Saving Queries With Randomness*

Pankaj Rohatgi

TR 91-1259
December 1991

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*This research was supported in part by NSF Research Grant CCR 88-23053.

Saving Queries with Randomness *

Pankaj Rohatgi

Department of Computer Science
Cornell University
Ithaca, NY 14853

December 20, 1991

Abstract

In this paper, we provide *tight* bounds on the success probabilities of randomized reductions between various classes in the Boolean and Bounded Query Hierarchies. The $\mathrm{P}^{\mathrm{SAT}\|[k]}$ $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete language randomly reduces to a language in $\mathrm{P}^{\mathrm{SAT}\|[k-1]}$ with a one-sided error probability of $1/\lceil k/2 \rceil$. If two-sided error is allowed, then we show that a much lower error probability of $1/(k+1)$ can be achieved. We prove that both these reductions are almost optimal by showing that the error probabilities cannot be reduced by even $1/poly$, unless the PH collapses. These tight bounds precisely characterize the power and limitations of randomness in saving a query to SAT.

These results can be used to identify optimal probability *thresholds* which determine when languages complete under randomized reductions inherit the hardness properties associated with $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete languages. Using these thresholds we prove hardness properties for some languages in these classes which are not $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete in certain relativized worlds.

We also explore the relationship between randomization and functions computable using bounded queries to SAT. For any function $h(n) = \mathrm{O}(\log n)$, we show that there is a function f computable using $h(n)$ *nonadaptive* queries to SAT, which cannot be computed correctly with probability $1/2 + 1/poly$ by any randomized machine which makes less than $h(n)$ *adaptive* queries to any oracle, unless PH collapses.

1 Introduction

Randomness is a useful computational resource due to its ability to enhance the capabilities of other resources. Its interaction with resources such as time, space, *interaction with provers* and its role in areas such as algorithm design, parallel computing and circuit complexity have been studied extensively. We examine how randomness interacts with another well studied computational resource - the number of queries allowed to an oracle.

Treating the number of queries to SAT as a computational resource gained acceptance with Krentel's work [Kre88] on NP Optimization problems. It is well known that there is a wide variation in the complexity of typical NP Optimization problems although their decision versions have the same complexity. Krentel was able to explain these differences by characterizing the complexity of optimization problems by the number of SAT-queries required to compute the optimal solution.

Often, one is only interested in finding out whether the optimal solution satisfies some predicate. In complexity theoretic terms, recognizing these *languages* should be easier than computing the

*This research was supported in part by NSF Research Grant CCR 88-23053.

optimal solution. Wagner [Wag86] extended Krentel's approach to such languages by characterizing their complexity by the number of SAT-queries required to evaluate the predicate. This also extended the work of Papadimitriou and Yannakakis [PY82] who defined the class D^P while studying the complexity of the facets of the TSP polytope. D^P is the class of languages expressible as a difference of two NP languages and a canonical example of a language in D^P is $\{ \langle I, k \rangle \mid P(I,k) \}$ where I is an instance of some NP Optimization problem and $P(I,k)$ is the *simple* predicate $Optimum(I) = k$.

An example of a language based on a more complex predicate is CLIQUE(5) defined as follows: Given a graph G and five integers $a_1, \ldots, a_5$, is it the case that the size of the largest clique in G is one of the a_i's ? Assuming NP $\neq$ co-NP, CLIQUE(5) cannot be in NP although it is related to the NP Optimization problem MAXCLIQUE. Since the predicate $P(G,k) =$ "maxclique in G has size exactly k" is D^P-complete and CLIQUE(5) is based on *five* instances of this predicate, one would expect it to be more complex than the D^P-complete languages.

The complexity of languages such as CLIQUE(5) can be characterized by the number of non-adaptive queries to SAT required to recognize them (10 will suffice in this case). Another closely related way of measuring the complexity of such languages is to see how they can be expressed as a set theoretic combination of a minimum number of NP languages (CLIQUE(5) is a nested difference of 10 NP languages). The formulation based on the number of queries gives rise the Bounded Query Hierarchy and the formulation based on set theoretic operations gives rise to the Boolean Hierarchy[CGH+88].

Subsequently, the robustness of these two complexity measures was investigated by many researchers [Kre88, Kad88, Bei87, Bei88, ABG90]. Research was also conducted on how the usefulness of bounded queries as a resource was affected by the complexity of the oracle itself. [AG88, Bei, GJY87, Bei87, Cha89, ABG90]. Of particular significance are results that state that, for all constants k, under usual complexity theoretic assumptions, k nonadaptive (adaptive) queries to SAT are more powerful than $k-1$ nonadaptive (adaptive) queries [Kre88, Kad88]. For instance, these results imply that CLIQUE(5) is not in $P^{SAT||[9]}$, unless PH collapses. In light of these negative results, it is natural to ask whether randomization can be used to bridge this gap and if so, to what extent. This question was first examined in [CKR91] where nontrivial upper and lower bounds were provided on the error probability of randomized reductions in the Boolean and Query Hierarchies. However there was a large gap between these bounds. For instance, the lower bound on the error in reducing $P^{SAT||[k]}$ to $P^{SAT||[k-1]}$ was roughly $1/exp(k)$ whereas the upper bound was roughly $1/linear(k)$.

We extend this work by proving *tight* bounds on various randomized reductions in the Boolean and Bounded Query classes. Any language in $P^{SAT||[k]}$ randomly reduces to a language in $P^{SAT||[k]}$ with one-sided error probability of $1/\lceil k/2 \rceil$. If two-sided error is allowed then we show that the error probability can be reduced to $1/(k+1)$. We prove that it is not possible to reduce these error probabilities even by $1/poly$, unless PH collapses. Observe that as k increases, the error bound *decreases*, thus giving a precise mathematical justification to our intuition that, for language recognition, the value of an additional nonadaptive query decreases as the number of queries increase. It was not possible to formalize this intuition without randomness because the deterministic result states that $\forall k$, $P^{SAT||[k]} \neq P^{SAT||[k-1]}$, unless PH collapses[Kad88].

Thus, CLIQUE(5) can be recognized with a two-sided error probability of only 1/11, using randomness and 9 nonadaptive queries to SAT. The randomized computation of CLIQUE(5) in $P^{SAT||[9]}$ is simple and elegant and it is surprising that more complex computations can't decrease the error significantly, unless PH collapses.

Our proof techniques exploit the rich structure within the Boolean and Query Hierarchies and rely on the important observation that some structure can be imposed even on arbitrary randomized reductions between classes in the Boolean Hierarchy.

Our bounds can be used to establish sharp probability *thresholds* above which languages complete under randomized reductions in these hierarchies inherit most of the hardness properties of the $\leq_m^P$-complete languages. As a consequence we can show that several languages in this hierarchy which are not $\leq_m^P$-complete in certain relativized worlds, nevertheless, behave almost like the $\leq_m^P$-complete languages.

Finally we explore the relationship between randomization and functions computable using bounded queries to SAT and we show that randomization is not helpful in this case.

2 Definitions, Notation and Background

We assume familiarity with the classes NP, co-NP, the NP-complete set SAT, the Polynomial time Hierarchy (PH) and the usual probabilistic and nonuniform classes.

Notation Let $\{0,1\}^n$ denote the set of n-bit strings. For any set A, let $A^{=n}$ denote the set of n-bit strings in A.

We now define the Bounded Query Hierarchy.

Definition We write $\mathrm{P}^{\mathrm{SAT}||[k]}$ for the k^{th} level of the Bounded Query Hierarchy. The class $\mathrm{P}^{\mathrm{SAT}||[k]}$ consists of all languages recognized by polynomial time Turing machines which are allowed at most k parallel (or non-adaptive) queries to the SAT oracle.

In this paper, we work with the *finer* query hierarchy based on *nonadaptive* queries; the k'th level of the query hierarchy based on *adaptive* queries being exactly $\mathrm{P}^{\mathrm{SAT}||[2^k-1]}$ [Bei87]. Bounded Query classes lacks structure. Results about these classes often rely on the structural properties of the closely related Boolean Hierarchy. The Boolean Hierarchy [CGH+88] is a natural generalization of the class $\mathrm{D^P}$. We now define this hierarchy. Let $L_1, \ldots, L_n$ denote arbitrary languages. Define the operator C as

$$C(L_1) \stackrel{\text{def}}{=} L_1$$

$$C(L_1, \ldots, L_n) \stackrel{\text{def}}{=} \begin{cases} C(L_1,\ldots,L_{n-1})\bigcup L_n & \text{if } n \text{ is odd} \\ C(L_1,\ldots,L_{n-1})\bigcap \overline{L_n} & \text{otherwise} \end{cases}$$

Definition We write BH(k) and co-BH(k) for the k^{th} levels of the Boolean hierarchy, defined as:

$$\mathrm{BH}(k) \stackrel{\text{def}}{=} \{L \mid L = C(L_1,\ldots,L_n) \text{ for some } L_1,\ldots,L_k \in \mathrm{NP}\}$$

$$\text{co-BH}(k) \stackrel{\text{def}}{=} \{L \mid \overline{L} \in \mathrm{BH}(k)\}$$

Note that BH(1) corresponds to NP and BH(2) corresponds to $\mathrm{D^P}$. A prominent member of $\mathrm{D^P}$ is is the set of uniquely satisfiable boolean formulas (USAT). Also, the set of all (G,k) such that the maximum clique size in graph G is *exactly* k, is complete for $\mathrm{D^P}$.

