On Properties of Random Reductions

Pankaj Rohatgi
Ph.D Thesis

93-1386
September 1993

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

ON PROPERTIES OF RANDOM REDUCTIONS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Pankaj Rohatgi

January 1994

ON PROPERTIES OF RANDOM REDUCTIONS

Pankaj Rohatgi, Ph.D.

Cornell University 1994

Randomness is widely accepted as a powerful computational resource because the most elegant and efficient solutions to several computational problems are randomized. A recurrent theme in the theory of randomized computation is the notion of a random reduction. Random reductions are similar to many-one ($\leq^{\mathrm{P}}_{\mathrm{m}}$) reductions except for the fact that they are carried out by probablistic transducers which may make errors with small probability. Such reductions are used explicitly in many basic results in complexity theory and implicitly in several randomized algorithms.

This thesis investigates the properties of random reductions as a tool towards understanding the power and limitations of randomness. We first prove some startling results which indicate that random reductions can be quite successful in reducing harder problems to simpler ones. We then propose the thesis that in many situations there is a sharp *probability threshold* which governs just how successful random reductions can be in this regard. As evidence, we prove that for several complexity classes $\mathcal{C}$, under standard assumptions, there exist corresponding *probability thresholds* $\mathcal{C}_T$, such that random reductions with success probability below $\mathcal{C}_T$ can reduce the hardest languages in $\mathcal{C}$ to simpler ones but reductions with

success probability above C_T cannot do so. Based on these thresholds, we then propose a meaningful definition of completeness under random reductions which resolves several anomalies caused by the traditional definitions which did not place much emphasis on the success probability.

The results described above depend on standard but unproven complexity-theoretic assumptions. In order to show that such behavior is inherent to random reductions and not an artifact introduced by these assumptions, we also prove that it is present in very high complexity classes *without any assumptions.* In this thesis we also examine other basic aspects of random reducibility. We prove several *absolute* separation results between the notions of completeness under various polynomial-time random reductions with different success probabilities and between various random reductions and deterministic polynomial-time reductions. In addition, we also prove new results on the consequences of having random reductions from NP-complete sets to sparse sets.

BIOGRAPHICAL SKETCH

Pankaj Rohatgi was born on July 5, 1967 in the older section of Delhi which is a few miles north of the capital city of New Delhi. He attended St. Xavier's School in Delhi for several years before moving to Kuala Lumpur, Malaysia where he obtained his GCE 'O' Levels from the University of London. He then returned to Delhi to complete his schooling and in 1984 he joined the Indian Institute of Technology at New Delhi where he received his Bachelor's degree in Computer Science and Engineering in May 1988. He then joined the graduate school at Cornell University and obtained his Master's degree in computer science in May 1991 and a Ph.D. in September 1993.

To my parents

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to Prof. Juris Hartmanis for introducing me to research and guiding me through this thesis. It would have been impossible for me to complete this work without his constant encouragement and support. An outstanding researcher, teacher and a very warm human being, he has been both a source of inspiration and a guide on academic and non-academic matters.

I thank Prof. Gerard Salton and Prof. Anil Nerode for serving on my special committee and providing many helpful suggestions. I thank Prof. Dexter Kozen and Prof. S. N. Maheshwari for introducing me to the mathematical beauty of Theoretical Computer Science.

It has been a pleasure working with my coauthors Jim Kadin, Richard Chang, Desh Ranjan, Suresh Chari and Aravind Srinivasan and their contributions are gratefully acknowledged. Special thanks to Richard for introducing me to the the intricacy of the Boolean and Query Hierarchies and collaborating with me on the central theme of this thesis. I have also benefited greatly from discussions with Bill Gasarch, Radhakrishnan Jagadeesan, Dexter Kozen, Steve Mitchell, David Shmoys and Vijay Vazirani.

I would like to thank all my friends who helped me endure five long winters in

Ithaca. Special thanks to Capt. Sofoklis Efremidis who has been my officemate for the past five years.

I gratefully acknowledge the generous financial support provided by the National Science Foundation (Research Grants #CCR-8823053 and #CCR-9123730), the IBM Graduate Fellowship and the Mathematical Sciences Institute of Cornell University.

Last but not the least, I thank my family for their support.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

The aim of complexity theory is to characterize the inherent difficulty of problems in terms of the computational resources required for their solution. Thus, identifying important computational resources and exploring relationships between them is a fundamental area of research in this field. In the early days, research in complexity theory focused on traditional computational resources such as Time, Space and Nondeterminism. More recently, however, research on non-traditional resources such as Randomness, Queries to oracles and Interaction with all powerful provers, has turned out to be very fruitful. For instance, very recently, the study of the interaction between two non-traditional resources: Randomness and Interaction with a prover, led to several breakthrough results which imply that even obtaining good approximations to many NP-Optimization problems such as the Travelling Salesman Problem on a plane, the Maximum Clique and the Chromatic Number of a Graph, is as hard as solving these problems exactly [FGL+91,AS92,ALM+92, LY93].

A key resource that has received much attention lately is access to a source of randomness. It has proved to be a valuable computational resource because some of the most elegant and efficient solutions to several computational problems are randomized. In several cases, e.g., solving the connectivity problem in undirected graphs in logarithmic space (LOGSPACE) [AKL+79], testing primality in polynomial time (P) [Rab80] or finding a perfect matching in a graph in parallel using polylogarithmic time and polynomial number of processors (NC) [KUW86, MVV87,CRS93], the use of randomness is the only known tool which allows these problems to be solved in the given resource bounds. Sometimes, as is the case with interactive protocols [Bab85,GMR89] between a polynomial time machine (a verifier) and an all powerful prover, the introduction of randomness results in a dramatic increase in computational power. If the verifier is deterministic then these protocols capture the class NP (Nondeterministic Polynomial time) but if the verifier is allowed access to randomness then these protocols capture the much larger complexity class PSPACE (Polynomial Space) [Sha90].

Due to the importance of randomness as a computational resource, a lot of research has focussed on characterizing the the power and limitations of randomness by studying its interaction with other computational resources. The interaction between Randomness and traditional resources such as Nondeterminism, Time and Space has been well studied. Since the existence of true sources of randomness is still the object of an ongoing philosophical debate and sources which are believed to random are quite slow, there has also been extensive research on optimizing the amount of randomness required to solve problems efficiently and several techniques such as the use of limited independence between random variables and the use of smaller sample spaces which approximate the behavior of larger distributions have been developed.

A recurrent theme which appears explicitly and implicitly in the literature dealing with randomized computations is the notion of a random reduction. A random reduction is quite similar to the classical polynomial time many-one ($\leq_m^P$) reduction except for the fact that it is carried out by a probablistic (rather than deterministic) transducer which may make mistakes with very small probability. These type of reductions were introduced by Adleman and Manders [AM77] to resolve the complexity of several number theoretic problems. For a long time, it was suspected that these number theoretic problems were NP-complete but no one was able to come up with the required deterministic polynomial time reduction from SAT (the canonical NP-complete Boolean Satisfiability problem) or provide any other evidence that these problems were intractable. Although, Adleman and Manders did not succeed in obtaining a deterministic reduction, they were able to reduce SAT to these problems using a *random* reduction. This immediately established that these number-theoretic problems cannot have efficient solutions unless all problems in NP also have efficient randomized solutions. A similar approach was also successfully applied by Valiant and Vazirani [VV86] to settle the complexity of the Unique Satisfiability problem (USAT), i e., the problem of deciding whether a given boolean formula has exactly one satisfying assignment. They succeeded in obtaining a random reduction from SAT to USAT, even though Blass and Gurevich [BG82] had earlier proved that most techniques will fail to produce a $\leq_m^P$ reduction. Subsequently, random reductions such as the Valiant-Vazirani reduction have also turned out to be very useful in resolving some basic issues in complexity theory such as the complexity of counting problems based on the number of accepting paths of NP machines [Tod89,TO91,BF90].

Apart from these explicit uses in complexity theory, random reductions have also been used implicitly in the design of several randomized algorithms. As an example, consider the problem of finding a maximal matching in a graph. Although polynomial time sequential algorithms were developed for this problem decades ago, it is still open whether this problem can be solved in NC, i.e., in parallel using polylogarithmic time and polynomial number of processors. However, if randomness is allowed then this can be done [KUW86,MVV87,CRS93]. Although not explicitly stated in the literature, these algorithms actually work by randomly reducing the given instance of the matching problem to an instance of a matrix inversion problem for which efficient NC algorithms are known.

In this thesis we provide fresh insights into the the power and limitations of randomness by investigating the properties of random reductions. Intuitively, if a problem A *deterministically* many-one ($\leq_{\mathrm{m}}^{\mathrm{P}}$) reduces to a problem B in polynomial time then we know that A is not much harder than B. Thus, it is not possible for $\leq_{\mathrm{m}}^{\mathrm{P}}$ reductions to reduce a hard problem to a substantially simpler one. However, in Chapter 3 (the technical backbone of this thesis) we will prove several surprising results which show that random reductions can succeed in reducing harder problems to simpler ones with good probability of success. For instance, we will show that by using randomness with good probability one can recognize languages in bounded query classes such as $\mathrm{P}^{\mathrm{SAT}\|[k]}$ (languages recognizable in polynomial time using k *nonadaptive* queries to a SAT oracle) using fewer than k *nonadaptive* queries. This is in sharp contrast to the results of Kadin [Kad88] and Chang and Kadin [CK90a] which state that under a reasonable hypothesis, one cannot save even a single query deterministically. We argue that these surprising results, instead of being anomalies, in fact demonstrate the power of randomness. In Chapter 3, we also show under reasonable hypothesis that certain problems do not randomly reduce to simpler ones beyond a certain success probability and this, we argue, demonstrates the limitations of randomness.

Based on the technical results presented in Chapter 3, in Chapter 4 we propose the thesis that, in several situations, the power and limitations of randomness is precisely characterized by a *probability threshold* for random reductions. As evidence, we will show that for several complexity classes, under standard assumptions, there exist corresponding *probability thresholds* such that random reductions with success probability below the *threshold* can reduce the hardest languages in these classes to simpler ones but random reductions with success probability above the *threshold* cannot do so. We believe that the *threshold* is a fundamental attribute of every complexity class which precisely characterizes the tradeoff involved in the use of randomness to reduce the complexity of problems within it, i.e., it quantifies the minimal amount of error that must be introduced in order to reduce the complexity of the hardest problems in the class using randomness. Our results on *probability thresholds* also highlight for the first time the crucial role played by the success probability of random reductions. Traditionally, random reductions were defined without giving much consideration to the success probability of the reduction. Our results immediately imply that any meaningful definition of completeness under random reductions must take into account the success probability: languages complete under random reductions with success probability above the threshold behave almost like the hardest languages in the class whereas languages complete under lower probability reductions could be much simpler.

