JOURNAL OF COMPUTER AND SYSTEM SCIENCES 38, 68–85 (1989)

With Probability One, a Random Oracle Separates *PSPACE* from the Polynomial-Time Hierarchy

JIN-YI CAI*

Department of Computer Science, Yale University, New Haven, Connecticut 06520

Received September 11, 1986

We consider how much error a fixed depth Boolean circuit must make in computing the parity function. We show that with an exponential bound of the form $\exp(n^\lambda)$ on the size of the circuits, they make a 50% error on all possible inputs, asymptotically and uniformly. As a consequence, we show that a random oracle set A separates *PSPACE* from the entire polynomial-time hierarchy with probability one.

1. INTRODUCTION

The relationship between time and space, as complexity measures, has been one of the primary concerns in complexity theory research. It is well known that the entire polynomial-time hierarchy *PH* is contained in *PSPACE*. However, despite convincing heuristic evidence and persistent effort, no proof is yet available for separating the polynomial-time hierachy from polynomial space.

A proof that $PH \neq PSPACE$ would be an extremely strong separation of time and space. In this paper, we show that *PH* is properly contained in *PSPACE* in almost all relativized worlds.

THEOREM 1.1. *With probability one, a random oracle separates PSPACE from the entire polynomial-time hierarchy.*

The present work is a continuation of the work pioneered by Furst, Saxe, Sipser, and Yao. For the definitions of some basic notions we refer the reader to Refs. [FSS84; Sip83; Yao85].

In 1978 Furst, Saxe, and Sipser showed that the Boolean function *Parity* (see the definition below) cannot be computed in a fixed depth polynomial size Boolean circuit. They also observed that an exponential poly-logarithmic lower bound (i.e., bounded below by $\exp((\log)^k)$ for all k) would establish the existence of an oracle separating *PSPACE* from the polynomial hierarchy. Later Sipser extended this

* The research was supervised by Professor J. Hartmanis as part of this author's Ph.D. thesis. It was supported by a Sage Fellowship from Cornell University and NSF Grant DCR-8301766.

0022-0000/89 $3.00

work in [Sip83]. Finally, in 1985, a breakthrough came with the following theorem by Yao, which influenced our research immensely.

THEOREM 1.2 (Yao). *There exists an oracle A such that*

$$P^A \neq NP^A \neq \Sigma_2^{P,A} \neq \cdots \neq PH^A \neq PSPACE^A.$$

Our strong separation result is obtained by looking at *how much* error is present in the supposed circuit computation (instead of the existence of a *single* error). This question is interesting in its own right in the theory of circuit computation. As a nice byproduct of the proof of this strong separation we have the following corollary.

COROLLARY 1.3. *Fixed depth Boolean circuits with a bound of the form* $\exp(n^\lambda)$ *on the size, for some* λ, *make a* 50% *error, aymptotically and uniformly, when they compute the Boolean function Parity.*

The proof in this paper is organized as follows:

1. Use the alternating Turning machine model [CKS81] to reduce the problem to a Boolean circuit computation problem.

2. Employ certain probabilistic and game theoretic techniques to crack a shallow circuit.

3. Inductively prove a theorem in the general case and then adapt it to resolve the problem on circuit computation in step 1.

2. INITIAL REDUCTIONS AND TECHNIQUES

We proceed with some definitions. Let X be the set of n Boolean variables $\{x_1, x_2, ..., x_n\}$. A $\Sigma_{0,n}$-formula (circuit) is the constant 0, and a $\Pi_{0,n}$-formula (circuit) is the constant 1. A $\Sigma_{1,n}$-formula (circuit) is a sum of the form $\sum_k \overline{x_{i_k}} + \sum_k x_{j_k}$, where $x_{i_k}, x_{j_k} \in X$. Without loss of generality, we assume that the variables are distinct. The number of literals is its size. A $\Pi_{1,n}$-formula (circuit) is the negation of a $\Sigma_{1,n}$-formula, with the same size, i.e., a product of the form $\prod_k \overline{x_{i_k}} \cdot \prod_k x_{j_k}$.

For $k > 1$, a $\Sigma_{k,n}$-formula (circuit) H is a sum of $\Pi_{k-1,n}$-formulae, $\sum_i G_i$, with $\text{size}(H) = \sum_i \text{size}(G_i)$. A $\Pi_{k,n}$-formula is the negation of a $\Sigma_{k,n}$-formula, with the same size. Inductively, a subcircuit of H is H or any of the subcircuits of the G_i's. The depth of a $\Sigma_{k,n}$-formula (circuit) or a $\Pi_{k,n}$-formula is k. The bottom fan-in (*bfi*) of a Boolean circuit is the maximum size of the depth one subcircuits.

For any $\Pi_{2,n}$-formula G, $G = \prod_{i=1}^{l} C_i$, where

$$C_i = \overline{x_{i_1}} + \overline{x_{i_2}} + \cdots + \overline{x_{i_s}} + x_{i_{s+1}} + x_{i_{s+2}} + \cdots + x_{i_{s+t}} \quad \text{and} \quad s, t \geqslant 0.$$

We let $J_{i-} = \{i_1, i_2, ..., i_s\}$ $J_{i+} = \{i_{s+1}, i_{s+2}, ..., i_{s+t}\}$, and $J_i = J_{i-} \cup J_{i+}$.

A (partial) assignment of X is an n-tuple $\sigma=(\sigma_1, \sigma_2, ..., \sigma_n)\in\{0, 1, *\}^n$. If $\sigma\in\{0, 1\}^n$, then σ is a total assignement. Let F be a Boolean function on X; then $F|_\sigma$ denotes the Boolean function after the assignment σ, i.e., assign $x_i=0$, 1, or unassigned, if $\sigma_i=0$, 1, or $*$.

To generalize a bit, we also consider random assignments of X. For $0\leqslant p\leqslant 1$, let $\mathscr{R}_p$ denote the probability space $\{0, 1, *\}^\omega$ with a product measure ν, where (independently) for each coordinate i, $1\leqslant i<\omega$, $\nu\{(a_1, ..., a_i, ...) \mid \alpha_i=\alpha\}=(1-p)/2$, if $\alpha=0$ or 1; and p if $\alpha=*$. That such a product measure exists is a well known result of probability theory. A random assignement is simply a point in the measure space $\mathscr{R}_p$. We will write it as $A=(a_1, ..., a_i, ...)$.

If F is a Boolean function on free variables $\{x_{i_1}, ..., x_{i_s}\}\subseteq X$, then a random assignment A taken from $\mathscr{R}_p$ (denoted as $A\in\mathscr{R}_p$) assigns the variables x_{i_j} to 0, 1, or leaves it unassigned, according to a_j of A.

We denote by $F\|_A$ the Boolean function that resulted from the assignement.