Every level of the Boolean Hierarchy has complete languages including languages based on important NP Optimization problems [CGH+88]. For example, the language CLIQUE(5) defined earlier is complete for BH(10). From the definition of the classes BH(k) and co-BH(k) it is not hard to show that the following languages are complete for the respective levels of the Boolean Hierarchy:

Definition We write $\mathrm{L}_{\mathrm{BH}(k)}$ for the canonical complete language for $\mathrm{BH}(k)$ and $\mathrm{L}_{\text{co-BH}(k)}$ for the complete language for co-$\mathrm{BH}(k)$:

$$\mathrm{L}_{\mathrm{BH}(1)} \stackrel{\text{def}}{=} \mathrm{SAT}$$

$$\mathrm{L}_{\mathrm{BH}(2k)} \stackrel{\text{def}}{=} \{\langle x_1,\ldots,x_{2k}\rangle \mid \langle x_1,\ldots,x_{2k-1}\rangle \in \mathrm{L}_{\mathrm{BH}(2k-1)} \text{ and } x_{2k} \in \overline{\mathrm{SAT}}\}$$

$$\mathrm{L}_{\mathrm{BH}(2k+1)} \stackrel{\text{def}}{=} \{\langle x_1,\ldots,x_{2k+1}\rangle \mid \langle x_1,\ldots,x_{2k}\rangle \in \mathrm{L}_{\mathrm{BH}(2k)} \text{ or } x_{2k+1} \in \mathrm{SAT}\}$$

$$\mathrm{L}_{\text{co-BH}(1)} \stackrel{\text{def}}{=} \overline{\mathrm{SAT}}$$

$$\mathrm{L}_{\text{co-BH}(2k)} \stackrel{\text{def}}{=} \{\langle x_1,\ldots,x_{2k}\rangle \mid \langle x_1,\ldots,x_{2k-1}\rangle \in \mathrm{L}_{\text{co-BH}(2k-1)} \text{ or } x_{2k} \in \mathrm{SAT}\}$$

$$\mathrm{L}_{\text{co-BH}(2k+1)} \stackrel{\text{def}}{=} \{\langle x_1,\ldots,x_{2k+1}\rangle \mid \langle x_1,\ldots,x_{2k}\rangle \in \mathrm{L}_{\text{co-BH}(2k)} \text{ and } x_{2k+1} \in \overline{\mathrm{SAT}}\}$$

Since NP and co-NP are closed under boolean ANDs and ORs, one can prove the following interesting fact about languages in the Boolean Hierarchy over NP sets.

Fact 1 In the definition of the class $\mathrm{BH}(k)$ we can also assume that the NP languages $L_1, \ldots, L_k$ are such that

$$L_k \subseteq L_{k-1} \subseteq \cdots \subseteq L_2 \subseteq L_1.$$

One of the interesting consequences of Fact 1 is the following:

Notation Let π_j denote j^{th} projection function, and $\pi_{(i,j)}$ denote the function that selects the i^{th} through j^{th} elements of a k-tuple. For example,

$$\pi_j(\langle x_1,\ldots,x_k\rangle) = x_j \quad\text{and}\quad \pi_{(i,j)}(\langle x_1,\ldots,x_k\rangle) = \langle x_i,\ldots,x_j\rangle.$$

Fact 2 Let L be any language in $\mathrm{BH}(k)$. Then $L \le_{\mathrm{m}}^{\mathrm{P}}$-reduces to $\mathrm{L}_{\mathrm{BH}(k)}$ via a reduction h which has the following properties:

- For any x, $h(x)$ is the k-tuple $\langle h_1(x),\ldots,h_k(x)\rangle$.
- $h(x) \in \mathrm{L}_{\mathrm{BH}(k)} \iff x \in L$.
- $\forall x, i,\ (2 \le i \le k)\ \ h_i(x) \in \mathrm{SAT} \implies h_{i-1}(x) \in \mathrm{SAT}$

One such h can be obtained by observing that $L = C(L_1, \ldots, L_k)$, where $L_1, \ldots, L_k$ are NP languages such that

$$L_k \subseteq L_{k-1} \subseteq \cdots \subseteq L_2 \subseteq L_1.$$

We can therefore define $h_i(x)$ to be the formula obtained by applying Cook's reduction to the $x \in L_i$? question.

The Bounded Query Hierarchy and the Boolean Hierarchy are closely related[Bei87]. In fact,

$$\mathrm{BH}(k) \cup \text{co-BH}(k) \subseteq \mathrm{P}^{\mathrm{SAT}||[k]} \subseteq \mathrm{BH}(k+1) \cap \text{co-BH}(k+1)$$

Figure 1 shows the relationships between the various levels of the Boolean and Query Hierarchies. Both the hierarchies are proper unless PH collapses [Kad88]

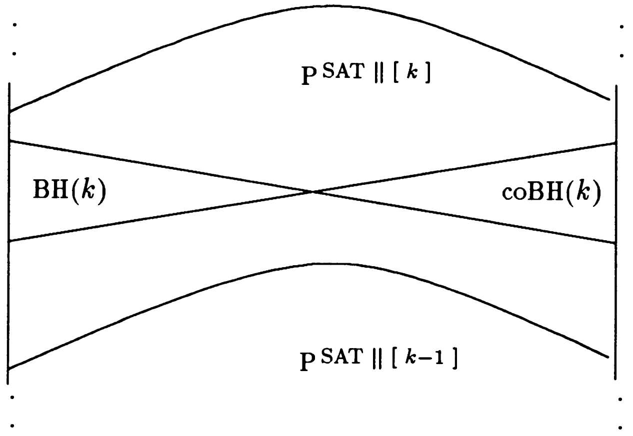

Figure 1: The structure of the Boolean and Query Hierarchies

A beautiful result due to Beigel [Bei87] shows that the $\leq_m^P$-complete language for $P^{SAT||[k]}$ is just the tagged union of the canonical $\leq_m^P$-complete languages of the k^{th} level of the Boolean Hierarchy, i.e., $L_{BH(k)}$ and $L_{co\text{-}BH(k)}$. We shall call the canonical $P^{SAT||[k]}$ $\leq_m^P$-complete language $L_{BH(k)} \oplus L_{co\text{-}BH(k)}$ and a formal definition follows:

Definition The canonical $P^{SAT||[k]}$ $\leq_m^P$-complete language is denoted by $L_{BH(k)} \oplus L_{co\text{-}BH(k)}$ and is defined as

$$L_{BH(k)} \oplus L_{co\text{-}BH(k)} \stackrel{\text{def}}{=} \{0y \mid y \in L_{BH(k)}\} \bigcup \{1y \mid y \in L_{co\text{-}BH(k)}\}$$

We now define random reductions between sets. These definitions are generalizations of the definitions given in [AM77] and [VV86].

Definition $A \leq_m^{rp} B$ with probability δ, if there exists a polynomial time function f such that

$$x \in A \Longrightarrow \text{Prob}_z[\ f(x,z) \in B\ ] \geq \delta$$
$$x \notin A \Longrightarrow \text{Prob}_z[\ f(x,z) \notin B\ ] = 1$$

where z is chosen uniformly at random from $\{0,1\}^{q(|x|)}$, for some polynomial q.

Definition We say that $A \leq_m^{co-rp} B$ with probability δ, if $\overline{A} \leq_m^{rp} \overline{B}$ with probability δ. $\leq_m^{co-rp}$-reductions are similar to $\leq_m^{rp}$-reductions except that the error occurs on the other side.

Definition $A \leq_m^{bpp} B$ with probability δ, if there exists a polynomial time function f such that

$$\text{Prob}_z[\ x \in A \iff f(x,z) \in B\ ] \geq \delta$$

where z is chosen uniformly at random from $\{0,1\}^{q(|x|)}$, for some polynomial q and $\delta > 1/2$.

3 Randomization and Bounded Query Languages

We now examine the power and limitation of randomization in reducing harder languages in the Query and Boolean Hierarchies to simpler ones. Figure 2 summarizes the results.

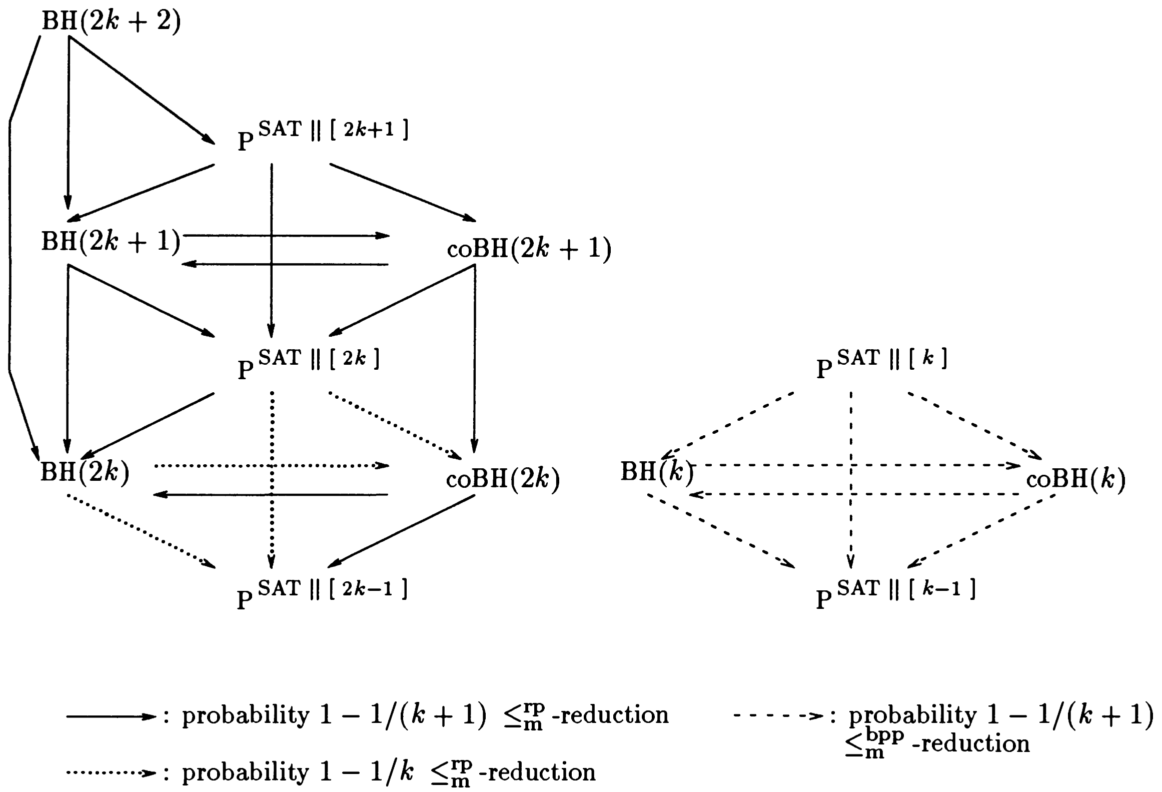

Figure 2: Almost Optimal $\leq_{\mathrm{m}}^{\mathrm{rp}}$ and $\leq_{\mathrm{m}}^{\mathrm{bpp}}$ reductions

3.1 Almost Optimal Randomized Reductions

First we show that randomized reductions can reduce harder languages to simpler ones with the probabilities shown in Figure 2. The $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions are simpler than the $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reductions but their success probabilities are lower.

Lemmas 1 and 2 can be used to derive all the $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions in Figure 2. The basic idea behind these reductions is to express the harder language as a union of t sets such that excluding any one of these sets from the union gives rise to a simpler language. Then by randomly excluding one of the sets from the union we get an $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction to a simpler language with error $1/t$.