The notion of probability thresholds developed in this thesis also fits well with some earlier results of Graham and Yao [GY89] on probabilistic protocols for the Byzantine Generals Agreement Problem. It is well known that no deterministic protocol can solve the n-processor Byzantine Generals Agreement Problem if the number of faulty processors t is $n/3$ or more. However, randomized protocols can succeed in solving this problem with good probability of success. Graham and Yao were the first to prove *tight* bounds on the success probability of some of these randomized protocols in terms of n and t and thus were able to exactly quantify the tradeoff involved in the use of randomness to solve this class of problems.

The results presented in Chapters 3 and 4 are based on the assumption that the Polynomial Time Hierarchy (PH) does not collapse to its third level. Although, this is a standard assumption in complexity theory and in fact, it is widely believed that the PH does not collapse at all, still, the validity of our conclusions in Chapters 3 and 4 can be questioned on this account. So, in Chapter 5, we prove that random reductions exhibit similar behavior in very high complexity classes *without any assumptions.* This provides strong evidence that such behavior is inherent to random reductions and not a consequence of some quirk in our assumptions.

Apart from threshold behavior, this thesis also examines several basic aspects of random reducibility. In Chapter 5, we also prove several *absolute* separation results between the notions of completeness under various polynomial-time random reductions with different success probabilities and between various random reductions and deterministic polynomial-time reductions. This work can be seen as a natural extension of the seminal work of Ladner, Lynch and Selman [LLS75] and Watanabe [Wat87] who provided the first *absolute* separation results for various deterministic reductions. In Chapter 6 we prove some new results on the consequences of having random reductions from NP to sparse sets which builds upon earlier work by Mahaney [Mah82], Ogiwara and Watanabe [OW90] and others on deterministic reductions from NP to sparse sets and improves upon prior work of Mahaney and Simon [MS81] on random reductions to sparse sets.

Chapter 2

Preliminaries

In this chapter we will go over the definitions of various types of deterministic and random reductions that will be used in this thesis. We will also use this as an opportunity to fix some notation. We assume that the reader is already familiar with classes such as polynomial time (P), polynomial space (PSPACE), nondeterministic polynomial time (NP) and its complementary class co-NP, the NP-complete set SAT and the Polynomial time Hierarchy (PH). If not, then any of the standard texts on Theory of Computing or Complexity Theory would serve as an excellent reference [HU79,LP81,BDG90]. In this thesis we will also deal with other complexity classes in the Boolean, Bounded Query and Weak Exponential Hierarchies but we will defer the definitions of these classes to later chapters where they are used.

We also assume that the reader is familiar with the Turing machine model and in particular with the notion of oracle Turing machines. We shall distinguish between two types of access mechanisms to the oracle, the usual *adaptive* mechanism and the more restrictive *nonadaptive* mechanism. In the usual *adaptive* mechanism, the Turing machine makes queries to its oracle sequentially and each query can depend on the answers provided by the oracle to prior queries. In the case of the *nonadaptive* access mechanism, the Turing machine is forced to make all its queries to the oracle at the same time. That is, the machine computes independently for some time, then it asks a vector of queries to the oracle and upon receiving the corresponding vector of responses its access to the oracle is terminated. Thus, the queries made by the machine are independent of the response of the oracle and hence the use of the term *nonadaptive* to describe this access mechanism.

Notation We will denote the output of an oracle Turing machine M on input x and *adaptive* access to oracle O as $M^O(x)$.

Notation We will denote the output of an oracle Turing machine M on input x and *nonadaptive* access to oracle O as $M^{O||}(x)$.

Sometimes we will be dealing with Turing machines with further restrictions on the oracle access to their oracle. A common restriction is to limit the number of queries allowed to the oracle and in this thesis we will be dealing with several instances where the number of *adaptive* or *nonadaptive* queries allowed to the oracle are bounded by some constant k. In such cases we will denote the output of an oracle Turing machine M on input x with a bound k on the number of *adaptive* or *nonadaptive* queries to its oracle O as $M^{O[k]}(x)$ and $M^{O||[k]}(x)$ respectively.

Since this thesis deals with the properties of random reductions we now go over the definitions of various deterministic and random polynomial-time reductions.

2.1 Reductions Defined

One of the most basic concepts in recursion theory as well as in complexity theory is the notion of reductions between languages. First let us review various deterministic polynomial time reductions. The classical polynomial time deterministic reductions are the Many-one ($\leq_{\mathrm{m}}^{\mathrm{P}}$), Truth-Table ($\leq_{tt}^{p}$) and Turing ($\leq_{\mathrm{T}}^{\mathrm{P}}$) reductions which are defined as follows:

Definition We say that $A \leq_{\mathrm{m}}^{\mathrm{P}} B$, if there exists a polynomial time computable function f such that

$$x \in A \iff f(x) \in B.$$

Definition We say that $A \leq_{tt}^{p} B$, if there exists a polynomial time machine M such that

$$x \in A \iff M^{B\|}(x) \text{ accepts.}$$

Definition We say that $A \leq_{\mathrm{T}}^{\mathrm{P}} B$, if there exists a polynomial time machine M such that

$$x \in A \iff M^{B}(x) \text{ accepts.}$$

It is obvious from these definitions that if $A \leq_{\mathrm{m}}^{\mathrm{P}} B$ then that implies that $A \leq_{tt}^{p} B$ which in turn implies that $A \leq_{\mathrm{T}}^{\mathrm{P}} B$. In seminal work, Ladner Lynch and Selman [LLS75] showed that, in E (exponential time), $\leq_{\mathrm{m}}^{\mathrm{P}}$-reductions are strictly stronger than $\leq_{tt}^{p}$-reductions which are in turn strictly stronger than $\leq_{\mathrm{T}}^{\mathrm{P}}$-reductions, i.e., there are languages A, B, C and D such that $A \leq_{tt}^{p} B$ but $A \not\leq_{\mathrm{m}}^{\mathrm{P}} B$ and $C \leq_{\mathrm{T}}^{\mathrm{P}} D$ but $C \not\leq_{tt}^{p} D$. Apart from separating these basic reducibilities, Ladner, Lynch and Selman also defined and separated several specialized instances of $\leq_{tt}^{p}$reductions such as the the k-truth table ($\leq_{k-tt}^{p}$), conjunctive ($\leq_{\mathrm{conj}}^{\mathrm{P}}$) and disjunctive ($\leq_{\mathrm{disj}}^{\mathrm{P}}$) reductions. Since we will also be dealing with these specialized $\leq_{tt}^{p}$ reductions we shall define them here.

A k-truth table reduction (for any constant k) is a special case of a ($\leq_{tt}^{p}$) reduction in which the machine carrying out the $\leq_{tt}^{p}$ reduction makes no more than k *nonadaptive* queries to its oracle. More formally:

Definition We say that $A \leq_{k-tt}^{p} B$, if there exists a polynomial time machine M such that

$$x \in A \iff M^{B\|[k]}(x) \text{ accepts.}$$

We now define the conjunctive and disjunctive reductions. These can be viewed as specialized $\leq_{tt}^{p}$ reductions in which the polynomial time oracle machine carrying out the reduction accepts if and only if all (or at least one) of its oracle queries are answered as yes respectively.

Definition We say that $A \leq^{\mathrm{p}}_{\mathrm{conj}} B$, if there exists a polynomial time machine M which on input x outputs a non-empty set M_x such that

$$x \in A \;\Rightarrow\; M_x \cap \overline{B} = \phi,$$

$$x \notin A \;\Rightarrow\; M_x \cap \overline{B} \neq \phi.$$

Definition We say that $A \leq^{\mathrm{p}}_{\mathrm{disj}} B$, if there exists a polynomial time machine M which on input x outputs a non-empty set M_x such that

$$x \in A \;\Rightarrow\; M_x \cap B \neq \phi,$$

$$x \notin A \;\Rightarrow\; M_x \cap B = \phi.$$

We now define various types of random reductions between sets. These definitions are based on the definitions given in [AM77,VV86] and can be viewed as generalizations of the notion of a $\leq^{\mathrm{p}}_{\mathrm{m}}$-reduction. Depending on the type of error allowed, three types of random reductions (the $\leq^{\mathrm{rp}}_{\mathrm{m}}$, $\leq^{\mathrm{co-rp}}_{\mathrm{m}}$ and $\leq^{\mathrm{bpp}}_{\mathrm{m}}$ reductions) have been defined in the literature. The $\leq^{\mathrm{rp}}_{\mathrm{m}}$ and $\leq^{\mathrm{co-rp}}_{\mathrm{m}}$ random reductions are one-sided error reductions whereas the $\leq^{\mathrm{bpp}}_{\mathrm{m}}$-reduction is a two-sided error reduction.

Definition We say that $A \leq^{\mathrm{rp}}_{\mathrm{m}} B$ with probability δ, if there exists a polynomial time function f such that

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

$$x \notin A \Rightarrow \mathrm{Prob}_z[\ f(x,z) \notin B\ ] = 1,$$

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

Definition We say that $A \leq^{\mathrm{co-rp}}_{\mathrm{m}} B$ with probability δ, if there exists a polynomial time function f such that

$$x \in A \Rightarrow \mathrm{Prob}_z[\ f(x,z) \in B\ ] = 1,$$

$$x \notin A \Rightarrow \mathrm{Prob}_z[\ f(x,z) \notin B\ ] \geq \delta,$$

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

Thus, a $\leq_{\mathrm{m}}^{\mathrm{co-rp}}$-reduction is quite similar to a $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction except for the fact that it errs on the opposite side. Next we define a $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reduction in which the error may occur on either side.

Definition We say that $A \leq_{\mathrm{m}}^{\mathrm{bpp}} B$ with probability δ, if there exists a polynomial time function f such that

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

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

With these definitions in place we can now proceed to the rest of this thesis with a good understanding of various types of deterministic and random reductions. However, before we proceed let us also fix some additional notation that we will use throughout the thesis.

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. Similarly, let $A^{\leq n}$ and $A^{>n}$ denote the subsets of strings in A of length less than or equal to n and greater than n respectively.

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,

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

Chapter 3

Saving Queries with Randomness[1]

Randomness is a useful computational resource due to its ability to enhance the capabilities of other resources. Very often, the best known randomized algorithms are faster than the best known deterministic solutions. In several cases, e.g., solving the connectivity problem in undirected graphs in LOGSPACE [AKL+79] or finding a perfect matching in a graph in NC [KUW86,MVV87,CRS93], the use of randomness is the only known tool for solving problems in the given resource bounds. In some cases, the introduction of randomness results in a dramatic increase in computational power. For instance, interactive protocols between a polynomial time deterministic machine (a verifier) and an all powerful prover are as powerful as NP whereas protocols in which the verifier has access to randomness are as powerful as PSPACE [Sha90].

The interaction between randomness and resources such as time, space and interaction with provers has been studied extensively. In this chapter we will investigate how randomness interacts with another well studied computational resource - the number of queries allowed to an NP-complete oracle. The results

[1]The results presented in this chapter have been drawn from [CKR91] and [Roh92]

proved in this chapter will form the technical backbone for the thesis proposed in Chapter 4.