Similarly we define one-sided random assignements. A random B taken from $\mathscr{R}_p^+$ (denoted as $B\in\mathscr{R}_p^+$) assigns independently to each x_{i_j} in F to 1 with probability p, and leaves it unassigned with probability $1-p$, respectively. $\mathscr{R}_p^-$ is defined in the same way with 0 substituting for 1. Note that all random assignments affect only free variables, when they are applied to a formula.

Consider a sequence of random assignements $R_1, ..., R_s$. $F\|_{R_1, ..., R_s}$ is defined to be $(F\|_{R_1, ..., R_{s-1}})\|_{R_s}$. For instance, let R and S be two random assignements, $R=(a_1, ..., a_j, ...)$, $S=(b_1, ..., b_k, ...)$. Let $a_{j_1}, a_{j_2}, ...$ be those a_j in R which are equal to $*$. As before let F be a Boolean function on free variables $\{x_{i_1}, ..., x_{i_s}\}$. Then in $F\|_{RS}$, x_{i_j} is assigned a_j if a_j is not a $*$. Otherwise, suppose a_j is a_{jk}, the k th $*$ in R; then x_{i_j} is assigned b_k, provided b_k is not a $*$. Finally if b_k is a $*$, then x_{i_j} is left unassigned. The successive random assignments act only on the variables left untouched by previous assignments. In what follows, when we make a statement such as "take two random assignements R and S from probability spaces $\mathscr{R}$ and $\mathscr{S}$, respectively, with probability p, event E occurs," we assert the product measure of the set $\{(R, S) \mid E \text{ occurs}\}\subseteq\mathscr{R}\times\mathscr{S}$ is p. We also denote $R_1\cdots R_s$ as $A=(a_1, ..., a_j, ...)$, where $a_j=0$, 1, or $*$, depending on whether $R_1\cdots R_s$ assigns the jth variable to 0, 1, or unassigned, respectively.

Note, however, that a partial assignement σ is applicable to $F\|_{R_1, ..., R_s}$ iff σ assigns 0 or 1 to only those variables that are $*$-valued by $R_1\cdots R_s$. In this case, we denote the resulting function by $(F\|_{R_1, ..., R_s})|_\sigma$.

Fix an alphabet $\{0, 1\}$ and an integer n. Define the parity function $Parity_n$:

$$Parity_n(x_1, x_2, ..., x_n)=\sum_{i=1}^{n} x_i \pmod 2.$$

We will consider circuit computation in relation to the parity function $Parity_n$. The parity function is chosen for the following property: the value of $Parity_n$ is *vitally* dependent on each variable x_i.

For $A \subseteq \{0, 1\}^*$, define the parity language

$$Parity^A = \{1^n \mid \text{there are odd number of strings of length } n \text{ in } A\}.$$

Clearly, we have $Parity^A \in PSPACE^A$, for all A.

We study separation of the polynomial-time hierarchy from *PSPACE* in almost all the relativized world. Intuitively, a random oracle set A is generated as follows. For each string $x \in \{0, 1\}^*$, we flip a fair coin, and depending on the outcome, we put x in A or not. Formally, we may represent each A by its characteristic function, and then map to a real number in the binary expansion $\in [0, 1]$. Now we define the probability measure μ on the oracle space to be the Lebesgue measure on $[0, 1]$. The readers may easily verify that the formal definition represents our intuitive notion of a random set as described above.

We aim to prove that $\mu\{A \mid Parity^A \notin PH^A\} = 1$. Surely this implies that a (random) oracle separates *PSPACE* from the entire polynomial-time hierarchy, with probability one.

There are only countably many levels $\Sigma_k^{P,A}$ in PH^A. Each level Σ_k^P has a recursive enumeration as the class of languages accepted by polynomial time alternating machines at that level [CKS81]. Let $M_1, M_2, \ldots$ be an enumeration of Σ_k^P alternating machines; then it is sufficient to show that

$$\forall i, \qquad [\mu\{A \mid Parity^A \neq L(M_i^A)\} = 1].$$

According to a theorem by Bennet and Gill [3], we only need to show that for each level k,

$$\exists \varepsilon_0 > 0, \forall i, \qquad [\mu\{A: Parity^A \neq L(M_i^A)\} > \varepsilon_0].$$

Now we reduce alternating machines to Boolean circuits. This reduction is from Furst, Saxe, and Sipser in [FSS84].

For a fixed alternating machine with oracle M_i^A at level Σ_k^P, consider its computation on 1^m. We claim that it is always possible to postpone the queries of strings. The trick is to guess the answers and verify them at a later stage. For example, at an existential stage, whenever a query is needed, we instead guess the answer and proceed until the succeeding universal stage. At the beginning of this universal stage we verify with the orable the guesses at the previous existential stage. If any of the guesses is wrong, we abort this path; otherwise we proceed. Similarly we may delay the queries of a universal stage and verify them at the suceeding existential stage. This time if any guess is wrong we simply accept. It is easily shown that this transformation preserves the notion of acceptance by the alternating Turing machine. By adding one more level of the alternation using the same method, we may obtain an equivalent polynomial-time bounded alternating machine that queries only at the bottom level and queries only once on every computation path. Thus, the computation tree structure is *independent* of the oracle and, therefore, can be frozen to yield a Σ_{k+1}-circuit, G.

The set of input Boolean variables of G corresponds to those strings that are queried by the modified alternating machine, on 1^m. (A queried string is in A *iff* the corresponding Boolean variable is set to be true.) Empirically, there should be precisely 2^m input variables, corresponding to 2^m strings of length m. This is because whether $1^m \in Parity^A$ is independent of any string of length unequal to m; furthermore, for any string x of length m, whether $1^m \in Parity^A$ depends on whether or not $x \in A$. Does the machine have to query precisely those strings of length m, no more and no less? We show that this is indeed the case, without loss of generality, in the sense that one can always replace the machine with one that does. (Strictly speaking, we only replace the machine with a nonuniform circuit family. Thus there is no uniformity concern.)

Suppose then for some x, $|x| = m$, and x is never queried. Then clearly the machine errs with probability 1/2 (under μ) at length m. That is enough.

Now suppose all x's with $|x| = m$ are queried, but so are $y_1, \ldots, y_l$ of length unequal to m.

Consider all 2^l many possible assignements σ for the y_i's. For any such σ, consider the $\Sigma_{k+1,2^m}$-circuit $G|_\sigma$. Pick the best σ_0, in the sense that $G|_{\sigma_0}$ makes the least error for parity. Clearly the original G makes no less error than that made by $G|_{\sigma_0}$, percentage-wise. Formally speaking,

$$\mu\{A \mid Parity^A(1^m) \neq M_i^A(1^m)\} \geqslant \text{error rate of } G|_{\sigma_0} \text{ for parity}$$
$$\equiv \frac{|\{\tau \in \{0,1\}^n : G|_{\sigma_0}|_\tau \neq Parity_n|_\tau\}|}{2^n},$$

where $n = 2^m$.

A remark on the size: Since $M_i(1^m)$ runs at most $p(m)$ steps, for some polynomial $p(\cdot)$, the size of the circuit is bounded by an exponential polylog in n, $\exp(O(p(\log(n))))$, where n is the input size to the circuit.