Lemma 1 [CKR91] For any $k \geq 2$, $\mathrm{L}_{\mathrm{BH}(2k)} \leq_{\mathrm{m}}^{\mathrm{rp}} \mathrm{L}_{\mathrm{BH}(2k-2)}$ with probability $1 - 1/k$.

Proof: For all k, it is known that a language $L \in \mathrm{BH}(2k)$ if and only if it can be expressed as a union of k $\mathrm{D^P}$ sets[CGH+88]. If we omit one of the k $\mathrm{D^P}$ sets in the union, we will be left with a language in $\mathrm{BH}(2k-2)$ which $\leq_{\mathrm{m}}^{\mathrm{P}}$-reduces to the complete language $\mathrm{L}_{\mathrm{BH}(2k-2)}$. We can therefore reduce $\mathrm{L}_{\mathrm{BH}(2k)}$ to $\mathrm{L}_{\mathrm{BH}(2k-2)}$ by randomly omitting one of the k $\mathrm{D^P}$ sets from the union. This type of reduction has a one-sided error of up to $1/k$.

Example: The language CLIQUE(5) described earlier is BH(10)-complete. Note that CLIQUE(5) can be represented as a union of 5 $\mathrm{D^P}$ sets ($\mathrm{Maxcliquesize}(G) = a_i$) for $1 \leq i \leq 5$. By randomly excluding one of the 5 sets from this union we can reduce CLIQUE(5) to the simpler language CLIQUE(4) with one sided error of 1/5.

Lemma 2 For any $k \geq 1$, $\mathrm{L}_{\text{co-BH}(2k)} \leq_{\mathrm{m}}^{\mathrm{rp}} \mathrm{L}_{\mathrm{BH}(2k-1)} \oplus \mathrm{L}_{\text{co-BH}(2k-1)}$ with probability $1 - 1/(k+1)$.

Proof: For all $k \geq 1$, it is known that a language $L \in$ co-BH($2k$) if and only if it can be expressed as a union of $k-1$ $\mathrm{D^P}$ sets, an NP set and a co-NP set [CGH+88]. Omitting one of these $k+1$ sets from the union results in either a language in co-BH($2k-2$) (a $\mathrm{D^P}$ set is omitted) or a language in BH($2k-1$) (the co-NP set is omitted) or a language in co-BH($2k-1$) (the NP set is omitted). In all cases, the language will be in $\mathrm{P}^{\mathrm{SAT}||[2k-1]}$. We can therefore reduce $\mathrm{L}_{\text{co-BH}(2k)}$ to $\mathrm{L}_{\mathrm{BH}(2k-1)} \oplus \mathrm{L}_{\text{co-BH}(2k-1)}$ by randomly omitting one of the $k+1$ sets from the union. This type of reduction has a one-sided error of up to $1/k+1$.

Example: Note that $\overline{\mathrm{CLIQUE}(5)}$ is co-BH(10)-complete. Assuming that the integers $a_1, \ldots, a_5$ are in increasing order, it is easy to see that $\overline{\mathrm{CLIQUE}(5)}$ is the union of 4 $\mathrm{D^P}$ sets ([$a_i <$ Maxcliquesize(G) $< a_{i+1}$] for $1 \leq i \leq 4$), an NP set (Maxcliquesize(G) $> a_5$) and a co-NP set (Maxcliquesize(G) $< a_1$). By Lemma 2, $\overline{\mathrm{CLIQUE}(5)}$ can be recognized using randomness and only 9 nonadaptive queries SAT with one-sided error of $1/6$.

Some reductions from $\mathrm{P}^{\mathrm{SAT}||[2k]}$ are based on the fact that the language $\mathrm{L}_{\mathrm{BH}(2k)} \oplus \mathrm{L}_{\text{co-BH}(2k)}$ is complete for that class [Bei87]. In that case, the reduction would be based on both lemmas.

Randomized reductions with two-sided error ($\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reductions) are better at reducing harder languages to simpler ones because they are less constrained. The following lemma along with structural properties of the classes can be used to derive the $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reductions in Figure 2.

Lemma 3 $\mathrm{L}_{\mathrm{BH}(k)} \leq_{\mathrm{m}}^{\mathrm{bpp}} \mathrm{L}_{\mathrm{BH}(k-1)} \oplus \mathrm{L}_{\text{co-BH}(k-1)}$ with probability $1 - 1/(k+1)$.

Proof: Let A denote the language $\mathrm{L}_{\mathrm{BH}(k)}$ and let B denote $\mathrm{L}_{\mathrm{BH}(k-1)} \oplus \mathrm{L}_{\text{co-BH}(k-1)}$ The analysis for odd k and even k is slightly different and we will only deal with the even case. The same ideas are applicable to the odd case.

<u>Case 1:</u> k is even: Let $k = 2t$. Then by Lemma 1 we know that $A \leq_{\mathrm{m}}^{\mathrm{rp}} B$ via a reduction R_1 with probability $1 - 1/t$. Also by Lemma 2 we know that $\overline{A} \leq_{\mathrm{m}}^{\mathrm{rp}} B$ with probability $1 - 1/(t+1)$ via a reduction R_2. Since, the class $\mathrm{P}^{\mathrm{SAT}||[k-1]}$ is closed under complementation, this also means that $\overline{A} \leq_{\mathrm{m}}^{\mathrm{rp}} \overline{B}$, i.e., $A \leq_{\mathrm{m}}^{\mathrm{co-rp}} B$ with probability $1 - 1/(t+1)$ via some reduction R_3.

Now the $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reduction R from A to B will work as follows: On input x, with probability $t/(2t+1)$, R uses the reduction R_1 on x and with the remaining probability [$(t+1)/(2t+1)$], R uses R_3 on x.
Analysis: If $x \in A$ then the R can make an error only when it chooses to use R_1. If it chooses R_1, then it will err with probability at most $1/t$. So the probability of error is at most $t/(2t+1) * 1/t = 1/(2t+1)$. If $x \notin A$, then R can only make an error if it chooses to use R_3. So the error probability is at most $(t+1)/(2t+1) * 1/(t+1) = 1/(2t+1)$. Thus R is a probability $1 - 1/(k+1)$ $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reduction.

<u>Case 2:</u> k is odd. Similar to Case I, details omitted. □

Example: Consider the language CLIQUE(5) which is a union of 5 $\mathrm{D^P}$ sets. We can obtain five simpler languages by removing a set from this union. We can form six more simpler languages by "adding" to CLIQUE(5) one of the six sets whose union constitutes $\overline{\mathrm{CLIQUE}(5)}$. The $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reduction from CLIQUE(5) to a language in $\mathrm{P}^{\mathrm{SAT}||[9]}$ would be to randomly choose one of these *eleven* languages.

3.2 Proofs of Optimality

We now prove that the reductions presented earlier are almost optimal. By almost optimal we mean that there cannot exist reductions whose correctness probability is better by $1/poly$, unless the PH collapses. Minor variants of the same technique and the relationships between the various classes can be used to to derive all the 25 bounds in Figure 2. Therefore, we only prove the result for $\leq_m^{rp}$-reductions from languages in BH($2k$) to languages in co-BH($2k$). As shown earlier, it is possible to reduce any language in BH($2k$) to a language in co-BH($2k$) via a probability $1 - 1/k$ $\leq_m^{rp}$-reduction. The following theorem shows that the probability bound is almost optimal.

Theorem 4 Let L be any language in co-BH($2k$). If $\mathrm{L}_{\mathrm{BH}(2k)} \leq_m^{rp} L$ with probability $1-1/k+1/p(n)$, for some polynomial p, then the PH collapses to Σ_3^{P}.

We will need some definitions and lemmas before we can prove this theorem. Note that the hypothesis of Theorem 4 implies that there *exists* a probability $1 - 1/k + 1/p(n)$ $\leq_m^{rp}$-reduction from $\mathrm{L}_{\mathrm{BH}(2k)}$ to $\mathrm{L}_{\text{co-BH}(2k)}$. Arbitrary $\leq_m^{rp}$-reductions from $\mathrm{L}_{\mathrm{BH}(2k)}$ to $\mathrm{L}_{\text{co-BH}(2k)}$ are quite difficult to work with because, unlike the complete languages in the Boolean Hierarchy, arbitrary reductions lack structure. However, we can establish that if there is some $\leq_m^{rp}$-reduction to a complete language in the Boolean Hierarchy, then there is a $\leq_m^{rp}$-reduction which has very nice structural properties. We will call such reductions *nested* $\leq_m^{rp}$-reductions. We define these reductions as follows:

Definition Let h be a reduction from some language A to $\mathrm{L}_{\mathrm{BH}(\ell)}$ ($\mathrm{L}_{\text{co-BH}(\ell)}$) for some ℓ. We say that h is a *nested* $\leq_m^{rp}$-reduction from A to $\mathrm{L}_{\mathrm{BH}(\ell)}$ ($\mathrm{L}_{\text{co-BH}(\ell)}$ resp.) iff all of the following conditions hold.

1. For all x, $h(x)$ is an ℓ-tuple.
2. $A \leq_m^{rp} \mathrm{L}_{\mathrm{BH}(\ell)}$ ($\mathrm{L}_{\text{co-BH}(\ell)}$ resp.) via h.
3. $\forall x, i,\ (2 \leq i \leq \ell)\ \ \pi_i \circ h(x) \in \mathrm{SAT} \implies \pi_{i-1} \circ h(x) \in \mathrm{SAT}$

The following lemma based on Fact 2 allows us to work with the much nicer *nested* $\leq_m^{rp}$-reductions instead of arbitrary $\leq_m^{rp}$-reductions.

Lemma 5 Let A be any set and let L be an arbitrary language in BH(ℓ) (co-BH(ℓ)). If $A \leq_m^{rp} L$ with probability δ, then there exists a *nested* $\leq_m^{rp}$-reduction h such that $A \leq_m^{rp} \mathrm{L}_{\mathrm{BH}(\ell)}$ ($A \leq_m^{rp} \mathrm{L}_{\text{co-BH}(\ell)}$ resp.) via h with probability δ.

Thus, if the hypothesis of Theorem 4 is true, then there exists a probability $1 - 1/k + 1/p(n)$ *nested* $\leq_m^{rp}$ reduction from $\mathrm{L}_{\mathrm{BH}(2k)}$ to $\mathrm{L}_{\text{co-BH}(2k)}$.

