With Probability One, A Random Oracle Separates PSPACE from The Polynomial-Time Hierarchy

Jin-yi Cai

TR 85-715
December 1985

Department of Computer Science
Cornell University
Ithaca, NY 14853

With Probability One, A Random Oracle Separates PSPACE From The Polynomial -Time Hierarchy

Jin-yi Cai

Computer Science Department
Cornell University
Ithaca, NY 14853

Draft : August 1985

Revised : November 1985

Key word : PSPACE, Polynomial-Time Hierarchy, Parity, Boolean Cirsuit, Random Oracle.

abstract

We consider how much error a fixed depth Boolean circuit has to make for 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 asymptotically *50%* error on all possible input, uniformly.

As a consequence, we show that with a random oracle set A, $Pr.(\text{PSPACE}^A \supsetneq \text{PH}^A) = 1$.

§ 1. Introductions and Preliminaries

We assume the readers are familiar with the notions P, NP, Σ_k^P,..., and PSPACE. The union of all Σ_k^P is the polynomial-time hierarchy PH [MS] [St]. It is unknown, although generally conjectured, that all of these classes are distinct.

There has been much work in this area, separating the classes by oracles. Most notably the work by Baker, Gill and Solovay [BGS], by Baker and Selman [BS], by Bennet and Gill [BG], and most recently by Yao [Y2].

In this paper, we prove the following theorems.

Theorem 1.1. Fix depth Boolean circuits, when computing parity, err on *50%* of all the cases, asymptotically and uniformly, provided that the sizes of the circuits are bounded by $exp(n^\lambda)$, for some contant $\lambda>0$. (λ depends only on the depth.)

Theorem 1.2. For a random oracle A, PSPACE$^A \supsetneq$ PH^A, with probability 1.

We proceed with some definitions.

Let X be the set of n Boolean variables $\{x_1, ..., x_n\}$. A $\Sigma_{1,n}$-formula (circuit) is a sum of the form $\Sigma_k(\neg x_{i_k}) + \Sigma_k x_{j_k}$, where $x_{i_k}, x_{j_k} \in X$. WOLOG, we assume 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 $\Pi_k(\neg x_{i_k}) \cdot \Pi_k x_{j_k}$.

For $k>1$, a $\Sigma_{k,n}$ -formula (circuit) H is a sum of $\Pi_{k-1,n}$-formulae. $\Sigma_{1\le i\le l}\, G$, with size (H) $=\Sigma_{1\le i\le l}$ size (G_i). A $\Pi_{k,n}$-formula is the negation of a $\Sigma_{k,n}$-formula, with the same size.

The bottom fan-in (*bfi.*) of a Boolean circuit is the largest of sizes of depth 1 subcircuits.

Fix any $\Pi_{2,n}$-formula G , $G = \Pi_{1\le i\le l} C_i$, where

$$C_i = \overline{x_{i_1}} + \overline{x_{i_2}} + \cdots + \overline{x_{i_s}} + x_{i_{s+1}} + x_{i_{s+2}} + \cdots + x_{i_{s+t}}, \qquad s,t\ge 0$$

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

We will consider circuit computation in relation to the parity function, $Parity_n$, where $Parity_n(x_1, ..., x_n) = \Sigma_{1\le i\le n}\, x_i \pmod 2$. x_i = 0 or 1. We choose the parity function for the following property: the value of $Parity_n$ is *vitally* dependent on every single variable x_i.

A (partial) assignment of X is an n-tuple $\sigma = (\sigma_1, \dots , \sigma_n) \in \{ 0, 1, * \}^n$. If $\sigma \in \{ 0, 1 \}^n$, then σ is a total assignment. 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 $0 \leq p \leq 1$, let $\mathbf{R}_p$ denote the probability space $\{ 0,1,* \}^\omega$ with a product measure μ, where (independently) for each coordinate i, $1 \leq i < \omega, \mu\{ (a_1, \dots , a_i, \dots) \mid a_i = a \} = (1-p)/2$, if $a = 0$ or 1: and p if $a = *$. A random assignment is denoted as $A = (a_1, \dots , a_i, \dots)$.

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

We denote $F\|_A$ the Boolean function resulted from the assignment.

Similarly we define onesided random assignments. A random B taken from $\mathbf{R}_{p+}$ (denoted as $B \in \mathbf{R}_{p+}$) assigns independently to each x_{i_j} in F to 1 with probability p , and unassigned with probability $1-p$, respectively. $\mathbf{R}_{p-}$ is defined in the same way. Note, all random assignments only affect free variables, when they are applied to a formula.

