m_j}$, which Φ_j does not change. Hence Φ_j acts on the finitely many coordinates generating $\mathcal{F}$ by flipping one of them, determined by the others, and therefore maps each atom of $\mathcal{F}$ (a cylinder on these coordinates) onto another atom. All these cylinders fix the same number of coordinates, so they have equal, positive probability under the uniform product measure.

(ii) An orbit has one or two elements. If an atom C satisfied $\Phi_j(C) = C$, then picking $A \in C$ we would get $H_j(\Phi_j A) = H_j(A)$, because H_j is constant on atoms of $\mathcal{F}$; but $H_j(\Phi_j A) = 1 - H_j(A)$ by the definition (138) of Φ_j, a contradiction. So every orbit has exactly two elements. A set belongs to $\mathcal{G}$ exactly when it is a Φ_j-invariant union of atoms of $\mathcal{F}$, equivalently, when it is a union of sets of the form $C \cup \Phi_j(C)$. Since every orbit has two elements, these orbit-unions are precisely the atoms of $\mathcal{G}$.

(iii) T_j is $\mathcal{F}$-measurable and Φ_j-invariant; by (ii) it is therefore a union of orbits. □

Lemma 60. *Let $\mathcal{F}, \mathcal{G}$ be as in Lemma 59, and let*

$$P_j = \Pr(E_j \mid \mathcal{G}), Q_{j,b} = \Pr(B_{j,b} \mid \mathcal{G}). \tag{139}$$

Then, for every $b \in \{0,1\}$,

$$Q_{j,b} \geq \frac{1}{2}(P_j - \mathbf{1}_{T_j}). \tag{140}$$

Proof. Fix an atom D of $\mathcal{G}$. By Lemma 59(ii), there are two distinct atoms C_0, C_1 of $\mathcal{F}$ such that $D = C_0 \cup C_1$, $\Phi_j(C_0) = C_1$, $\Phi_j(C_1) = C_0$.

By Lemma 59(i), C_0 and C_1 have equal positive probability. Consequently, conditionally on D, the choice between the two $\mathcal{F}$-atoms is uniform.

We now pass explicitly to the associated two-point quotient. Let $\widehat{\Omega}_D = \{0,1\}$ with the uniform probability measure, and define the involution $\widehat{\Phi}_D(r) = 1-r$. Since E_j, T_j, and H_j are $\mathcal{F}$-measurable, they are constant on each C_r. Define $\widehat{E}_D = \{r \in \widehat{\Omega}_D : C_r \subseteq E_j\}$, $\widehat{T}_D = \{r \in \widehat{\Omega}_D : C_r \subseteq T_j\}$, and define $\widehat{H}_D : \widehat{\Omega}_D \to \{0,1\}$ by $\widehat{H}_D(r) = H_j(A)$, for any $A \in C_r$. These definitions do not depend on the choice of $A \in C_r$.

The hypotheses of Lemma 15 hold on $\widehat{\Omega}_D$. Indeed, from $H_j(\Phi_j A) = 1 - H_j(A)$ we obtain $\widehat{H}_D(\widehat{\Phi}_D(r)) = 1 - \widehat{H}_D(r)$. The Φ_j-invariance of T_j gives $\widehat{\Phi}_D(\widehat{T}_D) = \widehat{T}_D$, and the inclusion $E_j \triangle \Phi_j(E_j) \subseteq T_j$ descends to $\widehat{E}_D \triangle \widehat{\Phi}_D(\widehat{E}_D) \subseteq \widehat{T}_D$.

Put $\widehat{B}_{D,b} = \widehat{E}_D \cap \{\widehat{H}_D \neq b\}$.

Lemma 15, applied to the two-point probability space $\widehat{\Omega}_D$, therefore gives

$$\Pr_{\widehat{\Omega}_D}(\widehat{B}_{D,b}) \geq \frac{1}{2}\left(\Pr_{\widehat{\Omega}_D}(\widehat{E}_D) - \Pr_{\widehat{\Omega}_D}(\widehat{T}_D)\right). \tag{141}$$

Because C_0 and C_1 have equal probability, the quotient probabilities are exactly the corresponding probabilities conditioned on D. Thus

$$\Pr(B_{j,b} \mid D) \geq \frac{1}{2}\left(\Pr(E_j \mid D) - \Pr(T_j \mid D)\right). \tag{142}$$

Finally, $T_j \in \mathcal{G}$ by Lemma 59(iii). Hence, on the $\mathcal{G}$-atom D, $\Pr(T_j \mid D) = \mathbf{1}_{T_j}$. Moreover, on D, $P_j = \Pr(E_j \mid D)$, $Q_{j,b} = \Pr(B_{j,b} \mid D)$. It follows that $Q_{j,b} \geq \frac{1}{2}(P_j - \mathbf{1}_{T_j})$ on D. Since D was an arbitrary atom of $\mathcal{G}$, the inequality holds everywhere. □

We next prove that the exceptional events T_j occur with summable total probability:

Lemma 61. *Fix a deterministic polynomial-time oracle machine M. Then there is a probabilistic polynomial-time inverter I_M such that, for every sufficiently large j,*

$$\mathbf{Pr}_A[T_j] = \mathbf{Pr}_{y \leftarrow U_{m_j}, r}\left[f_{m_j}(I_M(1^{m_j}, y, r)) = y\right]. \tag{143}$$

Proof. Given 1^m and $y \in \{0,1\}^m$, the inverter first sets $n = 4m$. It tests whether $n = t_j$ for some $j \geq 2$. If not, it fails.

Suppose that indeed $n = t_j$ and hence $m = m_j$. The inverter simulates $M(1^n)$ together with a lazy random oracle. For the programmed coordinates $u_{j,r}$, this lazy oracle responds

$$A(u_{j,r}) = y_r. \tag{144}$$

All other oracle queries are answered by independent unbiased bits, with a table used to maintain consistency on repeated queries. Whenever M asks a query of the form $v_{j,x}, (x \in \{0,1\}^m)$, the inverter computes $f_m(x)$. If $f_m(x) = y$, it outputs x.

Because f_m is a permutation, there is exactly one such string, namely $x = f_m^{-1}(y)$. Thus the inverter succeeds precisely when the simulated machine asks the distinguished query $v_{j,f_m^{-1}(y)}$.

It remains to compare distributions. When y is uniform, the programmed coordinates (144) are themselves independent unbiased bits. Every other oracle coordinate is generated independently and uniformly by lazy sampling. Hence the simulated oracle has exactly the random-oracle distribution. This proves (143).

Finally, since $n = 4m$, simulating a polynomial-time computation on input 1^n takes time polynomial in m. The lazy oracle table contains only polynomially many strings of polynomial length. Therefore I_M is probabilistic polynomial time. □

Corollary 62. *For every polynomial-time oracle machine M,*

$$\sum_{j=2}^{\infty} \mathbf{Pr}_A[T_j] < \infty. \tag{145}$$

Consequently,

$$\mathbf{Pr}_A[T_j \text{ infinitely often}] = 0. \tag{146}$$

Proof. By one-wayness and Lemma 61, there is a negligible function ν_M such that $\mathbf{Pr}_A[T_j] \leq \nu_M(m_j)$. Since ν_M is negligible, eventually $\nu_M(k) \leq \frac{1}{k^2}$. The convergence of $\zeta(2)$ completes the proof.

The first Borel-Cantelli lemma gives (146). □

Remark 63. Corollary 62 is the precise point at which cryptographic average-case one-wayness is used. A purely worst-case noninvertibility assumption would not imply (145).

Fix now a clocked polynomial-time oracle machine M, and let q_M be an increasing polynomial $q_M(n) \geq n$ bound both its running time and the lengths of its oracle queries.

Lemma 64. *Fix $b \in \{0,1\}$. With probability one over A,*

$$L(M^A) \subseteq L_b^A \quad \Longrightarrow \quad L(M^A) \text{ is finite.} \tag{147}$$

Proof. As in previous proofs, discard a finite initial range so that $q_M(t_j) < t_{j+1}$.

Define

$$\mathcal{F}_j = \sigma\big(A(z) : |z| < t_{j+1}\big). \tag{148}$$

Then the acceptance event E_j belongs to $\mathcal{F}_j$.

Let Φ_j be the involution from (138), and define

$$\mathcal{G}_j = \left\{ C \in \mathcal{F}_j : \Phi_j^{-1}(C) = C \right\}. \tag{149}$$

Since $q_M(t_j) < t_{j+1}$ and the coordinates $u_{j,r}, v_{j,x}$ have length t_j, the pair $(\mathcal{F}_j, \mathcal{G}_j)$ satisfies the hypotheses of Lemma 59.