Notation Henceforth we will refer to this *nested* reduction from $\mathrm{L}_{\mathrm{BH}(2k)}$ to $\mathrm{L}_{\text{co-BH}(2k)}$ as h. Let $r(n)$ be the size of the random input to h and let $q(m)$ be the size of $2k$-tuples of strings in $\{0,1\}^m$. Let z denote a randomly & uniformly chosen string of size $r(q(m))$ and let ε denote $1/p(q(m))$.

To explain the importance of *nested* $\leq_m^{rp}$-reductions we provide a rough overview of our proof: The proof is nonuniform; we consider every length. A random reduction between two complementary classes in the Boolean Hierarchy with the appropriate success probability may either result in an adverse structural consequence for the given length or induce a weaker randomized reduction between the two complementary classes one level below. This argument can be used recursively and terminates when one reaches the lowest level of the hierarchy.

If we start with an arbitrary reduction, then the probability loss while going down the hierarchy is so great that we need to start with a probability much above the bound (roughly $1-1/exp(k)+1/poly$ instead of $1-1/k+1/poly$) in order to cause an adverse structural consequences at the lower levels of the hierarchy[CKR91]. If we use a *nested* reduction then the structural properties of the reduction allows us to use the following lemma (**The Probability Recovery Lemma**) to recover some of the probability lost while going down the hierarchy. Also, at intermediate stages of our proof we deal with induced randomized reductions between lower classes in the Boolean Hierarchy which work with certain probabilities. In many instances there exist *better* reductions between these classes *without any assumptions*! We can still prove our result because the reductions induced by the initial *nested* reduction *have the potential* to recover large amounts of lost probability as the proof proceeds whereas the existing reductions between the classes don't. Thus, by using a *nested* reduction we don't need to start with very high success probability and we can in fact obtain tight bounds.

Lemma 6 [Probability Recovery Lemma] Suppose $\mathrm{L}_{\mathrm{BH}(2k)} \leq_{\mathrm{m}}^{\mathrm{rp}} \mathrm{L}_{\text{co-BH}(2k)}$ via h defined above. Then the following proposition $P(j)$ holds for all j, $0 \leq j \leq 2k-1$:

<u>Proposition $P(j)$:</u> Let $x_1, \ldots, x_j$ be any collection of j formulas in $\overline{\mathrm{SAT}}^{=m}$. Then for all $y_1, \ldots, y_\ell \in$ <u>$\{0,1\}^m$ (where $\ell = 2k - j$):</u>
If j is odd:

$$\langle y_1, \ldots, y_\ell \rangle \in \mathrm{L}_{\text{co-BH}(\ell)} \implies \mathrm{Prob}_z[\pi_{(1,\ell)} \circ h(\langle y_1, \ldots, y_\ell, x_j, \ldots, x_1 \rangle, z) \in \mathrm{L}_{\mathrm{BH}(\ell)}] = 1$$

If j is even:

$$\langle y_1, \ldots, y_\ell \rangle \in \mathrm{L}_{\mathrm{BH}(\ell)} \implies \mathrm{Prob}_z[\pi_{(1,\ell)} \circ h(\langle y_1, \ldots, y_\ell, x_j, \ldots, x_1 \rangle, z) \in \mathrm{L}_{\text{co-BH}(\ell)}] \geq 1 - \tfrac{1}{k} + \varepsilon$$

Proof: (by induction on j)

<u>Base Cases $P(0)$ & $P(1)$:</u> These follow trivially from the hypothesis of the lemma and the definition <u>of $\mathrm{L}_{\mathrm{BH}(2k)}$</u>.

<u>Assume $P(j)$, Induction case $P(j+2)$</u> We divide this proof into two subcases depending on the <u>parity of j.</u>

Case 1: j is odd. Since $P(j)$ holds we know that for any collection of j formulas $x_1, \ldots, x_j \in \overline{\mathrm{SAT}}^{=m}$ and for all $y_1, \ldots, y_\ell \in \{0,1\}^m$ where $\ell = 2k - j$:

$$\begin{aligned} &\langle y_1, \ldots, y_\ell \rangle \in \mathrm{L}_{\text{co-BH}(\ell)} \\ &\implies \mathrm{Prob}_z(\pi_{(1,\ell)} \circ h(\langle y_1, \ldots, y_\ell, x_j, \ldots, x_1 \rangle, z) \in \mathrm{L}_{\mathrm{BH}(\ell)}) = 1 \end{aligned}$$

Therefore, for any given set of formulas $x_1, \ldots, x_{j+2} \in \overline{\mathrm{SAT}}^{=m}$, by setting $y_{\ell-1} = x_{j+2}$ and $y_\ell = x_{j+1}$ we have for all $y_1, \ldots, y_{\ell-2} \in \{0,1\}^m$

$$\begin{aligned} &\langle y_1, \ldots, y_{\ell-2}, x_{j+2}, x_{j+1} \rangle \in \mathrm{L}_{\text{co-BH}(\ell)} \\ &\implies \mathrm{Prob}_z(\pi_{(1,\ell)} \circ h(\langle y_1, \ldots, y_{\ell-2}, x_{j+2}, \ldots, x_1 \rangle, z) \in \mathrm{L}_{\mathrm{BH}(\ell)}) = 1 \end{aligned}$$

Since both x_{j+2} and x_{j+1} are in $\overline{\mathrm{SAT}}$, it follows from the definition of $\mathrm{L}_{\text{co-BH}(\ell)}$ (ℓ is odd) that

$$\langle y_1, \ldots, y_{\ell-2}, x_{j+2}, x_{j+1} \rangle \in \mathrm{L}_{\text{co-BH}(\ell)} \iff \langle y_1, \ldots, y_{\ell-2} \rangle \in \mathrm{L}_{\text{co-BH}(\ell-2)}$$

and thus we have that for all $y_1, \ldots, y_{\ell-2} \in \{0,1\}^m$

$$\langle y_1, \ldots, y_{\ell-2} \rangle \in \mathrm{L}_{\text{co-BH}(\ell-2)} \implies \tag{1}$$

$$\mathrm{Prob}_z(\pi_{(1,\ell)} \circ h(\langle y_1, \ldots, y_{\ell-2}, x_{j+2}, \ldots, x_1 \rangle, z) \in \mathrm{L}_{\mathrm{BH}(\ell)}) = 1$$

We claim that

$$\pi_{(1,\ell)} \circ h(\langle y_1, \ldots, y_{\ell-2}, x_{j+2}, \ldots, x_1 \rangle, z) \in \mathrm{L}_{\mathrm{BH}(\ell)} \implies \tag{2}$$

$$\pi_{(1,\ell-2)} \circ h(\langle y_1, \ldots, y_{\ell-2}, x_{j+2}, \ldots, x_1\rangle, z) \in \mathrm{L}_{\mathrm{BH}(\ell-2)}$$

Proof: Let us denote the ℓ-tuple $\pi_{(1,\ell)} \circ h(\langle y_1, \ldots, y_{\ell-2}, x_{j+2}, \ldots, x_1\rangle, z)$ as $\langle h_1, \ldots, h_\ell\rangle$. To prove equation 2, let us assume that $\langle h_1, \ldots, h_\ell\rangle \in \mathrm{L}_{\mathrm{BH}(\ell)}$. We show that this means that $\langle h_1, \ldots, h_{\ell-2}\rangle \in \mathrm{L}_{\mathrm{BH}(\ell-2)}$. By the definition of $\mathrm{L}_{\mathrm{BH}(\ell)}$ for odd ℓ we know that since $\langle h_1, \ldots, h_\ell\rangle \in \mathrm{L}_{\mathrm{BH}(\ell)}$,

$$((\langle h_1, \ldots, h_{\ell-2}\rangle \in \mathrm{L}_{\mathrm{BH}(\ell-2)}) \text{ and } h_{\ell-1} \in \overline{\mathrm{SAT}}) \text{ or } h_\ell \in \mathrm{SAT}$$

Now if $h_\ell \in \overline{\mathrm{SAT}}$ then we can say that

$$\begin{aligned} &((\langle h_1, \ldots, h_{\ell-2}\rangle \in \mathrm{L}_{\mathrm{BH}(\ell-2)}) \text{ and } h_{\ell-1} \in \overline{\mathrm{SAT}}) \\ &\Longrightarrow \langle h_1, \ldots, h_{\ell-2}\rangle \in \mathrm{L}_{\mathrm{BH}(\ell-2)} \end{aligned}$$

If on the other hand, $h_\ell \in \mathrm{SAT}$, then, since h is a *nested* reduction, we know that $h_{\ell-2} \in \mathrm{SAT}$. Note that $j+2 < 2k \Longrightarrow (\ell-2) \geq 1$. If $(\ell-2) = 1$ then that means that

$$\langle h_1, \ldots, h_{\ell-2}\rangle = \langle h_1\rangle \in \mathrm{L}_{\mathrm{BH}(\ell-2)} = \mathrm{BH}(1)$$

since $h_1 \in \mathrm{SAT}$. If, on the other hand, $(\ell-2) > 1$ then we know that since

$$\begin{aligned} &\langle h_1, \ldots, h_{\ell-2}\rangle \in \mathrm{L}_{\mathrm{BH}(\ell-2)} \iff \\ &\langle h_1, \ldots, h_{\ell-3}\rangle \in \mathrm{L}_{\mathrm{BH}(\ell-3)} \text{ or } h_{\ell-2} \in \mathrm{SAT} \end{aligned}$$

and since $h_{\ell-2} \in \mathrm{SAT}$, $\langle h_1, \ldots, h_{\ell-2}\rangle \in \mathrm{L}_{\mathrm{BH}(\ell-2)}$.