We have shown that in order to prove Theorem 1.1, the following theorem on Boolean circuit computation would suffice.

THEOREM 2.1. *For all $k \geqslant 2$, there is a sequence α_n, where $\alpha_n \to \frac{1}{2}$ as $n \to \infty$, such that all depth k Boolean circuits with n inputs and size bounded by $\exp(n^{1/4(k+1)})$ err on more than $\alpha_n \cdot 2^n$ of the 2^n many inputs when computing $Parity_n$.*

We now define a notion that is central to our exposition.

A Boolean function G is given. Consider the following class of two-man games, played between a master and a player: the general mode of the game is a cycle; the master gives a Boolean variable (unassigned so far) and asks the player to assign it. The player may assign it either 0 or 1. The master may repeat the cycle zero or more times, until he declares the end of the game. The rule dictates that when the master declares the end of the game, the assignment made by the player so far makes G a constant.

A Boolean function G is *k-monochromatic* iff there is a two-man game of the defined class, in which the master has a winning strategy in the following sense: the master can declare the end of the game after no more than $\lceil k \rceil$ many variables are assigned.

To put it differently, it is guaranteed that, no matter how the player plays, the master can force the function G to be constant, after at most $\lceil k \rceil$ variables are assigned.

Here we emphasize two points:

1. The k variables are *not* given out in a batch; rather the master makes up his mind as to which variable to give next, depending on how the player has assigned the variables so far.

2. Even in play following a winning strategy, the master is (technically) *not* required to declare the end of the game at the earliest possible moment.

We finish the section with the following lemma, which essentially states that under a *monochromaticity* condition, a *conjunction* of a *disjunction* and a *disjunction* of a *conjunction* are interchangeable.

LEMMA 2.2. *If G is k-monochromatic, then G is equivalent to a $\Sigma_{2,n}$-circuit (as well as a $\Pi_{2,n}$-circuit) with bfi $\leqslant k$. (It is a constant if bfi $= 0$.)*

Proof. The proof is by induction on k. The case $k = 0$ is trivial. Suppose $k > 0$, and the lemma is true for all values less than k. Let G be k-monochromatic, but not $(k-1)$-monochromatic. Let us play the game; suppose x_i is the first variable the master puts out when following the strategy given by k-monochromaticity. Then $G = [x_i \wedge G|_{x_i = T}] \vee [\overline{x_i} \wedge G|_{x_i = F}]$, where T stands for *true* and F stands for *false*. Now both $G|_{x_i = T}$ and $G|_{x_i = F}$ are $(k-1)$-monochromatic; we apply the inductive hypothesis once more, and the result follows. Q.E.D.

3. DEPTH TWO CIRCUITS

We wish to prove a theorem concerning depth two Boolean circuits.

THEOREM 3.1. *Fix $0 < \varepsilon < \frac{1}{5}$. Then there exists a constant C, such that, for any $G \in \Pi_{2,n}$ with bfi $\leqslant n^{\varepsilon}$, and for any q with $0 \leqslant q \leqslant n^{-1.05\varepsilon}$, and a random $Q_1 \in \mathscr{R}_{1-q}^{-}$ and a random $Q_2 \in \mathscr{R}_{1-q}^{+}$, the probability that $G \| _{Q_1 Q_2}$ is n^{ε}-monochromatic is $1 - \varepsilon_n$, where $\varepsilon_n \leqslant Ce^{-n^{\varepsilon}}$.*

The idea of the proof is as follows: We will define a two-man game associated with the circuit $G \| _{Q_1 Q_2}$, for which we claim that the master most probably has a

winning strategy (cf. Section 2). The game is designed so that each play creates a *record* of how the game was played. In the rare case in which $G\|_{Q_1Q_2}$ is not n^ε-monochromatic, the *record* will be "large." Now we define another procedure, Recording (Q_1Q_2, *record*), which will reproduce the game play. On the other hand, given a "large" *record*, the event that a random assignment R will *survive* the procedure Recording (R, *record*) is so unlikely that even if the probability is summed over all "large" *records*, it is still of measure near 0.

3.1. *The Game and the Recording*

We denote an assignment Q_1Q_2 as $A=(a_1, ..., a_i, ...)$, as in Section 2. Suppose $G=C_1 \wedge \cdots \wedge C_l$. Let $N=\{1, ..., \lceil n^\varepsilon \rceil\}$, $\mathscr{X}=\{\langle Z, s\rangle \mid Z\subseteq N,\ s\in\{0,1\}^{|Z|}\}$. Define $\|\langle Z, s\rangle\|=|Z|$, the cardinality of Z. A record $\mathscr{S}$ is a finite sequence $\langle X_1, ..., X_l\rangle$, where $X_i\in\mathscr{X}$. Define the norm $\|\mathscr{S}\|=\sum_{i=1}^{l}\|X_i\|$.

Intuitively, when a record element $X_i=\langle Z, s\rangle$ is generated in a certain round of the game, Z codes the variables to be assigned and s codes the assignment made by the player, in that round.

The coding scheme in Z is an indirect addressing. Specifically, if $J=\{i_1, i_2, ..., i_\alpha\}$, and $Z=\{z_1, z_2, ..., z_\beta\}$, where $\alpha, \beta\geqslant 0$, $1\leqslant i_1<\cdots<i_\alpha$, $1\leqslant z_1<\cdots<z_\beta$, then Z codes the subset of J:

$$\{i_{z_1}, ..., i_{z_\beta}\}, \qquad \text{if} \quad z_\beta\leqslant\alpha.$$

We denote this set as $J\downarrow Z$. If $Z=\varnothing$, then $J\downarrow Z=\varnothing$. If $z_\beta>\alpha$, $J\downarrow Z$ is undefined. Conversely for $\Delta=\{i_{z_1}, ..., i_{z_\beta}\}\subseteq J$, we denote

$$J\uparrow\Delta=\{z_1, ..., z_\beta\}.$$

Note that $J\uparrow\varnothing=\varnothing$. Clearly for $\Delta\subseteq J$, $J\downarrow(J\uparrow\Delta)=\Delta$.

Our game is played in rounds. The master executes the program, and in certain rounds, he asks the player to assign a few Boolean variables. Then the master continues, until the program halts. When the program halts, it halts in "result" or in "abort."

The procedure Recording is similar; for technical reasons, we first present Recording. A record $\mathscr{S}=\langle X_1, ..., X_l\rangle$ is given. There are four essential variables Θ, Y^+, Y^*, and N^*, respectively, representing our knowledge about the assignement A at any given point in the execution (more accurately, our knowledge about A which we can be *forced* to acknowledge). Here are some intuitive ideas behind the procedure (they should be taken as such only). The variable Y^+ will collect indices which correspond to variables that are assigned to be true by the given assignment Q_1Q_2. Similarly, Y^* and N^* will correspond to variables that are assigned to $*$ by the given assignment but assigned to be true and false, respectively, by the player recorded in $\mathscr{S}$. And Θ will collect subsets of indices which contain variables that are assigned to be false by the given assignment. Our goal is to show that an *authentic large record* that was produced by a game play rarely occurs.