Since Φ_j changes only an oracle bit of length exactly t_j, $\mathcal{F}_{j-1} \subseteq \mathcal{G}_j \subseteq \mathcal{F}_j$.

Likewise, $\mathcal{F}_j \subseteq \mathcal{G}_{j+1}$, because Φ_{j+1} changes a coordinate of length t_{j+1}, while $\mathcal{F}_j$ contains only coordinates of smaller length. Hence

$$\mathcal{F}_{j-1} \subseteq \mathcal{G}_j \subseteq \mathcal{F}_j \subseteq \mathcal{G}_{j+1} \subseteq \cdots \tag{150}$$

is an increasing interleaved filtration.

We now apply Lemma 29 to filtration (150), inserting the empty event at odd indices. Thus, if

$$P_j = \Pr(E_j \mid \mathcal{G}_j), \tag{151}$$

then almost surely

$$E_j \text{ i.o.} \quad \Longleftrightarrow \quad \sum_j P_j = \infty. \tag{152}$$

Similarly, putting

$$Q_{j,b} = \Pr(B_{j,b} \mid \mathcal{G}_j), \tag{153}$$

we obtain almost surely

$$B_{j,b} \text{ i.o.} \quad \Longleftrightarrow \quad \sum_j Q_{j,b} = \infty. \tag{154}$$

We now complete the proof of Lemma 64. Take the intersection of the following probability-one events:

(*). by Corollary 62, only finitely many T_j occur;
(**). the equivalence (152) holds;
(***). the equivalence (154) holds.

Fix an oracle A in this intersection. We now regard $P_j(A)$ and $Q_{j,b}(A)$ as ordinary nonnegative real numbers.

There are two cases:

- **Case 1:** $\sum_j P_j(A) < \infty$.
  By conditional Borel-Cantelli, only finitely many events E_j occur. Hence M^A accepts only finitely many support strings 1^{t_j}. If $L(M^A) \subseteq L_b^A$ then M^A accepts no string outside T, because $L_b^A \subseteq T$. Thus $L(M^A)$ is finite.
- **Case 2:** $\sum_j P_j(A) = \infty$.
  Because of (*), $T_j(A) = 0$ for all sufficiently large j. The query-or-error inequality (140) gives eventually

$$Q_{j,b}(A) \geq \frac{1}{2} P_j(A). \tag{155}$$

  Therefore

$$\sum_j Q_{j,b}(A) = \infty. \tag{156}$$

  Conditional Borel-Cantelli Lemma 29 implies that $B_{j,b}$ occurs infinitely often. In particular, for some j, $M^A(1^{t_j})$ accepts but $H_j(A) \neq b$. Hence $1^{t_j} \in L(M^A) \setminus L_b^A$, so $L(M^A) \not\subseteq L_b^A$.

Thus, whenever $L(M^A) \subseteq L_b^A$, only Case 1 is possible, and then $L(M^A)$ is finite.

□

Intersecting the probability 1 events in Lemma 64 over the countably many clocked machines M and both values of $b \in \{0,1\}$, and intersecting once more with the event from Lemma 58 yields a measure-one set of oracles such that both L_0^A and L_1^A are infinite and have no infinite subset in P^A, completing the proof of Theorem 7.

6. Proof of the Main Theorem

We now prove Theorem 1. Our approach does not use directly the Sipser functions from [41], but a family $G_{d,m}$ based on a minor reparametrization $\widehat{Sipser}_{d,m}$ of the RST family. More precisely

$$G_{d,m} = \begin{cases} \widehat{Sipser}_{d,m}, & d \text{ even}, \\ 1 - \widehat{Sipser}_{d,m}, & d \text{ odd}. \end{cases} \tag{157}$$

The reason for this choice is technical: we want that the boolean circuit representing our test functions has an OR gate at the top. In the second case, pushing the complement to the leaves exchanges AND and OR and negates every input literal.

As for the modified Sipser functions $\widehat{Sipser}_{d,m}$ used in the definition of $G_{d,m}$:

- The depth, alternating AND/OR structure, read-once property, root orientation, bottom fan-in m and intermediate fan-in w are the same as for the family $Sipser_{d,m}$ in [41].
- RST's top fan-in w_0, defined indirectly through their restriction parameters, is replaced by a new fan-in $\widehat{w}_0$ that is computable in polynomial time by elementary rational arithmetic, and that agrees with w_0 up to a *relative* error $2^{-\Omega_d(m)}$. The net result will be a class of formulas $\widehat{Sipser}_{d,m}$, that will, in general have a different number of variables than $Sipser_{d,m}$, so a direct comparison between the two formulas is not possible. However, we will show that there is a way to couple their input variables, with uniform marginals, such that

$$\mathbf{Pr}_{X,Y,Z}[\widehat{Sipser}_{d,m}(X,Y) \neq Sipser_{d,m}(X,Z)] \leq 2^{-\Omega_d(m)} \tag{158}$$

  This will allow us to transfer the correlation upper bound from [41] (which is valid for the family $Sipser_{d,m}$) to the new class of functions $\widehat{Sipser}_{d,m}$ and, ultimately, to functions $G_{d,m}$.
- The reason we use the modified version of the Sipser function is that we need to ensure the polynomial time computability of the fan-ins, in order to prove that the test language we construct actually belongs to the claimed level of the relativized polynomial hierarchy. For the correlation bounds in [41] ignoring this computability was harmless, but in our case it may affect the complexity of the constructed test language. Whereas for the intermediate fan-ins used in [41] we could prove that they are computable in polynomial time *exactly*, for the top fan-in the situation is more delicate, and motivated the use of a polynomial time computable approximation $\widehat{w_0}$ to w_0. It is worth stressing at the outset why, in our proof strategy, an *exact* value is indispensable for the intermediate fan-in w but a relative approximation suffices for the top fan-in: the coupling of Lemma 71 matches root subtrees *together with all the leaves below them*, so a discrepancy in w (or in the bottom fan-in m) would perturb every one of the $\approx 2^m$ root subtrees simultaneously, whereas a discrepancy Δ in the top fan-in leaves Δ subtrees unmatched, each of which contributes at most $p_{d-1} \leq 2^{1-m}$ to the disagreement probability[9].

6.1. Polynomial time computability of the intermediate fan-ins.

We recall the following standard number-theoretic input (with a deliberately weak exponent, since that's all that we need):

Lemma 65. *There exist effectively computable constants $c > 0$ and $Q_0 \geq 1$ such that, for all integers P and all integers $Q \geq Q_0$,*

$$\left| \ln 2 - \frac{P}{Q} \right| \geq c \cdot Q^{-5}.$$

Proof. One can take $c = (2000)^{-1}$ and $Q_0 = 1$. See [1], Corollary 2 of Theorem 3, page 150. □

[9] One can, indeed, use approximate fan-ins at all levels, at the expense of a more complicated coupling strategy.

Lemma 66. *The integer $w(m) = \lfloor m2^m \ln 2 \rfloor$ is computable in deterministic time polynomial in m.*

Proof. Put $q_m = m \cdot 2^m$. Lemma 65 gives, for every integer P and all sufficiently large m,

$$|q_m \ln 2 - P| \geq c \cdot q_m^{-4}. \tag{159}$$

Since $q_m \leq 2^{2m}$ for $m \geq 2$, there are effectively computable constants $C > 0$ and m_0 such that $\operatorname{dist}(q_m \ln 2, \mathbb{Z}) \geq 2^{-Cm}$ for $(m \geq m_0)$.

We use the positive-term expansion

$$\ln 2 = \sum_{r=1}^{\infty} \frac{1}{r2^r}. \tag{160}$$

For $s \geq 1$, define

$$\ell_s = \sum_{r=1}^{s} \frac{1}{r2^r}, \qquad u_s = \ell_s + \frac{1}{(s+1)2^s}. \tag{161}$$

The tail satisfies

$$0 < \ln 2 - \ell_s = \sum_{r=s+1}^{\infty} \frac{1}{r2^r} \leq \frac{1}{s+1} \sum_{r=s+1}^{\infty} 2^{-r} = \frac{1}{(s+1)2^s}, \tag{162}$$

and hence $\ell_s < \ln 2 \leq u_s$.

Take $s = \lceil (C+3)m \rceil$. Then

$$q_m(u_s - \ell_s) \leq q_m \cdot 2^{-s} \leq 2^{2m-(C+3)m} < 2^{-Cm}. \tag{163}$$

Thus the rational interval $I_m = [q_m\ell_s, q_m u_s]$ contains $q_m \ln 2$ and has length strictly smaller than $\operatorname{dist}(q_m \ln 2, \mathbb{Z})$. It follows that I_m contains no integer. Consequently, $\lfloor q_m\ell_s \rfloor = \lfloor q_m u_s \rfloor = \lfloor q_m \ln 2 \rfloor = w(m)$.