Thus we have proved equation 2. Therefore, by equations 1 & 2 we establish $P(j+2)$ for odd j, i.e.,

$\underline{P(j+2)}$:For any collection of $j+2$ formulas $x_1, \ldots, x_{j+2} \in \overline{\mathrm{SAT}}^{=m}$ and for all $y_1, \ldots, y_{\ell-2} \in \{0,1\}^m$ where $\ell = 2k - j$:

$$\begin{aligned} &\langle y_1, \ldots, y_{\ell-2}\rangle \in \mathrm{L}_{\text{co-BH}(\ell-2)} \\ &\Longrightarrow \mathrm{Prob}_z(\pi_{(1,\ell-2)} \circ h(\langle y_1, \ldots, y_{\ell-2}, x_{j+2}, \ldots, x_1\rangle, z) \in \mathrm{L}_{\mathrm{BH}(\ell-2)}) = 1 \end{aligned}$$

Now let us consider the case when j is even

Case 2: j is even. Since $P(j)$ holds we know that for any collection of j formulas $x_1, \ldots, x_j \in \overline{\mathrm{SAT}}^{=m}$ and for all $y_1, \ldots, y_\ell \in \{0,1\}^m$ where $\ell = 2k - j$:

$$\begin{aligned} &\langle y_1, \ldots, y_\ell\rangle \in \mathrm{L}_{\mathrm{BH}(\ell)} \\ &\Longrightarrow \mathrm{Prob}_z(\pi_{(1,\ell)} \circ h(\langle y_1, \ldots, y_\ell, x_j, \ldots, x_1\rangle, z) \in \mathrm{L}_{\text{co-BH}(\ell)}) = 1 - 1/k + \varepsilon \end{aligned}$$

By an analysis very similar to that of Case 1, we can establish from this that $P(j+2)$ holds for even j, i.e.,

$\underline{P(j+2)}$:For any collection of $j+2$ formulas $x_1, \ldots, x_{j+2} \in \overline{\mathrm{SAT}}^{=m}$ and for all $y_1, \ldots, y_{\ell-2} \in \{0,1\}^m$ where $\ell = 2k - j$:

$$\begin{aligned} &\langle y_1, \ldots, y_{\ell-2}\rangle \in \mathrm{L}_{\mathrm{BH}(\ell-2)} \\ &\Longrightarrow \mathrm{Prob}_z(\pi_{(1,\ell-2)} \circ h(\langle y_1, \ldots, y_{\ell-2}, x_{j+2}, \ldots, x_1\rangle, z) \in \mathrm{L}_{\text{co-BH}(\ell-2)}) = 1 - 1/k + \varepsilon \end{aligned}$$

Hence we have proved the lemma by induction. □

We use a variant of the hard/easy proof technique [Kad88] to prove the main theorem. The argument extends the technique used in [CKR91]. We first define a *hard* sequence of formulas.

Definition Suppose $\mathrm{L}_{\mathrm{BH}(2k)} \leq_{\mathrm{m}}^{\mathrm{rp}} \mathrm{L}_{\text{co-BH}(2k)}$ via h. Then, we call $\langle 1^m, x_1, \ldots, x_j\rangle$ a *hard sequence* with respect to h if $j = 0$ or if all of the following hold:

1. $1 \leq j \leq 2k-1$.
2. $|x_j| = m$.
3. $x_j \in \overline{\mathrm{SAT}}$.
4. $\langle 1^m, x_1, \ldots, x_{j-1}\rangle$ is a hard sequence with respect to h.
5. For all $y_1, \ldots, y_\ell \in \{0,1\}^m$ (where $\ell = 2k - j$)

$$\mathrm{Prob}_z(\pi_{\ell+1}\circ h(\langle y_1, \ldots, y_\ell, x_j, \ldots, x_1\rangle, z) \in \mathrm{SAT}) \leq \frac{\lfloor j/2 \rfloor}{k} + (j \bmod 2) * \frac{\varepsilon}{2}$$

If $\langle 1^m, x_1, \ldots, x_j\rangle$ is a hard sequence, then we refer to it as a hard sequence of order j for length m. Also, we call a hard sequence *maximal* if it cannot be extended to a hard sequence of higher order. The following lemma shows that given a *nested* $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction from $\mathrm{L}_{\mathrm{BH}(2k)}$ to $\mathrm{L}_{\text{co-BH}(2k)}$, a hard sequence of order j for length m induces an asymmetric probabilistic reduction from $\mathrm{L}_{\mathrm{BH}(2k-j)}$ to $\mathrm{L}_{\text{co-BH}(2k-j)}$ for tuples of strings of length m.

Lemma 7 Suppose $\mathrm{L}_{\mathrm{BH}(2k)} \leq_{\mathrm{m}}^{\mathrm{rp}} \mathrm{L}_{\text{co-BH}(2k)}$ via h. Then, the following proposition $Q(j)$ holds for all j, $0 \leq j \leq 2k-1$:

Proposition $Q(j)$: If $\langle 1^m, x_1, \ldots, x_j\rangle$ is a hard sequence w.r.t. h, then for all $y_1, \ldots, y_\ell \in \{0,1\}^m$ (where $\ell = 2k - j$):
If ℓ is even:

$$\langle y_1, \ldots, y_\ell\rangle \in \mathrm{L}_{\mathrm{BH}(\ell)} \implies \mathrm{Prob}_z[\pi_{(1,\ell)}\circ h(\langle y_1, \ldots, y_\ell, x_j, \ldots, x_1\rangle, z) \in \mathrm{L}_{\text{co-BH}(\ell)}] \geq 1 - \tfrac{1}{k} + \varepsilon$$

$$\langle y_1, \ldots, y_\ell\rangle \in \mathrm{L}_{\text{co-BH}(\ell)} \implies \mathrm{Prob}_z[\pi_{(1,\ell)}\circ h(\langle y_1, \ldots, y_\ell, x_j, \ldots, x_1\rangle, z) \in \mathrm{L}_{\mathrm{BH}(\ell)}] \geq 1 - \tfrac{j}{2k}$$

If ℓ is odd:

$$\langle y_1, \ldots, y_\ell\rangle \in \mathrm{L}_{\text{co-BH}(\ell)} \implies \mathrm{Prob}_z[\pi_{(1,\ell)}\circ h(\langle y_1, \ldots, y_\ell, x_j, \ldots, x_1\rangle, z) \in \mathrm{L}_{\mathrm{BH}(\ell)}] = 1$$

$$\langle y_1, \ldots, y_\ell\rangle \in \mathrm{L}_{\mathrm{BH}(\ell)} \implies \mathrm{Prob}_z[\pi_{(1,\ell)}\circ h(\langle y_1, \ldots, y_\ell, x_j, \ldots, x_1\rangle, z) \in \mathrm{L}_{\text{co-BH}(\ell)}] \geq 1 - \tfrac{j+1}{2k} + \tfrac{\varepsilon}{2}$$

Proof: (by induction on j)

Base Case $Q(0)$: This follows trivially from the hypothesis of the lemma.

Induction Case $Q(j+1)$: Suppose $Q(j)$ holds. Let $\ell = 2k - j$ and let $\langle 1^m, x_1, \ldots, x_{j+1}\rangle$ be a hard sequence. Consider the cases where ℓ is even or odd separately.

Case 1: ℓ is even. Since $\langle 1^m, x_1, \ldots, x_j\rangle$ is also a hard sequence, by the induction hypothesis, for all $y_1, \ldots, y_\ell \in \{0,1\}^m$

$$\langle y_1, \ldots, y_\ell\rangle \in \mathrm{L}_{\mathrm{BH}(\ell)}$$
$$\implies \mathrm{Prob}_z(\pi_{(1,\ell)}\circ h(\langle y_1, \ldots, y_\ell, x_j, \ldots, x_1\rangle, z) \in \mathrm{L}_{\text{co-BH}(\ell)}) \geq 1 - \tfrac{1}{k} + \varepsilon$$

In particular, for $y_\ell = x_{j+1}$ we have

$$\langle y_1, \ldots, y_{\ell-1}, x_{j+1}\rangle \in \mathrm{L}_{\mathrm{BH}(\ell)}$$
$$\implies \mathrm{Prob}_z(\pi_{(1,\ell)}\circ h(\langle y_1, \ldots, y_\ell, x_j, \ldots, x_1\rangle, z) \in \mathrm{L}_{\text{co-BH}(\ell)}) \geq 1 - \tfrac{1}{k} + \varepsilon$$

Using the definitions of $\mathrm{L}_{\mathrm{BH}(\ell)}$ and $\mathrm{L}_{\text{co-BH}(\ell)}$ for even ℓ, for all $y_1, \ldots, y_{\ell-1} \in \{0,1\}^m$

$$\langle y_1, \ldots, y_{\ell-1}\rangle \in \mathrm{L}_{\mathrm{BH}(\ell-1)} \text{ and } x_{j+1} \in \overline{\mathrm{SAT}} \implies \tag{3}$$

$$\text{Prob}_z \left(\begin{array}{c} \pi_{(1,\ell-1)} \circ h(\langle y_1, \ldots, y_{\ell-1}, x_{j+1}, \ldots, x_1 \rangle, z) \in \mathrm{L}_{\text{co-BH}(\ell-1)} \\ \text{or} \\ \pi_\ell \circ h(\langle y_1, \ldots, y_{\ell-1}, x_{j+1}, \ldots, x_1 \rangle, z) \in \text{SAT} \end{array} \right) \geq 1 - \frac{1}{k} + \varepsilon$$

Since $\langle 1^m, x_1, \ldots, x_{j+1} \rangle$ is a hard sequence, we know conditions 1 and 5 of the definition hold. That is, $x_{j+1} \in \overline{\text{SAT}}$ and for all $y_1, \ldots, y_{2k-j-1} \in \{0,1\}^m$

$$\text{Prob}_z(\pi_{k-j} \circ h(\langle y_1, \ldots, y_{2k-j-1}, x_{j+1}, \ldots, x_1 \rangle, z) \in \text{SAT}) \leq \frac{\lfloor (j+1)/2 \rfloor}{k} + \frac{\varepsilon}{2}$$

i.e. for $\ell = 2k - j$

$$\text{Prob}_z(\pi_\ell \circ h(\langle y_1, \ldots, y_{\ell-1}, x_{j+1}, \ldots, x_1 \rangle, z) \in \text{SAT}) \leq \frac{\lfloor (j+1)/2 \rfloor}{k} + \frac{\varepsilon}{2}$$

So, if $\langle y_1, \ldots, y_{\ell-1} \rangle \in \mathrm{L}_{\text{BH}(\ell-1)}$, then by equation (3) and the fact that $x_{j+1} \in \overline{\text{SAT}}$, we have

$$\text{Prob}_z \left(\begin{array}{c} \pi_{(1,\ell-1)} \circ h(\langle y_1, \ldots, y_{\ell-1}, x_{j+1}, \ldots, x_1 \rangle, z) \in \mathrm{L}_{\text{co-BH}(\ell-1)} \\ \text{or} \\ \pi_\ell \circ h(\langle y_1, \ldots, y_{\ell-1}, x_{j+1}, \ldots, x_1 \rangle, z) \in \text{SAT} \end{array} \right) \geq 1 - \frac{1}{k} + \varepsilon$$

Moreover, by condition 5 described above, we can say that

$$\begin{array}{l} \text{Prob}_z(\pi_{(1,\ell-1)} \circ h(\langle y_1, \ldots, y_{\ell-1}, x_{j+1}, \ldots, x_1 \rangle, z) \in \mathrm{L}_{\text{co-BH}(\ell-1)}) \\ \quad \geq 1 - \frac{1}{k} + \varepsilon - \frac{\lfloor (j+1)/2 \rfloor}{k} - \frac{\varepsilon}{2} \quad = 1 - \frac{j+2}{2k} + \frac{\varepsilon}{2} \end{array}$$

(since j is even, $\lfloor (j+1)/2 \rfloor = j/2$).