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. For instance, solving the Traveling Salesman Problem (TSP), i.e., finding the length of the shortest tour in a given weighted graph, appears to be much harder than finding the size of the largest clique in a graph. However, the decision versions of these problems, i.e., deciding whether there is a tour of a given length in a weighted graph or deciding whether there is a clique of a certain size in a graph, are all NP-complete and have the same complexity. Thus, the theory of NP-completeness does not adequately explain the complexity of optimization problems. Krentel resolved this issue by characterizing the complexity of optimization problems by the number of SAT queries required by a polynomial time machine to find the optimal solution. For instance, solving TSP requires $\mathrm{O}(n)$ queries whereas computing the size of the largest clique requires only $\mathrm{O}(\log n)$ queries. Krentel also proved that problems requiring $\mathrm{O}(n)$ queries cannot be solved using only $\mathrm{O}(\log n)$ queries unless P = NP, thus establishing the validity of the number of queries as a complexity measure.

Often, one is only interested in finding out whether the optimal solution satisfies some predicate. For instance, Papadimitriou and Yannakakis [PY84] linked the complexity of the facets of the TSP polytope to the class $\mathrm{D^P}$ which consists of languages expressible as a difference of two NP languages. A canonical example of a language in $\mathrm{D^P}$ is $\{ \langle I, k \rangle \mid P(I,k) \}$ where I is an instance of an NP Optimization problem and $P(I,k)$ is the *simple* predicate $\mathit{Optimum}(I) = k$. As an example, the language CLIQUE(1) consisting of all tuples $\langle G, k \rangle$ such that k is the size of the largest clique in the graph G is $\mathrm{D^P}$-complete. One can easily construct language based on more complex predicates, e.g., we could define CLIQUE(5) to be the set of all 6-tuples $\langle\ G, a_1, \ldots, a_5\ \rangle$ where the size of the largest clique in the graph G is one of the five integers $a_1, \ldots, a_5$. Intuitively, both CLIQUE(1) and CLIQUE(5) appear to be more complex than the typical languages in NP or co-NP and in fact one can easily show that this is indeed the case unless NP = co-NP. Also, CLIQUE(5) appears to be more complex than CLIQUE(1) since it is based on a more complex predicate. Furthermore, since both CLIQUE(5) and CLIQUE(1) are based on simple predicates, one would also suspect that their complexity should be less than the complexity of first computing the maximum clique size and then checking the predicate.

A systematic study of such predicate based languages began with the work of Cai, Hemachandra [CH86] and Wagner [Wag86]. Wagner extended Krentel's approach to these languages by characterizing their complexity by the number of nonadaptive SAT-queries required to evaluate the predicate. Thus, a language which requires k nonadaptive queries to SAT i.e., a language which $\leq^p_{k\text{-}tt}$-reduces to SAT is placed in the complexity class $\mathrm{P}^{\mathrm{SAT}||[k]}$ which is the $k'th$ level of the Bounded Query Hierarchy. Thus, the complexity of CLIQUE(5) and CLIQUE(1) can be characterized by the fact that the lowest Bounded Query classes in which these languages can be placed are $\mathrm{P}^{\mathrm{SAT}||[10]}$ and $\mathrm{P}^{\mathrm{SAT}||[2]}$ respectively. 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). This can be viewed as a generalization of the definition of D^{P} (which consists of languages expressible as nested difference of 2 NP languages) and it gives rise to the closely related Boolean Hierarchy [CGH$^+$88] [Bei91].

Subsequently, the robustness of these two complexity measures was investigated by many researchers [Kre88,Kad88,Bei91,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,Bei91,Cha89,ABG90]. Of particular significance to this work 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 $\mathrm{P}^{\mathrm{SAT}||[9]}$, unless PH collapses.

In light of these separation results, it is natural to ask whether randomization can be used to bridge these gaps and if so, to what extent. This this chapter we resolve this question by providing tight bounds on the ability of randomness to save a nonadaptive query to SAT. We do this by providing tight bounds on the success probability of random reductions between successive levels of the Query and Boolean Hierarchies. For instance, we will show that any language in $\mathrm{P}^{\mathrm{SAT}||[k]}$ $\leq_{\mathrm{m}}^{\mathrm{rp}}$-randomly to a language in $\mathrm{P}^{\mathrm{SAT}||[k-1]}$ with probability of $1-1/\lceil k/2 \rceil$. If two-sided error ($\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reduction) is allowed then we show that the success probability can be increased to $1-1/(k+1)$. We will prove that this is the best that randomization can achieve by proving that it is not possible to increase these probabilities even by $1/poly$, unless the Polynomial time hierarchy (PH) collapses. Notice that as k increases, the error probability *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 $\forall k$, $\mathrm{P}^{\mathrm{SAT}||[k]} \neq \mathrm{P}^{\mathrm{SAT}||[k-1]}$, unless PH collapses [Kad88]. Our proof technique is based on the rich structure within the Boolean and Query Hierarchies and relies on the important observation that it is possible to convert arbitrary random reductions between classes in the Boolean Hierarchy into reductions which have very nice structural properties.

Thus, CLIQUE(5) can be recognized correctly with probability 10/11 using randomness and only 9 nonadaptive queries to SAT. Moreover the randomized computation of CLIQUE(5) in $\mathrm{P}^{\mathrm{SAT}||[9]}$ is simple and elegant and it is surprising that more complex computations can't decrease the error significantly, unless PH collapses.

The results described above hold for *languages* computable using bounded queries to SAT. For the sake of completeness, we also examine the ability of randomness to save a query in bounded query *function* computations and prove strong negative results about the ability of randomness to do so.

The rest of the chapter is organized as follows. In Section 3.1, we provide the necessary background and state the important properties of the Boolean and Query Hierarchies. In Section 3.2 we state our main results. In Section 3.3 we will show how randomization can be used to reduce harder languages in the Boolean and Query Hierarchies to simpler ones. In Section 3.4 we will prove that our results in Section 3.3 are optimal. First, in Section 3.4.1 we will illustrate the main ideas behind the optimality results by means of an example; the complete proof for a general case will appear in Section 3.4.2 and in Section 3.4.3 we state the results for the other cases without proof. In Section 3.5, we study the (non)ability of randomness to save a query in bounded query function computations and in Section 3.6 we list some open problems.

3.1 Background

First we formally 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.

From the above definition it is easy to see that the k'th level of the Bounded Query Hierarchy is actually the class of all languages which $\leq^p_{k-tt}$-reduce to SAT. In this chapter, we will deal with the Query Hierarchy based on *nonadaptive* queries and not with the Query Hierarchy based on *adaptive* queries because the *nonadaptive* Query Hierarchy is just a finer version of the *adaptive* Query Hierarchy. In fact, a beautiful result due to Beigel [Bei91] shows that the k'th level of the *adaptive* query hierarchy, i.e., $\mathrm{P}^{\mathrm{SAT}[k]}$ is exactly $\mathrm{P}^{\mathrm{SAT}||[2^k-1]}$.

Results about Bounded Query classes often rely on the structural properties of the closely related Boolean Hierarchy [CGH$^+$88] which is a natural generalization of the class $\mathrm{D^P}$. Most uninitiated readers, when first confronted with the definitions of the the Boolean Hierarchy find them to be quite nonintuitive and arbitrary. However, behind these nonintuitive definitions lies a very rich theory which often gets overlooked when one uses properties of a specific definition of the Boolean Hierarchy to prove results. Thus, before we provide a specific definition of this hierarchy which will be useful in our proofs we would first like to give the general picture of what the Boolean Hierarchy is all about.

Consider the complexity class $\mathrm{P}^{\mathrm{SAT}||[k]}$ consisting of languages recognized by polynomial time machines which make at most k nonadaptive queries to SAT. In general, when a machine has made its k queries to the oracle, then the future behavior of the machine (acceptance or rejection) can be completely characterized by a boolean function of the responses to the oracle queries (or more formally a truth-table condition). For a machine making k queries on some input, the acceptance function or truth-table condition could be any one of the 2^{2^k} boolean functions on k boolean variables representing the oracle responses to its k parallel queries! However SAT is a very robust oracle language and it is closed under boolean operations such as ANDs and ORs, i.e., given any collection of boolean formulae, $F_1, \ldots, F_k$, one can construct a single reasonably sized boolean formula $\mathcal{F}$ which is satisfiable if and only if all (or at least one) of the formulae $F_1, \ldots, F_k$ are satisfiable, respectively. Thus, many of the 2^{2^k} truth table conditions happen to be equivalent in the sense that with very little (polynomial time) effort a machine asking queries $P_1, \ldots, P_k$ with truth-table condition T_1 could be made to ask a slightly different set of queries $Q_1, \ldots, Q_k$ and truth table condition T_2 (where T_2 is in T_1's equivalence class) without changing it acceptance/rejection behavior. As an example, if a machine asks queries $F_1, \ldots, F_k$ and accepts if all the queries are answered yes then another machine could ask only one query $\mathcal{F}$ constructed as above and accept if it is answered yes and the acceptance behavior of both machines would identical. In fact, a beautiful theory based on Hausdorff's theorems on the boolean closure of set rings shows that, in fact, the 2^{2^k} possible truth-table conditions can be partitioned into only $2k$ different equivalence classes [Hau14, WW85]! These $2k$ classes are actually composed of k different classes and their complements. It is these equivalence classes which form the basis of the Boolean Hierarchy. For a general $\mathrm{P}^{\mathrm{SAT}||[k]}$ computation, the actual equivalence class that the truth-table will fall into will depend on the input. However, if a language in $\mathrm{P}^{\mathrm{SAT}||[k]}$ is recognized by a machine which uses a truth table from the same equivalence class on all inputs then we say that the language is also in the Boolean Hierarchy class defined by that equivalence class. We now formally provide one definition of the Boolean Hierarchy which will prove to be useful in our results. It is important to remember that this definition actually picks out one particular representative from each equivalence class of truth-tables, there are many other ways of defining this hierarchy based on other representatives. 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:

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

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

Thus BH(1) corresponds to NP and BH(2) corresponds to D^P. Among the prominent members of the class D^P is the language CLIQUE(1) defined earlier which is $\leq_m^P$-complete and the language of uniquely satisfiable boolean formulas (USAT) which is easily seen to be in D^P but not known to be $\leq_m^P$-complete. In fact, every level of the Boolean Hierarchy has $\leq_m^P$-complete languages including languages based on important NP Optimization problems [CGH+88]. For example, the language CLIQUE(5) defined earlier is $\leq_m^P$-complete for BH(10). From the definition of the classes BH(k) and co-BH(k) it follows that the following languages are $\leq_m^P$-complete for the respective levels of the Boolean Hierarchy:

Definition We write BL_k for the canonical complete language for BH(k) and co-BL_k for the complete language for co-BH(k):

$$
\begin{aligned}
\mathrm{BL}_1 &\stackrel{\mathrm{def}}{=} \mathrm{SAT}, \\
\mathrm{BL}_{2k} &\stackrel{\mathrm{def}}{=} \{\langle\, x_1, \ldots, x_{2k} \,\rangle \mid \langle\, x_1, \ldots, x_{2k-1} \,\rangle \in \mathrm{BL}_{2k-1} \text{ and } x_{2k} \in \overline{\mathrm{SAT}}\}, \\
\mathrm{BL}_{2k+1} &\stackrel{\mathrm{def}}{=} \{\langle\, x_1, \ldots, x_{2k+1} \,\rangle \mid \langle\, x_1, \ldots, x_{2k} \,\rangle \in \mathrm{BL}_{2k} \text{ or } x_{2k+1} \in \mathrm{SAT}\}, \\
\text{co-}\mathrm{BL}_1 &\stackrel{\mathrm{def}}{=} \overline{\mathrm{SAT}}, \\
\text{co-}\mathrm{BL}_{2k} &\stackrel{\mathrm{def}}{=} \{\langle\, x_1, \ldots, x_{2k} \,\rangle \mid \langle\, x_1, \ldots, x_{2k-1} \,\rangle \in \text{co-}\mathrm{BL}_{2k-1} \text{ or } x_{2k} \in \mathrm{SAT}\}, \\
\text{co-}\mathrm{BL}_{2k+1} &\stackrel{\mathrm{def}}{=} \{\langle\, x_1, \ldots, x_{2k+1} \,\rangle \mid \langle\, x_1, \ldots, x_{2k} \,\rangle \in \text{co-}\mathrm{BL}_{2k} \text{ and } x_{2k+1} \in \overline{\mathrm{SAT}}\}.
\end{aligned}
$$

Since NP and co-NP are closed under boolean ANDs and ORs, one can also prove the following interesting fact about languages in the Boolean Hierarchy over NP sets [CGH+88].

Fact 3.1 In the definition of the class BH(k) we can also assume that the NP languages $L_1, \ldots, L_k$ are such that $L_k \subseteq L_{k-1} \subseteq \ldots \subseteq L_2 \subseteq L_1$.

One of the interesting consequences of Fact 3.1 is the following:

Fact 3.2 Let L be any language in BH(k). Then $L \leq_{\mathrm{m}}^{\mathrm{P}}$-reduces to BL_k via a reduction h which has the following property:

$$\forall x, i,\ (2 \leq i \leq k)\ \ \pi_i \circ h(x) \in \mathrm{SAT} \ \Rightarrow\ \pi_{i-1} \circ h(x) \in \mathrm{SAT}.$$

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 $L_k \subseteq L_{k-1} \subseteq \ldots \subseteq L_2 \subseteq L_1$. We can therefore define $h(x) = \langle\ h_1(x), \ldots, h_k(x)\ \rangle$ where $h_i(x)$ is the boolean formula obtained by applying Karp's reduction to the $x \in L_i$? question.
Another fact about the Boolean Hierarchy which will be useful in some of our proofs is:

Fact 3.3 ([CGH+88], Thm 2.1.2)

$$\{L \mid L = C(\overline{L}_1, \ldots, \overline{L}_k) \text{ for some } L_1, \ldots, L_k \in \mathrm{NP}\} = \begin{cases} \mathrm{BH}(k) & k \text{ even,} \\ \text{co-BH}(k) & k \text{ odd.} \end{cases}$$

Using Fact 3.3 and the earlier definition of the Boolean Hierarchy we can obtain the following useful characterization of the classes co-BH($2k+1$) and co-BH($2k$).

Fact 3.4

$$\begin{aligned} \text{co-BH}(2k) &\stackrel{\text{def}}{=} \{\ L \mid L = A \cup B \cup C \\ &\qquad\qquad \text{where } A \in \mathrm{BH}(2k-2),\ B \in \text{co-NP and } C \in \mathrm{NP}\ \}, \\ \text{co-BH}(2k+1) &\stackrel{\text{def}}{=} \{\ L \mid L = A \cup B \text{ where } A \in \mathrm{BH}(2k) \text{ and } B \in \text{co-NP}\ \}. \end{aligned}$$

Proof: From Fact 3.3 it follows that a language L is in co-BH($2k+1$) if and only if it can be expressed as $C(\overline{L}_1, \ldots, \overline{L}_{2k+1})$ for some NP languages $L_1, \ldots, L_{2k+1}$. By definition of the operator C it follows that $L \in$ co-BH($2k+1$) if and only if $L = C(\overline{L}_1, \ldots, \overline{L}_{2k}) \cup \overline{L_{2k+1}}$ for some NP languages $L_1, \ldots, L_{2k+1}$. But by Fact 3.3, $C(\overline{L}_1, \ldots, \overline{L}_{2k})$ is an language in BH($2k$). Thus, a language $L \in$ co-BH($2k+1$) if and only if $L = A \cup B$, where $A \in$ BH($2k$) and $B \in$ co-NP. Now let us consider the class co-BH($2k$). By definition, a language $L \in$ co-BH($2k$) if and only if $L =$

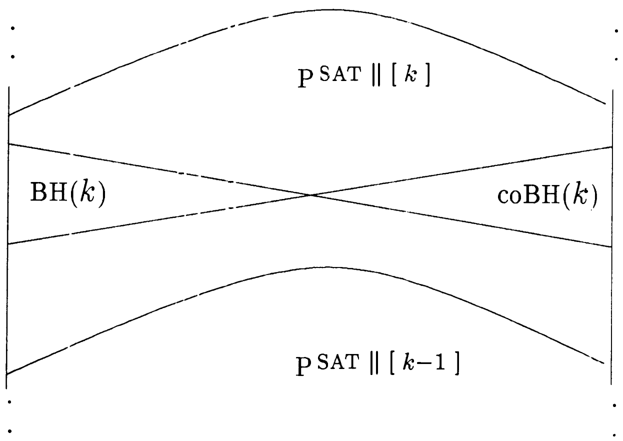


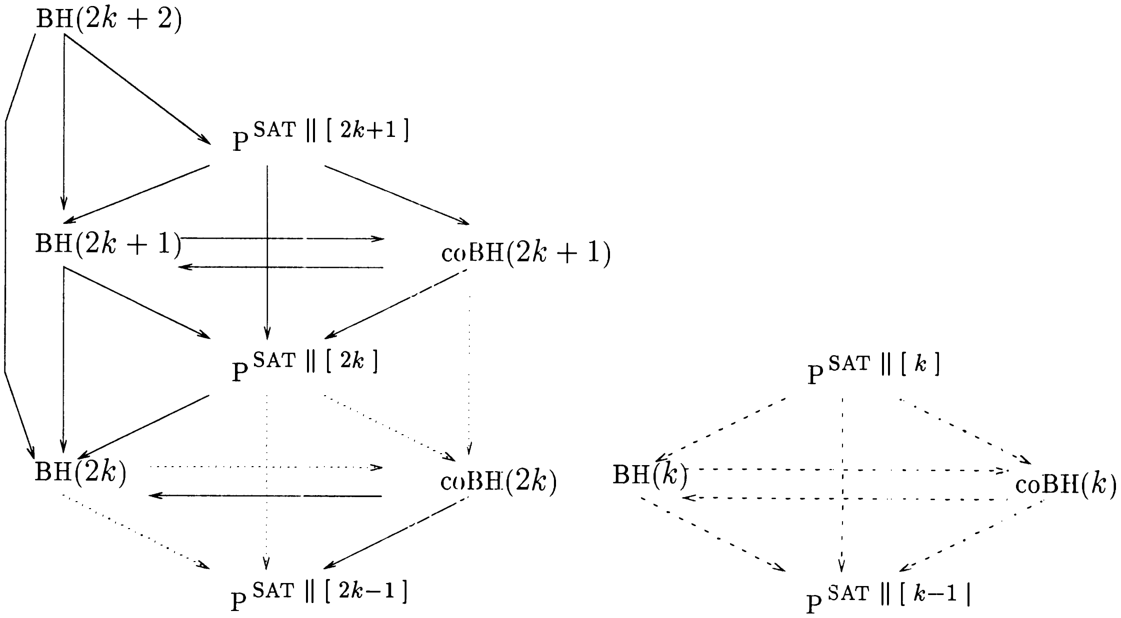
Figure 3.1: The structure of the Boolean and Query Hierarchies

$\overline{C(L_1, \ldots, L_{2k})}$ for some NP languages $L_1, \ldots, L_{2k}$. By definition of operator C is follows that $L \in$ co-BH($2k$) if and only if $L = \overline{C(L_1, \ldots, L_{2k-1})} \cup L_{2k}$ for NP languages $L_1, \ldots, L_{2k}$. But $\overline{C(L_1, \ldots, L_{2k-1})}$ represents a language in co-BH($2k-1$) which can be decomposed into a union of a language $A \in$ BH($2k-2$) and a language $B \in$ co-NP as proved earlier. Thus, a language $L \in$ co-BH($2k$) if and only if it can be expressed as a union of a language in BH($2k-2$), a language in co-NP and a language in NP. $\square$

As we mentioned earlier, the Bounded Query Hierarchy and the Boolean Hierarchy are closely related [Bei91]. 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 3.1 shows the relationships between the various levels of the Boolean and Query Hierarchies. Both hierarchies are proper unless PH collapses [Kad88]. A beautiful result due to Beigel [Bei91] shows that the $\leq_m^P$-complete language for $\mathrm{P}^{\mathrm{SAT}\|[k]}$ is just the tagged union of the canonical $\leq_m^P$-complete languages of the k^{th} level of the Boolean Hierarchy, i.e., BL_k and co-BL_k. We shall call the canonical $\mathrm{P}^{\mathrm{SAT}\|[k]}$ $\leq_m^P$-complete language $\mathrm{BL}_k \oplus$ co-BL_k.



⟶: probability $1 - \frac{1}{k+1}$ $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction

⋯⋯>: probability $1 - \frac{1}{k}$ $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction

- - - ->: probability $1 - \frac{1}{k+1}$ $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reduction

Figure 3.2: Optimal $\leq_{\mathrm{m}}^{\mathrm{rp}}$ and $\leq_{\mathrm{m}}^{\mathrm{bpp}}$ reductions

3.2 Summary of Results

In this section, we present a summary of our results on the power and limitation of randomization in reducing harder languages in the Query and Boolean Hierarchies to simpler ones. We considered both $\leq_{\mathrm{m}}^{\mathrm{rp}}$ and $\leq_{\mathrm{m}}^{\mathrm{bpp}}$ reductions between nearby classes in the Boolean and Bounded Query Hierarchies (26 reductions in all) and in each case we provide a tight bound on the success probability that can be achieved by the reduction. Figure 3.2 summarizes these bounds. To obtain each of these bounds we first proved that reductions achieving these success probabilities exist. Some of these proofs will be presented in Section 3.3. Then we showed that there cannot exist reductions which achieve significantly higher (higher by $1/poly$) success probability unless the Polynomial time hierarchy (PH) collapses. Some of these proofs will be presented in Section 3.4. Our bounds also provide upper and lower bounds on the success probabilities of random reductions between classes which are further apart. We conjecture that tight performance bounds exist for those reductions as well.