The rationals ℓ_s and u_s can be formed exactly using $s = O(m)$ additions and multiplications of integers of polynomially many bits. Even the naive common-denominator computation uses only $O(s^2)$ bits, and therefore runs in time polynomial in m. The finitely many cases $m < m_0$ may be hardwired. □

For the top fan-in $w_0 = \lceil R_m \rceil$ (see equation (35)) no analogue of Lemma 66 is available, and none is obviously within reach: unlike w, the number R_m is not an explicit closed-form expression but the output of a chain of $\Theta(d)$ transcendental operations, so an exact computation would require a separation-from-the-integers statement for R_m of the kind that Lemma 65 provides for $\ln 2$.

Fortunately we do not need one. As explained in the third bullet above, the top fan-in enters the argument only through the number of unmatched root subtrees in the coupling of Lemma 71, and each unmatched subtree contributes only $p_{d-1} \leq 2^{1-m}$ to the disagreement probability. It therefore suffices to approximate R_m to *relative* accuracy $2^{-\Omega_d(m)}$. That is an easy task, because the leading term of R_m is the single explicit constant $2^m \ln 2$:

Lemma 67. *For every fixed $d \geq 2$ and all sufficiently large m,*

$$\frac{q}{2} \leq t_r \leq 2q \qquad (1 \leq r \leq d-1), \tag{164}$$

$$2^{-m-1} \leq D_m \leq 2^{2-m}, \tag{165}$$

and

$$R_m = 2^m \ln 2 \left(1 + O_d\left(2^{-m/20}\right)\right). \tag{166}$$

Proof. Recall that

$$\rho = 2^{-m}, \qquad q = 2^{-m/2}, \qquad w = \lfloor m2^m \ln 2 \rfloor, \qquad \lambda = \frac{(\log_2 w)^{3/2}}{w^{5/4}}, \tag{167}$$

and

$$t_{d-1} = \frac{\rho - \lambda}{q}, \qquad t_{r-1} = \frac{(1-t_r)^{qw} - \lambda}{q} \quad (2 \le r \le d-1), \tag{168}$$

By [41, Lemma 7.1] (whose proof is given in [41, Appendix A]) there is a universal constant $c > 0$ such that

$$|t_r - q| \le q^{1.1} \qquad (1 \le r \le d-1) \tag{169}$$

whenever $2 \le d \le cm/\log m$. Since d is fixed, this condition holds for all sufficiently large m. Furthermore, $q^{0.1} = 2^{-m/20} \le 1/2$ for $m \ge 20$. Consequently,

$$t_r \ge q - q^{1.1} = q(1-q^{0.1}) \ge \frac{q}{2} \qquad \text{and} \qquad t_r \le q + q^{1.1} = q(1+q^{0.1}) \le \frac{3q}{2} \le 2q,$$

which is (164); in particular, after increasing $m_0 = m_0(d)$ if necessary, $2q \le 1/2$.

The inequalities

$$x \le -\ln(1-x) \le \frac{x}{1-x} \qquad (0 \le x < 1) \tag{170}$$

applied with $x = t_1$ therefore give

$$\frac{1}{2}q^2 \le D_m \le \frac{2q^2}{1-2q} \le 4q^2, \tag{171}$$

that is, (165). For the sharper statement (166), write $-\ln(1-t_1) = t_1(1+\xi)$ with, by (170), $0 \le \xi \le t_1/(1-t_1) \le 4q$. Since $t_1 = q\big(1+O(q^{0.1})\big)$ by (169) and $q \le q^{0.1} = 2^{-m/20}$, we get

$$D_m = q\,t_1(1+\xi) = q^2\big(1+O(2^{-m/20})\big) = \rho\big(1+O(2^{-m/20})\big),$$

and (166) follows from $R_m = \ln 2 / D_m$ and $\rho^{-1} = 2^m$. □

Note that (166) carries an absolute error $O(2^{19m/20})$ and therefore says nothing whatsoever about the integer w_0 itself. What it does deliver is precisely the relative accuracy that the coupling can absorb, and that is what we exploit in the next subsection.

6.2. **The reparameterized Sipser functions.** For integers $m \ge 2$, $d \ge 2$, let ℓ_m denote the $2m$-th partial sum of the series for $\ln 2$ already used in the proof of Lemma 66, and let $\widehat{w}_0$ be the integer it determines:

$$\widehat{\ell_m} = \ell_{2m} = \sum_{r=1}^{2m} \frac{1}{r2^r}, \qquad \widehat{w}_0 = \left\lfloor 2^m \widehat{\ell_m} \right\rfloor. \tag{172}$$

By the tail estimate already recorded in the proof of Lemma 66 (with $s = 2m$),

$$0 < \ln 2 - \widehat{\ell_m} \le \frac{1}{(2m+1)2^{2m}} < 2^{-2m}. \tag{173}$$

Let $\widehat{Sipser}_{d,m}$ be the regular, read-once, monotone formula with d alternating layers, bottom AND gates, and fan-ins

$$\widehat{w_0}, \qquad \underbrace{w, \ldots, w}_{d-2 \text{ intermediate levels}} \qquad , m. \tag{174}$$

The intermediate range is, of course, empty when $d = 2$. Thus the root is OR when d is even and AND when d is odd. Write

$$N(d,m) = m \cdot \widehat{w_0} \cdot w^{d-2}. \tag{175}$$

6.3. **Useful properties of the reparameterized Sipser functions.** Next, we record some useful properties of the family $\widehat{Sipser}_{d,m}$:

First of all, the change in top-width is, as promised, small enough to make the upper bound work:

Lemma 68. *The integer $\widehat{w}_0$ is computable in deterministic time polynomial in m, and $\frac{1}{2} < \widehat{\ell_m} < \ln 2 < 1$. Moreover, for every fixed $d \geq 2$ there exists a constant $C_d > 0$ such that, for all sufficiently large m,*

$$|\widehat{w}_0 - w_0| \leq \Delta_m := C_d \cdot 2^{\frac{19}{20}m}. \tag{176}$$

Proof. Computability. The rational $\widehat{\ell_m}$ has reduced denominator dividing $\mathrm{lcm}(1,\ldots,2m) \cdot 2^{2m}$, a number of $O(m)$ bits by the elementary bound $\mathrm{lcm}(1,\ldots,n) \leq 4^n$. Summing the $2m$ terms over this common denominator and performing a single integer division computes $\lfloor 2^m \widehat{\ell_m} \rfloor$ exactly, using $O(m)$ arithmetic operations on $O(m)$-bit integers.

Size. The first two terms already give $\widehat{\ell_m} \geq \frac{1}{2} + \frac{1}{8} = \frac{5}{8} > \frac{1}{2}$, and $\widehat{\ell_m} < \ln 2 < 1$ by (173).

Accuracy. By (173), $|2^m \widehat{\ell_m} - 2^m \ln 2| \leq 2^m \cdot 2^{-2m} = 2^{-m} \leq 1$, and taking the floor costs at most 1 further, so $|\widehat{w}_0 - 2^m \ln 2| \leq 2$. By (166), $|R_m - 2^m \ln 2| \leq 2^m \ln 2 \cdot O_d(2^{-m/20})$, so there exists $K_d > 0$ such that for all sufficiently large m, $|R_m - 2^m \ln 2| \leq K_d \cdot 2^{\frac{19}{20}m}$. On the other hand $|w_0 - R_m| \leq 1$, because $w_0 = \lceil R_m \rceil$. Adding the three estimates gives $|\widehat{w}_0 - w_0| \leq 3 + K_d \cdot 2^{\frac{19}{20}m} \leq C_d \cdot 2^{\frac{19}{20}m}$, for all sufficiently large m, where e.g. $C_d = K_d + 3$. □

Lemma 69. *$\widehat{w}_0$ and all the other fan-ins of $\widehat{\mathrm{Sipser}}_{d,m}$, $N(d,m)$, validity of a child index, the rank of a leaf, and the rank of a valid root-to-leaf path are computable in time polynomial in m (that may depend on d).*

Finally, for every fixed $d \geq 2$ and all sufficiently large m,

$$2^{-(d-1)} m^{d-1} 2^{(d-1)m} \leq N(d,m) \leq m^{d-1} 2^{(d-1)m}. \tag{177}$$

Consequently,

$$2^m \leq N(d,m) \leq 2^{2dm} \tag{178}$$

for all sufficiently large m.

