Complexity Theory Column 89: The Polynomial Hierarchy, Random Oracles, and Boolean Circuits[1]

Benjamin Rossman, Rocco A. Servedio, Li-Yang Tan

Abstract

We give an overview of a recent result [RST15] showing that the polynomial hierarchy is infinite relative to a random oracle. Since the early 1980s it has been known that this result would follow from a certain "average-case depth hierarchy theorem" for Boolean circuits. In this article we present some background and history of related relativized separations; sketch the argument showing how the polynomial hierarchy result follows from the circuit lower bound; and explain the techniques underlying the new circuit lower bound.

1 Introduction

An overarching goal in complexity theory is the classification of computational problems according to their inherent difficulty. One of the most intensively studied such classifications is provided by the *polynomial hierarchy*, which was introduced by Albert Meyer and Larry Stockmeyer in 1972 [MS72] (see also [Sto76, Wra76]). Informally, this hierarchy classifies problems according to a natural notion of logical complexity, and is defined with an infinite number of levels: problems at the zeroth level are the "easiest", and for every integer k, problems at the $(k+1)$-st level have logical complexity "one notch higher" than those at level k. Formally, we define $\Sigma_0^{\mathrm{P}} := \mathsf{P}$, and for every $k \in \mathbb{N}$, a language L is in the k-th level Σ_k^{P} of the hierarchy iff there exists a polynomial-time Turing machine M and a polynomial p such that

$$x \in L \iff \exists\, y_1 \in \{0,1\}^{p(|x|)}\ \forall\, y_2 \in \{0,1\}^{p(|x|)} \cdots Q_k\, y_k \in \{0,1\}^{p(|x|)}\ M(x, y_1, \ldots, y_k) = 1,$$

where Q_k is $\exists$ if k is odd, and $\forall$ if k is even. The polynomial hierarchy is $\mathsf{PH} := \bigcup_{k \in \mathbb{N}} \Sigma_k^{\mathrm{P}}$, the class of all languages that can be defined with a constant number of alternating quantifiers.

Recall that $\Sigma_1^{\mathrm{P}} = \mathsf{NP}$, the class of languages involving a single existential quantifier: for example, the Boolean Satisfiability problem asks if there exists a satisfying assignment to a Boolean formula, and the Traveling Salesman problem asks if there exists a short tour visiting every city once. Higher levels of the polynomial hierarchy capture natural decision and optimization problems that are not known to be in NP. For example, the FORMULA MINIMIZATION problem is in Σ_2^{P} since it asks a question with two quantifiers: "Given a Boolean formula ϕ and an integer k, does there exist a formula φ of size at most k such that for all assignments x, we have $\phi(x) = \varphi(x)$?" As another example, consider the VC DIMENSION problem: "Given a collection $\mathcal{S} = \{S_1, \ldots, S_m\}$ of subsets of a finite set U (concisely represented by $\mathrm{poly}(m)$ bits) and an integer k, does there exist a subset $Y \subseteq U$ of size at least k, such that for all $X \subseteq Y$, there exists $i \in [m]$ satisfying $S_i \cap Y = X$"? This problem is in Σ_3^{P} since it asks a question with three quantifiers. Indeed, FORMULA MINIMIZATION is Σ_2^{P}-complete [BU11] and VC DIMENSION is Σ_3^{P}-complete [Sch99]; see Schäfer and Umans's surveys [SU02a, SU02b] (the 37th and 38th edition of this Complexity Theory Column) for a compendium of problems that are complete for various levels of the polynomial hierarchy.

A central conjecture in complexity theory posits a far-reaching generalization of $\mathsf{P} \neq \mathsf{NP}$ (i.e. $\Sigma_0^{\mathrm{P}} \neq \Sigma_1^{\mathrm{P}}$): *all* the infinitely many levels of the polynomial hierarchy are distinct. Just as BOOLEAN SATISFIABILITY is conjectured not to be in P, it is conjectured that FORMULA MINIMIZATION is not in NP, VC DIMENSION is not in Σ_2^{P}, and so on: for every $k \in \mathbb{N}$, it is conjectured that Σ_k^{P}-complete problems are not in $\Sigma_{k-1}^{\mathrm{P}}$. Several important results in complexity theory are built on this conjecture: for example, if the polynomial hierarchy is indeed infinite then NP does not have small circuits (i.e. $\mathsf{NP} \not\subseteq \mathsf{P/poly}$) [KL80] and GRAPH ISOMORPHISM is not NP-complete [BHZ87, Sch88].

1.1 The polynomial hierarchy in relativized worlds

More than forty years after Meyer and Stockmeyer's paper, we remain far from separating even the zeroth and first levels of the hierarchy (showing $\mathsf{P} \neq \mathsf{NP}$), much less showing that all its levels are distinct. However, there has been significant success in understanding the structure of the hierarchy in *relativized* worlds. The following question, attributed to Meyer, was listed as a main open problem in the seminal paper of Baker, Gill, and Solovay [BGS75] which initiated the study of relativization:

Meyer's Question. *Is there a relativized world within which the polynomial hierarchy is infinite?*

Meyer's question quickly became the central open problem in relativized complexity. However progress was modest for a decade, with only the first three levels shown to be distinct [BGS75, BS79]. In 1985 breakthrough results of Andrew Yao [Yao85] and Johan Håstad [Hås86a] finally answered Meyer's question in the affirmative: there is an oracle A relative to which $\Sigma_{k-1}^{\mathrm{P}} \neq \Sigma_k^{\mathrm{P}}$ for all $k \in \mathbb{N}$. (We refer the reader to [Kol85], written shortly after the announcement of [Yao85], for a popular account of this result and the surrounding excitement then.)

The polynomial hierarchy relative to a random oracle. Yao and Håstad's celebrated result may be viewed as an important piece of evidence in favor of the conjecture that the hierarchy is infinite in our actual, oracle-less world. However, it does not provide much information about the oracle witnessing this separation… could it be that this oracle is a particularly degenerate one, craftily engineered so that the hierarchy has infinitely many distinct levels in this atypical relativized world? Could the truth actually be the opposite relative to most other oracles, in most other relativized worlds? Indeed, in [Hås86a, Hås86b, Hås89] Håstad asked whether his result can be strengthened to address these concerns: is the hierarchy infinite relative to not just *some* oracle, but a *random* oracle?

In this column we give an overview of a recent result which answers Håstad's question in the affirmative:

Theorem 1 ([RST15]). *The polynomial hierarchy is infinite relative to a random oracle: with probability* 1*, a random oracle* $\boldsymbol{A}$ *satisfies* $\Sigma_{k-1}^{\mathrm{P},\boldsymbol{A}} \neq \Sigma_{k}^{\mathrm{P},\boldsymbol{A}}$ *for all* $k \in \mathbb{N}$*.*

We recall that a random oracle $\boldsymbol{A} \leftarrow \{0,1\}^*$ is one in which every string is independently included with probability $1/2$; this induces a probability measure on $\{0,1\}^*$ that corresponds to the Lebesgue measure on $[0,1]$. As observed by Bennett and Gill in their paper [BG81] initiating the study of random oracles, the set of oracles relative to which complexity-theoretic statements hold satisfy Kolmogorov's zero-one law: these statements hold with probability one or with probability zero relative to a random oracle. Understanding the relationship of complexity classes relative to a random oracle often offers useful intuition regarding the unrelativized case — the truth of a statement relative to a random oracle may even be viewed as evidence of its truth in the unrelativized setting — though we emphasize that precise relationship between the "random oracle world" and our actual unrelativized world remains poorly understood.

1.2 Organization of this column

We begin in Section 2 by presenting the background and history of relativized separations leading up to [Yao85, Hås86a] and [RST15]. We especially highlight the pivotal role played by the work of Furst, Saxe, and Sipser [FSS81], which draws a close connection between the relativized polynomial hierarchy and small-depth Boolean circuits. In Section 3 we present the [FSS81] framework in the context of separating P from NP relative to an oracle; describe how it extends to the setting of random oracles; and outline how both settings extend to higher levels of the polynomial hierarchy. In particular, we sketch how Yao and Håstad's resolution of Meyer's question follows from a certain *depth hierarchy theorem* for Boolean circuits, and how Theorem 1 follows from an *average-case extension* of such a depth hierarchy theorem. Finally, in Section 4 we explain the techniques used in [RST15] to prove the average-case depth hierarchy theorem, and thereby establish Theorem 1.