Consider a sequence of random assignments $R_1, \dots ,R_s$. $F\|_{R_1 \dots R_s}$ is defined to be $(F\|_{R_1 \dots R_{s-1}})\|_{R_s}$. The successive random assignments only act on the variables left untouched by previous assignments. We also denote $R_1 \cdots R_s$ as $A = (a_1, \dots , a_j, \dots)$, where $a_j = 1, 0$ or $*$, depending on $R_1 \cdots R_s$ assigns x_j to 1, 0 or unassigned, respectively.

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

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

A Boolean function is given, G. Consider the following class of 2-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 must make G a constant.

A Boolean function G is *k-monochromatic iff* there is a 2-man game of the defined class, in which the master has a winning strategy in the following sense : the master can declare the end 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 assigned so far.

2. Even in a 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.

Lemma 1.3. If G is k-monochromatic, then G is a constant, or equivalent to a $\Sigma_{2,n}$ - circuit (as well as a $\Pi_{2,n}$ - circuit) with $bfi \leq k$.

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 $< k$. Let G be k-monochromatic, but not $(k-1)$-monochromatic. Let's play the game; suppose x_i is the first variable the master puts out. Then $G = [\, x_i \wedge G|_{x_i = \mathrm{T}} \,] \vee [\, (\neg x_i) \wedge G|_{x_i = \mathrm{F}} \,]$. Now both $G|_{x_i = \mathrm{T}}$ and $G|_{x_i = \mathrm{F}}$ are $(k-1)$-monochromatic; we apply the inductive hypothesis, the result follows.$\square$

§ 2. Initial Reductions

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

$$L^A = \{\, 1^n \mid \textit{there are odd number of strings of length } n \textit{ in } A \,\}$$

Clearly, $L^A \in \mathrm{PSPACE}^A$, $\forall A$.

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 binary expansion $\in [0,1]$. Now we define the probability measure ν on the oracle space to be the Lebesque measure on $[0,1]$.

We aim to prove that $\nu\{\, A : L^A \notin \mathrm{PH}^A \,\} = 1$. Surely this implies that a random oracle separates PSPACE from the entire polynomial hierarchy , with probability 1.

There are only countably many levels Σ_i^p in PH. Each Σ_i^p has an r.e. presentation as the class of languages accepted by polynomial-time alternation machines at that level [CKS]. Let $M_1, M_2, \ldots$ be an enumeration of alternation machines , then it is sufficient to show that

$$\forall i: \ \nu\{\, A : L^A \neq L(M_i^A) \,\} = 1 \qquad (\mathrm{I})$$

A theorem in [BG] makes things even easier : they showed that the following is sufficient

$$\exists \varepsilon_0 > 0 \ \forall i: \ \nu\{A : L^A \neq L(M_i^A)\} > \varepsilon_0 \qquad \text{(II)}$$

A careful reader will notice that our proof in this paper actually proves (I) (by iterating at different lengths); but we choose only to verify (II) to simplify the proof somewhat.

Now we reduce alternation machines to Boolean circuits. [FFS] has a nice account on this reduction; we will only give a sketch here and refer the readers to their paper for more detail.

For a fixed alternation machine with oracle M^A, at level Σ_k^p, consider its computation on 1^m. It is always possible to postpone the queries of strings. The trick is to guess the answer and verify them in a later stage. By adding one more level of the alternation, we may obtain an equivalent polynomial time bounded alternetion machine, that only queries at the bottom level and only queries once on every computation path. Thus the computation tree structrue 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 alternation 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 $1^m \in {}^? L^A$ is independent of any string of length $\neq m$; furthermore, fix any string x of length m, it does depend on x whether 1^m belongs to the parity language, L^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.

Suppose then $\exists\, x, |x| = m$, and x is never queried . Then clearly the machine is making error precisely half the probability (under ν) 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 $\neq$ m.

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

$$\nu\{A : L^A(1^m) \neq (M_i^A(1^m))\} \geq \textit{error rate of } G|_{\sigma_0} \textit{ for parity} \equiv \frac{\textit{card.}(\{\iota \in \{0,1\}^n \mid G|_{\sigma_0}|_\iota \neq Parity_n|_\iota\})}{2^n}$$

where $n = 2^m$.

Finally a remark on the size. Since $M_i(1^m)$ runs at most $p(m)$ steps, for some polynomial p, the size of the circuit is bounded by exponential polylog in n, the input size to the circuit.