Proof. Polynomial-time computability of fan-ins follows from Lemma 66, Lemma 68, and the fact that the bottom fan-in is m.

To index bottom leaves we use zero-based child indices. Denote by f_r the fan-in of the r-th level from the top, according to equation (174). A path $(i_0,\ldots,i_{d-1})$ from the root to the given leaf is valid exactly when $0 \leq i_r < f_r$ for every r, and its rank is

$$\mathrm{rank}_m(i_0,\ldots,i_{d-1}) = \sum_{r=0}^{d-1} i_r \prod_{s=r+1}^{d-1} f_s. \tag{179}$$

This is a bijection from the valid paths onto $\{0,\ldots,N(d,m)-1\}$. Since d is fixed, its evaluation and all validity tests use a fixed number of arithmetic operations on $O_d(m)$-bit integers.

Directly from (172) and $\frac{1}{2} < \ell_m < 1$ (Lemma 68) we obtain, for every $m \geq 1$,

$$2^{m-1} \leq \widehat{w}_0 \leq 2^m, \tag{180}$$

since $2^m \ell_m > 2^{m-1}$ forces $\lfloor 2^m \ell_m \rfloor \geq 2^{m-1}$ (as 2^{m-1} is an integer), while $\widehat{w}_0 \leq 2^m \ell_m < 2^m$. Moreover, $w = \lfloor m \cdot 2^m \ln 2 \rfloor$, and therefore, again using $1/2 < \ln 2 < 1$,

$$\frac{1}{2} \cdot m \cdot 2^m \leq w \leq m \cdot 2^m \tag{181}$$

for all sufficiently large m. Recalling that $N(d,m) = m \cdot \widehat{w_0} \cdot w^{d-2}$, we obtain

$$N(d,m) \geq m\, 2^{m-1} \left(\frac{1}{2} m 2^m\right)^{d-2} = 2^{-(d-1)} m^{d-1} 2^{(d-1)m}, \tag{182}$$

$$N(d,m) \leq m\, 2^m (m 2^m)^{d-2} = m^{d-1} 2^{(d-1)m}. \tag{183}$$

This proves (177). In particular, $N(d,m) = \Theta_d\Big(m^{d-1}2^{(d-1)m}\Big)$, $\log N(d,m) = \Theta_d(m)$, and all fan-ins and $N(d,m)$ have $O_d(m)$ bits. □

Lemma 70. *For fixed d,*

$$\frac{N(d,m)}{N_{RST}(d,m)} = 1 + O_d\Big(2^{-m/20}\Big). \tag{184}$$

In particular $N(d,m) \leq 2N_{RST}(d,m)$ and $w_0 \geq 2^{m-2}$ for all sufficiently large m.

Proof. The two formulas differ only in their top fan-in, so

$$\frac{N(d,m)}{N_{RST}(d,m)} = \frac{\widehat{w}_0}{w_0}.$$

By (180) and Lemma 68, $w_0 \geq \widehat{w}_0 - \Delta_m \geq 2^{m-1} - C_d \cdot 2^{\frac{19}{20}m} \geq 2^{m-2}$ for all sufficiently large m. Hence

$$\left|\frac{\widehat{w}_0}{w_0} - 1\right| \leq \frac{\Delta_m}{w_0} \leq \frac{C_d \cdot 2^{19m/20}}{2^{m-2}} = 4C_d \cdot 2^{-m/20}.$$

which is (184). □

The most important result is that the reparameterized class of formulas can be coupled with the original one:

Lemma 71. *There is a way to couple (some of) the variables in the two formulas so that, if X denotes the variables in common to both formulas, Y the set of variables private to the first formula and Z the set of variables private to the second formula*

$$\mathbf{Pr}_{X,Y,Z}[\widehat{Sipser}_{d,m}(X,Y) \neq Sipser_{d,m}(X,Z)] \leq 2^{-\Omega_d(m)} \tag{185}$$

Proof. Define the following coupling:

Definition 72. Put $r = \min(w_0, \widehat{w}_0)$. For each $1 \leq i \leq r$, identify the ith root child of $\mathrm{Sipser}_{d,m}$ with the ith root child of $\widehat{\mathrm{Sipser}}_{d,m}$, including all leaves below it. Assign the same independent unbiased bits to the corresponding leaves. The $|\widehat{w}_0 - w_0|$ remaining root children of whichever formula has the larger top fan-in are left *unmatched*; assign independent unbiased bits to every leaf in each unmatched subtree.

In the following result we use the following conventions: an input to a gate *activates its parent* if the gate is an OR and the input is one or if the gate is an AND and the input is zero.

Claim 73. *Let p_h be the probability that a height-h non-root RST subformula takes the value activating its parent. For fixed d,*

$$p_h = \rho\left(1 + O_d(m^{2d}\rho)\right) \qquad (1 \leq h \leq d-1). \tag{186}$$

In particular,

$$p_{d-1} \leq 2\rho \tag{187}$$

for all sufficiently large m.

Proof. We prove the following stronger assertion by induction on h: for every fixed $h \geq 1$ there is a constant $C_h > 0$ such that, for all sufficiently large m,

$$p_h = \rho(1+\varepsilon_h), \qquad |\varepsilon_h| \leq C_h m^{h-1}\rho. \tag{188}$$

The constants and the threshold for m may depend on h. Since only $1 \leq h \leq d-1$ are considered and d is fixed, they may eventually be chosen uniformly in terms of d.

For $h = 1$ we have $p_1 = 2^{-m} = \rho$, so (188) holds with $\varepsilon_1 = 0$.

Now let $2 \leq h \leq d-1$ and suppose that $p_{h-1} = \rho(1+\varepsilon_{h-1}), |\varepsilon_{h-1}| \leq C_{h-1}m^{h-2}\rho$. Since $m^{h-2}\rho = m^{h-2}2^{-m} = o(1)$, after increasing the threshold for m we have $|\varepsilon_{h-1}| \leq 1/2$ and hence $0 \leq p_{h-1} \leq \frac{3}{2}\rho \leq 2\rho$. Thus $p_{h-1} \leq 1/2$ for sufficiently large m, and we may use $\ln(1-x) = -x + O(x^2), (0 \leq x \leq 1/2)$, with an absolute implied constant.

Write $w = m(\ln 2)\rho^{-1} + \theta_m, |\theta_m| \leq 1$, which follows from $w = \lfloor m2^m \ln 2 \rfloor$ and $\rho = 2^{-m}$. Using the recursion $p_h = (1-p_{h-1})^w$, we obtain

$$\begin{aligned}\ln\frac{p_h}{\rho} &= w\ln(1-p_{h-1}) + m\ln 2 = -wp_{h-1} + m\ln 2 + O(wp_{h-1}^2) = \\ &= -m(\ln 2)\varepsilon_{h-1} - \theta_m\rho(1+\varepsilon_{h-1}) + O(w\rho^2).\end{aligned}$$

Here we used $p_{h-1} = O(\rho)$. Moreover, $w\rho^2 = \big(m(\ln 2)\rho^{-1} + O(1)\big)\rho^2 = O(m\rho)$. Consequently,

$$\ln\frac{p_h}{\rho} = O_h\big(m|\varepsilon_{h-1}|\big) + O(m\rho) = O_h\big(m^{h-1}\rho\big).$$

Because $m^{h-1}\rho = o(1)$, exponentiation and $e^x = 1 + O(x)$ for $x = o(1)$ give

$$p_h = \rho \cdot \exp\big(O_h(m^{h-1}\rho)\big) = \rho\big(1 + O_h(m^{h-1}\rho)\big).$$

This proves (188) and completes the induction.