2 The relativized polynomial hierarchy: an abbreviated history

It is well known that the 1980s were a "golden age" for small-depth circuit lower bounds, during which many landmark results were established [FSS81, Ajt83, Yao85, Hås86a, Raz87, Smo87]. What is sometimes not so well remembered is that much of this pioneering work was largely motivated by a connection between small-depth Boolean circuits and the polynomial hierarchy that was first discovered by Furst, Saxe, and Sipser [FSS81]. They gave a super-polynomial size lower bound for constant-depth circuits, proving that depth-k circuits computing the n-variable parity function must have size $\Omega(n^{\log^{(3k-6)} n})$, where $\log^{(i)} n$ denotes the i-th iterated logarithm. They also showed that an improvement of this lower bound to super-quasipolynomial for constant-depth circuits (i.e. $\Omega_k\big(2^{(\log n)^c}\big)$ for all constants c) would yield an oracle A such that $\mathsf{PSPACE}^A \neq \mathsf{PH}^A$. Ajtai independently proved a stronger lower bound of $n^{\Omega_k(\log n)}$ [Ajt83]; his motivation came from finite model theory. Yao gave the first super-quasipolynomial lower bounds on the size of constant-depth circuits computing the parity function [Yao85], and shortly after Håstad proved the optimal lower bound of $\exp(\Omega(n^{1/(k-1)}))$ via his influential Switching Lemma [Hås86a].

Yao's relativized separation of PSPACE from PH was improved qualitatively by Cai, who showed that the separation holds even relative to a random oracle [Cai86]. Leveraging the connection made by [FSS81], Cai accomplished this by proving *average-case* lower bounds against constant-depth circuits, showing that constant-depth circuits of sub-exponential size agree with the parity function only on a $(1/2 + o_n(1))$ fraction of inputs. (Independent work of Babai [Bab87] gave a simpler proof of the same relativized separation.)

Together, these results paint a fairly complete picture of the status of the PSPACE versus PH question in relativized worlds: not only does there exist an oracle A such that $\mathsf{PSPACE}^A \neq \mathsf{PH}^A$, this separation holds relative to almost all oracles. A natural next step is to seek analogous results showing that the relativized polynomial hierarchy is infinite; recall that the polynomial hierarchy being infinite implies $\mathsf{PSPACE} \neq \mathsf{PH}$, and furthermore, this implication relativizes. We recall Meyer's question from the introduction:

Meyer's Question. *Is there a relativized world within which the polynomial hierarchy is infinite? Equivalently, does there exist an oracle A such that $\Sigma_{k-1}^{\mathrm{P},A} \neq \Sigma_k^{\mathrm{P},A}$ for all $k \in \mathbb{N}$?*

Early work on Meyer's question predates [FSS81]. It was first considered by Baker, Gill, and Solovay in their paper initiating the study of relativization [BGS75], in which they prove the existence of an oracle A such that $\mathsf{P}^A \neq \mathsf{NP}^A \neq \mathsf{coNP}^A$, answering Meyer's question in the affirmative for $k \in \{1, 2\}$. Subsequent work of Baker and Selman proved the $k = 3$ case [BS79]. Following [FSS81], Sipser noted the analogous connection between Meyer's question and circuit lower bounds [Sip83]: to answer Meyer's question in the affirmative, it suffices to prove a *depth hierarchy theorem* for AC^0: for every constant $k \in \mathbb{N}$, there exists a Boolean function F computable by a depth-$(k+1)$ AC^0 circuit such that any depth-k circuit computing F requires super-quasipolynomial size. (This is a significantly more delicate task than proving super-quasipolynomial size lower bounds for the parity function; see Section 4.1 for a detailed discussion.) Sipser also constructed a family of Boolean functions — depth-$(k+1)$ read-once monotone formulas with alternating layers of AND and OR gates of fan-in $n^{1/(k+1)}$ — for which he proved an n versus $\Omega(n^{\log^{(3k+3)} n})$ separation. These came to be known as the *Sipser functions*, and they play the same central role in Meyer's question as the parity function does in the relativized PSPACE versus PH problem.

In 1986, Håstad gave the first proof of a strong depth hierarchy theorem for small-depth circuits, by proving the following near-optimal separation for (a slight variant of) the Sipser functions:

Theorem 2 ([Hås86a]; see also [Hås86b, Hås89])**.** *For all $k \leq \frac{c \log n}{\log \log n}$ where $c > 0$ is a universal constant, there exists an n-variable Boolean function F computable by a linear-size depth-$(k+1)$ circuit which is such that no depth-k circuit of size $\exp(n^{O(1/k)})$ computes F correctly on all inputs.*

This answered Meyer's original question in the affirmative for all $k \in \mathbb{N}$.

2.1 Recent progress: The polynomial hierarchy is infinite relative to a random oracle

Given Håstad's result, a natural goal is to complete our understanding of Meyer's question by showing that the polynomial hierarchy is not just infinite with respect to *some* oracle, but in fact with respect to *almost all* oracles. Indeed, in [Hås86a, Hås86b, Hås89], Håstad poses the problem of extending his result to show this as an open problem:

	$\mathsf{PSPACE}^A \neq \mathsf{PH}^A$	$\Sigma_{k-1}^{\mathrm{P},A} \neq \Sigma_k^{\mathrm{P},A}$ for all $k \in \mathbb{N}$
Connection to lower bounds for constant-depth circuits	[FSS81]	[Sip83]
Hard function(s)	Parity	Sipser functions
Relative to *some* oracle A	[Yao85, Hås86a]	[Yao85, Hås86a]
Relative to *random* oracle $\boldsymbol{A}$	[Cai86, Bab87]	[RST15]

Table 1: Previous work and the result of [RST15] on the relativized polynomial hierarchy

Question 1 (Meyer's Question for Random Oracles [Hås86a, Hås86b, Hås89])**.** *Is the polynomial hierarchy infinite relative to a random oracle? Equivalently, does a random oracle $\boldsymbol{A}$ satisfy $\Sigma_{k-1}^{\mathrm{P},\boldsymbol{A}} \neq \Sigma_k^{\mathrm{P},\boldsymbol{A}}$ for all $k \in \mathbb{N}$?*

Question 1 also appears as the main open problem in [Cai86, Bab87]; as mentioned above, an affirmative answer to Question 1 would imply Cai and Babai's result showing that $\mathsf{PSPACE}^{\boldsymbol{A}} \neq \mathsf{PH}^{\boldsymbol{A}}$ for a random oracle $\boldsymbol{A}$. Further motivation for studying Question 1 comes from a surprising result of Book, who proved that the *unrelativized* polynomial hierarchy collapses if it collapses relative to a random oracle [Boo94]. Over the years Question 1 was discussed in a wide range of surveys [Joh86, Ko89, Hem94, ST95, HRZ95, VW97, Aar], textbooks [DK00, HO02], and research papers [Tar89, For99, Aar10a]. However, progress on the conjecture was almost glacially slow; the $k \in \{1, 2\}$ cases were proved by Bennett and Gill in their paper initiating the study of random oracles [BG81], but larger values of k stubbornly resisted attack. Before the results of [RST15], the only progress that we are aware of subsequent to [BG81] was due to Aaronson; motivated by the problem of obtaining relativized separations in quantum structural complexity, he showed that a random oracle $\boldsymbol{A}$ separates Π_2^{P} from P^{NP} [Aar10b, Aar10a], and he conjectured in [Aar10a] that his techniques could be extended to resolve the $k = 3$ case of Question 1.

A resolution of Question 1. Recent work gives an average-case extension of Håstad's worst-case depth hierarchy theorem (Theorem 2 from the previous subsection) for small-depth circuits:

Theorem 3 ([RST15])**.** *For all $k \leq \frac{c\sqrt{\log n}}{\log\log n}$ where $c > 0$ is a universal constant, there exists an n-variable Boolean function F computable by a linear-size depth-$(k+1)$ circuit which is such that any depth-k circuit of size $\exp(n^{O(1/k)})$ agrees with F on at most a $(1/2 + n^{-\Omega(1/k)})$ fraction of inputs.*

As we show in the next section, Theorem 1 (an affirmative answer to Question 1 for all $k \in \mathbb{N}$) follows as a consequence of Theorem 3. We emphasize that the high-level approach of proving Theorem 1 by establishing an average-case depth hierarchy theorem for Boolean circuits is certainly *not* a new contribution of [RST15]. Indeed, already in 1986 Håstad explicitly pointed to an average-case extension of his depth hierarchy theorem as a possible approach towards Theorem 1 [Hås86a]. Similarly, in the fifth edition of this Complexity Theory Column [Hem94], Lane Hemaspaandra stated Question 1 (in the form of a conjecture), and commented that "The 'obvious' route to a proof of the conjecture is to build on [Yao85, Hås86a, Cai86, Bab87]. However, there are already a number of bodies along that route."