We have shown that *Thm 1.2.* follows from the following, which is an exact account of *Thm 1.1.*

Theorem 2.1. $\forall k \geq 2$, $\exists \{ a_n \}$, $a_n \rightarrow 1/2$, such that, all depth k Boolean circuits, with n input, size $\leq \exp(n^{1/4(k+1)})$, when computing Parity$_n$, make error on $\geq a_n$ of all 2^n possible input.

The rest of this paper is devoted to the proof of *Theorem 2.1.* We will start with depth 2 circuits in the next section.

§ 3. Depth two circuits : The Game and the Recording

We wish to prove a theorem concerning depth 2 Boolean circuits.

Theorem 3.1. Fix $0<\varepsilon<1/5$, then $\exists$ C, such that , for any $G \in \Pi_{2,n}$ with $bfi \leq n^{\varepsilon}$, $0 \leq q \leq n^{-1.05\varepsilon}$, take random $Q_1 \in \mathbf{R}_{l\text{-}q}{}_{-}$, $Q_2 \in \mathbf{R}_{l\text{-}q}{}_{+}$, then $G\|_{Q_1Q_2}$ is n^{ε}-monochromatic, with probability $1-\varepsilon_n$, where $\varepsilon_n \leq C\, e^{-n^{\varepsilon}}$.

The idea of the proof is as follows: We will define a 2-man game , which we claim the master has a winning strategy (cf. § 1). The game is designed so that each play creates a *record* as to how the game was played. In the rare case $G\|_{Q_1Q_2}$ 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*, a random assignment R *survives* Recording (R, *record*) is so unlikely, that even summing up the probability over all "large" *records*, it is still of measure near 0.

We denote an assignment Q_1Q_2 as $A = (a_1 , a_2 , \ldots)$, as in § 1. Suppose $G = C_1 \wedge \ldots \wedge C_l$. Let $N = \{ 1, \ldots, \lceil n^{\varepsilon} \rceil \}$, $X = \{ \langle Z, s \rangle \mid Z \subseteq N,\ s \in \{0,1\}^{|Z|} \}$. Define $\| \langle Z, s \rangle \| =$ card. (Z) . A record $\boldsymbol{S}$ is a finite sequence $\langle X_1 , \ldots, X_l \rangle$; where $X_i \in X$. Define the norm $\| \boldsymbol{S} \| = \Sigma_{i=1\ldots l} \| X_i \|$.

Intuitively, when a record element $X_i = \langle Z, s \rangle$ is generated in 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 $J = \{i_1, \ldots, i_a\}$, $Z = \{z_1, \ldots, z_\beta\}$, where $a, \beta \geq 0$, $1 \leq i_1 < \ldots < i_a$, $1 \leq z_1 < \ldots < z_\beta$, then Z codes the subset of J:

$$\left\{ i_{z_1}, \ldots, i_{z_\beta} \right\}, \textit{ if } z_\beta \leq a.$$

We denote this set as $J \oplus Z$. If $Z = \varnothing$, then $J \oplus Z = \varnothing$. If $z_\beta > a$, $J \oplus Z$ is undefined .

Conversely for $\Delta = \{ i_{z_1}, \ldots, i_{z_\beta} \} \subseteq J$, we denote ,

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

Note that $J \ominus \varnothing = \varnothing$. Clearly for $\Delta \subseteq J$, $J \oplus (J \ominus \Delta) = \Delta$.

The 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 $\boldsymbol{S} = \langle X_1, \dots, X_l \rangle$ is given. There are 4 essential variables κ, $Y(1)$, $Y(*)$ and $N(*)$, respectively representing our knowledge about the assignment A at any given point in the execution. (More accurately, our knowledge about A for which we can be *forced* to acknowledge).

```
Procedure Recording ( A, S )
0     Y(1), Y(*), N(*), κ := ∅;  t := 0;  List := [ C_1, ... , C_l ];
Repeat
1     if List = ∅ then case 1 : t < l ⇒ "abort"; case 2 : t ≥ l ⇒ "result" fi
2     let C_i be on the top of List
3     if ( ∃ t ∈ J_{i-}, a_t = 0 ) then κ := κ ∪ {J_{i-}}, delete C_i from List
4     else [ critical round ]
5           t := t + 1
6           F := J_i − ( Y(1) ∪ Y(*) ∪ N(*) )
7           get X_t = ⟨ Z, s ⟩ from S , "abort" if nonexistent
8           D := F ∘ Z, "abort" if undefined
9           if ( D ≠ { j ∈ F | a_j = * } ) then "abort"
10          else  Y(1) := Y(1) ∪ { u ∈ F | a_u = 1 }
                  κ := κ ∪ { { u } | u ∈ F , a_u = 0 }
                  Y(*) := Y(*) ∪ { u ∈ D : s assigns u to 1 }
                  N(*) := N(*) ∪ { u ∈ D : s assigns u to 0 }
            fi
11          if ( D = ∅ ) then
12                if ( ∀ u ∈ J_{i+} − N(*), a_u = 0 ) then "abort" fi
            fi
13          Delete any C_k from List with J_{k-} ∩ N(*) ≠ ∅ or J_{k+} ∩ ( Y(1) ∪ Y(*) ) ≠ ∅
      fi [ c.r. ]
End [ Repeat ]
```