Let us now consider the case when $\langle y_1, \ldots, y_{\ell-1} \rangle \in \mathrm{L}_{\text{co-BH}(\ell-1)}$ By conditions 3 & 4 in the definition of a *hard* sequence, we know that $x_1, \ldots, x_{j+1} \in \overline{\text{SAT}}$. Since h is a *nested* $\leq_{\text{m}}^{\text{rp}}$-reduction, by a direct application of the Probability Recovery Lemma we know that in this case

$$\text{Prob}_z(\pi_{(1,\ell-1)} \circ h(\langle y_1, \ldots, y_{\ell-1}, x_{j+1}, \ldots, x_1 \rangle, z) \in \mathrm{L}_{\text{BH}(\ell-1)}) = 1$$

Thus, we have proved $Q(j+1)$ for the case when $\ell = 2k - j$ is even.

Case 2: $\ell = k - j$ is odd. Using a proof similar to the proof of *Case 1* we can show that $Q(j+1)$ holds in this case as well. This completes the proof of the lemma. □

The next lemma states that if $\mathrm{L}_{\text{BH}(2k)} \leq_{\text{m}}^{\text{rp}} \mathrm{L}_{\text{co-BH}(2k)}$ via h, then a *maximal* hard sequence for a given length m allows us to differentiate between the cases $y \in \overline{\text{SAT}}$ and $y \in \text{SAT}$, where y is a formula of length m.

Lemma 8 Suppose $\mathrm{L}_{\text{BH}(2k)} \leq_{\text{m}}^{\text{rp}} \mathrm{L}_{\text{co-BH}(2k)}$ via h. Let $\langle 1^m, x_1, \ldots, x_j \rangle$ be a maximal hard sequence with respect to h. Define $\ell = 2k - j$. Then,

$$y \in \overline{\text{SAT}} \Longrightarrow \\ \left(\begin{array}{l} \exists\, y_1, \ldots, y_{\ell-1} \in \{0,1\}^m, \\ \text{Prob}_z[\pi_\ell \circ h(\langle y_1, \ldots, y_{\ell-1}, y, x_j, \ldots, x_1 \rangle, z) \in \text{SAT}] \ \geq \ \frac{\lfloor (j+1)/2 \rfloor}{k} \ + \ ((j+1) \bmod 2) * \frac{\varepsilon}{2} \end{array} \right)$$

and

$$y \in \text{SAT} \Longrightarrow \\ \left(\begin{array}{l} \forall\, y_1, \ldots, y_{\ell-1} \in \{0,1\}^m, \\ \text{Prob}_z[\pi_\ell \circ h(\langle y_1, \ldots, y_{\ell-1}, y, x_j, \ldots, x_1 \rangle, z) \in \text{SAT}] \ \leq \ \frac{\lfloor (j+1)/2 \rfloor}{k} \ - \ (j \bmod 2) * \frac{\varepsilon}{2} \end{array} \right)$$

Proof:
If $j = 2k - 1$ ($\langle y_1, \ldots, y_{\ell-1} \rangle$ is the empty sequence), then, by Lemma 7, for all $y \in \{0,1\}^m$

$$y \in \overline{\mathrm{SAT}} \Longrightarrow \mathrm{Prob}_z(\pi_1 \circ h(\langle y, x_j, \ldots, x_1 \rangle, z) \in \mathrm{SAT}) = 1$$

and

$$y \in \mathrm{SAT} \Longrightarrow \mathrm{Prob}_z(\pi_1 \circ h(\langle y, x_j, \ldots, x_1 \rangle, z) \in \overline{\mathrm{SAT}}) \geq \frac{\varepsilon}{2}$$

Thus, the lemma holds when $j = 2k - 1$ (i.e. when $y_1, \ldots, y_{\ell-1}$ is the empty sequence).

Consider the case when $j < 2k - 1$. Let $\ell = 2k - j$

Suppose $y \in \overline{\mathrm{SAT}}$. Since $\langle 1^m, x_1, \ldots, x_j \rangle$ is maximal, $\langle 1^m, x_1, \ldots, x_j, y \rangle$ is not a hard sequence. However, $j+1 \leq 2k-1$, $|y| = m$, $y \in \overline{\mathrm{SAT}}$ and $\langle 1^m, x_1, \ldots, x_j \rangle$ is a hard sequence. So, $\langle 1^m, x_1, \ldots, x_j, y \rangle$ must fail to be a hard sequence by failing to satisfy condition 5 of the definition of hard sequences. Thus,

$$\exists\, y_1, \ldots, y_{\ell-1} \in \{0,1\}^m$$
$$\mathrm{Prob}_z(\pi_\ell \circ h(\langle y_1, \ldots, y_{\ell-1}, y, x_j, \ldots, x_1 \rangle, z) \in \mathrm{SAT}) \;>\; \tfrac{\lfloor (j+1)/2 \rfloor}{k} \;+\; ((j+1) \bmod 2) * \tfrac{\varepsilon}{2}$$

Suppose $y \in \mathrm{SAT}$. Let $\ell = 2k - j$. By Lemma 7, for all $\langle y_1, \ldots, y_\ell \rangle$
If ℓ is even:

$$\langle y_1, \ldots, y_\ell \rangle \in \mathrm{L}_{\text{co-BH}(\ell)}$$
$$\Longrightarrow \mathrm{Prob}_z(\pi_{(1,\ell)} \circ h(\langle y_1, \ldots, y_\ell, x_j, \ldots, x_1 \rangle, z) \in \mathrm{L}_{\mathrm{BH}(\ell)}) \geq 1 - \tfrac{j}{2k}$$

and if ℓ is odd:

$$\langle y_1, \ldots, y_\ell \rangle \in \mathrm{L}_{\mathrm{BH}(\ell)}$$
$$\Longrightarrow \mathrm{Prob}_z(\pi_{(1,\ell)} \circ h(\langle y_1, \ldots, y_\ell, x_j, \ldots, x_1 \rangle, z) \in \mathrm{L}_{\text{co-BH}(\ell)}) \geq 1 - \tfrac{j+1}{2k} + \tfrac{\varepsilon}{2}$$

Now let us consider the even and odd cases separately. If ℓ is even, then by the definition of $\mathrm{L}_{\text{co-BH}(\ell)}$, we know that $y \in \mathrm{SAT} \Longrightarrow \forall y_1, \ldots, y_{\ell-1},\ \langle y_1, \ldots, y_{\ell-1}, y \rangle \in \mathrm{L}_{\text{co-BH}(\ell)}$. Therefore,

$$\forall y_1, \ldots, y_{\ell-1} \mathrm{Prob}_z(\pi_{(1,\ell)} \circ h(\langle y_1, \ldots, y_{\ell-1}, y, x_j, \ldots, x_1 \rangle, z) \in \mathrm{L}_{\mathrm{BH}(\ell)}) \geq 1 - \frac{j}{2k}$$

Then, by the definition of $\mathrm{L}_{\mathrm{BH}(\ell)}$ for even ℓ, we know that $\langle u_1, \ldots, u_\ell \rangle \in \mathrm{L}_{\mathrm{BH}(\ell)} \Longrightarrow u_\ell \in \overline{\mathrm{SAT}}$. Thus, we get the desired result

$$\forall y_1, \ldots, y_{\ell-1},\ \mathrm{Prob}_z(\pi_\ell \circ h(\langle y_1, \ldots, y_{\ell-1}, y, x_j, \ldots, x_1 \rangle, z) \in \overline{\mathrm{SAT}}) \geq 1 - \tfrac{j}{2k}$$

i.e.,

$$\forall y_1, \ldots, y_{\ell-1},\ \mathrm{Prob}_z(\pi_\ell \circ h(\langle y_1, \ldots, y_{\ell-1}, y, x_j, \ldots, x_1 \rangle, z) \in \mathrm{SAT})$$
$$\leq \tfrac{j}{2k} \;=\; \tfrac{\lfloor (j+1)/2 \rfloor}{k} \;+\; (j \bmod 2) * \tfrac{\varepsilon}{2}$$

If ℓ is odd then again by unfolding the definition of $\mathrm{L}_{\mathrm{BH}(\ell)}$, we know that $y \in \mathrm{SAT} \Longrightarrow \forall y_1, \ldots, y_{\ell-1}$, $\langle y_1, \ldots, y_{\ell-1}, y \rangle \in \mathrm{L}_{\mathrm{BH}(\ell)}$. Therefore,

$$\forall y_1, \ldots, y_{\ell-1}$$
$$\mathrm{Prob}_z(\pi_{(1,\ell)} \circ h(\langle y_1, \ldots, y_{\ell-1}, y, x_j, \ldots, x_1 \rangle, z) \in \mathrm{L}_{\text{co-BH}(\ell)}) \geq 1 - \tfrac{j+1}{2k} + \tfrac{\varepsilon}{2}$$

By definition of $\mathrm{L}_{\text{co-BH}(\ell)}$ for odd ℓ we have that

$$\pi_{(1,\ell)}{\circ}h(\langle y_1,\ldots,y_{\ell-1},y,x_j,\ldots,x_1\rangle,z) \in \mathrm{L}_{\text{co-BH}(\ell)}$$
$$\Longrightarrow \pi_\ell{\circ}h(\langle y_1,\ldots,y_{\ell-1},y,x_j,\ldots,x_1\rangle,z) \in \overline{\mathrm{SAT}}$$

Thus, we get the required result for odd ℓ:

$$\forall y_1,\ldots,y_{\ell-1}\mathrm{Prob}_z(\pi_\ell{\circ}h(\langle y_1,\ldots,y_{\ell-1},y,x_j,\ldots,x_1\rangle,z) \in \overline{\mathrm{SAT}}) \geq 1 - \frac{j+1}{2k} + \frac{\varepsilon}{2}$$

i.e.,

$$\forall y_1,\ldots,y_{\ell-1}\mathrm{Prob}_z(\pi_\ell{\circ}h(\langle y_1,\ldots,y_{\ell-1},y,x_j,\ldots,x_1\rangle,z) \in \overline{\mathrm{SAT}})$$
$$\geq \tfrac{j+1}{2k} + \tfrac{\varepsilon}{2} \;=\; \tfrac{\lfloor (j+1)/2 \rfloor}{k} \;+\; (j \bmod 2) * \tfrac{\varepsilon}{2}$$

This completes the proof of the Lemma. □

Now we are in a position to prove the main theorem:

Theorem 4 Let L be any language in co-BH($2k$). If $\mathrm{L}_{\mathrm{BH}(2k)} \leq_{\mathrm{m}}^{\mathrm{rp}} L$ with probability $1-1/k+1/p(n)$, for some polynomial p, then the PH collapses to Σ_3^{P}.