Since what is obvious to Lane may not be obvious to everyone, in the next section we describe the connection between lower bounds against constant-depth circuits (in the worst-case and the average-case) and proofs about the polynomial hierarchy (relative to some oracle and to random oracles). Finally, in Section 4 we present the main ideas behind the circuit lower bound of Theorem 3, and explain how the method of *random projections* — an extension of the classical method of random restrictions — enabled the authors of [RST15] to avoid leaving their bodies on the route to Theorem 1.

3 From constant-depth circuits to the polynomial hierarchy

The structure of this section is as follows: First, in Section 3.1 we give a simple proof that $\mathsf{P} \neq \mathsf{NP}$ relative to some oracle A. Our presentation highlights how a basic fact from circuit complexity — namely, that polylog-depth decision trees cannot compute the "tribes" DNF — is at the heart of this oracle result; indeed, the key intuition here is an analogy in which decision trees correspond to P and DNF formulas correspond to NP. Next, in Section 3.2, we extend the circuit–oracle connection by showing how an *average-case* version of this circuit complexity fact — more precisely, that polylog-depth decision trees cannot even *approximate* the tribes DNF — yields a proof that $\mathsf{P} \neq \mathsf{NP}$ relative to a *random* oracle $\boldsymbol{A}$. Finally, in Section 3.3 we sketch how these worst- and average-case oracle separations extend to higher levels of the polynomial hierarchy. The key analogy here, extending the analogy sketched above, is between depth-$(k+1)$ circuits and the k-th level of the polynomial hierarchy.

We note that the proofs of $\mathsf{P} \neq \mathsf{NP}$ relative to an oracle and a random oracle that we present in this section are not the original ones [BGS75, BG81]; those were not based on circuit lower bounds. We present alternative proofs based on the circuit-oracle connection because this framework extends naturally to higher levels of the hierarchy; in particular, it is through this connection that we obtain Theorem 1 as a consequence of Theorem 3. The circuit-oracle framework is due to Furst, Saxe, and Sipser [FSS81], who originally stated the connection in the context of separating PH from PSPACE relative to an oracle A (showing that such a separation follows from sufficiently strong lower bounds against constant-depth circuits computing the parity function). Sipser noted in [Sip83] that this connection extends to the context of showing that the relativized polynomial hierarchy is infinite (i.e. that such a separation would follow from a sufficiently strong depth hierarchy theorem for constant-depth circuits); a complete proof of this claim appears in Chapter §7 of Håstad's thesis [Hås86b]. Our presentation is based on Håstad's thesis and the survey of Ko [Ko89].

3.1 $\mathsf{P} \neq \mathsf{NP}$ relative to some oracle A

For $s, w \in \mathbb{N}$ satisfying $(1-2^{-w})^s = \frac{1}{2} \pm o(1)$, the $\mathsf{Tribes}_{s,w}$ function [BOL90] is an s-term read-once monotone DNF with all s terms having width exactly w. Therefore $\mathsf{Tribes}_{s,w}$ computes a Boolean function over $N := sw$ many variables, and we sometimes write Tribes_N instead of $\mathsf{Tribes}_{s,w}$. Note that $w = \Theta(\log N)$, $s = \Theta(N/\log N)$, and $\mathbf{Pr}[\mathsf{Tribes}_N(\boldsymbol{x}) = 1] = (1-2^{-w})^s = \frac{1}{2} \pm o(1)$ where $\boldsymbol{x}$ is a uniform random input from $\{0,1\}^N$.

Somewhat surprisingly, the relativized separation of P from NP follows from an elementary fact in circuit complexity:

Fact 3.1. Tribes_N *cannot be computed by a* $\mathrm{polylog}(N)$*-depth decision tree.*

(In fact Tribes_N is *evasive*: any decision tree computing it must have depth N, as can be seen from an easy adversary argument.) Given an oracle $A \subseteq \{0,1\}^*$ and string $y \in \{0,1\}^*$, we write $A(y)$ to denote the Boolean value $\mathbf{1}[y \in A] \in \{0,1\}$; we can therefore denote an oracle call of an oracle Turing machine M^A by $A(y)$ for some $y \in \{0,1\}^*$. We begin by describing how the Tribes function defines, for every oracle A, a language $L(A) \in \mathsf{NP}^A$. Let A be an oracle and consider

$$L(A) := \{1^n \colon \mathsf{Tribes}_{2^n}(A(y^{1,n}), A(y^{2,n}), \ldots, A(y^{2^n,n})) = 1\}, \tag{1}$$

where $y^{i,n}$ is the lexicographically i-th string of length n. To see that $L(A) \in \mathsf{NP}^A$, suppose $1^n \in L(A)$. $\mathsf{Tribes}_{2^n}(A(y^{1,n}), \ldots, A(y^{2^n,n})) = 1$ iff at least one of the terms in the DNF is satisfied, i.e. there exists an index $i \in [2^n]$ which is an integer multiple of w and is such that

$$A(y^{i+1,n}) = A(y^{i+2,n}) = \cdots = A(y^{i+w,n}) = 1, \quad \text{where } w = \Theta(\log 2^n) = \Theta(n).$$

Therefore $L(A) \in \mathsf{NP}^A$ since $i \in [2^n]$ can be encoded with $\log 2^n = n$ bits, and given i, the condition above can be verified with $w = \Theta(n)$ oracle calls to A.

It remains to argue the existence of an oracle $A^\dagger$ such that $L(A^\dagger) \notin \mathsf{P}^{A^\dagger}$; we construct such an $A^\dagger$ by diagonalizing against all polynomial-time oracle Turing machines $\{M_j\}_{j\in\mathbb{N}}$. For each machine M_j with running time at most $p_j(n)$ for some polynomial p_j, input $x \in \{0,1\}^*$ to M_j, and oracle A, we have that M_j^A either accepts or rejects x after making at most $p_j(|x|)$ many oracle calls to A. It follows that there exists a decision tree $T_{j,x}$ of depth at most $p_j(|x|)$, with internal nodes branching on oracle queries, such that

$$\forall \text{ oracles } A,\ M_j^A \text{ accepts } x \iff T_{j,x}(A) = 1. \tag{2}$$

Here $T_{j,x}(A) \in \{0,1\}$ denotes the output of $T_{j,x}$ when its oracle queries are answered according to A. Since $\text{depth}(T_{j,1^n}) \le p_j(n) \le \text{polylog}(2^n)$, it follows from Fact 3.1 that there exists $n_j \in \mathbb{N}$ such that

$$\forall n \ge n_j,\ \exists \text{ oracle } A_{j,n} \subseteq \{0,1\}^n \text{ s.t. } \mathsf{Tribes}_{2^n}(A_{j,n}(y^{1,n}), \ldots, A_{j,n}(y^{2^n,n})) \ne T_{j,1^n}(A_{j,n}), \tag{3}$$

We may assume that $A_{j,n}$ only contains strings of length n since Tribes_{2^n} depends only on $A_{j,n}(y)$'s where y has length n. Indeed, for the same reason, the above holds for all oracles A that agree with $A_{j,n}$ on all strings of length n:

$$\begin{gathered}\forall n \ge n_j,\ \forall \text{ oracles } A \subseteq \{0,1\}^* \text{ s.t. } A(y^{i,n}) = A_{j,n}(y^{i,n}) \text{ for all } i \in [2^n],\\ \mathsf{Tribes}_{2^n}(A(y^{1,n}), \ldots, A(y^{2^n,n})) \ne T_{j,1^n}(A).\end{gathered} \tag{4}$$

We will use (3) and (4) to diagonalize against all polynomial-time oracle Turing machines $\{M_j\}_{j\in\mathbb{N}}$. At a high level, we begin with $A^\dagger = \emptyset$ and for each $j \in \mathbb{N}$, we commit to including and excluding in $A^\dagger$ strings of a certain length m_j (at least n_j and greater than any length considered so far) according to A_{j,m_j} as defined in (3). By (4) this ensures that M_j^A does not recognize $L(A)$ for any oracle A that agrees with $A^\dagger$ on these strings. Since the m_j's are distinct (i.e. the strings A_{j,m_j} that "defeat" M_j have different lengths than the strings A_{k,m_k} that defeat M_k), we conclude that there exists $A^\dagger$ such that $M_j^{A^\dagger}$ does not recognize $L(A^\dagger)$ for all $j \in \mathbb{N}$ (and hence $L(A^\dagger) \notin \mathsf{P}^{A^\dagger}$).