Some properties of Recording are easily verified. Define $List_{out}$ to be the set of j such that C_j has been deleted from $List$. (c.r. is a short hand for "critical round"). We have:

Lemma 3.2. (1) κ, $Y(1)$, $Y(*)$, $N(*)$ and $List_{out}$ are monotonically non-decreasing.

Everytime the **Repeat** loop is entered, the followings are true:

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

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

(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 $J_{j+} \cap (Y(1) \cup Y(*)) = \varnothing$.

(7) $\forall j \in List_{out}$, $J_{j-} \in \kappa$ or $J_{j-} \cap N(*) \neq \varnothing$ or $J_{j+} \cap (Y(1) \cup Y(*)) \neq \varnothing$.

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

Proof: A straight forward check.□

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

- In the initialization part (line 0), add $\boldsymbol{S} := \varnothing$.
- Change line 1 to: **if** $List = \varnothing$ **then** "result" $\boldsymbol{S}$
- Change lines 7 and 8 to: Create X_t: $Z := F \ominus \{j \in F \mid a_j = *\}$: $D := F - Z$ ($= \{j \in F \mid a_j = *\}$)

  for $j \in D$ **do** let the player assign x_j and record in $\boldsymbol{S}$ with a binary string s of length $|D|$

  $X_t := \langle Z, s \rangle$.

Let ρ be the assignment made by the player : $\rho_d = 0, 1$ or $*$: if $d \in N(*)$, $Y(*)$ or o.w.

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 $\boldsymbol{S}$ be the record it created, then Recording (A, $\boldsymbol{S}$) will run precisely the same as Game (A, $\boldsymbol{S}$), 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 $D = \varnothing$. In the case $D = \varnothing$ 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 C_i is the current clause. But then C_i must be deleted at line 13.

Therefore the Game will eventually halt. Let $\boldsymbol{S_0}$ be the record created when the Game halts.

We prove by induction, that Recording (A, $\boldsymbol{S_0}$) will reproduce the game play: Suppose they both enter a new round with all the variables the same. (certainly true initially). Also assume t = the length of $\boldsymbol{S_0}$ constructed so far in the game.

If $List = \varnothing$, then the game halts in result. Since $t = l$, the length of $\boldsymbol{S_0}$, Recording (A, $\boldsymbol{S_0}$) will also halt in result.

Suppose $List \neq \varnothing$. Then they get the same C_i, 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 it is never altered later in the Game, it is what Recording obtains from $\boldsymbol{S_o}$. From the way X_t is created in the Game, Recording will not halt at line 7,8 and 9. Now for the rest they have the same code. The induction is completed.

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

Followed from *Lemma 3.2.* (4) and (8), 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) and (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.*, $F = [\, J_{i-} - (Y(1) \cup Y(*))\,] \cup [\, J_{i+} - N(*)\,]$, 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 line 3,9 and 11; and if $t \in J_{i+} - N(*)$, $a_t = 0$, by line 12. So $C_i\|_A|_\rho \equiv 0$. □

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

If $A \in \mathcal{A}$, 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.

Being not constant 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 $\boldsymbol{S}$, where $\| \boldsymbol{S} \| > \lceil n^\varepsilon \rceil$.

Therefore $\mathcal{A} \subseteq \bigcup \mathcal{A}_{\boldsymbol{S}}$, where the union is over all $\boldsymbol{S}$, with $\| \boldsymbol{S} \| > \lceil n^\varepsilon \rceil$.