Finally, since $1 \leq h \leq d-1$ and d is fixed, the finitely many implied constants may be replaced by a single constant depending only on d. Furthermore, $m^{h-1}\rho \leq m^{d-2}\rho \leq m^{2d}\rho$. Therefore

$$p_h = \rho\big(1 + O_d(m^{2d}\rho)\big), \qquad 1 \leq h \leq d-1,$$

which proves (186). Since $m^{2d}\rho = m^{2d}2^{-m} = o(1)$, it also follows that $p_{d-1} \leq 2\rho$ for all sufficiently large m. □

Now we're in position to prove the lemma: given the coupling described in Definition 72, every matched root subtree agrees *identically* with its coupled match, not merely with high probability. Hence the two root outputs can differ only if some *unmatched* root child takes the value activating the root. By Lemma 68 there are at most $\Delta_m = C_d \cdot 2^{\frac{19}{20}m}$ unmatched children, and each takes the activating value with probability $p_{d-1} \leq 2\rho = 2^{1-m}$ by Claim 73. A union bound (no independence is needed) therefore gives

$$\mathbf{Pr}_{X,Y,Z}[\widehat{Sipser}_{d,m}(X,Y) \neq Sipser_{d,m}(X,Z)] \leq \Delta_m \cdot 2^{1-m} = 2C_d \cdot 2^{-m/20}, \tag{189}$$

which proves (185). □

6.4. **Hardness result for family $G_{d,m}$.** The following is the specific hardness result we employ:

Lemma 74 (Hardness of the uniform family $G_{d,m}$)**.** *For every fixed $d \geq 2$, there are constants $\alpha_d, \beta_d > 0$ such that, for all sufficiently large m, every normal form circuit*

$$C : \{0,1\}^{N(d,m)} \longrightarrow \{0,1\}$$

with an AND root, depth at most d and $\mathrm{size}(C) \leq 2^{N(d,m)^{\alpha_d}}$ *satisfies*

$$\mathbf{Pr}[C(U) = G_{d,m}(U)] \leq \frac{1}{2} + N(d,m)^{-\beta_d}. \tag{190}$$

Here $U = (X, Y)$ is a random input in $\{0,1\}^{N(d,m)}$, X consists of the first $min(N(d,m), N_{RST}(d,m))$ bits, and Y of the remaining (if any) bits.

The dual statement holds for $1 - G_{d,m}$ against OR-root circuits.

Proof. We first treat the case where the top gate is an OR, that is $G_{d,m}(U) = \widehat{Sipser}_{d,m}(U)$.

Couple the exact and modified formulas $Sipser_{d,m}(X, Z)$ and $\widehat{Sipser}_{d,m}(X, Y)$ as in Definition 72. Denote their outputs by S and $\widehat{S}$.

On the other hand, circuit $C(X, Y)$ and the formula $Sipser_{d,m}(X, Z)$ may as well have input sets of slightly different sizes, $N(d,m)$ variables for C, $N_{RST}(d,m)$ for $Sipser_{d,m}$. We will couple them as well, as follows:

Claim 75. *We can couple the circuit $C(X, Y)$ and the formula $Sipser_{d,m}(X, Z)$ so that*

$$\mathbf{Pr}[C(X,Y) = Sipser_{d,m}(X,Z)] \leq \frac{1}{2} + N_{RST}(d,m)^{-b_d}, \tag{191}$$

where $b_d > 0$ is supplied by Proposition 38.

Proof. There are two cases:

- If the modified formula $\widehat{Sipser}_{d,m}(X, Y)$ has the unmatched root subtrees (i.e. $Y \neq \emptyset$, $Z = \emptyset$) fix all variables in Y. For each fixing u, the restricted circuit C_u has no larger depth or size. C_u is in normal form (Definition 36), possibly with constant leaves. Thus the bound from Corollary 39 applies to every C_u. Averaging over the independent uniform choice of u yields (191).
- The same reasoning applies when the exact formula $Sipser_{d,m}(X, Z)$, rather than the modified formula $\widehat{Sipser}_{d,m}(X, Y)$, has the unmatched subtrees, that is $Z \neq \emptyset$ and $Y = \emptyset$: in that case $C(X, Y)$ simply ignores the variables in Z, and we apply again Corollary 39.

□

Now triangle inequality gives:

$$\mathbf{1}_{\{C(X,Y)=\widehat{Sipser}_{d,m}(X,Y)\}} \leq \mathbf{1}_{\{C(X,Y)=Sipser_{d,m}(X,Z)\}} + \mathbf{1}_{\{Sipser_{d,m}(X,Z)\neq\widehat{Sipser}_{d,m}(X,Y)\}}, \tag{192}$$

so Lemma 71, in the quantitative form (189), gives

$$\mathbf{Pr}[C(U) = \widehat{Sipser}_{d,m}(U)] \leq \mathbf{Pr}[C(X,Y) = Sipser_{d,m}(X,Z)] + 2C_d \cdot 2^{-m/20}. \tag{193}$$

By Lemma 70 we may take $\alpha_d := a_d/2$: indeed $N(d,m) \leq 2N_{RST}(d,m)$ for all sufficiently large m, so $N(d,m)^{\alpha_d} \leq (2N_{RST}(d,m))^{a_d/2} \leq N_{RST}(d,m)^{a_d}$ and hence $2^{N(d,m)^{\alpha_d}} \leq 2^{N_{RST}(d,m)^{a_d}}$. This justifies the appeal to Corollary 39 in Claim 75. Combining (191) with (193) gives

$$\mathbf{Pr}[C(U) = \widehat{Sipser}_{d,m}(U)] \leq \frac{1}{2} + N_{RST}(d,m)^{-b_d} + 2C_d \cdot 2^{-m/20}. \tag{194}$$

It remains to absorb both error terms into a single power of $N(d,m)$. Set

$$\theta_d := \min\left\{b_d(d-1),\ \tfrac{1}{10}\right\}, \qquad \beta_d := \frac{\theta_d}{8d} > 0. \tag{195}$$

By (178), $\log_2 N(d,m) \leq 2dm$, so $N(d,m)^{-\beta_d} \geq 2^{-2d\beta_d m} = 2^{-\theta_d m/4}$. On the other hand $w_0 \geq 2^{m-2}$ by Lemma 70, so

$$N_{RST}(d,m) = m\, w_0\, w^{d-2} \geq m \cdot 2^{m-2}\left(\tfrac{1}{2}m2^m\right)^{d-2} = 2^{-d} m^{d-1} 2^{(d-1)m},$$

whence $\log_2 N_{RST}(d,m) \geq (d-1)m - d \geq (d-1)m/2$ for $m \geq 4$, and therefore $N_{RST}(d,m)^{-b_d} \leq 2^{-b_d(d-1)m/2} \leq 2^{-\theta_d m/2}$. Since $\theta_d \leq \frac{1}{10}$ we also have $2C_d \cdot 2^{-m/20} \leq 2C_d \cdot 2^{-\theta_d m/2}$. Consequently

$$N_{RST}(d,m)^{-b_d} + 2C_d \cdot 2^{-m/20} \leq 2^{-\theta_d m/2}(1 + 2C_d) \leq 2^{-\theta_d m/4} \leq N(d,m)^{-\beta_d} \tag{196}$$

for every large enough m. This proves (190) for the uncomplemented orientation.

If the formula $Sipser_{d,m}$ has an AND root, complement both exact and modified targets. For an AND-root approximating circuit C, its complement $1 - C$ can be converted using de Morgan's law, to have an OR root, unchanged depth and size, and

$$\mathbf{Pr}[C = 1 - \widehat{Sipser}_{d,m}] = \mathbf{Pr}[1 - C = \widehat{Sipser}_{d,m}].$$

Applying the estimate to the AND-rooted formula $\widehat{Sipser}_{d,m}$ and the OR-rooted circuit $1-C$ (using the same type of couplings described above) proves the normalized claim. □

6.5. **Deriving the Main Theorem from the Hardness Lemma.** Fix $k \geq 1$ and set $d = k + 1$. We construct an infinite language in $\Sigma_k^{\mathsf{P},A}$ that is $\Pi_k^{\mathsf{P},A}$-immune for almost all oracles A.

Recall that, by Lemma 19, for every fixed polynomial q, there is $j_0 \geq 1$ such that

$$q(t_j) < t_{j+1} \qquad \text{for all } j \geq j_0. \tag{197}$$

Let $m_*(d)$ be sufficiently large that Lemmas 67, 68, 70, 74, as well as the estimates (177)–(178), hold for every $m \geq m_*(d)$. Choose

$$J_d := \min\left\{ j \geq 1 : \left\lfloor \frac{t_j}{8d} \right\rfloor \geq m_*(d) \right\}. \tag{198}$$

For every $j \geq J_d$, define

$$m_j = \left\lfloor \frac{t_j}{8d} \right\rfloor \tag{199}$$

and write

$$N_j = N(d, m_j). \tag{200}$$

By (177), there is a constant $\gamma_d > 0$ such that for $j \geq J_d$

$$2^{\gamma_d t_j} \leq N_j \leq 2^{t_j/2} < 2^{t_j}. \tag{201}$$

We may therefore identify the N_j leaves of G_{d,m_j} with a uniformly computable set $V_j \subseteq \{0,1\}^{t_j}$, $|V_j| = N_j$. For example, order the leaves lexicographically by their tuples of child indices and map the ℓth leaf to the t_j-bit binary representation of ℓ. Since $\log N_j = O(t_j)$ and d is fixed, this address is computable in time polynomial in t_j.