In more detail, we define

$$m_j := \begin{cases} n_j & \text{if } j = 1 \\ \max\{n_j, m_{j-1}+1\} & \text{if } j \ge 2 \end{cases}$$

and consider

$$A^\dagger := \bigcup_{j \in \mathbb{N}} A_{j,m_j}, \tag{5}$$

where $A_{j,m_j} \subseteq \{0,1\}^{m_j}$ is the oracle defined in (3). Fix $j \in \mathbb{N}$; we claim that $M_j^{A^\dagger}$ does not recognize $L(A^\dagger)$. Since the m_j's are distinct (indeed $m_1 < m_2 < m_3 < \cdots$), we have that for all j, $A^\dagger$ agrees with A_{j,m_j} on all strings of length m_j. Furthermore, since $m_j \geq n_j$, it follows from (4) that

$$\mathsf{Tribes}_{2^{m_j}}(A^\dagger) \neq T_{j,1^{m_j}}(A^\dagger). \tag{6}$$

Recalling the definition (1) of $L(A^\dagger)$, we have that

$$1^{m_j} \in L(A^\dagger) \iff \mathsf{Tribes}_{2^{m_j}}(A^\dagger(y^{1,m_j}), \ldots, A^\dagger(y^{2^{m_j},m_j})) = 1. \tag{7}$$

On the other hand, by the definition (2) of the decision tree $T_{j,1^{m_j}}$ we have that

$$M_j^{A^\dagger} \text{ accepts } 1^{m_j} \iff T_{j,1^{m_j}}(A^\dagger) = 1 \tag{8}$$

Together (6), (7), and (8) imply that $M_j^{A^\dagger}$ does not recognize $L(A^\dagger)$, and the proof is complete. □

3.2 Random oracles and average-case hardness

This proof from Section 3.1 extends quite easily to show that a *random* oracle $\boldsymbol{A}$ separates P from NP. The key difference is that we need an *average-case* extension of the underlying circuit lower bound (Fact 3.1):

Fact 3.2. *Any* $\mathrm{polylog}(N)$*-depth decision tree agrees with* Tribes_N *on at most a* 0.9*-fraction of inputs.*

Roughly speaking, the intuition behind Fact 3.2 is that almost all inputs in $\mathsf{Tribes}_N^{-1}(1)$ satisfy much less than a $1/\mathrm{polylog}(N)$ fraction of the $\Theta(N/\log N)$ terms of the DNF, and conditioned on a random input x not satisfying a given term, the distribution of x restricted to that term is identical whether or not x satisfies Tribes_N. Hence a $\mathrm{polylog}(N)$-depth decision tree, which commits to an output after reading only $\mathrm{polylog}(N)$ many coordinates of any input x, can have only an $o(1)$ advantage over random guessing in predicting the value of $\mathsf{Tribes}_N(x)$. (Indeed, it can be shown that any decision tree that has at least 51% agreement with Tribes_N must have exponentially many nodes at depth $\Omega(N/\log N)$.)

With Fact 3.2 in hand, the proof that a random oracle $\boldsymbol{A}$ separates P from NP proceeds almost identically to the argument in Section 3.1. First, Fact 3.2 translates into the following strengthening of (3): there exists an $n_j \in \mathbb{N}$ such that

$$\forall n \geq n_j, \quad \mathop{\mathbf{Pr}}_{\boldsymbol{A}_{j,n} \leftarrow \{0,1\}^n}\Big[\mathsf{Tribes}_{2^n}(\boldsymbol{A}_{j,n}(y^{1,n}), \ldots, \boldsymbol{A}_{j,n}(y^{2^n,n})) \neq T_{j,1^n}(\boldsymbol{A}_{j,n})\Big] \geq 0.1, \tag{9}$$

where $\boldsymbol{A}_{j,n}$ is a uniform random string in $\{0,1\}^n$. Next, we observe that although $A^\dagger$ is defined in (5) to be the union of $A_{j,m_j} \subseteq \{0,1\}^{m_j}$ for all $j \in \mathbb{N}$, the essential property needed for the argument to go through is that $A^\dagger$ agrees with A_{j,m_j} on all strings of length m_j. In other words, while we defined $A^\dagger$ so that it does not include any string of length m where $m \notin \{m_j\}_{j \in \mathbb{N}}$, the presence or absence of these strings is inconsequential: the same proof shows that $L(A) \notin \mathsf{P}^A$ for all oracles A such that $A(y) = A^\dagger(y)$ for all $y \in \bigcup_{j\in\mathbb{N}}\{0,1\}^{m_j}$. This observation, together with (9) and Kolmogorov's zero-one law, implies that a random oracle $\boldsymbol{A}^\dagger$ satisfies

$$\underset{\boldsymbol{A}^\dagger \leftarrow \{0,1\}^*}{\mathbf{Pr}}\left[L(\boldsymbol{A}^\dagger) \notin \mathsf{P}^{\boldsymbol{A}^\dagger}\right] = 1, \quad \text{and hence} \quad \underset{\boldsymbol{A}^\dagger \leftarrow \{0,1\}^*}{\mathbf{Pr}}\left[\mathsf{P}^{\boldsymbol{A}^\dagger} \neq \mathsf{NP}^{\boldsymbol{A}^\dagger}\right] = 1.$$

3.3 Separating higher levels of the polynomial hierarchy

In this section we describe how the Furst–Saxe–Sipser framework, presented above for P versus NP, extends to higher levels of the polynomial hierarchy. In particular, we describe how the arguments from Sections 3.1 and 3.2 (establishing $\mathsf{P} \neq \mathsf{NP}$ relative to an oracle A and a random oracle $\boldsymbol{A}$ respectively) extend to show that a relativized separation of the hierarchy follows from a certain *depth hierarchy theorem* for Boolean circuits, and a random oracle separation follows from an *average-case depth hierarchy theorem.*

We follow the structure of the proof in Section 3.1 closely, highlighting the essential differences. First observe that the language $L(A)$ defined in (1) remains in NP^A if $\{\mathsf{Tribes}_N\}_{N\in\mathbb{N}}$ is replaced by any family $\{F_N\}_{N\in\mathbb{N}}$ of Boolean functions where F_N is computed by an N-variable DNF with $s = \mathrm{poly}(N)$ terms of width $w = \mathrm{polylog}(N)$. (To certify that $1^n \in L(A)$, one provides the verifier with $\log(s) = \mathrm{poly}(n)$ bits encoding the index $i \in [s]$ of a satisfied term in F_N, and given i, the verifier checks that this i-th term is indeed satisfied with $w = \mathrm{polylog}(N) = \mathrm{poly}(n)$ oracle calls to A.) This connection extends easily to higher levels of the polynomial hierarchy: for each level $k \in \mathbb{N}$, let $\{F_N^{k+1}\}_{N\in\mathbb{N}}$ be a family of Boolean functions such that F_N^{k+1} is computed by an N-variable depth-$(k+1)$ circuit with $\mathrm{poly}(N)$ many gates and bottom fan-in $\mathrm{polylog}(N)$. By a straightforward extension of the argument above, the language

$$L(A) := \{1^n \colon F_{2^n}^{k+1}(A(y^{1,n}), A(y^{2,n}), \ldots, A(y^{2^n,n})) = 1\} \tag{10}$$

is in $\Sigma_k^{\mathrm{P},A}$ for all oracles A. We again construct an oracle $A^\dagger$ such that $L(A^\dagger) \notin \Sigma_{k-1}^{\mathrm{P},A^\dagger}$ (and hence $\Sigma_{k-1}^{\mathrm{P},A^\dagger} \neq \Sigma_k^{\mathrm{P},A^\dagger}$) by diagonalizing against all $\Sigma_{k-1}^{\mathrm{P}}$ oracle Turing machines $\{M_j\}_{j\in\mathbb{N}}$. Here we need the analogue of the decision tree representation (2) of a polynomial-time oracle Turing machine: for $k \geq 2$, for each $\Sigma_{k-1}^{\mathrm{P}}$ oracle Turing machine M_j and input $x \in \{0,1\}^*$, there exists a depth-k circuit $C_{j,x}^k$ of size $2^{p_j(|x|)}$ and bottom fan-in $p_j(|x|)$ for some polynomial p_j, such that

$$\forall \text{ oracles } A,\ M_j^A \text{ accepts } x \iff C_{j,x}^k(A) = 1.$$

To apply the next step (3) of the argument, i.e. to assert the existence of $n_j \in \mathbb{N}$ such that

$$\forall n \geq n_j,\ \exists \text{ oracle } A_{j,n} \subseteq \{0,1\}^n \text{ s.t. } F_{2^n}^{k+1}(A_{j,n}(y^{1,n}), \ldots, A_{j,n}(y^{2^n,n})) \neq C_{j,1^n}^k(A_{j,n}),$$

we need the analogue of Fact 3.1: $F_{2^n}^{k+1}$ cannot be computed by any depth-k circuit C of size $2^{\mathrm{poly}(n)} = \mathrm{quasipoly}(2^n)$ and bottom fan-in $\mathrm{poly}(n) = \mathrm{polylog}(2^n)$. We have arrived at the circuit lower bound — a *depth hierarchy theorem* for Boolean circuits — that is the pith of Yao and Håstad's separation of the relativized polynomial hierarchy:

Depth Hierarchy Theorem. *For all constants $k \in \mathbb{N}$ there exists a family $\{F_N^{k+1}\}_{N\in\mathbb{N}}$ of Boolean functions such that*

1. *F_N^{k+1} is an N-variable Boolean function computable in depth-$(k+1)$ AC^0, and yet*

2. *No depth-k circuit C of* $\mathrm{quasipoly}(N)$ *size and bottom fan-in* $\mathrm{polylog}(N)$ *can compute* F_N^{k+1}.

Note that Håstad obtained significantly stronger quantitative parameters (cf. Theorem 2 in Section 2) than what is sought above: Håstad showed that no depth-k circuit C of subexponential size $\exp(N^{O(1/k)})$, regardless of bottom fan-in, can compute F_N^{k+1}. Furthermore, his theorem holds not just for constant values of k, but for all k up to $\Theta(\frac{\log N}{\log \log N})$. With such a depth hierarchy theorem in hand the remainder of the proof proceeds exactly as in Section 3.1 to yield the existence of an oracle $A^\dagger$ such that $\Sigma_{k-1}^{\mathsf{P},A^\dagger} \neq \Sigma_k^{\mathsf{P},A^\dagger}$.[2]

In sharp contrast with the elementary fact (Fact 3.1) underlying the oracle separation of P from NP, Håstad's proof of Theorem 2 is a technical tour de force, culminating a long line of work on the problem [FSS81, Ajt83, Sip83, KPPY84, Yao85]. At the heart of his proof is a delicate application of the *method of random restrictions*, a common essential ingredient underlying many of the landmark lower bounds in Boolean circuit complexity. We discuss Håstad's proof and the method of random restrictions in detail in Section 4.1.

3.3.1 Separating the hierarchy relative to a random oracle

Just as an average-case extension of Fact 3.1 underlies the separation of P from NP relative to a random oracle (as outlined in Section 3.2), to show that a random oracle separates the polynomial hierarchy using the above framework, we prove an *average-case depth hierarchy theorem* for Boolean circuits:

Average-Case Depth Hierarchy Theorem. *For all constants $k \in \mathbb{N}$ there exists a family $\{F_N^{k+1}\}_{N\in\mathbb{N}}$ of Boolean functions such that*

1. F_N^{k+1} *is an N-variable Boolean function computable in depth-$(k+1)$ AC^0, and yet*

2. *Any depth-k circuit C of* $\mathrm{quasipoly}(N)$ *size and bottom fan-in* $\mathrm{polylog}(N)$ *agrees with* F_N^{k+1} *on at most a 0.9-fraction of inputs.*

Again we remark that the result in [RST15] achieves stronger quantitative parameters (cf. Theorem 3 in Section 2) than what is sought above: we show that any depth-k circuit C of sub-exponential size $\exp(N^{O(1/k)})$, regardless of bottom fan-in, agrees with F_N^{k+1} on at most a $(1/2 + N^{-\Omega(1/k)})$ fraction of inputs. Note that a *constant* function achieves 50% agreement with F_N^{k+1}; we show that depth-k circuits of sub-exponential size can barely do any better. Furthermore, the theorem holds for all values of k up to $\Theta(\frac{\sqrt{\log N}}{\log \log N})$.[3]

A key component of our proof is an extension of the method of random restrictions, which we call the *method of random projections.* While restrictions work by fixing variables to 0, to 1, or

[2] The alert reader will notice that the depth hierarchy theorem as stated above does not *quite* sync up perfectly with the discussion that precedes it: in addition to being computable in depth-$(k+1)$ AC^0, the hard function F_N^{k+1} must also have bottom fan-in $\mathrm{polylog}(N)$ in order for $L(A)$ as defined in (10) to be in $\Sigma_k^{\mathsf{P},A}$. Indeed, Håstad's variant of the Sipser function is computed by a depth-$(k+1)$ circuit which has bottom fan-in $N^{\Theta(1/k)} \gg \mathrm{polylog}(N)$. However, since every depth-$(k+1)$ circuit is certainly also a depth-$(k+2)$ circuit with bottom fan-in 1, the depth hierarchy theorem as stated above (as well as Håstad's theorem) translates into the separation $\Sigma_{k-1}^{\mathsf{P},A} \neq \Sigma_{k+1}^{\mathsf{P},A}$ for some oracle A, which in turn implies $\Sigma_{k-1}^{\mathsf{P},A} \neq \Sigma_k^{\mathsf{P},A}$.

[3] Regarding the technical issue mentioned in the previous footnote, we mention that the F_N^{k+1} functions that [RST15] considers *do* have bottom fan-in $\mathrm{polylog}(N)$ (unlike Håstad's variant of the Sipser functions). Therefore the average-case depth hierarchy theorem translates directly into the separation $\Sigma_{k-1}^{\mathsf{P},\boldsymbol{A}} \neq \Sigma_k^{\mathsf{P},\boldsymbol{A}}$ for a random oracle $\boldsymbol{A}$, without the need for the additional step described in the previous footnote.

leaving them unchanged, projections work by fixing variables to 0, to 1, or *identifying* groups of many variables — "projecting" them all to the same new variable, so that they must all take the same value. Very roughly speaking, we show that (like random restrictions) random projections simplify Boolean circuits, but the identification of variables helps maintain "useful structure" that we exploit in our lower bound arguments. We elaborate on this in the next section.

4 The [RST15] average-case depth hierarchy theorem

In this section we describe the high-level structure of the proof of Theorem 3 from [RST15]. To do so, we first describe the general framework for proving worst- and average-case lower bounds against small-depth circuits via the method of random restrictions in Section 4.1. Within this framework, we sketch the now-standard proof of average-case lower bounds against the parity function based on Håstad's Switching Lemma. We also recall why the lemma is not well-suited for proving a depth hierarchy theorem for small-depth circuits, hence necessitating the "blockwise variant" of the lemma that Håstad developed and applied to prove Theorem 2, his (worst-case) depth hierarchy theorem. In Section 4.2 we highlight the difficulties that arise in extending Håstad's depth hierarchy theorem to the average-case, and explain how our techniques — specifically, the notion of random *projections* — allow us to overcome these difficulties.

Before delving into the details of [RST15], we mention that the first progress towards an average-case depth hierarchy theorem for small-depth circuits was made by Ryan O'Donnell and Karl Wimmer [OW07]. They constructed a Boolean function F computable by a linear-size depth-3 circuit and proved that any depth-2 circuit that approximates F must have exponential size:

Theorem 4 ([OW07])**.** *Let* $\mathsf{Tribes}^{\dagger}_N$ *denote the Boolean dual of* Tribes_N *and consider the* $2N$*-variable Boolean function*

$$F(x) := \mathsf{Tribes}(x_1, \ldots, x_N) \vee \mathsf{Tribes}^{\dagger}(x_{N+1}, \ldots, x_{2N}).$$

Any depth-2 *circuit* C *on* $2N$ *variables that has size* $\exp(O(N/\log N))$ *agrees with* F *on at most a* 0.9*-fraction of the* 2^{2N} *many inputs.*

With the [FSS81] circuit-oracle framework in mind, we note that Theorem 4 recovers Bennett and Gill's [BG81] separation of Σ_1^{P} from Σ_2^{P} relative to a random oracle $\boldsymbol{A}$ (though the authors of [OW07] do not discuss this application in their paper). In Section 4.2.1 we highlight a key idea from [OW07] that plays an important role in our proof.