Procedure Recording(A, $\mathscr{S}$)
0 Θ, Y^+, Y^*, $N^* := \varnothing$; $t := 0$; $List := [C_1, ..., C_l]$;
Repeat
1 **if** $List = \varnothing$ **then** case 1: $t < l \Rightarrow$ "abort";
case 2: $t \geqslant l \Rightarrow$ "result"
fi
2 let C_i be on the top of $List$
3 **if** $(\exists u \in J_{i-}, a_u = 0)$ **then** $\Theta := \Theta \cup \{J_{i-}\}$, delete C_i from $List$
4 **else** [critical round]
5 $t := t + 1$
6 $F := J_i - (Y^+ \cup Y^* \cup N^*)$
7 get $X_t = \langle Z, s \rangle$ from $\mathscr{S}$, "abort" if nonexistent
8 $D := F \downarrow Z$, "abort" if undefined
9 **if** $(D \neq \{j \in F \mid a_j = *\})$ **then** "abort"
10 **else** $Y^+ := Y^+ \cup \{u \in F \mid a_u = 1\}$
$\Theta := \Theta \cup \{\{u\} \mid u \in F, a_u = 0\}$
$Y^* := Y^* \cup \{u \in D \mid s$ assigns x_u to $1\}$
$N^* := N^* \cup \{u \in D \mid s$ assigns x_u to $0\}$
fi
11 **if** $(D = \varnothing)$ **then**
12 **if** $(\forall u \in J_{i+} - N^*, a_u = 0)$ **then** "abort" **fi**
fi
13 Delete any C_k from $List$ with
$J_{k-} \cap N^* \neq \varnothing$ or $J_{k+} \cap (Y^+ \cup Y^*) \neq \varnothing$
fi
End[Repeat]

Some properties of Recording are easily verified. Define $List_{\text{out}}$ to be the set of j such that C_j has been deleted from $List$. (In the following, c.r. is shorthand for "critical round"). We have

LEMMA 3.2. (1) Θ, Y^+, Y^*, N^*, *and* $List_{\text{out}}$ *are monotonically non-decreasing.*

Every time the **Repeat** *loop is entered, the following are true*:

(2) $\forall K \in \Theta, \exists u \in K, a_u = 0$.

(3) $\forall u \in Y^+, a_u = 1$.

(4) $\forall u \in Y^* \cup N^*, a_u = *$.

(5) $t = \#$ *of c.r. completed so far.*

(6) $\forall j \in List, J_{j-} \cap N^* = \varnothing$ *or* $J_{j+} \cap (Y^+ \cup Y^*) = \varnothing$.

(7) $\forall j \in List_{\text{out}}, J_{j-} \in \Theta$ *or* $J_{j-} \cap N^* \neq$ *or* $J_{j+} \cap (Y^+ \cup Y^*) \neq \varnothing$.

(8) $Y^* \cap N^* = \varnothing$.

Proof. A straightforward check. Q.E.D.

Next we define our Game. The Game is very much like Recording, except that the record $\mathscr{S}$ is produced as we go along, one slot per critical round. Spicifically,

- In the initialization part (line 0), add $\mathscr{S} := \varnothing$.
- Change line 1 to: **if** $List = \varnothing$ **then** "result" S.
- Change lines 7 and 8 to:

  Create X_t as follows:

  $Z := F\uparrow \{j \in F \mid a_j = *\}$;

  $D := F\downarrow Z (= \{j \in F \mid a_j = *\})$;

  for $j \in D$ **do** let the player assign x_j, and record the assignment in $\mathscr{S}$ with a binary string s of length $|D|$ (in the obvious way). $X_t := \langle Z, s \rangle$.

Let ρ be the assignment made by the player: $\rho_d = 0, 1$, or $*$; if $d \in N^*$, Y^*, or otherwise.

We wish to prove the following:

LEMMA 3.3. *The Game will eventually halt. When the Game halts,* $G \|_A |_\rho \equiv 1$ *(at line* 1*),* $or \equiv 0$ *(at line* 12*). Furthermore, let* $\mathscr{S}$ *be the record it created; then Recording*$(A, \mathscr{S})$ *will run in precisely the same way as Game*$(A, \mathscr{S})$*, until halting.*

Proof. We claim that every completed round of the Game either deletes a clause or assigns a variable. The only nontrivial case is in a c.r. with $D = \varnothing$. If $D = \varnothing$ and the round is completed, the condition at line 12 must be false. Thus $\exists t \in J_{i+} - N^*$, $a_t = 1$ (there is no $*$ in F), where i is the index of the current clause C_i. But then C_i must be deleted at line 13.

Therefore the Game will eventually halt. Let $\mathscr{S}_0$ be the record created when the Game halts.

We prove by induction that Recording $(A, \mathscr{S}_0)$ will reproduce this play of the game. Suppose they both enter a new round with all the variables having the same value. (This is certainly true initially.) Also assume $t =$ the length of $\mathscr{S}_0$ constructed so far in the game.

If $List = \varnothing$, then the game halts as a "result." Since $t = l$, the length of $\mathscr{S}_0$, Recording$(A, \mathscr{S}_0)$ will also halt as a "result."

Suppose $List \neq \varnothing$. Then they get the same C_i and the same condition at line 3 (same A!). If the condition is satisfied, then the induction is completed. Suppose not. They come to line 7. The Game creates the next X_t. Since X_t is never altered later in the Game, it is what Recording obtains from $\mathscr{S}_0$. From the way X_t is created in the Game, Recording will not halt at lines 7, 8, and 9. Now for the rest of this round they have the same code. The induction is completed.

We have proved that Recording will reproduce the play of the game, and hence Lemma 3.2 applies to the procedure Game. In fact we proved something more, namely, that the Game can halt only at line 1 or line 12.

It follows from Lemma 3.2(4), 8) that ρ is a valid assignment to $G\|_A$. If the Game halts at line 1, then $G\|_A|_\rho \equiv 1$, by Lemma 3.2(2), 3), 7). If the Game halts at line 12, then we claim that $C_i\|_A|_\rho \equiv 0$, hence $G\|_A|_\rho \equiv 0$.

By (6) of Lemma 3.2, $F = [J_{i-} - (Y^+ \cup Y^*)] \cup [J_{i+} - N^*]$, at line 6. Since $D = \varnothing$, Y^* and N^* are unchanged at line 10. Clearly the only way to satisfy C_i is in F. But if $t \in J_{i-} - Y^*$, $a_t = 1$, by lines 3, 9, and 11; and if $t \in J_{i+} - N^*$, $a_t = 0$, by line 12. So $C_i\|_A|_\rho \equiv 0$. Q.E.D.