For an oracle A and $j \geq J_d$, define

$$X_j^A = G_{d,m_j}((A(z))_{z \in V_j}) \tag{202}$$

and

$$L_k^A = \left\{ 1^{t_j} : X_j^A = 1, j \geq J_d. \right\} \tag{203}$$

Lemma 76. *For every oracle A,*

$$L_k^A \in \Sigma_k^{\mathsf{P},A}. \tag{204}$$

Proof. By Lemma 18, the support $T_d = \{1^{t_j} : j \geq J_d\}$ is decidable in polynomial time. Thus it remains to evaluate G_{d,m_j} on input 1^{t_j}.

The formula has depth $d = k+1$ and an OR root. The machine existentially chooses a child of the root, universally chooses a child of the next layer, and continues alternating in this fashion. After $d - 1 = k$ such choices, a bottom gate has been reached. Its $m_j = O(t_j)$ leaves are then evaluated deterministically by computing their addresses in V_j, querying A, and evaluating the bottom AND or OR gate. All choices have length $O(t_j)$.

One aspect to take care of in the algorithm above is that of invalid (guessed) indices. By Lemma 69 they can be recognized in polynomial time, and are handled as follows:

- on existential levels invalid indices lead to rejection.
- on universal levels invalid indices lead to acceptance.

This is a $\Sigma_k^{\mathsf{P},A}$ computation. □

Lemma 77. *Put*

$$\varepsilon_{d,m} = N(d,m)^{-\beta_d}. \tag{205}$$

Then, for $m \geq m_*(d)$,

$$\left|\mathbf{Pr}_X[G_{d,m}(X)=1] - \frac{1}{2}\right| \leq \varepsilon_{d,m}. \tag{206}$$

Proof. Indeed, apply Lemma 74 to the constant zero and constant one functions, using the gadgets in Remark 37 if necessary to obtain the required syntactic depth and top-gate orientation. □

Lemma 78. *Define $\varepsilon_j = N_j^{-\beta_d}$. Then:*

$$\sum_{j=J_d}^{\infty} \varepsilon_j < \infty. \tag{207}$$

Proof. By (201),

$$\varepsilon_j \leq 2^{-\beta_d \gamma_d t_j} = 2^{-\beta_d \gamma_d \cdot 2^{j!}} \tag{208}$$

The sum of right-hand bounds is finite since, e.g. $2^{j!} \geq j!$. □

Lemma 79. *For every $k \geq 1$ the language L_k^A is almost surely infinite.*

Proof. Indeed, the blocks V_j are pairwise disjoint, so the variables X_j^A are independent. By (206),

$$\mathbf{Pr}[X_j^A = 1] = \frac{1}{2} \pm \varepsilon_j. \tag{209}$$

In particular,

$$\mathbf{Pr}[X_j^A = 1] \geq \frac{1}{3} \tag{210}$$

for all sufficiently large j. The ordinary second Borel–Cantelli lemma therefore gives

$$\mathbf{Pr}[X_j^A = 1 \text{ infinitely often}] = 1. \tag{211}$$

□

It remains to prove immunity:

Lemma 80. *For each fixed Π_k^P clocked oracle machine M,*

$$\mathbf{Pr}_A\left[L(M^A) \subseteq L_k^A \Longrightarrow L(M^A) \text{ is finite}\right] = 1. \tag{212}$$

Proof. There is a polynomial q_M such that on every input x the acceptance predicate of $M^A(x)$, viewed as a Boolean function of the oracle bits, is represented by a depth-d circuit $C_{M,x}$ satisfying

$$\mathrm{depth}(C_{M,x}) = d = k+1, \tag{213}$$

$$\mathrm{top}(C_{M,x}) = \mathrm{AND}, \tag{214}$$

$$\mathrm{size}(C_{M,x}) \leq 2^{q_M(|x|)}, \tag{215}$$

$$|z| \leq q_M(|x|) \quad \text{for every oracle variable } A(z) \text{ occurring in } C_{M,x}. \tag{216}$$

Fix the machine M. After padding with unused quantified bits if necessary, there are a polynomial p, polynomially bounded strings $y_1, \ldots, y_k$, and a deterministic polynomial-time oracle machine R such that

$$M^A(x) = 1 \quad \Longleftrightarrow \quad \forall y_1\, \exists y_2 \cdots Q_k y_k\; R^A(x, y_1, \ldots, y_k) = 1,$$

where the quantifiers alternate, beginning universally. The running time, number of oracle queries, and every query length of R are at most $p(|x|)$ after enlarging p.

For fixed $x, y_1, \dots, y_k$, expand the adaptive computation of R^A into its binary oracle decision tree. This tree has depth at most $p(|x|)$ and at most $2^{p(|x|)}$ leaves. It can be expressed either as a DNF, by taking one term for every accepting leaf, or as a CNF, by taking one clause for every rejecting leaf. Repeated oracle variables may be retained or identified; in either case all negations occur only on input literals.

Choose the DNF representation when $Q_k = \exists$, and the CNF representation when $Q_k = \forall$. Its top gate then has the same type as the gate representing the last quantifier and can be merged with that gate. The remaining term or clause layer has the opposite type. The k quantifier layers together with this final term/clause layer therefore form an alternating circuit of depth exactly $k + 1 = d$, with an AND root. All gates at a fixed level have the same type. Constant decision-tree leaves are permitted by Definition 36; vacuous quantifier blocks and degenerate formulas can be padded by unary gates or the constant gadgets of Remark 37 without changing the root type.

The number of assignments to all quantified strings and the total size of all decision-tree formulas are $2^{p'(|x|)}$ for a polynomial p' depending only on M. Any remaining layered-normal-form conversion has at most a fixed-degree polynomial cost and can be absorbed by enlarging p'. Consequently there is a polynomial q_M for which

$$\operatorname{depth}(C_{M,x}) = k + 1, \qquad \operatorname{top}(C_{M,x}) = \text{AND}, \qquad \operatorname{size}(C_{M,x}) \le 2^{q_M(|x|)},$$

and every oracle variable $A(z)$ occurring in $C_{M,x}$ satisfies $|z| \le q_M(|x|)$.

By (197), for $j \ge j_0$, $q_M(t_j) < t_{j+1}$. Moreover, (201) implies $q_M(t_j) < N_j^{\alpha_d}$ for all sufficiently large j, and hence

$$2^{q_M(t_j)} \le 2^{N_j^{\alpha_d}}. \tag{217}$$

Define

$$\mathcal{F}_j = \sigma\big(A(z) : |z| < t_{j+1}\big), \tag{218}$$

$$\mathcal{G}_j = \sigma\big(A(z) : |z| < t_{j+1},\ z \notin V_j\big). \tag{219}$$

Then

$$\mathcal{F}_{j-1} \subseteq \mathcal{G}_j \subseteq \mathcal{F}_j \subseteq \mathcal{G}_{j+1} \tag{220}$$

The last inclusion follows since V_{j+1} consists of strings of length exactly t_{j+1}, whereas $\mathcal{F}_j$ sees only strings of shorter length.

Conditional on $\mathcal{G}_j$, the bits $(A(z))_{z \in V_j}$ remain mutually independent unbiased bits. Let

$$E_j = \{M^A(1^{t_j}) \text{ accepts}\}. \tag{221}$$

By (197), $E_j \in \mathcal{F}_j$.

After conditioning on $\mathcal{G}_j$, all oracle variables of $C_{M,1^{t_j}}$ outside V_j become constants. We perform this restriction syntactically, without simplifying away gates. Consequently the alternating layered structure and the AND root are preserved. Thus, on each atom of $\mathcal{G}_j$, E_j is represented as a function of the fresh block V_j by a normal-form depth-d, AND-root circuit of size at most $2^{N_j^{\alpha_d}}$. Lemma 74 therefore yields

$$\mathbf{Pr}[\mathbf{1}_{E_j} = X_j^A \mid \mathcal{G}_j] \le \frac{1}{2} + \varepsilon_j, \qquad \varepsilon_j = N_j^{-\beta_d}. \tag{222}$$

At the same time,

$$\left|\mathbf{Pr}[X_j^A = 1 \mid \mathcal{G}_j] - \frac{1}{2}\right| \le \varepsilon_j. \tag{223}$$

For $j \geq J_d$ set

$$P_j = \mathbf{Pr}[E_j \mid \mathcal{G}_j], \tag{224}$$

$$B_j = E_j \cap \{X_j^A = 0\}, \tag{225}$$

$$Q_j = \mathbf{Pr}[B_j \mid \mathcal{G}_j]. \tag{226}$$

The event B_j is a false positive: M accepts 1^{t_j} although $1^{t_j} \notin L_k^A$.