§ 4. A Probability Analysis

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

Define $\mathcal{A}^{m\gamma} = \{ A$: Recording ($A, \boldsymbol{S}$) will come to its mth c.r. with (κ, $Y(1)$, $Y(*)$, $N(*)$) $= \gamma \}$, $\Gamma^m = \{ \gamma : \mathcal{A}^{m\gamma} \neq \varnothing \}$, $\mathcal{A}^m = \bigcup_{\gamma \in \Gamma^m} \mathcal{A}^{m\gamma}$.

We first derive a conditon for $A \in \mathcal{A}^{m\gamma}$.

Lemma 4.1. $\forall \gamma = (\gamma^{\kappa}, \gamma^{1}, \gamma^{Y}, \gamma^{N}) \in \Gamma'''$, $\exists\, i_{\gamma}$, such that $A \in \mathcal{A}^{m\gamma} \Leftrightarrow A$ satisfies the following conditions :

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

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

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

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

Let's prove *Lemma 4.2.* first : Pick any $A^0 \in \mathcal{A}^{m\gamma}$.

Lemma 4.2. A satisfies Conditions (I), (II) and (III) $\Rightarrow$ Recording for A^0 and A will run precisely the same (with all the variables κ, Y (1), Y (*), N (*), *List*, s being the same at corresponding moments) upto line 1 of the m^{th} c.r. of A^0.

Proof: By induction. Suppose they are at line 1 of the m_0 th 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 m th c.r. for A^0, then the induction is completed. Suppose not. Hence A^0 will complete this round without halting. Thus $List \neq \emptyset$, and they pick the same C_i.

If A^0 satisfies the conditon at line 3 , then $J_{i-} \in \gamma^{\kappa}$, since the m th 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 m th yet. A^0 will successfully record all $u \in J_{i-} \cup J_{i+}$ in (κ, Y (1), Y (*), N (*)), which will later become γ. In particular, $\forall u \in J_{i-}$, $u \in \gamma^{1} \cup \gamma^{Y} \cup \gamma^{N}$. By (II) and (III), A must also fail the condition at line 3.

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

In particular, by (I), (II) and (III), A must also find D to be precisely the set of * 's F, and thus update Y (*) and N (*) exactly the same. Similarly , A must update κ and Y (1) the same way as A^0 does, by (I) and (II).

Now if $D \neq \emptyset$, we are done. If $D = \emptyset$, then A^0 will find the condition at line 12 to be false. i.e. $\exists t_0 \in J_{i+} - N(*)$, $a_{t_0}{}^0 \neq 0$. But $D = \emptyset \Rightarrow a_{t_0}{}^0 = 1$. Hence $t_0 \in Y(1)$, which is the same for both A and A^0. Hence $a_{t_0} = 1$ as well. Therefore A won't halt there . The induction is completed. $\square$

Now we prove *Lemma 4.1.* pick $A^0 \in \mathcal{A}^{m\gamma}$, run Recording (A^0, $\boldsymbol{S}$) , let $C_{i_{\gamma}}$ be the clause under question in its m th c.r.

$\Rightarrow$ Since $A \in \mathcal{A}^{m\gamma}$, A satisfies (I), (II) and (III). By *Lemma 4.2.* A and A^0 will reach line 1 of the m th c.r. of A^0, with all the variables the same. Since A^0 comes to line 4, $List \neq \emptyset$, which is the same for

A; so they both pick up C_{i_γ}. Since this is the m th c.r. for A^0, A^0 will fail the condition at line 3, and enter its c.r. with κ unchanged. Thus γ^κ is the common value for κ 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 m th c.r. is yet to come . Since $J_{i_\gamma}{}_-$ is now added to κ by A, $J_{i_\gamma}{}_- \in \gamma^\kappa$. In other words, $J_{i_\gamma}{}_- \in \kappa$ 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 4.2.* we can assume they arrive at line 1 of the m th c.r. of A^0, with exactly the same history.

$A^0 \in A^{m\gamma} \Rightarrow List \neq \emptyset$ and A picks up C_{i_γ}. Then (IV) says that this is also a c.r. for A. Since this is the m th c.r. for A^0, $A^0 \in A^{m\gamma}$, and so far they are the same, this is also the m th c.r. for A, with $(\kappa, Y(1), Y(*), N(*)) = \gamma$. Hence $A \in A^{m\gamma}$.$\square$

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

$$Pr.(A_{\boldsymbol{S}}) \le Pr.(A^1)\cdot \Pi_{1 \le m < l} Pr.(A \in A^{m+1} \mid A \in A^m)\cdot Pr.(E^l \mid A \in A^l)$$

$$\le [\Pi_{1 \le m < l} Pr.(E^m \mid A \in A^m)]\cdot Pr.(E^l \mid A \in A^l)$$

$$= \Pi_{1 \le m \le l} Pr.(E^m \mid A \in A^m)$$

We show the following:

Lemma 4.3. $\forall \gamma \in \Gamma^m$, $Pr.(E^m \mid A \in A^{m\gamma}) \le q^{\|X_m\|}$ if $\|X_m\| \neq 0$

$\le q\cdot n^\varepsilon$ o.w.