Let $\mathscr{A} = \{A: G\|_A$ is *not* n^ε-monochromatic$\}$. For a record $\mathscr{S}$, let $\mathscr{A}[\mathscr{S}] = \{A:$ Recording$(A, \mathscr{S})$ "results"$\}$.

If $A \in \mathscr{A}$, then for any game in particular for our Game, the master has no winning strategy. Hence there is a play in which the player assigned $\lceil n^\varepsilon \rceil$ many variables and still the circuit is not constant.

Because the circuit is not constantly 0, there is a satisfying assignment σ. Now for the rest of the Game, the player adopts the following strategy: assign any new variable according to σ. Since this strategy keeps the circuit satisfiable and the Game eventually halts, the Game must halt with the circuit equal to constant 1. Hence, the Game "results" with some $\mathscr{S}$, where $\|\mathscr{S}\| > \lceil n^\varepsilon \rceil$. Therefore,

$$\mathscr{A} \subseteq \bigcup \mathscr{A}[\mathscr{S}],$$

where the union is over all $\mathscr{S}$, with $\|\mathscr{S}\| > \lceil n^\varepsilon \rceil$.

3.2. *A Probability Analysis*

In this section, we will focus on Recording $(A, \mathscr{S})$. For a fixed $\mathscr{S}$ with $\|\mathscr{S}\| > \lceil n^\varepsilon \rceil$, we consider the probability that Recording$(A, \mathscr{S})$ results, where $A = Q_1 Q_2$, $Q_1 \in \mathscr{R}^-_{1-q}$, and $Q_2 \in \mathscr{R}^+_{1-q}$.

Define $\mathscr{A}^{m\gamma} = \{A:$ Recording$(A, \mathscr{S})$ will come to its mth c.r. with $(\Theta, Y^+, Y^*, N^*) = \gamma\}$, $\Gamma^m = \{\gamma: \mathscr{A}^{m\gamma} \neq \varnothing\}$, $\mathscr{A}^m = \bigcup_{\gamma \in \Gamma^m} \mathscr{A}^{m\gamma}$.

We first derive a condition for $A \in \mathscr{A}^{m\gamma}$.

LEMMA 3.4. *For any* $\gamma = (\gamma^\Theta, \gamma^+, \gamma^Y, \gamma^N) \in \Gamma^m$, *there exists* i_γ, *such that* $A \in \mathscr{A}^{m\gamma} \Leftrightarrow A$ *satisfies the following conditions*:

(I) $\forall K \in \gamma^\Theta$, $\exists u \in K$, $a_u = 0$.

(II) $\forall u \in \gamma^+$, $a_u = 1$.

(III) $\forall u \in \gamma^Y \cup \gamma^N$, $a_u = *$.

(IV) $\forall u \in J_{i_\gamma -}$, $a_u \neq 0$.

Let us prove the following lemma first: Pick any $A^0 \in \mathscr{A}^{m\gamma}$.

LEMMA 3.5. *A satisfies conditions* (I), (II), *and* (III) $\Rightarrow$ *Recording for* A^0 *and* A *will run precisely the same* (*with all the variables* Θ, Y^+, Y^*, N^*, *List, and* t *the same at corresponding moments*) *up to line* 1 *of the* m*th c.r. of* A^0.

Proof. By induction. Suppose they are at line 1 of the m_0th round of A^0 (including m' c.r. and $m' < m$), and so far they are all the same (trivially true for the base case $m_0 = 1$).

If this is the mth c.r. for A^0, then the induction is completed. Suppose it is not. Hence A^0 will complete this round without halting. Thus $List \neq \varnothing$ and they pick the same C_i.

If A^0 satisfies the condition at line 3, then $J_{i-} \in \gamma^{\Theta}$, since the mth c.r. of A^0 is yet to come. By (I), A satisfies the same condition at line 3.

If A^0 fails the condition at line 3, this is a c.r. of A^0, but not the mth yet. A^0 will successfully record all $u \in J_{i-} \cup J_{i+}$ in (Θ, Y^+, Y^*, N^*), which will later become γ. In particular, $\forall u \in J_{i-}$, $u \in \gamma^+ \cup \gamma^Y \cup \gamma^N$. By (II) and (III), A must also fail the condition at line 3.

Hence either A^0 and A both finish the current round at line 3, in which case the induction is completed; or they both advance to line 4. Suppose then that this is a c.r. for both. They must find (the same) D well defined. As we noted, A^0 will record all $u \in J_{i-} \cup J_{i+}$ which will appear in γ.

In particular, by (I), (II), and (III), A must also find D to be precisely the set of $*$'s in F, and thus update Y^* and N^* in exactly the same way. Similarly, A must update Θ and Y^+ in the same way that A^0 does, by (I) and (II).

Now if $D \neq \varnothing$, we are done. If $D = \varnothing$, then A^0 will find the condition at line 12 to be false; i.e., $\exists t_0 \in J_{i+} - N^*$, $a^0_{t_0} \neq 0$. But $D = \varnothing \Rightarrow a^0_{t_0} = 1$. Hence $t_0 \in Y^+$, which is the same for both A and A_0. Hence $a_{t_0} = 1$ as well. Therefore A will not halt there. The induction is completed. Q.E.D.

Proof of Lemma 3.4. Pick $A^0 \in \mathscr{A}^{m\gamma}$ and run Recording$(A^0, \mathscr{S})$; let C_{i_γ} be the clause under consideration in its mth c.r.

$\Rightarrow$ Since $A \in \mathscr{A}^{m\gamma}$, A satisfies (I), (II), and (III). By Lemma 3.5, A and A^0 will reach line 1 of the mth c.r. of A_0, with all the variables the same. Since A^0 comes to line 4, $List \neq \varnothing$, which is the same as for A; so they both pick up C_{i_γ}. Since this is the mth c.r. for A^0, A^0 will fail the condition at line 3, and enter its c.r. with Θ unchanged. Thus γ^{Θ} is the common value for Θ when A and A^0 entered the current round at line 1.

If A were to satisfy the condition at line 3, then this is not a c.r. for A, and its mth c.r. is yet to come. Since $J_{i_\gamma -}$ is now added to Θ by A, $J_{i_\gamma -} \in \gamma^{\Theta}$. In other words, $J_{i_\gamma -} \in \Theta$ when they entered at line 1. But then A^0 must have satisfied the condition at line 3. A contradiction. Therefore A satisfies (IV).

$\Leftarrow$ Again by Lemma 3.5, we can assume they arrive at line 1 of the mth c.r. of A^0, with exactly the same history.

$A^0 \in \mathscr{A}^{m\gamma} \Rightarrow List \neq \varnothing$ and A picks up C_{i_γ}. Then (IV) says that this is also a c.r. for A. Since this is the mth c.r. for A^0, $A^0 \in \mathscr{A}^{m\gamma}$, and so far they are the same; this is also the mth c.r. for A, with $(\Theta, Y^+, Y^*, N^*) = \gamma$. Hence $A \in \mathscr{A}^{m\gamma}$. Q.E.D.