Let $J = J(M, d) \geq J_d$ be large enough so that (197), (217) and all asymptotic hardness results hold for $j \geq J$. Applying the rare-or-wrong inequality (Lemma 10) to $\mathbf{1}_{E_j}$ and X_j^A, $j \geq J$, implies

$$Q_j \geq \frac{1}{2} P_j - \varepsilon_j, \qquad \sum_{j \geq J} \varepsilon_j < \infty. \tag{227}$$

On the interlaced filtration $\mathcal{H}_{2j-2} = \mathcal{F}_{j-1}$, $\mathcal{H}_{2j-1} = \mathcal{G}_j$, $\mathcal{H}_{2j} = \mathcal{F}_j$, apply Lévy's lemma to the sequences $D_{2j} = E_j, D_{2j-1} = \varnothing$ and $D'_{2j} = B_j, D'_{2j-1} = \varnothing$. On a common event of probability one:

- if $\sum_{j \geq J} P_j < \infty$, only finitely many E_j occur;
- if $\sum_{j \geq J} P_j = \infty$, then $\sum_{j \geq J} Q_j = \infty$, so infinitely many B_j occur.

In the first case, $L(M^A) \subseteq L_k^A$ forces $L(M^A)$ to be finite because no word outside the support can be accepted. In the second case, B_j gives a false positive, so $L(M^A) \not\subseteq L_k^A$. Therefore

$$\mathbf{Pr}_A\big[L(M^A) \subseteq L_k^A \Longrightarrow L(M^A) \text{ is finite}\big] = 1. \tag{228}$$

□

There are only countably many syntactically clocked Π_k^P machines. Intersecting their probability-one events and using (211), we obtain

$$\mathbf{Pr}_A\left[L_k^A \in \mathbf{\Sigma}_k^{\mathsf{P},A} \text{ and } L_k^A \text{ is } \mathbf{\Pi}_k^{\mathsf{P},A}\text{-immune}\right] = 1. \tag{229}$$

For the second separation we consider the symmetric witness

$$R_k^A = \left\{1^{t_j} : X_j^A = 0, j \geq J_d.\right\} \tag{230}$$

Its target function on V_j is $1 - G_{d,m_j}$, an alternating depth-d formula with an AND root. Hence $R_k^A \in \mathbf{\Pi}_k^{\mathsf{P},A}$. For a Σ_k^P oracle machine the corresponding restricted circuit has an OR root, so the dual part of Lemma 74 applies. The entire rare-or-wrong, filtration, and conditional Borel-Cantelli argument is unchanged. Consequently

$$\mathbf{Pr}_A\left[R_k^A \in \mathbf{\Pi}_k^{\mathsf{P},A} \text{ and } R_k^A \text{ is } \mathbf{\Sigma}_k^{\mathsf{P},A}\text{-immune}\right] = 1. \tag{231}$$

Finally, there are only countably many integers $k \geq 1$. Intersecting the probability-one events (229) and (231) over all k proves Theorem 1 simultaneously at every level of the polynomial hierarchy.

7. Conclusions and Open Problems

Although a single general metatheorem covering all the applications of our general method may be hard to state and prove, we believe that (if you have made it thus far in the paper) you may be convinced that the approach we proposed is reasonably flexible, and can conceivably be employed to upgrade many other separation results under a random oracle to separations with immunity.

It is also somewhat instructive to scrutinize the limits of our method. Its clearest failure was Case (b) of Observation 9, but there isn't that much to learn from this particular case: there the obstruction was structural, rather than a limitation of our rare-or-wrong approach. Indeed, every language in the Boolean hierarchy is a finite union of DP languages (and this remains true relative to every oracle). Consequently, every infinite language in the Boolean hierarchy contains an infinite DP sublanguage, making the corresponding immunity statement impossible, independently of the choice of oracle or of the proof technique. This contrasts with the situation of Theorem 7, where

no analogous structural obstruction is present: there the difficulty is probabilistic, and consists in making sure that the exceptional events in the rare-or-wrong argument have summable probabilities. Thus, while the method cannot turn every random-oracle separation into a separation with immunity, its failures may reflect either genuine structural barriers or insufficient *local* hardness.

On the other hand the method, as described in this paper, does not currently produce the strongest possible separations. Hemaspaandra and Zimand [28] have studied the concept of *balanced immunity* and have shown that with probability 1 (relative to a random oracle A) NP^A contains a set that is P^A-balanced-immune.

The witness sets constructed in this paper are not even biimmune, much less balanced immune. The reason is that they include in the witnessing languages words of type 1^{t_j} only. The complement of the set $T = \{1^{t_j} : j \geq 1\}$ is polynomial time decidable and included in the complement of the constructed languages L^A. It remains to be seen whether Hemaspaandra and Zimand's more complicated approach to witness language construction can be combined with the "rare-or-wrong" technique from this paper to obtain the given separations with balanced immunity, when appropriate[10].

Another interesting open problem concerns *generic oracle separations*. However, generic separations *with immunity* are a topic with very few results: practically the only one we know is from the same paper of Hemaspaandra and Zimand [28].

On the other hand, a recent exciting approach has connected separations with respect to a generic oracle in (the different) *Ellentuck topology* [16] to unconditional separations. It is quite interesting (in this author's opinion) that methods based on Ramsey Theory and Descriptive Set Theory may have implications for separating complexity classes. Whether a version of this phenomenon extends to separations *with immunity* is a fascinating (and open) research question.

8. Acknowledgments

I acknowledge the use of AI assistants (precise names available on request) to help me oversell the results of this paper, for idea exploration, and for editorial assistance as a "virtual referee"[11]. Said LLMs and their parent companies are **not** responsible, however, for several aspects of the paper, including (i) the main technical ingredient of this paper, the "rare-or-wrong" approach, which existed in a preliminary form (Lemma 10, and the proof of Theorem 2) before LLMs were a thing; (ii) the results in this paper and the writing of the proofs, for which the author assumes full responsibility; (iii) some personal touches in the writing.

On the other hand, I would like to thank all the *people* whose results motivated this paper[12], especially Pavan Aduri, Jin-Yi Cai, Lance Fortnow, the late Juris Hartmanis, Lane Hemaspaandra, Elvira Mayordomo, Ken Regan, Jörg Rothe, Ben Rossman, the late Alan Selman, Marius Zimand. I have had the good fortune to meet every one of the people listed above, and to interact substantially with (and learn from) many of them.

[10]Not all strong separations can extend to separations with biimmunity: for instance an infinite set in $\Sigma_k^{P,A}$ cannot be $\Pi_k^{P,A}$-biimmune.

[11]In particular, the most significant contribution of the LLMs was to single out the issue of polynomial time computability of the fan-ins in an attempted proof of our Theorem 1 using the original family of Sipser functions from [41], and to suggest a way around it by employing an approximation of the top fan-in; the suggested fix was incomplete (it did not realize that the number of variables of the reparameterized formulas was different from those of the original ones, and was missing the coupling lemma, whose idea was supplied by the author) and more complicated: the approximation proposed by the LLM guaranteed that $|\widehat{w_0} - w_0| \leq 1$ and ensured polynomial time computability of $\widehat{w_0}$ by a rather baroque argument; the author realized that the coupling lemma allowed a looser approximation $\widehat{w_0}$, with a simplified complexity argument.

[12]I first asked myself the question settled by Theorem 1 in 1995, reading [46] as a graduate student; this is a paper I wish I had written back then.

References

[1] K. Alladi and M. L. Robinson. *Legendre polynomials and irrationality.* J. Reine Angewandte Mathematik, 318 (1980), pp. 137–155.

[2] S. Arora and B. Barak. *Computational complexity: a modern approach.* Cambridge University Press, 2009.

[3] J. Aspnes, R. Beigel, M. Furst, and S. Rudich. *The expressive power of voting polynomials*, Combinatorica **14** (1994), no. 2, pp. 135–148.

[4] L. Babai. *Random oracles separate PSPACE from the polynomial-time hierarchy*, Information Processing Letters **26** (1987), pp. 51–53. DOI: 10.1016/0020-0190(87)90036-6.

[5] J.L. Balcázar and U. Schöning. *Bi-immune sets for complexity classes.* Mathematical Systems Theory 18(1), 1985, pp. 1–10.

[6] R. Beigel. *Relativized counting classes: Relations among thresholds, parity, and mods.* Journal of Computer and System Sciences, 1991, 42.1: 76-96.