Clearly, *Lemma 4.3.* implies the same bound for $Pr.(E^m \mid A \in A^m)$, since it can be estimated as

$$\Sigma_{\gamma \in \Gamma^m} Pr.(E^m \mid A \in A^{m\gamma})\cdot Pr.(A \in A^{m\gamma} \mid A \in A^m), \text{ and } \Sigma_{\gamma \in \Gamma^m} Pr.(A \in A^{m\gamma} \mid A \in A^m) = 1,$$

Hence,

Corollary 4.4. $Pr.(A_{\boldsymbol{S}}) \le q^{\|\boldsymbol{S}\|}\cdot(qn^\varepsilon)^{l'}$, where $l' = \#$ of X_i in $\boldsymbol{S}$ with $\| X_i \| = 0$.

We use *Lemma 4.1.* to prove *Lemma 4.3.*

Assume $\|X_m\| \neq 0$. Consider a random assignment taken from $\mathsf{R}_{l\text{-}q}{}_-$, followed by one from $\mathsf{R}_{l\text{-}q}{}_+$, on the variables in $F = J_{i_\gamma} - (\gamma^l \cup \gamma^Y \cup \gamma^N)$. Clearly the conditions can be most favorably strengthened to that all variables in F be assigned * by the first round $\mathsf{R}_{l\text{-}q}{}_-$. (Since in order to remain * after two sweeps, it must remain * after the first). Note that originally the condition on F is with $\mathbf{R}_{1\text{-}q}\,-$ only.For $\mathbf{R}_{1\text{-}q}\,+$, a given $u \in F$ is assigned * with probability q. Thus the upper bound $q^{\|X_m\|}$.

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

$Pr.(\, D = \varnothing$ is all the * 's in $F \wedge \exists\, t \in J_{i_\gamma +} - \gamma^N,\, a_t \neq 0 \,|\, A \in A^{m\gamma}\,)$

$\leq Pr.(\,\exists\, t \in J_{i_\gamma +} - \gamma^N,\, a_t = 1 \,|\, A \in A^{m\gamma}\,)$

We consider two sweeps from $\mathbf{R}_{1\text{-}q}\,-$ followed by $\mathbf{R}_{1\text{-}q}\,+$, on $J_{i_\gamma +} - \gamma^N$. The conditions (II),(III) and (IV) are irrelavant now, (using independence). And condition (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 $\mathbf{R}_{1\text{-}q}\,-$ with probability q, hence the upper bound qn^ε, where n^ε comes from the condition $|J_{i_\gamma}| \leq n^\varepsilon$. □

Now we can finally estimate $Pr.(\,A\,)$. It is bounded above by $\sum_{\|S\| > \lceil n^\varepsilon \rceil} Pr.(\,A_S\,)$, which is bounded by

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

where N stands for norm, l = # of nonempty X_i in $\boldsymbol{S}$. l' = # of empty X_i in $\boldsymbol{S}$.

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

$$Pr.(\,A\,)$$

$$\leq \sum_{N > \lceil n^\varepsilon \rceil} 2^{N-1} (2\lceil n^\varepsilon \rceil)^N q^N \sum_{l=1}^{N} 2^l \sum_{l' \geq 0} 2^{l'} (q\,n^\varepsilon)^{l'}$$

$$\leq 2 \sum_{N > \lceil n^\varepsilon \rceil} 2^N (2\lceil n^\varepsilon \rceil)^N q^N 2^N$$

$$\leq 2 \sum_{N > \lceil n^\varepsilon \rceil} (16\, n^{-0.05\varepsilon})^N$$

$$\leq e^{-n^\varepsilon}$$

Hence $Pr.(\,A\,) \leq C e^{-n^\varepsilon}$, $\forall\, n$: where C only depends on ε.

Theorem 3.1 is proven.