4.1 Lower bounds via random restrictions

The method of random restrictions was originated by Subbotovskaya in the early 1960s [Sub61] and continues to be an indispensable technique in circuit complexity. Focusing only on small-depth circuits, we note that the random restriction method helped enable much of the rapid progress in the 1980s, and is the common essential ingredient underlying the landmark circuit lower bounds [FSS81, Ajt83, Sip83, Yao85, Hås86a, Cai86, Bab87] discussed in the previous sections. This technique has also contributed directly to important advances in other areas including computational learning theory, pseudorandomness, and proof complexity.

We begin by describing the general framework for proving worst- and average-case lower bounds against small-depth circuits via the random restriction method. Suppose we would like to show that a *target function* $F : \{0,1\}^N \to \{0,1\}$ has small correlation with any size-S depth-k *approximating circuit* C under the uniform distribution $\mathcal{U}$ over $\{0,1\}^N$. A standard approach is to construct a series of random restrictions $\{\mathcal{R}_\ell\}_{\ell \in \{2,\dots,k\}}$ satisfying three properties:

- **Property 1: Approximator C simplifies.** The randomly-restricted circuit $C \upharpoonright \boldsymbol{\rho}^{(k)} \cdots \boldsymbol{\rho}^{(2)}$, where $\boldsymbol{\rho}^{(\ell)} \leftarrow \mathcal{R}_\ell$ for $2 \le \ell \le k$, should "collapse to a simple function" with high probability. This is typically shown via iterative applications of an appropriate "Switching Lemma for the $\mathcal{R}_\ell$'s", which shows that each random restriction $\boldsymbol{\rho}^{(\ell)}$ decreases the depth of the circuit $C \upharpoonright \boldsymbol{\rho}^{(k)} \cdots \boldsymbol{\rho}^{(\ell-1)}$ by one with high probability. The upshot is that while C is a depth-k size-S circuit, $C \upharpoonright \boldsymbol{\rho}^{(k)} \cdots \boldsymbol{\rho}^{(2)}$ will be a small-depth decision tree, a "simple function", with high probability.

- **Property 2: Target F retains structure.** In contrast with the approximating circuit C, the target function F should (roughly speaking) be resilient against the random restrictions $\boldsymbol{\rho}^{(\ell)} \leftarrow \mathcal{R}_\ell$. While the precise meaning of "resilient" depends on the specific application, the key property we need is that $F \upharpoonright \boldsymbol{\rho}^{(k)} \cdots \boldsymbol{\rho}^{(2)}$ will with high probability be a "well-structured" function that is uncorrelated with any small-depth decision tree.

Together, these two properties imply that random restrictions of F and C are uncorrelated with high probability. Note that this already yields *worst-case* lower bounds, showing that $F : \{0,1\}^N \to \{0,1\}$ cannot be computed exactly by C. To obtain average-case lower bounds, we need to translate such a statement into the fact that F and C *themselves* are uncorrelated. For this we need the third key property of the random restrictions:

- **Property 3: Composition of $\mathcal{R}_\ell$'s completes to the uniform distribution $\mathcal{U}$.** Evaluating a Boolean function $G : \{0,1\}^N \to \{0,1\}$ on a random input $\mathbf{X} \leftarrow \mathcal{U}$ is equivalent to first applying random restrictions $\boldsymbol{\rho}^{(k)}, \dots, \boldsymbol{\rho}^{(2)}$ to G, and then evaluating the randomly-restricted function $G \upharpoonright \boldsymbol{\rho}^{(k)} \cdots \boldsymbol{\rho}^{(2)}$ on $\mathbf{X}' \leftarrow \mathcal{U}$.

Average-case lower bounds for parity. For uniform-distribution average-case lower bounds against constant-depth circuits computing the parity function, the random restrictions are all drawn from $\mathcal{R}(p)$, the "standard" random restriction which independently sets each free variable to 0 with probability $\frac{1}{2}(1-p)$, to 1 with probability $\frac{1}{2}(1-p)$, and keeps it free with probability p. The main technical challenge arises in proving that Property 1 holds — this is precisely Håstad's Switching Lemma — whereas Properties 2 and 3 are straightforward to show. For the second property, we note that

$$\mathsf{Parity}_n \upharpoonright \rho \equiv \pm \mathsf{Parity}(\rho^{-1}(*)) \quad \text{for all restrictions } \rho \in \{0,1,*\}^N,$$

and so $\mathsf{Parity}_n \upharpoonright \boldsymbol{\rho}^{(k)} \cdots \boldsymbol{\rho}^{(2)}$ computes the parity of a random subset $\mathbf{S} \subseteq [N]$ of coordinates (or its negation). With an appropriate choice of the $*$-probability p we have that $|\mathbf{S}|$ is large with high probability; recall that $\pm \mathsf{Parity}_t$ (the t-variable parity function or its negation) has zero correlation with any decision tree of depth at most $t-1$. For the third property, we note that for all values of $p \in (0,1)$, a random restriction $\boldsymbol{\rho} \leftarrow \mathcal{R}(p)$ specifies a uniform random subcube of $\{0,1\}^N$ (of dimension $|\boldsymbol{\rho}^{-1}(*)|$). Therefore, the third property is a consequence of the simple fact that a uniform random point within a uniform random subcube is itself a uniform random point from $\{0,1\}^N$.

Håstad's blockwise random restrictions. With the above framework in mind, we notice a conceptual challenge in proving an AC^0 depth hierarchy theorem via the random restriction method: even focusing only on the worst-case (i.e. ignoring Property 3), the random restrictions $\mathcal{R}_\ell$ will have to satisfy Properties 1 and 2 with the target function F being *computable in* AC^0. This is a significantly more delicate task than (say) proving $\mathsf{Parity} \notin \mathsf{AC}^0$ since, roughly speaking, in the latter case the target function $F \equiv \mathsf{Parity}$ is "much more complex" than the circuit $C \in \mathsf{AC}^0$ to begin with. In an AC^0 depth hierarchy theorem, *both* the target F and the approximating circuit C are constant-depth circuits; the target F is "more complex" than C in the sense that it has larger circuit depth, but this is offset by the fact that the circuit size of C is allowed to be exponentially larger than that of F (as is the case in both Håstad's and our depth hierarchy theorems). We refer the reader to Chapter §6.2 of Hastad's thesis [Hås86b] which contains a discussion of this very issue.

Håstad overcomes this difficulty by replacing the "standard" random restrictions $\mathcal{R}(p)$ with random restrictions *specifically suited to Sipser functions being the target*: his "blockwise" random restrictions are designed so that (1) they reduce the depth of the formula computing the Sipser function by one, but otherwise essentially preserve the rest of its structure, and yet (2) a switching lemma still holds for any circuit with sufficiently small bottom fan-in. These correspond to Properties 2 and 1 respectively. However, unlike $\mathcal{R}(p)$, Håstad's blockwise random restrictions are not independent across coordinates and do not satisfy Property 3: their composition does not complete to the uniform distribution $\mathcal{U}$ (and indeed it does not complete to any product distribution). This is why Håstad's construction establishes a worst-case rather than average-case depth hierarchy theorem.

4.2 The main technique of [RST15]: random projections

The crux of the difficulty in proving an average-case AC^0 depth hierarchy theorem therefore lies in designing random restrictions that satisfy Properties 1, 2, and 3 simultaneously, for a target f in AC^0 and an arbitrary approximating circuit C of smaller depth but possibly exponentially larger size. To recall, the "standard" random restrictions $\mathcal{R}(p)$ satisfy Properties 1 and 3 but not 2, and Håstad's blockwise variant satisfies Properties 1 and 2 but not 3.

We overcome this difficulty with *projections*, a generalization of restrictions. Given a set of formal variables $\mathcal{X} = \{x_1, \dots, x_N\}$, a restriction ρ either fixes a variable x_i (i.e. $\rho(x_i) \in \{0,1\}$) or keeps it alive (i.e. $\rho(x_i) = x_i$, often denoted by $*$). A *projection*, on the other hand, either fixes x_i or maps it to a variable y_j from a possibly different space of formal variables $\mathcal{Y} = \{y_1, \dots, y_M\}$. Restrictions are therefore a special case of projections where $\mathcal{Y} \equiv \mathcal{X}$, and each x_i can only be fixed or mapped to itself. Our arguments crucially employ projections in which $\mathcal{Y}$ is smaller than $\mathcal{X}$, and where moreover each x_i is only mapped to a specific element y_j where j depends on i in a carefully designed way that depends on the structure of the formula computing the target function. Such "collisions", where blocks of distinct formal variables in $\mathcal{X}$ are mapped to the same new formal variable $y_j \in \mathcal{Y}$, play a crucial role in the approach.