[7] C.H. Bennett and J. Gill. *Relative to a Random Oracle A, $P^A \neq NP^A \neq co\text{-}NP^A$ with Probability 1.* SIAM Journal on Computing, 10(1), 1981, pp. 96–113. DOI: 10.1137/0210008.

[8] D. Bruschi, *Strong separations of the polynomial hierarchy with oracles: constructive separations by immune and simple sets*, Theoretical Computer Science **102**(2) (1992), 215–252. DOI: 10.1016/0304-3975(92)90232-5.

[9] J.-Y. Cai. *With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy*, Journal of Computer and System Sciences **38** (1989), no. 1, pp. 68–85.

[10] J.-Y. Cai. *Probability One Separation of the Boolean Hierarchy*, In: Brandenburg, F.J., Vidal-Naquet, G., Wirsing, M. (eds) Proceedings of STACS 87. Lecture Notes in Computer Science, vol 247, pp. 148–158. Springer Verlag. doi:10.1007/BFb0039602

[11] J-Y. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner and G. Wechsung. *The Boolean Hierarchy I: Structural Properties.* SIAM Journal on Computing, 17(6), 1988, pp. 1232–1252.

[12] J-Y. Cai and L. Hemachandra. *The Boolean hierarchy: hardware over NP.* In: Selman, A.L. (eds) Structure in Complexity Theory. Lecture Notes in Computer Science, vol 223, pp. 105-124. Springer Verlag. doi:10.1007/3-540-16486-3_93

[13] J-Y. Cai, A. Selman. *Fine separation of average-time complexity classes.* SIAM Journal on Computing 28(4), 1999, pp. 1310–1325.

[14] T. K. Chandra. *The Borel-Cantelli Lemma*, SpringerBriefs in Statistics, Springer, 2012.

[15] R. Chang, B. Chor, O. Goldreich, J. Hartmanis, J. Håstad, D. Ranjan, and P. Rohatgi. *The Random Oracle Hypothesis is false.* Journal of Computer and System Sciences, **49**(1) (1994), 24–39. DOI: 10.1016/S0022-0000(05)80084-4.

[16] A. Creiner and S. Jackson. *Borel complexity and Ramsey largeness of sets of oracles separating complexity classes.* Mathematical Logic Quarterly 69.3 (2023): 267-286.

[17] R. Durrett. *Probability. Theory and Examples*, 4th Edition, Cambridge University Press, 2010.

[18] L. Fortnow, A. Pavan, and A. L. Selman. *Distributionally hard languages.* Theory of Computing Systems 34(3), 2001, pp. 245–261.

[19] R. Ghosal, Y. Ishai, A. Korb, E. Kushilevitz, P. Lou, and A. Sahai. *Hard languages in $NP \cap coNP$ and NIZK proofs from unstructured hardness.* in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC 2023)*, Association for Computing Machinery, New York, 2023, pp. 1243–1256. DOI: 10.1145/3564246.3585119. Full version: Cryptology ePrint Archive, Report 2023/1972.

[20] O. Goldreich, *Foundations of Cryptography, Volume 1: Basic Tools*, Cambridge University Press, Cambridge, 2001. DOI: 10.1017/CBO9780511546891.

[21] O. Goldreich, *On one-way permutations (a clarification).* technical note, Weizmann Institute of Science. Available from `https://www.wisdom.weizmann.ac.il/~oded/owp.html`.

[22] O. Goldreich, L. A. Levin, and N. Nisan, *On constructing 1-1 one-way functions.* in O. Goldreich (ed.), Studies in Complexity and Cryptography: Miscellanea on the Interplay between Randomness and Computation, Lecture Notes in Computer Science, vol. 6650, Springer, 2011, pp. 13–25. DOI: 10.1007/978-3-642-22670-0_3.

[23] S. Goldwasser, S. Micali, and C. Rackoff, *The knowledge complexity of interactive proof systems.* SIAM Journal on Computing, **18**(1) (1989), 186–208. DOI: 10.1137/0218012.

[24] S. Goldwasser and M. Sipser. *Private coins versus public coins in interactive proof systems* in Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC 1986), pp. 59–68. DOI: 10.1145/12130.12137.

[25] J. Håstad. *Computational Limitations for Small-Depth Circuits.* M.I.T. Press, 1987.

[26] J. Håstad. *On the correlation of parity and small-depth circuits.* SIAM Journal on Computing 43.5 (2014): 1699-1708.

[27] J. Håstad, B. Rossman, R. Servedio and L.Y. Tan. *An average-case depth hierarchy theorem for boolean circuits.* Journal of the A.C.M., 64(5), 2017, pp. 1–27.

[28] L. Hemaspaandra and M. Zimand. *Strong self-reducibility precludes strong immunity.* Mathematical systems theory 29.5 (1996): 535-548.
[29] J. Hitchcock, A. Pavan, and N. V. Vinodchandran. *Partial bi-immunity, scaled dimension, and NP-completeness.* Theory of Computing Systems 42(2), 2008, pp. 131–142.
[30] S. Homer and W. Maass. *Oracle-dependent properties of the lattice of NP sets.* Theoretical Computer Science 24.3 (1983), pp. 279–289.
[31] G. Istrate. *Game-theoretic Models of Moral and Other-Regarding Agents (extended abstract).* Proceedings of TARK 2021, 18th Conference on Theoretical Aspects of Rationality and Knowledge, Electronic Proceedings of Theoretical Computer Science, vol. 335. pp. 217–227.
[32] C. G. Jockusch, Jr. *The degrees of bi-immune sets.* Mathematical Logic Quarterly 15.7-12 (1969), pp. 135–140.
[33] J. Katz and Y. Lindell. *Introduction to Modern Cryptography, 3rd ed.* CRC Press, 2020.
[34] Ker-I Ko. *A note on separating the relativized polynomial time hierarchy by immune sets.* RAIRO. Informatique théorique et applications 24.3 (1990): pp. 229–240.
[35] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. *Algebraic methods for interactive proof systems*, Journal of the A.C.M. **39**(4) (1992), 859–868. DOI: 10.1145/146585.146605.
[36] E. Mayordomo. *Almost every set in exponential time is P-bi-immune.* Theoretical Computer Science 136(2), 1994, pp. 487–506.
[37] A. Pavan and A.L. Selman. *Complete distributional problems, hard languages, and resource-bounded measure.* Theoretical Computer Science 234.1-2 (2000): 273-286.
[38] A. Pavan and A.L. Selman. *Bi-immunity separates strong NP-completeness notions.* Information and Computation 188(1), 2004, pp. 116–126.
[39] E. L. Post. *Recursively enumerable sets of positive integers and their decision problems.* Bulletin of the A.M.S. 50(5), 1944, pp. 284–316.
[40] K. W. Regan and J. Royer. *On Closure Properties of Bounded Two-Sided Error Complexity Classes.* Mathematical Systems Theory **28**, 1995, pp. 229–243.
[41] B. Rossman, R. A. Servedio, and L.-Y. Tan, *An Average-Case Depth Hierarchy Theorem for Boolean Circuits.* In Proceedings of the 56th Annual Symposium on Foundations of Computer Science (FOCS'15), 2015, pp. 1030-1048. Full version: `https://arxiv.org/abs/1504.03398v1`.
[42] B. Rossman, R. Servedio, and L.-Y. Tan. *The Polynomial Hierarchy, Random Oracles, and Boolean Circuits.* ACM SIGACT News 46(4), 2015, pp. 50–68. DOI: 10.1145/2852040.2852052.
[43] Jörg Rothe. *Immunity and Simplicity for Exact Counting and Other Counting Classes.* RAIRO 33(2) (1999), pp. 159–176.
[44] A. Shamir, *IP = PSPACE*, Journal of the A.C.M.**39**(4) (1992), 869–877. DOI: 10.1145/146585.146609.
[45] G. Tardos, *Query complexity, or why is it difficult to separate $NP^A \cap coNP^A$ from P^A by random oracles A?. Combinatorica* **9**(4) (1989), 385–392. DOI: 10.1007/BF02125350.
[46] N. K. Vereshchagin. *Relationships between NP-sets, co-NP-sets, and P-sets relative to random oracles.* In Proceedings of the Eighth Annual Structure in Complexity Theory Conference. IEEE, 1993, pp. 132-138. doi:10.1109/SCT.1993.336533
[47] N.K. Vereshchagin, *NP-sets are Co-NP-immune relative to a random oracle.* Proceedings Third Israel Symposium on the Theory of Computing and Systems, pp. 40–45, 1995. doi:10.1109/ISTCS.1995.377048.

Faculty of Mathematics and Computer Science, University of Bucharest, Bucharest, Romania
Email address: `gabriel.istrate@unibuc.ro`

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il