§ 5. Depth k Circuits

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

Theorem 5.1. Fix $0<\varepsilon<1/5$. Then $\exists\ C$, such that for any circuit $G \in \Pi_{2,n}$ (or $\Sigma_{2,n}$) with $bfi \le n^{\varepsilon}$, $0\le p\le n^{-2.2\varepsilon}$, take $Q \in \mathbf{R}_p$, then $G\|_Q$ is n^{ε}-monochromatic, with probability $1-\varepsilon_n$, where $\varepsilon_n \le C\, e^{-n^{\varepsilon}}$.

Proof: Clearly we only need to show for the $\Pi_{2,n}$ case.

For $0\le p\le n^{-2.2\varepsilon}$, let

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

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

Now apply *Theorem 3.1.* to each $G\|_R$, the result follows. $\square$

Theorem 5.2. Let $k \ge 2$, $1\le j\le k-1$. Let $p = n^{-1/k}$, and

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

be equally spaced.

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

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

Proof: Fix $k \ge 2$, we prove the theorem by induction on j.

Base case $j = 1$. $G \in \Pi_{2,n}$ (or $\Sigma_{2,n}$) , with $bfi \le n^{1/3k}$. Take $\varepsilon = 1/3k < 1/5$, $p = n^{-1/k} \le n^{-2.2\varepsilon}$ in *Theorem 5.1.* we have $G\|_{A_1}$ is $n^{1/3k}$-monochromatic with probability $1-O(\,exp(\,-n^{1/3k}))$.

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

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

By induction hypothesis, with probability $1-O(\,exp(\,-n^{\varepsilon_{j-1}}))$, B_i is $n^{1/3k}$-monochromatic , for any i fixed, 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 only depends on k.

By *Lemma 1.3.*, all B_i are equivalent to Σ_2-formulae, and thus $G \|_{A_1 \ldots A_{j-1}}$ is equivalent to a $\Sigma_{2,n}$-formula with $bfi \leq n^{1/3k}$, with probability $1 - O\,(exp\,(-n^{(\varepsilon_{j-1}+\varepsilon_j)/2}))$. Apply *Theorem 5.1.* once more, we get $G \|_{A_1 \ldots A_j}$ is $n^{1/3k}$-monochromatic with probability $1 - O\,(\,exp\,(-n^{\varepsilon_j})\,)$.□

Take $j = k-1$ in *Theorem 5.2.* we obtain

Corollary 5.3. Let $k \geq 2$. $p = n^{-(k-1)/k}$. For any $G \in \Pi_{k,n}$ (or $\Sigma_{k,n}$) with $bfi \leq n^{1/3k}$ and size $(G) \leq e^{n^{1/4k}}$, take random $R \in \mathbf{R}_p$, then $G \|_R$ 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 $bfi \leq 1$.

§ 6. Circuits vs. Parity

In this section we complete the proof of *Theorem 2.1.* By the remark we made at the end of § 5, we need only to prove:

Theorem 6.1. Let $k \geq 2$. $\exists \{a_n\}$, $a_n \to 1/2$, such that, all depth k Boolean circuits, with n input, size $\leq \exp(\,n^{1/4k}\,)$, and $bfi \leq n^{1/3k}$, when computing $Parity_n$, make error on $\geq a_n$ of all 2^n possible input.

The strategy to prove *Theorem 6.1.* is the following: Fix $k \geq 2$. Consider any depth k circuit G satisfying the conditons. Take randomly a total assignment σ (all 2^n assignments from $\{0,1\}^n$ are equally likely). We wish to prove that $G|_\sigma \neq Parity_n|_\sigma$, with probability $1/2 - o(1)$. (Where $o(1)$ may depend on k, but it is independent of G).

Now we choose to 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 1 or 0 with equal probability, to obtain σ.

Theorem 6.1. 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

$$Pr.(\; G|_\sigma \neq Parity_n|_\sigma \;\big|\; G|_{\sigma_*} \equiv \text{constant}\;) = 50\,\%.$$

Now our "strategy " to generate a random $\sigma_* \in A_*$ is the following: Let $p = n^{-(k-1)/k}$, $q = n^{-1/2.5k}$.

Procedure Generate1 (σ_*)

Take a random $A \in \mathbf{R}_p$

if (A leaves $\leq n^{1/2k}$ variables in X unassigned) **then** take a random σ_*

else take a random $B \in \mathbf{R}_q$

  if (AB assigned every variable in X) **then** take a random σ_*

    else among the variables in X unassigned by AB, randomly pick one as *, and assign the others randomly to 1 or 0. Let σ_* be the result

  fi

fi Return (σ_*).

Clearly Generate1 does generate every $\sigma_* \in A_*$ equally likely.