At a high level, our overall approach is structured around a sequence $\mathbf{\Psi}$ of random projections satisfying Properties 1, 2, and 3 simultaneously, with the target being a function they denote Sipser, which is a slight variant of the Sipser function. We briefly outline how each of the three properties is established:

- **Property 1: Approximator C simplifies.** We first prove that depth-k approximating circuits C of size $\exp(N^{O(1/k)})$ "collapse to a simple function" with high probability under the sequence $\mathbf{\Psi}$

of random projections. Following the standard "bottom-up" approach to proving lower bounds against small-depth circuits, this is established by arguing that each of the individual random projections comprising $\boldsymbol{\Psi}$ "contributes to the simplification" of C by reducing its depth by (at least) one.

More precisely, we prove a *projection switching lemma*, showing that a small-width DNF or CNF "switches" to a small-depth decision tree with high probability under our random projections. (The depth reduction of C follows by applying this lemma to every one of its bottom-level depth-2 subcircuits.) Recall that the random projection of a depth-2 circuit over a set of formal variables $\mathcal{X}$ yields a function over a new set of formal variables $\mathcal{Y}$, and in our case $\mathcal{Y}$ is significantly smaller than $\mathcal{X}$. In addition to the structural simplification that results from setting variables to constants (as in Håstad's Switching Lemma for random *restrictions*), the proof of our projection switching lemma also crucially exploits the additional structural simplification that results from distinct variables in $\mathcal{X}$ being mapped to the same variable in $\mathcal{Y}$. For example, consider an AND gate (OR gate, respectively) in C that accesses x_i and $\overline{x}_j$, and suppose both x_i and x_j are projected to the same variable $y \in \mathcal{Y}$. This gate accesses both y and $\overline{y}$ in the projection of C and hence can be replaced by the constant 0 (1, respectively).

- **Property 2: Target Sipser retains structure.** Like Håstad's blockwise random restrictions, our random projections are defined with the target function Sipser in mind; in particular, they are carefully designed so as to ensure that Sipser "retains structure" with high probability under their composition $\boldsymbol{\Psi}$.

  Very roughly speaking, we show that with high probability, each of the individual random projections comprising $\boldsymbol{\Psi}$ have a "limited and well-controlled" effect on the structure of Sipser; equivalently, Sipser is resilient against these random projections. The high-level idea is that the variable identifications that take place under a random projection are engineered so as to reduce a Sipser function of depth d to a Sipser function of depth $d-1$. Combining this with Property 1 above we have that Sipser reduces under $\boldsymbol{\Psi}$ to a "well-structured" formula (more precisely, an OR of large fan-in), whereas the approximator C "collapses to a simple function" (more precisely, a decision tree of small depth), where both are high probability statements with respect to the randomness of $\boldsymbol{\Psi}$.

- **Property 3: $\boldsymbol{\Psi}$ completes to the uniform distribution.** Like Håstad's blockwise random restrictions (and unlike the "standard" random restrictions $\mathcal{R}(p)$), the distributions of our random projections are not independent across coordinates: they are carefully correlated in a way that depends on the structure of the formula computing Sipser. As discussed in the previous subsection, there is an inherent tension between the need for such correlations on one hand (to ensure that Sipser "retains structure"), and the requirement that their composition completes to the uniform distribution on the other hand (to yield average-case lower bounds with respect to the uniform distribution). We overcome this difficulty with projections: we prove that the composition $\boldsymbol{\Psi}$ of our sequence of random projections completes to the uniform distribution, despite the fact that every one of the individual random projections comprising $\boldsymbol{\Psi}$ is correlated among coordinates.

4.2.1 Completion to uniform via the O'Donnell–Wimmer trick

Establishing each of the three properties above requires significant work, and the notion of random projections plays an important role in the proofs of all three. Since Property 3 is the one that distinguishes our average-case lower bound from Håstad's worst-case lower bound, in the remainder of this section we elaborate on the third bullet above. We give a concrete example of a random projection (a simplified version of the ones employed in [RST15]), and we use this example to illustrate a key fact that underlies the proof of Property 3.

For $p \in (0,1)$ and symbols $\bullet, \circ$, we write $\{\bullet_{1-p}, \circ_p\}$ to denote the distribution over $\{\bullet, \circ\}$ which outputs $\circ$ with probability p and $\bullet$ with probability $1-p$. We write $\{\bullet_{1-p}, \circ_p\}^w$ to denote the product distribution over $\{\bullet, \circ\}^w$ in which each coordinate is distributed independently according to $\{\bullet_{1-p}, \circ_p\}$, and $\{\bullet_{1-p}, \circ_p\}^w \setminus \{\circ\}^w$ to denote the product distribution conditioned on not outputting $\{\circ\}^w$. (Note that $\{\bullet_{1-p}, \circ_p\}^w \setminus \{\circ\}^w$ is *not* a product distribution.) The following fact, implicit in [OW07]'s proof of Theorem 4, is key for us:

Fact 4.1 (the O'Donnell–Wimmer trick)**.** *Let* $\boldsymbol{\rho} \leftarrow \{*_{1/2}, 1_{1/2}\}^w \setminus \{1\}^w$ *and* $\boldsymbol{y} \leftarrow \{0_{1-2^{-w}}, 1_{2^{-w}}\}$. *The random string* $\boldsymbol{x} \in \{0,1\}^w$ *where*

$$\boldsymbol{x}_j := \begin{cases} \boldsymbol{y} & \text{if } \boldsymbol{\rho}_j = * \\ \boldsymbol{\rho}_j & \text{otherwise} \end{cases} \quad \text{for all } j \in [w]$$

is a uniform random string in $\{0,1\}^w$.

In words, Fact 4.1 says that one can generate a uniformly random string in $\{0,1\}^w$ via the following two-stage process: First set each coordinate to 1 independently with probability 1/2, conditioned on not setting all of them to 1. For the coordinates that remain unset (there is at least one such coordinate), collectively set *all* of them to 1 with probability 2^{-w} and all of them to 0 otherwise; equivalently, we "project" all the coordinates that remain unset to a single fresh formal variable, which we then set according to a 2^{-w}-biased random bit $\boldsymbol{y}$.

Though elementary, Fact 4.1 is at the heart of our proof that the composition $\boldsymbol{\Psi}$ of our random projections complete to the uniform distribution. For a sense of how this works, let $\mathcal{X} = \{x_{i,j} : i \in [u], j \in [w]\}$ and $\mathcal{Y} = \{y_i : i \in [u]\}$ be two sets of formal variables, and consider the distribution $\mathcal{D}$ over random restrictions $\boldsymbol{\rho} \in \{1,*\}^{\mathcal{X}}$ where $\boldsymbol{\rho}_i \leftarrow \{*_{1/2}, 1_{1/2}\}^w \setminus \{1\}^w$ independently for each $i \in [u]$. For a function F over the variables in $\mathcal{X}$ and $\boldsymbol{\rho} \leftarrow \mathcal{D}$, the $\boldsymbol{\rho}$*-random projection* of F is the function over the variables in $\mathcal{Y}$ defined by

$$(\mathrm{proj}_{\boldsymbol{\rho}} F)(y) := F(x) \quad \text{where } x_{i,j} = \begin{cases} y_i & \text{if } \boldsymbol{\rho}_{i,j} = * \\ \boldsymbol{\rho}_{i,j} & \text{otherwise.} \end{cases}$$

By Fact 4.1, for any two functions F and G over the variables in $\mathcal{X}$, we have that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \leftarrow \{0_{1/2}, 1_{1/2}\}^{uw}} [F(\boldsymbol{x}) \neq G(\boldsymbol{x})] = \mathop{\mathbf{Pr}}_{\substack{\boldsymbol{\rho} \leftarrow \mathcal{D} \\ \boldsymbol{y} \leftarrow \{0_{1-2^{-w}}, 1_{2^{-w}}\}^u}} [(\mathrm{proj}_{\boldsymbol{\rho}} F)(\boldsymbol{y}) \neq (\mathrm{proj}_{\boldsymbol{\rho}} G)(\boldsymbol{y})]. \tag{11}$$

In words, the correlation between F and G under the uniform distribution is equal to the correlation between their $\boldsymbol{\rho}$-random projections $\mathrm{proj}_{\boldsymbol{\rho}} F$ and $\mathrm{proj}_{\boldsymbol{\rho}} G$ under the 2^{-w}-biased product distribution. Intuitively, (11) is useful since $\mathrm{proj}_{\boldsymbol{\rho}} F$ and $\mathrm{proj}_{\boldsymbol{\rho}} G$ are "simpler" Boolean functions that are easier to reason about (for one, they are over $|\mathcal{Y}| = u$ many variables whereas F and G are over $|\mathcal{X}| = uw$ many variables).