Now we are ready to estimate the probability $\Pr(\mathscr{A}[\mathscr{S}])$. Let E^m denote the event that Recording$(A, \mathscr{S})$ completes its mth c.r. without halting:

$$\begin{aligned}\Pr(\mathscr{A}[\mathscr{S}]) &\leqslant \Pr(\mathscr{A}^1)\cdot \Pi_{1\leqslant m<l}\Pr(A\in\mathscr{A}^{m+1}\mid A\in\mathscr{A}^m)\cdot\Pr(E^l\mid A\in\mathscr{A}^l)\\ &\leqslant [\Pi_{1\leqslant m<l}\Pr(E^m\mid A\in\mathscr{A}^m)]\cdot\Pr(E^l\mid A\in\mathscr{A}^l)\\ &= \Pi_{1\leqslant m\leqslant l}\Pr(E^m\mid A\in\mathscr{A}^m).\end{aligned}$$

We show the following:

LEMMA 3.6. $\forall\gamma\in\Gamma^m$,

$$\begin{aligned}\Pr(E^m\mid A\in\mathscr{A}^{m\gamma}) &\leqslant q^{\|X_m\|} \qquad \text{if} \quad \|X_m\|\neq 0\\ &\leqslant q\cdot n^{\varepsilon} \qquad \text{otherwise.}\end{aligned}$$

Clearly, Lemma 3.6 implies the same bound for $\Pr(E^m\mid A\in\mathscr{A}^m)$, since it can be estimated as

$$\sum_{\gamma\in\Gamma^m}\Pr(E^m\mid A\in\mathscr{A}^{m\gamma})\cdot\Pr(A\in\mathscr{A}^{m\gamma}\mid A\in\mathscr{A}^m),$$

and

$$\sum_{\gamma\in\Gamma^m}\Pr(A\in\mathscr{A}^{m\gamma}\mid A\in\mathscr{A}^m)=1.$$

Hence,

COROLLARY 3.7. $\Pr(\mathscr{A}[\mathscr{S}])\leqslant q^{\|\mathscr{S}\|}\cdot(qn^{\varepsilon})^{l'}$, *where* $l'=\#$ *of* X_i *in* $\mathscr{S}$ *with* $\|X_i\|=0$.

We use Lemma 3.4 to prove Lemma 3.6.

Proof of Lemma 3.6. Assume $\|X_m\|\neq 0$. Consider a random assignment taken from $\mathscr{R}^-_{1-q}$, followed by one from $\mathscr{R}^+_{1-q}$, on the variables in $F=J_{i_\gamma}-(\gamma^+\cup\gamma^Y\cup\gamma^N)$. We refer to the procedure Recording. In order to survive the mth critical round, we must have $D=F\downarrow X_m=\{j\in F\mid a_j=*\}$. Clearly the conditions on the random assignment of Lemma 3.4 can be strengthened so that all variables in F are assigned $*$ by the first round $\mathscr{R}^-_{1-q}$ (since in order to remain $*$ after two sweeps, it must remain $*$ after the first.) Note that originally the conditions from Lemma 3.4 on F were with $\mathscr{R}^-_{1-q}$ only. For $\mathscr{R}^+_{1-q}$, a given $u\in F$ is assigned $*$ only with probability q. Thus we have the upper bound $q^{\|X_m\|}$.

In the case $\|X_m\|=0$, we estimate

$$\begin{aligned}&\Pr(D=\varnothing \text{ is all the } *\text{'s in } F\wedge\exists t\in J_{i_\gamma+}-\gamma^N, a_t\neq 0\mid A\in A^{m\gamma})\\ &\leqslant \Pr(\exists t\in J_{i_\gamma+}-\gamma^N, a_t=1\mid A\in\mathscr{A}^{m\gamma}).\end{aligned}$$

We consider two sweeps from $\mathscr{R}^-_{1-q}$ followed by one from $\mathscr{R}^+_{1-q}$, on $J_{i_\gamma+}-\gamma^N$. Conditions (II), (III), and (IV) are irrelevant now (using independence). And con-

dition (I) would only reduce the probability for a given $u \in J_{i_\gamma +} - \gamma^N$ to be assigned 1. Unconditionally, a given u is assigned $*$ by $\mathscr{R}_{1-q}^-$ with probability q, hence the upper bound qn^ε, where n^ε comes from the *bfi* condition $|J_{i_\gamma}| \leqslant N^\varepsilon$. Q.E.D.

Now we can finally estimate $\Pr(\mathscr{A})$. It is bounded above by

$$\sum_{\|\mathscr{S}\| > \lceil n \rceil} \Pr(\mathscr{A}[\mathscr{S}]),$$

which is bounded by

$$\sum_{N > \lceil n^\varepsilon \rceil} \sum_{l=1}^{N} \binom{N-1}{l-1} \sum_{l' \geqslant 0} \binom{l'+l}{l} (2\lceil n^\varepsilon \rceil)^N (qn^\varepsilon)^{l'} q^N,$$

where N runs through possible values of the norm of records, $l = \#$ of nonempty X_i in $\mathscr{S}$, and $l' = \#$ of empty X_i in $\mathscr{S}$.

Recall that $q \leqslant n^{-1.05\varepsilon}$. For a fixed ε, $\exists N_\varepsilon$, such that $\forall n > N_\varepsilon$, $16n^{-0.05\varepsilon} < 1/(2e)$. We get, for $n > N_\varepsilon$,

$$\begin{aligned}
\Pr(\mathscr{A}) &\leqslant \sum_{N > \lceil n^\varepsilon \rceil} 2^{N-1} (2\lceil n^\varepsilon \rceil)^N q^N \sum_{l=1}^{N} 2^l \sum_{l' \geqslant 0} 2^{l'} (qn^\varepsilon)^{l'} \\
&\leqslant 2 \sum_{N > \lceil n^\varepsilon \rceil} 2^N (2\lceil n^\varepsilon \rceil)^N q^N 2^N \\
&\leqslant 2 \sum_{N > \lceil n^\varepsilon \rceil} (16n^{-0.05\varepsilon})^N \\
&\leqslant e^{-n^\varepsilon}.
\end{aligned}$$

Hence $\Pr(\mathscr{A}) \leqslant Ce^{-n^\varepsilon}$, $\forall n$, where C only depends on ε. Theorem 3.1 is proven.