3.3 The Reductions

We now show that random reductions can reduce harder languages to simpler ones with the probabilities shown in Figure 3.2. The $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions are simpler than the $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reductions but their success probabilities are lower. Lemmas 3.1, 3.2 and 3.3 are sufficient to derive all the $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions in Figure 3.2. 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 3.1 For any $k \geq 2$, $\mathrm{BL}_{2k} \leq_{\mathrm{m}}^{\mathrm{rp}} \mathrm{BL}_{2k-2}$ with probability $1 - 1/k$.

Proof: For all k, it is well known that a language L is in BH($2k$) if and only if it can be expressed as a union of k D^P sets [CGH+88]. In fact this result has been used as a basis for defining the class BH($2k$) in the past. An easy way to see this is by means of an example. Consider the BH(10)-complete language CLIQUE(5) described earlier. Just by looking at its definition it is obvious that it can be represented as a union of the 5 D^P sets (Maxcliquesize$(G) = a_i$) for $1 \leq i \leq 5$. Thus, in general we can decompose a language in BL_{2k} as a union of k D^P sets. If we omit one of the k D^P sets in the union, we will be left with a language in BH($2k-2$) which $\leq_m^P$-reduces to the complete language BL_{2k-2}. We can therefore reduce BL_{2k} to BL_{2k-2} by randomly omitting one of the k D^P sets from the union. This type of reduction has a one-sided error of up to $1/k$. In the case of CLIQUE(5), one would randomly exclude one of the 5 D^P sets from the union to reduce CLIQUE(5) to the simpler language CLIQUE(4) with one sided error of 1/5. □

Lemma 3.1 immediately implies that for any two languages L_1 and L_2 such that $L_1 \leq_m^P BL_{2k}$ and $BL_{2k-2} \leq_m^P L_2$ we have that $L_1 \leq_m^{rp} L_2$ with probability $1 - 1/k$. Thus, we get the following corollary.

Corollary 3.1 $BL_{2k+2} \leq_m^{rp} BL_{2k}$, $BL_{2k+2} \leq_m^{rp} BL_{2k+1}$, $BL_{2k+2} \leq_m^{rp} BL_{2k+1} \oplus \text{co-}BL_{2k+1}$, $BL_{2k+1} \oplus \text{co-}BL_{2k+1} \leq_m^{rp} BL_{2k+1}$, $BL_{2k+1} \oplus \text{co-}BL_{2k+1} \leq_m^{rp} \text{co-}BL_{2k+1}$, $BL_{2k+1} \oplus \text{co-}BL_{2k+1} \leq_m^{rp} BL_{2k} \oplus \text{co-}BL_{2k}$, $BL_{2k+1} \leq_m^{rp} \text{co-}BL_{2k+1}$, $BL_{2k+1} \leq_m^{rp} BL_{2k}$, $BL_{2k+1} \leq_m^{rp} BL_{2k} \oplus \text{co-}BL_{2k}$, $\text{co-}BL_{2k+1} \leq_m^{rp} BL_{2k+1}$, $\text{co-}BL_{2k+1} \leq_m^{rp} BL_{2k} \oplus \text{co-}BL_{2k}$, $BL_{2k} \oplus \text{co-}BL_{2k} \leq_m^{rp} BL_{2k}$ and $\text{co-}BL_{2k} \leq_m^{rp} BL_{2k}$ with probability $1-1/(k+1)$. Also, $BL_{2k} \leq_m^{rp} BL_{2k-1} \oplus \text{co-}BL_{2k-1}$ and $BL_{2k} \leq_m^{rp} \text{co-}BL_{2k}$ with probability $1 - 1/k$.

Lemma 3.2 For any $k \geq 1$, $\text{co-}BL_{2k} \leq_m^{rp} BL_{2k-1} \oplus \text{co-}BL_{2k-1}$ with probability $1 - 1/(k+1)$.

Proof: For all $k \geq 1$, we know from Fact 3.4 that a language L is in co-BH($2k$) if and only if it can be expressed as a union of $k-1$ $\mathrm{D^P}$ sets (which form a set in BH($2k-2$)), an NP set and a co-NP set. As an example consider the language $\overline{\mathrm{CLIQUE}(5)}$ which 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) $< a_1$). From the definition of the Boolean Hierarchy it is not hard to show that omitting one of the $k+1$ 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). Thus, in all cases, the language will be in $\mathrm{P}^{\mathrm{SAT}||[2k-1]}$. We can therefore reduce co-BL_{2k} to $\mathrm{BL}_{2k-1} \oplus$ co-BL_{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 $1/k+1$. In the case of $\overline{\mathrm{CLIQUE}(5)}$ this means that it can be recognized using randomness and only 9 nonadaptive queries SAT with one-sided error of 1/6. □

The $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions from $\mathrm{P}^{\mathrm{SAT}||[2k]}$ to co-BL_{2k} and to $\mathrm{BL}_{2k-1} \oplus$ co-BL_{2k-1} are based on the fact that the language $\mathrm{BL}_{2k} \oplus$ co-BL_{2k} is complete for $\mathrm{P}^{\mathrm{SAT}||[2k]}$ [Bei91]. If a given instance of a problem in $\mathrm{P}^{\mathrm{SAT}||[2k]}$ $\leq_{\mathrm{m}}^{\mathrm{P}}$-reduces to an instance of BL_{2k} then we can use the generic reductions from BL_{2k} to co-BL_{2k} and $\mathrm{BL}_{2k-1} \oplus$ co-BL_{2k-1} respectively. If, on the other hand, the given instance $\leq_{\mathrm{m}}^{\mathrm{P}}$-reduces to an instance of co-BL_{2k} then the reduction to co-BL_{2k} is trivial and the reduction to and $\mathrm{BL}_{2k-1} \oplus$ co-BL_{2k-1} can be obtained by applying the generic reduction from co-BL_{2k} to this class. Thus, as a corollary to Lemmas 3.1 and 3.2 we also have:

Corollary 3.2 $\mathrm{BL}_{2k} \oplus$ co-$\mathrm{BL}_{2k} \leq_{\mathrm{m}}^{\mathrm{rp}}$ co-BL_{2k} with probability $1-1/k$ and $\mathrm{BL}_{2k} \oplus$ co-$\mathrm{BL}_{2k} \leq_{\mathrm{m}}^{\mathrm{rp}} \mathrm{BL}_{2k-1} \oplus$ co-BL_{2k-1} with probability $1-1/k$.

Lemma 3.3 For any $k \geq 1$, co-$\mathrm{BL}_{2k+1} \leq_{\mathrm{m}}^{\mathrm{rp}}$ co-BL_{2k} with probability $1-1/k$.

Proof: For all $k \geq 1$, we know from Fact 3.4 that a language L is in co-BH$(2k+1)$ if and only if it can be expressed as a union of k $\mathrm{D^P}$ sets (which form a set in BH$(2k)$) and a co-NP set. Also we know that the class co-BH$(2k)$ consists of languages expressible as a union of $k-1$ $\mathrm{D^P}$ sets (which form a set in BH$(2k-2)$), a co-NP set and an NP set. Thus, the random exclusion of one of the k $\mathrm{D^P}$ sets in the expression for co-BL_{2k+1} gives us a probability $1-1/k$ $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction from co-$\mathrm{BL}_{2k+1}$ to co-BL_{2k}. □

Random 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 simple properties of the classes can be used to derive the $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reductions in Figure 3.2.

Lemma 3.4 $\mathrm{BL}_k \leq_{\mathrm{m}}^{\mathrm{bpp}} \mathrm{BL}_{k-1} \oplus \text{co-}\mathrm{BL}_{k-1}$ with probability $1-1/(k+1)$.

Proof: Let A denote the language BL_k and let B denote $\mathrm{BL}_{k-1} \oplus \text{co-}\mathrm{BL}_{k-1}$ 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.

Case 1: k is even: Let $k=2t$. Then by Lemma 3.1 we know that $A \leq_{\mathrm{m}}^{\mathrm{rp}} B$ via a reduction R_1 with probability $1-1/t$. Also by Lemma 3.2 we know that $\overline{A} \leq_{\mathrm{m}}^{\mathrm{rp}} B$ with probability $1-1/(t+1)$ via a reduction R_2. Since, $B \leq_{\mathrm{m}}^{\mathrm{P}} \overline{B}$ because s 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 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 $1 - 1/(k+1)$ $\leq_m^{\text{bpp}}$-reduction.

Case 2: k is odd. Similar to Case I, details omitted. □

Example: Consider the BH(10)-complete language CLIQUE(5) which is a union of the 5 D^P sets (Maxcliquesize(G) $= a_i$) for $1 \leq i \leq 5$. We can obtain five simpler languages $L_1, \ldots, L_5$ by removing a set from this union. Also as mentioned before, assuming that the integers $a_1, \ldots, a_5$ are in increasing order, the language $\overline{\text{CLIQUE(5)}}$ is the union of 4 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) $< a_1$). If we take the union of CLIQUE(5) and any one of these six sets then the resulting language is actually simpler than CLIQUE(5) (in fact it is in $P^{SAT||[9]}$)! To see this, note that checking that Maxcliquesize(G) is in a given contiguous range of values takes 2 queries. Checking whether Maxcliquesize(G) is greater or less than a particular value takes 1 query. Thus, CLIQUE(5) requires 10 queries because 5 ranges of size 1 each ($a_i - a_i$, $1 \leq i \leq 5$) have to be checked simultaneously. The union of CLIQUE(5) with any of the four D^P sets $a_i <$ Maxcliquesize(G) $< a_{i+1}$ has only four ranges that have to be checked and this requires only 8 queries. If we take the union of CLIQUE(5) with the NP set Maxcliquesize(G) $> a_5$ then instead of using two queries to check in the range $a_5 - a_5$, we just need to check if Maxcliquesize(G) $\geq a_5$ and thus we save a query, Similarly the union of CLIQUE(5) with the co-NP set Maxcliquesize(G) $< a_1$ requires one less query than CLIQUE(5) because the $a_1 - a_1$ range check is replaced by the test Maxcliquesize(G) $\leq a_1$. Thus, we can form six simpler languages $R_1, \ldots, R_6$ by taking the union of CLIQUE(5) with one of the six sets whose union constitutes $\overline{\text{CLIQUE(5)}}$. The $\leq_m^{\text{bpp}}$-reduction from CLIQUE(5) to a language in $P^{SAT||[9]}$ would be to randomly choose one of these *eleven* simpler languages $L_1, \ldots, L_5, R_1, \ldots, R_6$.