Though the random projections employed in [RST15] are significantly more complicated than the ones considered above — necessarily so because they have to also satisfy Properties 1 and 2 — the proof that their composition $\mathbf{\Psi}$ completes to the uniform distribution is essentially based on iterated applications of (a generalization of) Fact 4.1.

5 Conclusion

As discussed in this column, the 1980s witnessed tremendous advances in our understanding of Boolean circuit lower bounds, the structure of relativized complexity classes, and the relationship between these two topics. In recent years there has been a resurgence of research activity in circuit complexity, with exciting progress on both old problems and new ones. It will be interesting to see whether, via the circuit-oracle connection (or new connections that we do not yet know about), this progress leads to corresponding progress in relativized complexity.

References

[Aar] Scott Aaronson. The Complexity Zoo. Available at http://cse.unl.edu/~cbourke/latex/ComplexityZoo.pdf. 2.1

[Aar10a] Scott Aaronson. A counterexample to the generalized Linial-Nisan conjecture. *Electronic Colloquium on Computational Complexity*, 17:109, 2010. 2.1

[Aar10b] Scott Aaronson. BQP and the polynomial hierarchy. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pages 141–150, 2010. 2.1

[Ajt83] Miklós Ajtai. Σ^1_1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1–48, 1983. 2, 3.3, 4.1

[Bab87] László Babai. Random oracles separate PSPACE from the polynomial-time hierarchy. *Information Processing Letters*, 26(1):51–53, 1987. 2, 2.1, 2.1, 2.1, 4.1

[BG81] Charles Bennett and John Gill. Relative to a random oracle A, $\mathsf{P}^A \neq \mathsf{NP}^A \neq \mathsf{coNP}^A$ with probability 1. *SIAM Journal on Computing*, 10(1):96–113, 1981. 1.1, 2.1, 3, 4

[BGS75] Theodore Baker, John Gill, and Robert Solovay. Relativizations of the P=?NP question. *SIAM Journal on computing*, 4(4):431–442, 1975. 1.1, 2, 3

[BHZ87] Ravi Boppana, Johan Håstad, and Stathis Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25(2):127–132, 1987. 1

[BOL90] Michael Ben-Or and Nati Linial. Collective coin flipping. In S. Micali, editor, *Randomness and Computation*, pages 91–115. Academic Press, 1990. 3.1

[Boo94] Ronald Book. On collapsing the polynomial-time hierarchy. *Information Processing Letters*, 52(5):235–237, 1994. 2.1

[BS79] Theodore Baker and Alan Selman. A second step toward the polynomial hierarchy. *Theoretical Computer Science*, 8(2):177–187, 1979. 1.1, 2

[BU11] David Buchfuhrer and Christopher Umans. The complexity of Boolean formula minimization. *Journal of Computer and System Sciences*, 77(1):142–153, 2011. 1

[Cai86] Jin-Yi Cai. With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 21–29, 1986. 2, 2.1, 2.1, 2.1, 4.1

[DK00] Ding-Zhu Du and Ker-I Ko. *Theory of Computational Complexity*. John Wiley & Sons, Inc., 2000. 2.1

[For99] Lance Fortnow. Relativized worlds with an infinite hierarchy. *Information Processing Letters*, 69(6):309–313, 1999. 2.1

[FSS81] Merrick Furst, James Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. In *Proceedings of the 22nd IEEE Annual Symposium on Foundations of Computer Science*, pages 260–270, 1981. 1.2, 2, 2.1, 3, 3.3, 4, 4.1

[Hås86a] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 6–20, 1986. 1.1, 1.2, 2, 2, 2.1, 2.1, 1, 2.1, 4.1

[Hås86b] Johan Håstad. *Computational Limitations for Small Depth Circuits*. MIT Press, Cambridge, MA, 1986. 1.1, 2, 2.1, 1, 3, 4.1

[Hås89] Johan Håstad. *Almost optimal lower bounds for small depth circuits*, pages 143–170. Advances in Computing Research, Vol. 5. JAI Press, 1989. 1.1, 2, 2.1, 1

[Hem94] Lane Hemaspaandra. Complexity Theory Column 5: The not-ready-for-prime-time conjectures. *ACM SIGACT News*, 25(2):5–10, 1994. 2.1, 2.1

[HO02] Lane Hemaspaandra and Mitsunori Ogihara. *The Complexity Theory Companion*. Springer, 2002. 2.1

[HRZ95] Lane Hemaspaandra, Ajit Ramachandran, and Marius Zimand. Complexity Theory Column 11: Worlds to die for. *ACM SIGACT News*, 26(4):5–15, 1995. 2.1

[Joh86] David Johnson. The NP-completeness column: An ongoing guide. *Journal of Algorithms*, 7(2):289–305, 1986. 2.1

[KL80] Richard Karp and Richard Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing*, pages 302–309, 1980. 1

[Ko89] Ker-I Ko. Constructing oracles by lower bound techniques for circuits. In *Combinatorics, computing and complexity (Tianjing and Beijing, 1988)*, volume 1 of *Math. Appl. (Chinese Ser.)*, pages 30–76. Kluwer Acad. Publ., Dordrecht, 1989. 2.1, 3

[Kol85] Gina Kolata. Must "Hard Problems" Be Hard? *Science*, 228(4698):479–81, 1985. 1.1

[KPPY84] Maria Klawe, Wolfgang Paul, Nicholas Pippenger, and Mihalis Yannakakis. On monotone formulae with restricted depth. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, pages 480–487, 1984. 3.3

[MS72] Albert Meyer and Larry Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pages 125–129, 1972. 1

[OW07] Ryan O'Donnell and Karl Wimmer. Approximation by DNF: examples and counterexamples. In *34th International Colloquium on Automata, Languages and Programming*, pages 195–206, 2007. 4, 4, 4.2.1

[Raz87] Alexander Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987. 2

[RST15] Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for Boolean circuits. In *Proceedings of the 56th Annual Symposium on Foundations of Computer Science*, 2015. To appear. (document), 1, 1.2, 2.1, 1, 2.1, 3, 2.1, 3.3.1, 3, 4, 4.2, 4.2.1, 4.2.1

[Sch88] Uwe Schöning. Graph isomorphism is in the low hierarchy. *Journal of Computer and System Sciences*, 37(3):312–323, 1988. 1

[Sch99] Marcus Schäefer. Deciding the Vapnik–Červonenkis dimension is Σ_3^{P}-complete. *Journal of Computer and System Sciences*, 58:177–182, 1999. 1

[Sip83] Michael Sipser. Borel sets and circuit complexity. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, pages 61–69, 1983. 2, 2.1, 3, 3.3, 4.1

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 77–82, 1987. 2

[ST95] David Shmoys and Éva Tardos. Computational Complexity. In *Handbook of Combinatorics (Ronald Graham, Martin Grötschel, and Lászlo Lovász, eds.)*, volume 2. North-Holland, 1995. 2.1

[Sto76] Larry Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976. 1

[SU02a] Marcus Schäefer and Chris Umans. Complexity Theory Column 37: Completeness in the Polynomial-Time Hierarchy: A Compendium. *ACM SIGACT News*, 33(3):32–49, 2002. 1

[SU02b] Marcus Schäefer and Chris Umans. Complexity Theory Column 38: Completeness in the Polynomial-Time Hierarchy: Part II. *ACM SIGACT News*, 33(4):22–36, 2002. 1

[Sub61] Bella Subbotovskaya. Realizations of linear functions by formulas using $\vee$, &, $\bar{\ }$. *Doklady Akademii Nauk SSSR*, 136(3):553–555, 1961. 4.1

[Tar89] Gábor Tardos. Query complexity, or why is it difficult to separate $\mathsf{NP}^A \cap \mathsf{coNP}^A$ from P^A by random oracles A? *Combinatorica*, 9(4):385–392, 1989. 2.1

[VW97] Heribert Vollmer and Klaus Wagner. *Measure One Results in Computational Complexity Theory*, pages 285–312. Advances in Algorithms, Languages, and Complexity. Springer, 1997. 2.1

[Wra76] Celia Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):23–33, 1976. 1

[Yao85] Andrew Yao. Separating the polynomial-time hierarchy by oracles. In *Proceedings of the 26th Annual Symposium on Foundations of Computer Science*, pages 1–10, 1985. 1.1, 1.2, 2, 2.1, 2.1, 3.3, 4.1