4. Depth k Circuits

Theorem 3.1 is proved under a "skewed" probabilistic assignment. We first "unskew" it:

Theorem 4.1. *Fix $0 < \varepsilon < \frac{1}{5}$. Then there exists a constant C, such that for any circuit $G \in \Pi_{2,n}$ (or $\Sigma_{2,n}$) with bfi $\leqslant n^\varepsilon$ and any p with $0 \leqslant p \leqslant n^{-2.2\varepsilon}$, and $Q \in \mathscr{R}_p$, $G \|_Q$ is n^ε-monochromatic, with probability $1 - \varepsilon_n$, where $\varepsilon_n \leqslant Ce^{-n^\varepsilon}$.*

Proof. Clearly we only need to prove the $\Pi_{2,n}$ case. For $0 \leqslant p \leqslant n^{-2.2\varepsilon}$, let

$$p' = 1 - \frac{1}{2} \frac{(1-p)^2}{1 - (p(2-p))^{1/2}},$$

$$q = \frac{(p(2-p))^{1/2} - p}{1-p}.$$

It is easy to verify that $0 \leqslant p' \approx \frac{1}{2} \leqslant 1$, $0 \leqslant q = O(p^{1/2}) \leqslant n^{-1.05\varepsilon}$. Take random $R \in \mathscr{R}_{p'}^{+}$, $Q_1 \in \mathscr{R}_{1-q}^{-}$, and $Q_2 \in \mathscr{R}_{1-q}^{+}$. It is straightforward to show that RQ_1Q_2 has the same distribution as $Q \in \mathscr{R}_p$.

Now we apply Theorem 3.1 to each $G \|_R$ and the result follows. Q.E.D.

THEOREM 4.2. *Let $k \geqslant 2$, $1 \leqslant j \leqslant k-1$. Let $p = n^{-1/k}$, and let*

$$0 < \frac{1}{4k} = \varepsilon_k < \varepsilon_{k-1} < \cdots < \varepsilon_1 = \frac{1}{3k}$$

be equally spaced.

For any $G \in \Pi_{j+1,n}$ (or $\Sigma_{j+1,n}$) with bfi $\leqslant n^{1/3k}$ and size$(G) \leqslant e^{n^{1/4k}}$, and random $A_1, \ldots, A_j$ from $\mathscr{R}_p$, $G \|_{A_1,\ldots,A_j}$ is $n^{1/3k}$-monochromatic with probability $1 - O(\exp(-n^{\varepsilon_j}))$.

Note. The constant in the O-notation depends only on k.

Proof. Fixing $k \geqslant 2$, we prove the theorem by induction on j. Base case $j = 1$. $G \in \Pi_{2,n}$, with $bfi \leqslant n^{1/3k}$. Taking $\varepsilon = 1/3k < \frac{1}{5}$, $p = n^{-1/k} \leqslant n^{-2.2\varepsilon}$ in Theorem 4.1, we have $G \|_{A_1}$ is $n^{1/3k}$-monochromatic with probability $1 - O(\exp(-n^{1/3k}))$. The proof is similar for $G \in \Sigma_{2,n}$.

Now suppose $j > 1$, and the theorem is true for $j-1$. We prove the theorem for the $G \in \Sigma_{j+1,n}$ case. The $\Pi_{j+1,n}$ case is dual.

Let $G = \Sigma_{i=1}^{l} K_i$, where $K_i \in \Pi_{j,n}$. Since G has $bfi \leqslant n^{1/3k}$, and $size(G) \leqslant e^{n^{1/4k}}$, $l \leqslant e^{n^{1/4k}}$, and every K_i inherits the condition on *bfi* and *size*. Let $B_i = K_i \|_{A_1,\ldots,A_{j-1}}$; then $G \|_{A_1,\ldots,A_{j-1}} = \sum_{i=1}^{l} B_i$.

By our inductive hypothesis, for any i fixed, we have B_i is $n^{1/3k}$-monochromatic, with probability $1 - O(\exp(-n^{\varepsilon_{j-1}}))$, where the constant is independent of B_i. Hence, with probability $1 - O(\exp(-n^{(\varepsilon_{j-1}+\varepsilon_j)/2}))$ all B_i are simultaneously $n^{1/3k}$-monochromatic. Again, the constant here depends only on k.

By Lemma 2.2, all B_i are equivalent to $\Sigma_{2,n}$-formulae, and thus with probability $1 - O(\exp(-n^{(\varepsilon_{j-1}+\varepsilon_j)/2}))$, $G \|_{A_1,\ldots,A_{j-1}}$ is equivalent to a Σ_2-formula with $bfi \leqslant n^{1/3k}$. Applying Theorem 4.1 once more, we get that $G \|_{A_1,\ldots,A_j}$ is $n^{1/3k}$-monochromatic with probability $1 - O(\exp(-n^{\varepsilon_j}))$. Q.E.D.

Taking $j = k-1$ in Theorem 4.2, we obtain:

COROLLARY 4.3. *Let $k \geqslant 2$, $p = n^{-(k-1)/k}$. For any $G \in \Pi_{k,n}$ (or $\Sigma_{k,n}$) with $bfi \leqslant n^{1/3k}$ and size$(G) \leqslant e^{n^{1/4k}}$, and a random $R \in \mathscr{R}_p$, $G \|_R$ is $n^{1/3k}$-monochromatic with probability $1 - o(1)$, uniformly.*

We note that the restriction on *bfi* is only technical; one may always extend one more level of alternation to have $bfi \leqslant 1$.

5. Circuits vs Parity

In this section we complete the proof of Theorem 2.1. By the remark at the end of last section, we need only prove:

THEOREM 5.1. *Let $k \geq 2$. There exists a sequence $\{\alpha_n\}$, $\alpha_n \to \frac{1}{2}$, such that all depth k Boolean circuits, with n inputs, size $\leq \exp(n^{1/4k})$, and bfi $\leq n^{1/3k}$, when computing Parity$_n$, make errors on $\geq \alpha_n$ of all 2^n possible inputs.*

The strategy to prove Theorem 5.1 is the following: Fix $k \geq 2$. Consider any depth k circuit G satisfying the conditions. Randomly take a total assignement σ (all 2^n many assignements from $\{0, 1\}^n$ are equally likely). We wish to prove that $G|_\sigma \neq Parity_n|_\sigma$, with probability $\frac{1}{2} - o(1)$, where $o(1)$ may depend on k, but it is independent of G.

Now we pick σ in two stages: First, randomly pick a "single $*$" σ^*, so that all $\sigma^* \in A^* \equiv \{\sigma \in \{0, 1, *\}^n \mid \exists$ a unique $d, \sigma_d = *\}$ are equally likely. Then assign the unique $*$ in σ^* to 0 or 1 with equal probability, to obtain our radom σ.

Theorem 5.1 will be proved if we can show that $G|_{\sigma^*} \equiv$ constant, with probability $1 - o(1)$, since for any σ^*, the conditional probability for failure is

$$\Pr(G|_\sigma \neq Parity_n|_\sigma \mid G|_{\sigma^*} \equiv \text{constant}) = 50\%.$$

Now our strategy to generate a random $\sigma^* \in A^*$ is the following: Let $p = n^{-(k-1)/k}$ and $q = n^{-1/2.5k}$. For a nonempty finite set of variables S, an "A^*-uniform assignement" on S is a random assignment that randomly picks one variable in S as $*$ and uniformly assigns the others to 0 or 1.