As mentioned before, by using Lemma 3.4 and simple relationships between the classes in the Boolean and Query Hierarchies we can get all the $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reductions shown in Figure 3.2. By definition, the $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reduction of Lemma 3.4 is also a reduction from $\overline{\mathrm{BL}_k}$ to $\overline{\mathrm{BL}_{k-1} \oplus \text{co-}\mathrm{BL}_{k-1}}$. Since co-$\mathrm{BL}_k \leq_{\mathrm{m}}^{\mathrm{P}} \overline{\mathrm{BL}_k}$ and $\overline{\mathrm{BL}_{k-1} \oplus \text{co-}\mathrm{BL}_{k-1}} \leq_{\mathrm{m}}^{\mathrm{P}} \mathrm{BL}_{k-1} \oplus \text{co-}\mathrm{BL}_{k-1}$, we get the required $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reduction from co-$\mathrm{BL}_k$ to $\mathrm{BL}_{k-1} \oplus \text{co-}\mathrm{BL}_{k-1}$. Also since $\mathrm{BL}_{k-1} \oplus \text{co-}\mathrm{BL}_{k-1}$ $\leq_{\mathrm{m}}^{\mathrm{P}}$-reduces to both BL_k and co-BL_k, the $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reductions from BL_k to co-BL_k and vice-versa are actually $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reductions to $\mathrm{BL}_{k-1} \oplus \text{co-}\mathrm{BL}_{k-1}$ followed by a $\leq_{\mathrm{m}}^{\mathrm{P}}$-reduction to the appropriate language. Also, as in the case of $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions, the $\leq_{\mathrm{m}}^{\mathrm{bpp}}$-reductions from languages in the class $\mathrm{P}^{\mathrm{SAT}||[k]}$ are based on the fact that the language $\mathrm{BL}_k \oplus \text{co-}\mathrm{BL}_k$ is $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete for it. Thus, for any given instance of a problem in $\mathrm{P}^{\mathrm{SAT}||[k]}$ the either the generic reduction from BL_k or the reduction from co-BL_k is used. Thus, we can state as a corollary:

Corollary 3.3 $\mathrm{BL}_k \leq_{\mathrm{m}}^{\mathrm{bpp}}$ co-BL_k, co-$\mathrm{BL}_k \leq_{\mathrm{m}}^{\mathrm{bpp}} \mathrm{BL}_k$, co-$\mathrm{BL}_k \leq_{\mathrm{m}}^{\mathrm{bpp}} \mathrm{BL}_{k-1} \oplus$ co-BL_{k-1}, $\mathrm{BL}_k \oplus \text{co-}\mathrm{BL}_k \leq_{\mathrm{m}}^{\mathrm{bpp}} \mathrm{BL}_k$, $\mathrm{BL}_k \oplus \text{co-}\mathrm{BL}_k \leq_{\mathrm{m}}^{\mathrm{bpp}}$ co-BL_k and $\mathrm{BL}_k \oplus \text{co-}\mathrm{BL}_k \leq_{\mathrm{m}}^{\mathrm{bpp}} \mathrm{BL}_{k-1} \oplus \text{co-}\mathrm{BL}_{k-1}$ with probability $1 - 1/(k+1)$.

3.4 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 derive all the bounds in Figure 3.2. Since the proofs of the general cases are quite tedious we will first illustrate our technique by means of an example and then provide the complete proof for a general case.

3.4.1 An Example

In this section, we will prove that the probability 2/3 $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction from the $\mathrm{P}^{\mathrm{SAT}||[5]}$-complete language to a language in $\mathrm{P}^{\mathrm{SAT}||[4]}$ is almost optimal. From the structure of the Boolean and Query Hierarchies (see Figure 3.1), it suffices to prove that the probability 2/3 $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction from BL_5 to co-BL_5 is almost optimal. The following theorem asserts that this is the case.

Theorem 3.1 Let L be any language in co-BH(5). If $\mathrm{BL}_5 \leq_{\mathrm{m}}^{\mathrm{rp}} L$ with probability $2/3 + 1/p(n)$, for some polynomial p, then the PH collapses to Σ_3^{P}.

We need some definitions and lemmas before we can prove this theorem. Note that the hypothesis of Theorem 3.1 implies that there *exists* a probability $2/3 + 1/p(n)$ $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction from BL_5 to co-BL_5. Arbitrary $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions from BL_5 to co-BL_5 are difficult to work with because they may lack structure. However, we can establish that if there is some $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction to co-$\mathrm{BL}_5$ then there is a $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction which has very nice structural properties. We call such reductions *nested* $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions. In general we define such reductions as follows:

Definition Let h be a $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction from some language A to BL_ℓ (co-BL_ℓ) for some ℓ. We say that h is a *nested* $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction from A to BL_ℓ (co-BL_ℓ resp.) if the following condition holds: For all x, z, i, $(2 \leq i \leq \ell)$,

$$\pi_i \circ h(x,z) \in \mathrm{SAT} \Rightarrow \pi_{i-1} \circ h(x,z) \in \mathrm{SAT}.$$

The following lemma based on Fact 3.2 allows us to work with the much nicer *nested* $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions instead of arbitrary $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions.

Lemma 3.5 Let A be any set and let L be an arbitrary language in BH(ℓ) (co-BH(ℓ)). If $A \leq_{\mathrm{m}}^{\mathrm{rp}} L$ with probability δ, then there exists a *nested* $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction h such that $A \leq_{\mathrm{m}}^{\mathrm{rp}} \mathrm{BL}_\ell$ ($A \leq_{\mathrm{m}}^{\mathrm{rp}}$ co-BL_ℓ, respectively) via h with probability δ.

Thus, if the hypothesis of Theorem 3.1 is true, then there exists a probability $2/3 + 1/p(n)$ *nested* $\leq_{\mathrm{m}}^{\mathrm{rp}}$ reduction from BL_5 to co-BL_5.

Notation Henceforth we will refer to this *nested* reduction from BL_5 to co-BL_5 as h. Let $r(n)$ be the size of the random input to h and let $q(m)$ be the size of 5-tuples of strings in $\{0,1\}^m$. Let z denote a randomly and uniformly chosen string of size $r(q(m))$ and let ε denote $1/p(q(m))$. For inputs $I = \langle\ x_1, \ldots, x_5\ \rangle$, where $x_1, \ldots, x_5 \in \{0,1\}^m$ and random string z, let $h_i(I,z)$ denote $\pi_i \circ h(I,z)$, i.e., the i'th component of the 5-tuple $h(I,z)$.

Using this notation we can say that for any 5-tuple $I = \langle\ x_1, \ldots, x_5\ \rangle$, where $x_1, \ldots, x_5 \in \{0,1\}^m$,

$$I \in \mathrm{BL}_5 \ \Rightarrow\ \mathrm{Prob}_z[h(I,z) \in \text{co-}\mathrm{BL}_5] \geq 2/3 + \varepsilon,$$

$$I \in \text{co-}\mathrm{BL}_5 \ \Rightarrow\ \mathrm{Prob}_z[h(I,z) \in \mathrm{BL}_5] = 1.$$

and since h is a *nested* reduction, in addition we can say that for $1 \leq i \leq 5$,

$$h_i(I,z) \in \mathrm{SAT} \ \Rightarrow\ h_1(I,z), \ldots, h_i(I,z) \in \mathrm{SAT}.$$

The reason for using *nested* reductions is that their structural properties allow us to prove the following powerful lemma (Probability Recovery Lemma) which plays a crucial role in our proofs.

Lemma 3.6 [Probability Recovery Lemma] Suppose $\mathrm{BL}_5 \leq_{\mathrm{m}}^{\mathrm{rp}}$ co-BL_5 via h defined above. Let H_2, H_3, H_4, H_5 be any collection of 4 formulas in $\overline{\mathrm{SAT}}^{=m}$ and let x_1, x_2, x_3 be any collection of 3 strings in $\{0,1\}^m$. Then the following propositions hold:

- $\mathcal{P}_1$: Let $I = \langle\ x_1, x_2, x_3, H_4, H_5\ \rangle$, then

$$\langle\ x_1, x_2, x_3\ \rangle \in \text{co-}\mathrm{BL}_3 \ \Rightarrow\ \mathrm{Prob}_z[\langle\ h_1(I,z), h_2(I,z), h_3(I,z)\ \rangle \in \mathrm{BL}_3] = 1.$$

- $\mathcal{P}_2$: Let $I = \langle\ x_1, x_2, H_3, H_4, H_5\ \rangle$, then

$$\langle\ x_1, x_2\ \rangle \in \mathrm{BL}_2 \ \Rightarrow\ \mathrm{Prob}_z[\langle\ h_1(I,z), h_2(I,z)\ \rangle \in \text{co-}\mathrm{BL}_2] \geq 2/3 + \varepsilon.$$

- $\mathcal{P}_3$: Let $I = \langle\ x_1, H_2, H_3, H_4, H_5\ \rangle$, then

$$x_1 \in \text{co-BL}_1 \ \Rightarrow\ \text{Prob}_z[h_1(I,z) \in \text{BL}_1] = 1.$$

Proof:

$\mathcal{P}_1$: Suppose $\langle\ x_1, x_2, x_3\ \rangle \in \text{co-BL}_3$. Consider $I = \langle\ x_1, x_2, x_3, H_4, H_5\ \rangle$. By definition of co-BL$_5$ we know that $I \in \text{co-BL}_5$ if and only if

$$(\ (\ \langle\ x_1, x_2, x_3\ \rangle \in \text{co-BL}_3 \ \text{ or } \ H_4 \in \text{SAT}\ ) \ \text{ and } \ H_5 \in \overline{\text{SAT}}\ ).$$

Since $H_4, H_5 \in \overline{\text{SAT}}$ we know that $I \in \text{co-BL}_5$. Therefore, by definition of h, we know that $\text{Prob}_z[h(I,z) \in \text{BL}_5] = 1$. Fix I and z. For $1 \le i \le 5$, let $t_i = h_i(I,z)$. We prove the result by showing that $h(I,z) \in \text{BL}_5$ implies $\langle\ t_1, t_2, t_3\ \rangle \in \text{BL}_3$. Assume $h(I,z) \in \text{BL}_5$. Then by definition,

$$(\ (\ (\ \langle\ t_1, t_2, t_3\ \rangle \in \text{BL}_3\ ) \ \text{ and } \ t_4 \in \overline{\text{SAT}}\ ) \ \text{ or } t_5 \in \text{SAT}).$$

From the above characterization, if $t_5 \in \overline{\text{SAT}}$ then $\langle\ t_1, t_2, t_3\ \rangle$ must be in BL$_3$. If $t_5 \in \text{SAT}$, then since h is a *nested* reduction, t_3 must be in SAT and hence $\langle\ t_1, t_2, t_3\ \rangle \in \text{BL}_3$ since

$$\langle\ t_1, \ldots, t_3\ \rangle \in \text{BL}_3 \iff \langle\ t_1, t_2\ \rangle \in \text{BL}_2 \text{ or } t_3 \in \text{SAT}.$$