Proof: By Lemma 5 we know that $\mathrm{L}_{\mathrm{BH}(2k)} \leq_{\mathrm{m}}^{\mathrm{rp}} \mathrm{L}_{\text{co-BH}(2k)}$ with probability $1 - 1/k + 1/p(n)$ via the *nested* $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction h. Thus Lemma 8 is applicable. Given h, let f be the advice function which on input 0^m outputs the lexically smallest maximal *hard* sequence for length m.

We define an NP machine N which on input $F\#a\#y\#z$ does the following. Suppose a is of the form $\langle 1^m, x_1, \ldots, x_j\rangle$ where $|x_i| = |F| = m$ and suppose that y is of the form $\langle y_1,\ldots,y_{2k-1}\rangle$ where $|y_i| = m$. Then, N accepts iff $\pi_{2k-j}{\circ}h(\langle y_1,\ldots,y_{2k-j-1},F,x_j,\ldots,x_1\rangle,z) \in \mathrm{SAT}$. It is easy to see that if $a = f(0^{|F|})$ is of order j then by Lemma 8,

$$F \in \overline{\mathrm{SAT}} \Longrightarrow \exists y,\ \mathrm{Prob}_z(N \text{ accepts } F\#a\#y\#z) \geq \frac{\lfloor (j+1)/2 \rfloor}{k} \;+\; ((j+1) \bmod 2) * \frac{\varepsilon}{2}$$

and

$$F \in \mathrm{SAT} \Longrightarrow \forall y,\ \mathrm{Prob}_z(N \text{ accepts } F\#a\#y\#z) \leq \frac{\lfloor (j+1)/2 \rfloor}{k} \;-\; (j \bmod 2) * \frac{\varepsilon}{2}$$

This shows that there exists a *nonuniform* Merlin-Arthur-Merlin game[Bab85] for $\overline{\mathrm{SAT}}$. The existence of such games for $\overline{\mathrm{SAT}}$ implies that the PH collapses to Σ_3^{P} [BHZ87, Yap83]. □

4 Completeness under Randomized Reductions

Randomized reductions were introduced by Adleman and Manders [AM77] in order to show that certain number theoretic problems are intractable unless NP = RP. Subsequently, randomized reductions, such as the Valiant-Vazirani reduction [VV86] from SAT to USAT have been widely used in complexity theory.

The notion of a probability *threshold* was introduced in [CKR91]. It was argued that for many complexity classes, there is probability *threshold* above which languages complete under randomized reductions inherit most of the hardness properties associated with the $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete languages. Below the probability *threshold* much simpler languages could also be complete under randomized reductions.

For instance, it was shown that the *threshold* for co-$\mathrm{D^P}$ $\leq_m^{rp}$-complete languages is $1/2 + 1/poly$. For higher levels of the Boolean and Query Hierarchies, a range was provided for the threshold.

By exhibiting *tight* bounds on the probabilities achievable by randomized reductions between classes in the Boolean and Query Hierarchies, we have determined the *exact* values of the thresholds for these classes. For instance, the *threshold* probability for $\mathrm{BH}(2k)$ $\leq_m^{rp}$-complete languages is $1 - 1/k + 1/poly$. Any language L which is $\leq_m^{rp}$-complete for $\mathrm{BH}(2k)$ with a probability above this threshold inherits the following hardness properties (assuming PH infinite) usually associated with $\leq_m^{P}$-complete languages:

- L is not in a simpler complexity class such as $\mathrm{P}^{\mathrm{SAT}||[2k-1]}$
- L is not in the complementary class co-$\mathrm{BH}(2k)$.
- L is not closed under polynomial ORs (or even binary OR if $k > 1$). Note that usual probability amplification techniques require closure under ORs.

For every $k \geq 1$, we exhibit a language in $\mathrm{BH}(2k)$ which is $\leq_m^{rp}$-complete with probability above the threshold. Let USAT denote the set of uniquely satisfiable boolean formulas. Define the languages $\mathrm{L}_{\mathrm{BH}(2k-2)} \vee \mathrm{USAT}$ as

$$\mathrm{L}_{\mathrm{BH}(2k-2)} \vee \mathrm{USAT} \stackrel{\mathrm{def}}{=} \{ \langle x_1, \ldots, x_{2k-2}, y \rangle \mid \langle x_1, \ldots, x_{2k-2} \rangle \in \mathrm{L}_{\mathrm{BH}(2k-2)} \text{ or } y \in \mathrm{USAT} \}.$$

Lemma 9 For any $k \geq 1$, the language $\mathrm{L}_{\mathrm{BH}(2k-2)} \vee \mathrm{USAT}$ is $\leq_m^{rp}$-complete for $\mathrm{BH}(2k)$ via a probability $1 - 1/k + 1/(4kn)$ reduction.

Proof: $\mathrm{L}_{\mathrm{BH}(2k-2)} \vee \mathrm{USAT} \in \mathrm{BH}(2k)$ because $\mathrm{USAT} \in \mathrm{D^P}$. The $\leq_m^{rp}$-reduction from $\mathrm{L}_{\mathrm{BH}(2k)}$ to $\mathrm{L}_{\mathrm{BH}(2k-2)} \vee \mathrm{USAT}$ can be obtained by combining the $\leq_m^{rp}$-reduction used in Lemma 1 with the probability $1/4n$ Valiant-Vazirani[VV86] reduction from the $\mathrm{D^P}$ $\leq_m^{P}$-complete language to USAT. □

Thus the languages $\mathrm{L}_{\mathrm{BH}(2k-2)} \vee \mathrm{USAT}$, by virtue of being $\leq_m^{rp}$-complete with the requisite probability inherit the hardness properties listed above. Note that we can show this *without* having to prove that the languages are $\leq_m^{P}$-complete. In fact, we can prove the following Theorem that shows that non-relativizing techniques will have to be used to decide whether these languages are $\leq_m^{P}$-complete.

Theorem 10 There is a recursive oracle O such that relative to O, $\forall k$, the relativized language corresponding to $\mathrm{L}_{\mathrm{BH}(2k-2)} \vee \mathrm{USAT}$ is not complete for the relativized class corresponding to $\mathrm{BH}(2k)$ under relativized $\leq_m^{P}$-reductions.

The oracle O can be constructed by combining the technique used to construct an oracle which separates the Boolean Hierarchy [CGH+88] with the technique used to construct oracle worlds where USAT is not $\leq_m^{P}$-complete for $\mathrm{D^P}$[BG82].

5 Randomization and Bounded Query Functions

In view of the results presented earlier, a natural question to ask is whether similar results hold for *functions* computable using bounded queries to SAT, i.e., can randomization be used to reduced the number of queries required to compute these functions ? In this section we address this issue and provide strong evidence that this is not the case.

In order to make the above question more precise we will first have to define Bounded Query function classes and the notion of computing function probabilistically using bounded number of queries.

Definition [AG88][Bei87] If A is a set and $k \in \mathcal{N}$ then $\mathrm{PF}^{\mathrm{A}}_{k-T}$ is the class of functions that can be computed by a polynomial time oracle Turing machine that can make at most k queries to the oracle. If the machine is only allowed to query the oracle nonadaptively, i.e., list all queries before making any of them, then the corresponding class of functions is denoted by $\mathrm{PF}^{\mathrm{A}}_{k-tt}$

In order to allow for a randomized bounded query function computation we shall define the notion of a polynomial time randomized bounded query (PRBQ) machine and the functions computed by such machines.

Definition A PRBQ machine is a polynomial time oracle TM which has access to a source of randomness and can query its oracle A a bounded number of times. On an input x, the machine first obtains a uniformly chosen, polynomial sized random string r and then based on x and r it queries the oracle a bounded number of times. At the end of this computation it outputs a string $g(x,r)$. Since the output of the machine depends on the random string r, it may not be a function of x. However, if for every input x, the machine outputs $f(x)$ with probability $> 1/2$, then we say that the PRBQ machine computes f.

We say that a given PRBQ machine G computes a function f with probability δ ($\delta > 1/2$), if for all inputs x, G outputs $f(x)$ with probability at least δ.

Definition For a set A and $k \in \mathcal{N}$, we say that a function f is in $\mathrm{RPF}^{\mathrm{A}}_{k-T}[\delta]$, if there is a PRBQ machine with oracle A which computes f with probability δ ($\delta > 1/2$) and makes no more than k queries to the oracle. If, in addition, the PRBQ machine always queries the oracle nonadaptively then f is also in $\mathrm{RPF}^{\mathrm{A}}_{k-tt}[\delta]$.

First we show that a PRBQ machine can trivially compute any function in $\mathrm{PF}^{\mathrm{SAT}}_{k-T}$ ($\mathrm{PF}^{\mathrm{SAT}}_{k-tt}$) with probability $1/2 + 1/exp$ using only $k-1$ adaptive (nonadaptive) queries. Note that k need not be a constant, it could be any function of the input.

Lemma 11 Any function f in $\mathrm{PF}^{\mathrm{SAT}}_{k-T}$ ($\mathrm{PF}^{\mathrm{SAT}}_{k-tt}$) is also in $\mathrm{RPF}^{\mathrm{SAT}}_{(k-1)-T}[1/2 + 1/2^{p(n)}]$ (respectively $\mathrm{RPF}^{\mathrm{SAT}}_{(k-1)-tt}[1/2 + 1/2^{p(n)}]$) where n is the input size and p is a polynomial which depends on f.

Proof Outline: We can simulate the machine computing f and instead of making one of the k queries to SAT we can randomly guess the oracle response. This way we will compute f correctly with probability $\geq 1/2$. To achieve a correctness probability greater than $1/2$ by the required amount we can use the fact that if a query $F \in SAT$, then with inverse exponential probability we can randomly guess a satisfying assingment of F.