Now we " realize " Generate1 by the following procedure, which will complete our proof:

Procedure Generate2 (σ_*, tag)

Take a random $A \in \mathbf{R}_p$

if (A leaves $\leq n^{1/2k}$ 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 $B \in \mathbf{R}_q$

    if (AB assigned every variable in X) **then** take a random σ_*

      else among the variables in X unassigned by AB, randomly pick one as *, and assign the others randomly to 1 or 0. Let σ_* be the result

    fi

  else play the game (as the player), assign any given variable with distribution $\mathbf{R}_q$

    if (the master ever gets a *) **then** stop the game; tag := failure; run through $\mathbf{R}_q$ for the rest; pick a * randomly (there must be one) and assign the other *'s to 1 or 0 randomly. Let σ_* be the result.

    else when the game is finished, run through $\mathbf{R}_q$ for the rest:

      if (no variable is assigned *) **then** tag := failure, take a random σ_*

      else tag := success

        pick a random * and assign other * randomly to 1 or 0. Let σ_* be the result

      fi

    fi

  fi

fi Return (σ_*, tag).

Clearly if we ignore the tag, Generate2 is the same as Generate1. If Generate2 returns (σ_*, success) , then $G|_{\sigma_*} \equiv$ constant. Let **F** denote the event that Generate2 returns with tag = failure. We claim:

$$Pr.(\mathbf{F}) = o(1)$$

We only need to verify:

1. *Pr.* ($A \in \mathbf{R}_p$ leaves $\leq n^{1/2k}$ 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 5.3*.

3. *Pr.* (the master ever gets a * under $\mathbf{R}_q$ | $G \|_A$ is $n^{1/3k}$-monochromatic)= o(1).

This is because $n^{1/3k} \cdot n^{-1/2.5k} \rightarrow 0$.

4. *Pr.* (AB leaves no * in X | A leaves $\geq n^{1/2k}$ variables unassigned in X) = o(1).

This is trivial.

§ 7. Final Remarks

The result concerning circuit and parity is of independent interests other than relativization. Notice that one can't do any worse than *50 %* error for parity.

The following *Corollary* grew out of a conversation with Prof. Y.Moschovakis.

Corollary : there is a recursive oracle A separating PSPACE and the Polynomial-Time Hierarchy.

The proof is simple. Observe that with probability 1 the Parity Language L^A is not in PH^A. Hence for any PH machine M_i, $L^A(M_i)$ differs from L^A infinitely often. Recall the definition of measure v, any initial segment of A corresponds to a small inteval of [0,1]. Now suppose we are given an initial segment A_m 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 work due to our probablity 1 separation.

Only a little over a month after this work is completed and presented at Cornell, the author learned with delight during his visit to MSRI in late September, that Mr. Hastad has obtained a simplification of Yao's proof. His proof is only 9 pages.

The author wishes to express his sincere gratitude to Prof. Juris Hartmanis, for his constant encouragement and inspiration, without which this work would be impossible. The author also wishes to thank Profs. R. Book, N. Immermen, D. Joseph, S. Machany, R. Shore and A. Yao and my fellow students L. Hemachandra, J. Jonesone and S. Smith for very stimulating conversations.

References

[BGS] T.Baker, J.Gill and R.Solovay, "Relativization of P=?NP question", SIAM J. on Computing 4 (1975), 431-442

[BS] T.Baker and A.Selman, "A second step toward the polynomial hierarchy", Theretical Computer Science 8 (1979), 177-187

[BG] C.Bennet and J.Gill, "Relative to a reandom oracle A, $P^A \neq NP^A \neq$ co-NP^A with probability 1", SIAM J. on Computing 10 (1981), 96-113

[CKS] A.Chandra, D.Kozen and L.Stockmeyer , "Alternation", JACM, 28,1981

[FSS] M.Furst, J.Saxe and M.Sipser, "Parity, circuits and the polynomial time hierarchy", Proc. of 22 nd FOCS, 1981, 260-270

[MS] A.Meyer and L.Stockmeyer, "The equivalence problem for regular expressions with squaring requires exponetial space. " Proc. of 13th Ann. IEEE Sym. on Switching & Aut. Th. 1972, 125-129

[St] L.Stockmeyer, "The polynomial-time hierarchy", Theo. Comp. Sc. 3 (1977) 1-22

[Y1] A. Yao, "Lower bounds by probabilistic arguments". Proc. of 24th FOCS, 1983

[Y2] A. Yao, "Separating the Polynomial -Time Hierarchy by Oracles", to appear on FOCS 1985