$\mathcal{P}_2$: Suppose $\langle\ x_1, x_2\ \rangle \in \text{BL}_2$. From the definition of BL$_5$ and the fact that $H_3, H_4, H_5 \in \overline{\text{SAT}}$, it follows that $I = \langle\ x_1, x_2, H_3, H_4, H_5\ \rangle \in \text{BL}_5$. Therefore

$$\text{Prob}_z[h(I,z) \in \text{co-BL}_5] \ge 2/3 + \varepsilon.$$

Fix any I and z such that $h(I,z) \in \text{co-BL}_5$. For $1 \le i \le 5$, let $t_i = h_i(I,z)$. We prove the result by showing that $\langle\ t_1, t_2\ \rangle \in \text{co-BL}_2$. By definition,

$$\begin{aligned} h(I,z) \in \text{co-BL}_5 \ &\Rightarrow\ \langle\ t_1, \ldots, t_4\ \rangle \in \text{co-BL}_4 \text{ and } t_5 \in \overline{\text{SAT}} \\ &\Rightarrow\ \langle\ t_1, \ldots, t_4\ \rangle \in \text{co-BL}_4 \\ &\Rightarrow\ (\ (\ (\ \langle\ t_1, t_2\ \rangle \in \text{co-BL}_2\ ) \text{ and } t_3 \in \overline{\text{SAT}}\ ) \text{ or } t_4 \in \text{SAT}\ ). \end{aligned}$$

Thus, if $t_4 \in \overline{\mathrm{SAT}}$ then $\langle\ t_1, t_2\ \rangle$ must be in co-BL_2. If $t_4 \in \mathrm{SAT}$, then since h is a *nested* reduction, t_2 must be in SAT and hence $\langle\ t_1, t_2\ \rangle \in$ co-BL_2 since

$$\langle\ t_1, t_2\ \rangle \in \text{co-BL}_2 \iff t_1 \in \overline{\mathrm{SAT}} \text{ or } t_2 \in \mathrm{SAT}.$$

$\mathcal{P}_3$: The proof of $\mathcal{P}_3$ is similar to the proof of $\mathcal{P}_1$ once $\mathcal{P}_1$ has been established. □

Now we can proceed with the proof of Theorem 3.1.

Proof of Theorem 3.1: The proof will use the Probability Recovery Lemma in conjunction with a variant of the *hard/easy* proof technique. The basic *hard/easy* proof technique pioneered by Kadin [Kad88] and further refined by Chang and Kadin [CK90a] was used to prove that the existence of $\leq_{\mathrm{m}}^{\mathrm{P}}$-reductions between complementary classes in the Boolean Hierarchy would imply the existence of an NP machine which for every length m, given a small (polynomial sized) amount of additional information (or advice), would recognize all $\overline{\mathrm{SAT}}$ formulas of length m. Apart from the small technicality introduced by the polynomial sized advice for every input length, this would mean that an NP machine could solve co-NP problems and that would collapse the PH. Thus, the basic hard/easy technique rules out $\leq_{\mathrm{m}}^{\mathrm{P}}$-reductions between complementary classes in the Boolean Hierarchy. Our variation of the hard/easy technique is geared towards proving the impossibility of having random reductions between complementary classes in this hierarchy which succeed beyond a certain probability. As is the case with the basic proof technique, our variant is also nonuniform; however we will show that if the hypothesis of Theorem 3.1 is true, then for every length m, given polynomial sized advice, one can construct *Merlin-Arthur-Merlin* games [Bab85] for recognizing all $\overline{\mathrm{SAT}}$ formulas of length m. A Merlin-Arthur-Merlin game is a 3-round game between two players, a probabilistic polynomial time player Arthur and an all powerful existential player Merlin. The game proceeds as follows: Given an input x which is visible to both players, in the first round Merlin existentially provides a polynomial (in size of x) sized string m_1 which is visible to both players. Arthur, in the second round, randomly produces a polynomial sized string a which is also visible to Merlin. In the third and final round Merlin again existentially provides a polynomial sized string m_2. The input x is then accepted if and only if an a priori fixed polynomial time referee machine accepts the string x, m_1, a, m_2. A language L is said to have an Merlin-Arthur-Merlin game if and only if there is polynomial time referee machine M such that whenever a string is in L, the probability that the referee M accepts the ensuing Merlin-Arthur-Merlin game is significantly higher than if the string was not in L. Thus, a Merlin-Arthur-Merlin game a type of interactive proof in which the number of rounds is bounded by 3 and all moves are visible. It has been known for quite some time that the existence of such games for $\overline{\text{SAT}}$ would imply that the PH collapses to Σ_3^{P} [BHZ87,Yap83] and it is straightforward to show that the same result holds even with the presence of advice [Cha91]. Thus, once we are able to construct advice-based Merlin-Arthur-Merlin games for $\overline{\text{SAT}}$ given the hypothesis of Theorem 3.1 the conclusion (PH collapsing) would follow immediately. Let us now proceed with the main part of the proof.

From the hypothesis of the Theorem, it follows that for any $I = \langle\ x_1, \ldots, x_5\ \rangle$, where $x_1, \ldots, x_5 \in \{0,1\}^m$,

$$I \in \text{BL}_5 \;\Rightarrow\; \text{Prob}_z[h(I,z) \in \text{co-BL}_5] \geq 2/3 + \varepsilon,$$

$$I \in \text{co-BL}_5 \;\Rightarrow\; \text{Prob}_z[h(I,z) \in \text{BL}_5] = 1.$$

Using the definitions of BL_5 and co-BL_5 we rewrite this as

$$\langle\ x_1, \ldots, x_4\ \rangle \in \text{BL}_4 \text{ or } x_5 \in \text{SAT} \quad \Rightarrow$$

$$\text{Prob}_z[\ \pi_{(1,4)} \circ h(I,z) \in \text{co-BL}_4 \ \text{ and }\ h_5(I,z) \in \overline{\text{SAT}}\ ] \geq 2/3 + \varepsilon,$$

$$\langle\ x_1, \ldots, x_4\ \rangle \in \text{co-BL}_4 \text{ and } x_5 \in \overline{\text{SAT}} \quad \Rightarrow$$

$$\text{Prob}_z[\ \pi_{(1,4)} \circ h(I,z) \in \text{BL}_4 \ \text{ or }\ h_5(I,z) \in \text{SAT}\ ] = 1.$$

Case 1: Suppose, the following condition $\mathcal{C}_1$ holds:

$\mathcal{C}_1$: $\forall x_5 \in \overline{\mathrm{SAT}}^{=m}$, $\exists x_1, \ldots, x_4 \in \{0,1\}^m$ such that for $I = \langle\, x_1, \ldots, x_5 \,\rangle$, $\mathrm{Prob}_z[h_5(I,z) \in \mathrm{SAT}] \geq 1/3$.

Then we can set up a Merlin-Arthur-Merlin game for $\overline{\mathrm{SAT}}$ formulas of length m as follows: Given a formula x_5 of length m, Merlin first provides $x_1, \ldots, x_4$. Arthur then provides the random z and asks Merlin to show that $h_5(I,z) \in \mathrm{SAT}$. If $x_5 \in \overline{\mathrm{SAT}}$, then condition $\mathcal{C}_1$ ensures that by a suitable choice of $x_1, \ldots, x_4$, Merlin can succeed with probability $\geq 1/3$. If on the other hand, $x_5 \in \mathrm{SAT}$, then irrespective of what $x_1, \ldots, x_4$ is, $I \in \mathrm{BL}_5$ and by the definition of h, $h_5(I,z) \in \overline{\mathrm{SAT}}$ with probability $\geq 2/3 + \varepsilon$. Thus, Merlin can succeed with probability at most $1/3 - \varepsilon$ in this case. This probability difference of ε is enough to distinguish between the two cases.

However, $\mathcal{C}_1$ may not hold, i.e., it may be the case that $\exists x_5 \in \overline{\mathrm{SAT}}^{=m}$ such that $\forall x_1, \ldots, x_4$, $\mathrm{Prob}_z[h_5(I,z) \in \mathrm{SAT}] < 1/3$. We then call the least such x_5 as H_5 and assume that it is provided as advice and proceed to the next case. Note that $H_5 \in \overline{\mathrm{SAT}}$.

Case 2: Given H_5 we know that for any 5-tuple $I = \langle\, x_1, \ldots, x_4, H_5 \,\rangle$, where $x_1, \ldots, x_4 \in \{0,1\}^m$

$$\langle\, x_1, \ldots, x_4 \,\rangle \in \mathrm{BL}_4 \text{ or } H_5 \in \mathrm{SAT} \quad \Rightarrow$$

$$\mathrm{Prob}_z[\ \pi_{(1,4)} \circ h(I,z) \in \text{co-}\mathrm{BL}_4 \ \text{ and } \ h_5(I,z) \in \overline{\mathrm{SAT}}\ ] \geq 2/3 + \varepsilon,$$

$$\langle\, x_1, \ldots, x_4 \,\rangle \in \text{co-}\mathrm{BL}_4 \text{ and } H_5 \in \overline{\mathrm{SAT}} \quad \Rightarrow$$

$$\mathrm{Prob}_z[\ \pi_{(1,4)} \circ h(I,z) \in \mathrm{BL}_4 \ \text{ or } \ h_5(I,z) \in \mathrm{SAT}\ ] = 1.$$

Since we know that $H_5 \in \overline{\text{SAT}}$ and $\forall x_1, \ldots, x_4$, $\text{Prob}_z[h_5(I,z) \in \text{SAT}] < 1/3$, we can simplify the above expression to get

$$\langle\ x_1, \ldots, x_4\ \rangle \in \text{BL}_4 \ \Rightarrow\ \text{Prob}_z[\pi_{(1,4)} \circ h(I,z) \in \text{co-BL}_4] \geq 2/3 + \varepsilon,$$

$$\langle\ x_1, \ldots, x_4\ \rangle \in \text{co-BL}_4 \ \Rightarrow\ \text{Prob}_z[\pi_{(1,4)} \circ h(I,z) \in \text{BL}_4] \geq 1 - 1/3 = 2/3.$$

Note that we now have a weaker reduction at one level lower in the Boolean Hierarchy. Again by unfolding the definition of BL_4 and co-BL_4, we can restate this as

$$\langle\ x_1, x_2, x_3\ \rangle \in \text{BL}_3 \text{ and } x_4 \in \overline{\text{SAT}} \quad \Rightarrow$$

$$\text{Prob}_z[\ \pi_{(1,3)} \circ h(I,z) \in \text{co-BL}_3 \ \text{ or } \ h_4(I,z) \in \text{SAT}\ ] \geq 2/3 + \varepsilon,$$

$$\langle\ x_1, x_2, x_3\ \rangle \in \text{co-BL}_3 \text{ or } x_4 \in \text{SAT} \quad \Rightarrow$$

$$\text{Prob}_z[\ \pi_{(1,3)} \circ h(I,z) \in \text{BL}_3 \ \text{ and } \ h_4(I,z) \in \overline{\text{SAT}}\ ] \geq 2/3.$$

Now, if condition $\mathcal{C}_2$ holds then just as in Case 1, we can design a Merlin-Arthur-Merlin game for $\overline{\text{SAT}}$ formulas of length m:

$\mathcal{C}_2$: $\forall x_4 \in \overline{\text{SAT}}^{=m}$, $\exists x_1, x_2, x_3 \in \{0,1\}^m$ such that for $I = \langle\ x_1, \ldots, x_4, H_5\ \rangle$, $\text{Prob}_z[h_4(I,z) \in \text{SAT}] \geq 1/3 + \varepsilon/2$.