Procedure Generate 1 (σ^*)
  Take a random $A \in \mathscr{R}_p$
  if (A leaves $\leq n^{1/2k}$ variables in X unassigned)
  then take a random σ^*
  else take a random $B \in \mathscr{R}_q$
    if (AB assigned every variable in X)
    then take a random σ^*
    else let $S = \{x_i \in X \mid AB$ assigns x_i to $*\}$,
      A^*-uniformly assign S,
      let σ^* be the result.
    fi
  fi Return (σ^*)

Clearly Generate 1 does generate every $\sigma^* \in A^*$ equally likely. Now we "realize" Generate 1 by the following procedure, which will complete our proof:

Procedure Generate 2 (σ^*, *tag*)
Take a random $A \in \mathscr{R}_p$
if (A leaves $\leqslant n^{1/2k}$ variables in X unassigned)
then *tag* := failure, take a random σ^*
else **if** ($G \|_A$ is not $n^{1/3k}$-monochromatic)
then *tag* := failure, take a random $B \in \mathscr{R}_q$
if (AB assigned every variable in X)
then take a random σ^*
else let $S = \{x_i \in X \mid AB$ assigns x_i to $*\}$,
A^*-uniformly assign S,
let σ^* be the result.
fi
else play the game (as the player),
assign any given variable with distribution $\mathscr{R}_q$
if (the master ever gets a $*$)
then stop the game, *tag* := failure,
run through $\mathscr{R}_q$ for the remaining variables,
let $S = \{x_i \in X \mid x_i$ is unassigned so far$\}$,
A^*-uniformly assign S,
Let σ^* be the result.
else when the game is finished, run through $\mathscr{R}_q$ for the rest,
if (no variable is assigned $*$)
then *tag* := failure, take a random σ^*
else *tag* := success,
let $S = \{x_i \in X \mid AB$ assigns x_i to $*\}$,
A^*-uniformly assign S,
Let σ^* be the result.
fi
fi
fi
fi Return (σ^*, *tag*)

Clearly if we ignore the *tag*, Generate 2 is the same as Generate 1. If Generate 2 returns (σ^*, success), then $G|_{\sigma^*} \equiv$ constant. Let F denote the event that Generate 2 returns with *tag* = failure. We claim:

$$\Pr(F) = o(1).$$

We only need to verify:

1. $\Pr(A \in \mathscr{R}_p$ leaves $\leqslant n^{1/2k}$ variables in X unassigned$) = o(1)$. This follows from Chebechev's inequality.

2. $\Pr(G \|_A$ is not $n^{1/3k}$-monochromatic$) = o(1)$. This is Corollary 4.3.

3. Pr(the master gets a $*$ under $\mathscr{R}_q \mid G \parallel_A$ is $n^{1/3k}$-monochromatic) $= o(1)$. This is because $n^{1/3k} \cdot n^{-1/2.5k} \to 0$.

4. Pr(AB leaves no $*$ in $X \mid A$ leaves $\geqslant n^{1/2k}$ variables unassigned in X) $= o(1)$. This is trivial.

6. Final Remarks

The result concerning circuit and parity is of interest independently of relativization. After all, one cannot do worse than 50% error for parity.

The following corollary is evident.

Corollary 6.1 (Yao). *There is a recursive oracle A separating PSPACE from the polynomial-time hierarchy.*

The proof is simple. Observe that with probability one the parity language $Parity^A$ is not in PH^A. Hence for those A, $L^A(M_i)$ differs from $Parity^A$ infinitely often for any PH machine M_i. By the definition of measure μ, any initial segment of A corresponds to a small interval of $[0, 1]$. Now suppose we are given an initial segment of A, the oracle constructed so far, and we want to diagonalize over M_i. What we do is simply look for an extension that kicks M_i out. The "brute force" method must succeed due to our probability one separation.

Shortly after this work, Hastad [9] obtained a simplification of Yao's proof, improving the bound on the circuit size from $\Omega(e^{n^{1/4k}})$ to $\Omega(e^{cn^{1/k}})$. Later Babai [Bab86] obtained the result in Theorem 1.1 by a short proof, assuming Yao's theorem and a result by Ajtai [Ajt83].

The following question is still open:

- Is it true that with probability one, a random oracle separates the polynomial-time hierachy PH into an infinite hierarchy?

Acknowledgments

The author expresses his sincere gratitude to Professor Juris Harmanis for his constant encouragement and inspiration, without which this work would have been impossible. The author also thanks Professors R. Book, N. Immerman, D. Joseph, S. Mahaney, Y. Moschovakis, R. Shore, and A. Yao and my fellow students L. Hemachandra, J. Johnstone, and S. Smith for very stimulating conversations. Thanks are also due to the anonymous referee, whose comments have enhanced the presentation.

References

[Ajt83] M. Ajtai, Σ_1^1-formulae on finite structures, *Ann. Pure Appl. Logic* **24** (1983), 1–48.

[Bab86] L. Babai, A random oracle separates *PSPACE* from polynomial hierarchy, Notes, 1986, to appear.

[BG81] C. BENNET AND J. GILL, Relative to a random oracle A, $P^A \neq NP^A$ with probability 1, *SIAM J. Comput.* **10** (1981), 96–113.

[BGS75] T. BAKER, J. GILL, AND R. SOLOVAY, Relativization of $P = ?NP$ question, *SIAM J. Comput.* **4** (1975), 431–442.

[BS79] T. BAKER AND A. SELMAN, A second step toward the Polynomial hierarchy, *Theoret. Comput. Sci.* **8** (1979), 177–187.

[CH86] J. CAI AND L. A. HEMACHANDRA, The Boolean hierarchy: Hardware over *NP*, *in* "Structure in Complexity Theory," pp. 105–124, Lecture Notes in Computer Science, Vol. 223, Springer-Verlag, New York/Berlin, 1986.

[CKS81] A. CHANDRA, D. KOZEN, AND L. STOCKMEYER, Alternation, *J. Assoc. Comput. Mach.* **26**, No. 1 (1981).

[FSS84] M. FURST, J. SAXE, AND M. SIPSER, Parity, circuits, and the polynomial-time hierarchy, *Math. Systems Theory* **17** (1984), 13–27.

[Has86] J. HASTAD, Almost optimal lower bounds for small depth circuits, *in* "Proceedings, ACM Symposium on Theory of Computation, 1986," pp. 6–20.

[Hu79] J. HOPCROFT AND J. ULLMAN, "Introduction to Automata Theory, Languages, and Computation," Addison–Wesley, Reading, MA, 1979.

[Sip83] M. SIPSER, Borel sets and circuit complexity, *in* "Proceedings, ACM Symposium on Theory of Computation, 1983," pp. 61–69.

[Sto77] L. STOCKMEYER, The polynomial-time hierarchy, *Theoret. Comput. Sci.* **3** (1977), 1–22.

[Yao85] A. YAO, Separating the polynomial-time hierarchy by oracles, *in* "Proceedings, IEEE Annual Symposium on Foundations of Computer Science, 1985," pp. 1–10.