Theorem 12 Let g be an increasing integer valued function such that $g(n) = \mathrm{O}(\log n)$ and let p be any polynomial. Then there is a function in $\mathrm{PF}^{\mathrm{SAT}}_{g(n)-tt}$ which is not in $\mathrm{RPF}^{\mathrm{X}}_{(g(n)-1)-T}[1/2 + 1/p(n)]$ for any oracle X, unless PH collapses.

Proof: Before we prove this theorem we shall need the following definition:

Definition [AG88] If A is any set and $k \in \mathcal{N}$ then

$$\mathrm{F}^{\mathrm{A}}_{k}(x_1, \ldots, x_k) = \langle \mathrm{A}(x_1), \ldots, \mathrm{A}(x_k) \rangle,$$

where $\langle \ldots \rangle$ is the standard pairing function and $\mathrm{A}(x)$ is the characteristic function of A, i.e., $\mathrm{A}(x) = 1$ if $x \in \mathrm{A}$, 0 otherwise.

In the proof of Theorem 12 we shall be using the following version of a Theorem proved in [ABG90].

Lemma 13 [ABG90]: Let $h(n)$ be any polynomially bounded function of n. If there is a function w in PF/*poly* such that on input $t = \langle x_1, x_2, \ldots, x_{h(n)}\rangle$, $w(t)$ is a $h(n)$-bit string such that, $w(t) \neq F^A_{h(n)}(x_1, x_2, \ldots, x_{h(n)})$, then $A \in \mathrm{NP}/poly \cap \text{co-NP}/poly$.

We now proceed with the proof of Theorem 12 . Suppose, for some $g(n) = O(\log n)$, it is the case that any function computable in $\mathrm{PF}^{\mathrm{SAT}}_{g(n)-tt}$ is also computable in $\mathrm{RPF}^{\mathrm{X}}_{(g(n)-1)-T}[1/2 + 1/p(n)]$ with for some oracle X and polynomial p.

Let $h(m) = g(m^2)$. Clearly, for large m, $h(m) \ll m$. Thus for sufficiently large m, the size of a collection of $h(m)$ boolean formulae of size m each, i.e., $\langle x_1, \ldots, x_{h(m)}\rangle$, will always be less than m^2. Let $\mathrm{pad}(x_1, x_2, \ldots, x_{h(m)})$ denote the collection of formulas $\langle x_1, \ldots, x_{h(m)}\rangle$ of size m each, padded up to size m^2. Define a function $\mathcal{T}$, which on input of the form $\mathrm{pad}(x_1, x_2, \ldots, x_{h(m)})$ outputs $\mathrm{F}^{\mathrm{SAT}}_{h(m)}(x_1, x_2, \ldots, x_{h(m)})$. It is easy to see that for an n-sized input of the correct form $\mathcal{T}$ can be computed using only $h(\sqrt{n}) = g(n)$ queries to SAT. Thus $\mathcal{T} \in \mathrm{PF}^{\mathrm{SAT}}_{g(n)-tt}$ and therefore, by assumption in $\mathrm{RPF}^{\mathrm{X}}_{(g(n)-1)-T}[1/2 + 1/p(n)]$ for some oracle X and polynomial p. Let M be the PRBQ machine computing $\mathcal{T}$ with probability $1/2 + 1/p(n)$, using only $g(n) - 1$ queries to X.

On input $I = \mathrm{pad}(x_1, x_2, \ldots, x_{h(m)})$ of size n, M obtains a random string r and it outputs $\mathcal{T}(I)$ with probability $1/2+1/p(n)$ and makes no more than $h(m)-1$ queries to X. Since $h(m) = O(\log n)$, it is possible to completely explore the entire query/computational tree of M on I and random string r. Since there are $h(m) - 1$ oracle queries in this tree, it has $2^{h(m)-1}$ leaves and the value output by M is one of the $2^{h(m)-1}$ possible values which are computed at these leaves.

Let $S_I(r)$denote the set of values output at the leaves of M's computation tree on input I and random string r. Clearly,

$$\forall r,\ |S_I(r)| \leq 2^{h(m)-1}$$

and

$$\mathrm{Prob}_r[\mathcal{T}(I) \in S_I(r)] \geq 1/2 + 1/p(n).$$

Choose $c(n) = 2 * [(p(n))^2 * n^2] + 1$ strings $r_1, \ldots, r_{c(n)}$ independently at random. By standard probability amplification techniques [Sch87], it can be shown that $\mathcal{T}(I)$ would belong to a majority of the sets $S_I(r_1), \ldots, S_I(r_{c(n)})$ with probability more than $1 - 1/2^{n^2}$. Thus, for every possible input J of size n, when strings $r_1, \ldots, r_{c(n)}$ are chosen independently at random, the probability that $\mathcal{T}(J)$ belongs to a majority of the sets $S_J(r_1), \ldots, S_J(r_{c(n)})$ is more than $1 - 1/2^{n^2}$. But there are at most 2^n inputs J of size n. Thus there is a sequence of random strings $r_1, \ldots, r_{c(n)}$, such that for all J of size n, $\mathcal{T}(J)$ belongs to a majority of the sets $S_J(r_1), \ldots, S_J(r_{c(n)})$. Let this sequence of random strings be encoded into the advice function $\mathcal{A}$, which outputs $r_1, \ldots, r_{c(n)}$ on input 0^n.

On input $J = \mathrm{pad}(x_1, x_2, \ldots, x_{h(m)})$ of size n and advice $\mathcal{A}(0^n) = r_1, \ldots, r_{c(n)}$ a polynomial time machine can easily compute the set MAJ of all $h(m)$-bit strings which are in a majority of the sets $S_J(r_1), \ldots, S_J(r_{c(n)})$. One of the strings in MAJ is $\mathcal{T}(J) = \mathrm{F}^{\mathrm{SAT}}_{h(m)}(x_1, x_2, \ldots, x_{h(m)})$.

Notice that MAJ cannot contain all possible $h(m)$-bit strings. If it did then then it must be the case that

$$|\bigcup_{i=1}^{c(n)} S_J(r_i)| \geq (c(n) + 1) * 2^{h(m)-1}$$

but we know that for each $S_J(r_i)$, $|S_J(r_i)| \leq 2^{h(m)-1}$ and therefore

$$|\bigcup_{i=1}^{c(n)} S_J(r_i)| \leq c(n) * 2^{h(m)-1}$$

Thus there exists at least one $h(m)$-bit string which is not in MAJ and the lexicographically least such string L can be output in polynomial time once MAJ has been computed. Clearly $L \neq \mathcal{T}(J)$.

Thus we have satisfied the hypothesis of Lemma 13. That is, there is a function w in PF/*poly* such that given a collection $c = \langle x_1, x_2, \ldots, x_{h(m)}\rangle$ of $h(m)$ ($h(m) = \mathrm{O}(\log m)$) formulae, w outputs a $h(m)$-bit string $w(c)$ such that, $w(c) \neq \mathrm{F}^{SAT}_{h(m)}(x_1, x_2, \ldots, x_{h(m)})$. Therefore, from the lemma it follows that SAT $\in$ NP/*poly* $\cap$ co-NP/*poly* and PH collapses to Σ_3^{P} [Yap83].

Acknowledgements

I am grateful to my advisor Juris Hartmanis for his guidance and support. This paper has benefited greatly from discussions with Richard Chang, Suresh Chari and Desh Ranjan. Special thanks to Richard for sharing key ideas leading to the first proof of Lemma 3 and Sanjay Gupta for suggesting Lemma 9.

References

[ABG90] A. Amir, R. Beigel, and W.I. Gasarch. Some connections between bounded query classes and non-uniform complexity. In *Proceedings of the 5th Structure in Complexity Theory Conference*, pages 232–243, July 1990.

[AG88] Amihood Amir and William I. Gasarch. Polynomial terse sets. *Information and Computation*, 77:37–56, April 1988.

[AM77] L. M. Adleman and K. Manders. Reducibility, randomness, and intractibility [*sic*]. In *ACM Symposium on Theory of Computing*, pages 151–163, 1977.

[Bab85] L. Babai. Trading group theory for randomness. In *ACM Symposium on Theory of Computing*, pages 421–429, 1985.

[Bei] Richard Beigel. Bi-immunity results for cheatable sets. *Theoretical Computer Science.* To appear.

[Bei87] R. Beigel. Bounded queries to SAT and the Boolean hierarchy. Technical Report 7, Department of Computer Science, The Johns Hopkins University, 1987. To appear in *Theoretical Computer Science.*

[Bei88] R. Beigel. NP-hard sets are p-superterse unless R = NP. Technical Report 4, Department of Computer Science, The Johns Hopkins University, 1988.

[BG82] A. Blass and Y. Gurevich. On the unique satisfiability problem. *Information and Control*, 55(1–3):80–88, 1982.

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

[CGH+88] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The Boolean hierarchy I: Structural properties. *SIAM Journal on Computing*, 17(6):1232–1252, December 1988.

[Cha89] R. Chang. On the structure of bounded queries to arbitrary NP sets. In *Proceedings of the 4th Structure in Complexity Theory Conference*, pages 250–258, June 1989.

[CKR91] R. Chang, J. Kadin, and P. Rohatgi. Connections between the complexity of unique satisfiability and the threshold behavior of randomized reductions. In *Proceedings of the 6th Structure in Complexity Theory Conference*, pages 255–269, July 1991.

[GJY87] Judy Goldsmith, Deborah Joseph, and Paul Young. A note on bi-immunity and p-closeness of p-cheatable sets in p/poly. Technical Report 87-11-05, Department of Computer Science, University of Washington, Seattle, November 1987. To appear in JCSS.

[Kad88] J. Kadin. The polynomial time hierarchy collapses if the Boolean hierarchy collapses. *SIAM Journal on Computing*, 17(6):1263–1282, December 1988.

[Kre88] Mark W. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3):490–509, 1988.

[PY82] C. H. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). In *ACM Symposium on Theory of Computing*, pages 255–260, 1982.

[Sch87] U. Schöning. Probabilistic complexity classes and lowness. In *Proceedings of the 2nd Structure in Complexity Theory Conference*, pages 2–8, 1987.

[VV86] L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47(1):85–93, 1986.

[Wag86] Klaus W. Wagner. More complicated questions about maxima and minima and some closures of NP. In *Proceedings of the 13th International Colloquium on Automata, Languages, and Programming*, pages 434–443, 1986. Volume 226 of *Lecture Notes in Computer Science*.

[Yap83] C. Yap. Some consequences of non-uniform conditions on uniform classes. *Theoretical Computer Science*, 26(3):287–300, 1983.