However, even $\mathcal{C}_2$ may not hold, i.e., it may be the case that $\exists x_4 \in \overline{\text{SAT}}^{=m}$ such that $\forall x_1, x_2, x_3$, $\text{Prob}_z[h_4(I,z) \in \text{SAT}] < 1/3 + \varepsilon/2$. We then call the least such x_4 as H_4 and assume that both H_4 and H_5 are provided as advice and proceed to the next case.

Case 3: Using the properties of H_4, we can simplify the expression we had in Case 2 to: Given any 5-tuple $I = \langle\ x_1, x_2, x_3, H_4, H_5\ \rangle$, where $x_1, x_2, x_3 \in \{0,1\}^m$

$$\langle\ x_1, x_2, x_3\ \rangle \in \text{BL}_3 \ \Rightarrow\ \text{Prob}_z[\pi_{(1,3)} \circ h(I,z) \in \text{co-BL}_3] \geq 1/3 + \varepsilon/2,$$

$$\langle\ x_1, x_2, x_3\ \rangle \in \text{co-BL}_3 \ \Rightarrow\ \text{Prob}_z[\pi_{(1,3)} \circ h(I,z) \in \text{BL}_3] \geq 2/3.$$

Note that we now have a very weak random reduction between the classes BH(3) and co-BH(3). Due to our analysis, the reduction h appears to have lost a lot of success probability. However, since h was a *nested* reduction, we can recover some of this lost probability because proposition $\mathcal{P}_1$ of the *Probability Recovery Lemma* is applicable and we actually have the following:

$$\langle\ x_1, x_2, x_3\ \rangle \in \mathrm{BL}_3 \ \Rightarrow\ \mathrm{Prob}_z[\pi_{(1,3)}\circ h(I,z) \in \text{co-BL}_3] \geq 1/3 + \varepsilon/2$$

$$\langle\ x_1, x_2, x_3\ \rangle \in \text{co-BL}_3 \ \Rightarrow\ \mathrm{Prob}_z[\pi_{(1,3)}\circ h(I,z) \in \mathrm{BL}_3] = 1$$

Even this reduction is not very powerful. In fact there exists a probability 1/2, $\leq_m^{rp}$-reduction from BL_3 to co-BL_3, without any assumptions! However, we can still obtain our result because the reduction based on h has the potential for large amounts probability recovery from propositions $\mathcal{P}_2$ and $\mathcal{P}_3$ of the Probability Recovery Lemma.

As usual, by unfolding the definitions of BL_3 and co-BL_3 we obtain: Given any 5-tuple $I = \langle\ x_1, x_2, x_3, H_4, H_5\ \rangle$, where $x_1, x_2, x_3 \in \{0,1\}^m$,

$$\langle\ x_1, x_2\ \rangle \in \mathrm{BL}_2 \text{ or } x_3 \in \mathrm{SAT} \quad \Rightarrow$$

$$\mathrm{Prob}_z[\ \pi_{(1,2)}\circ h(I,z) \in \text{co-BL}_2 \ \text{ and } \ h_3(I,z) \in \overline{\mathrm{SAT}}\ ] \geq 1/3 + \varepsilon/2.$$

$$\langle\ x_1, x_2\ \rangle \in \text{co-BL}_2 \text{ and } x_3 \in \overline{\mathrm{SAT}} \ \Rightarrow$$

$$\mathrm{Prob}_z[\ \pi_{(1,2)}\circ h(I,z) \in \mathrm{BL}_2 \ \text{ or } \ h_3(I,z) \in \mathrm{SAT}\ ] = 1.$$

Now, if condition $\mathcal{C}_3$ holds then just as in the earlier cases, we can design a Merlin-Arthur-Merlin game for $\overline{\mathrm{SAT}}$ formulas of length m:

$\mathcal{C}_3$: $\forall x_3 \in \overline{\mathrm{SAT}}^{=m}$, $\exists x_1, x_2 \in \{0,1\}^m$ such that for $I = \langle\ x_1, x_2, x_3, H_4, H_5\ \rangle$, $\mathrm{Prob}_z[h_3(I,z) \in \mathrm{SAT}] \geq 2/3$.

However, even $\mathcal{C}_3$ may not hold, i.e., it may be the case that $\exists x_3 \in \overline{\mathrm{SAT}}^{=m}$ such that $\forall x_1, x_2$, $\mathrm{Prob}_z[h_3(I,z) \in \mathrm{SAT}] < 2/3$. We then call the least such x_3 as H_3 and assume that H_3, H_4, H_5 are provided as advice and proceed to the next case.

Case 4: Using the properties of H_3 and proposition $\mathcal{P}_2$ of the Probability Recovery Lemma, we can now obtain that: Given any 5-tuple $I = \langle\, x_1, x_2, H_3, H_4, H_5 \,\rangle$, where $x_1, x_2 \in \{0,1\}^m$,

$$\langle\, x_1, x_2 \,\rangle \in \mathrm{BL}_2 \;\Rightarrow\; \mathrm{Prob}_z[\pi_{(1,2)} \circ h(I,z) \in \text{co-BL}_2] \geq 2/3 + \varepsilon,$$

$$\langle\, x_1, x_2 \,\rangle \in \text{co-BL}_2 \;\Rightarrow\; \mathrm{Prob}_z[\pi_{(1,2)} \circ h(I,z) \in \mathrm{BL}_2] \geq 1/3.$$

By unfolding the definitions of BL_2 and co-BL_2 we obtain:

$$x_1 \in \mathrm{SAT} \text{ and } x_2 \in \overline{\mathrm{SAT}} \;\Rightarrow$$
$$\mathrm{Prob}_z[\; h_1(I,z) \in \overline{\mathrm{SAT}} \;\text{ or }\; h_2(I,z) \in \mathrm{SAT}\;] \geq 2/3 + \varepsilon$$

$$x_1 \in \overline{\mathrm{SAT}} \text{ or } x_2 \in \mathrm{SAT} \;\Rightarrow\; \mathrm{Prob}_z[\; h_1(I,z) \in \mathrm{SAT} \;\text{ and }\; h_2(I,z) \in \overline{\mathrm{SAT}}\;] \geq 2/3$$

Now, if condition $\mathcal{C}_4$ holds then just as in the earlier cases, we can design a Merlin-Arthur-Merlin game for $\overline{\mathrm{SAT}}$ formulas of length m:

$\mathcal{C}_4$: $\forall x_2 \in \overline{\mathrm{SAT}}^{=m}$, $\exists x_1 \in \{0,1\}^m$ such that for $I = \langle\, x_1, x_2, H_3, H_4, H_5 \,\rangle$, $\mathrm{Prob}_z[h_2(I,z) \in \mathrm{SAT}] \geq 2/3 + \varepsilon/2$.

However, even $\mathcal{C}_4$ may not hold, i.e., $\exists x_2 \in \overline{\mathrm{SAT}}^{=m}$ such that $\forall x_1$, $\mathrm{Prob}_z[h_2(I,z) \in \mathrm{SAT}] < 2/3 + \varepsilon/2$. We then call the least such x_2 as H_2 and assume that $H_2, \ldots, H_5$ are provided as advice and proceed to the final case.

Case 5: Using the properties of H_2 and proposition $\mathcal{P}_3$ of the Probability Recovery Lemma, we now obtain that: Given any 5-tuple $I = \langle\, x_1, H_2, \ldots, H_5 \,\rangle$, where $x_1 \in \{0,1\}^m$,

$$x_1 \in \mathrm{SAT} \;\Rightarrow\; \mathrm{Prob}_z[h_1(I,z) \in \overline{\mathrm{SAT}}] \geq \varepsilon/2,$$

$$x_1 \in \overline{\mathrm{SAT}} \;\Rightarrow\; \mathrm{Prob}_z[h_1(I,z) \in \mathrm{SAT}] = 1.$$

We now have an Arthur-Merlin game for $\overline{\mathrm{SAT}}^{=m}$, given advice $H_2, \ldots, H_5$ and we are done. □

3.4.2 Proof for a General Case

To illustrate how one proves lower bounds on error probabilities of random reductions in the general cases, we prove the result for $\leq_{\rm m}^{\rm rp}$-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 $1 - 1/k$ $\leq_{\rm m}^{\rm rp}$-reduction. The following theorem shows that the probability bound is almost optimal.

Theorem 3.2 Let L be any language in co-BH($2k$). If $\mathrm{BL}_{2k} \leq_{\rm m}^{\rm rp} L$ with probability $1 - 1/k + 1/p(n)$, for some polynomial p, then the PH collapses to $\Sigma_3^{\rm P}$.

We will need some definitions and lemmas before we can prove this theorem. The hypothesis of Theorem 3.2 implies that there *exists* a probability $1 - 1/k + 1/p(n)$ $\leq_{\rm m}^{\rm rp}$-reduction from BL_{2k} to co-BL_{2k}. However, by applying Lemma 3.5 we obtain that there must exist a probability $1 - 1/k + 1/p(n)$ *nested* $\leq_{\rm m}^{\rm rp}$ reduction from BL_{2k} to co-BL_{2k}.

Notation Henceforth we will refer to this *nested* reduction from BL_{2k} to co-BL_{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 and uniformly chosen string of size $r(q(m))$ and let ε denote $1/p(q(m))$.

To understand the importance of *nested* $\leq_{\rm m}^{\rm rp}$-reductions one needs to look at a rough overview of our proof: As in the example, 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* random 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. However, if we use a *nested* reduction then the structural properties of the reduction allows us to use the **Probability Recovery Lemma** to recover some of the probability lost while going down the hierarchy. At intermediate stages of our proof we deal with induced random 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 3.7 [Probability Recovery Lemma] Suppose $\mathrm{BL}_{2k} \leq_{\mathrm{m}}^{\mathrm{rp}} \text{co-}\mathrm{BL}_{2k}$ via h defined above. Then the following proposition $\mathcal{P}_j$ holds for all j, $0 \leq j \leq 2k-1$:

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

$$\langle\ \vec{y}\ \rangle \in \text{co-}\mathrm{BL}_\ell \ \Rightarrow\ \mathrm{Prob}_z[\pi_{(1,\ell)} \circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z) \in \mathrm{BL}_\ell] = 1.$$

If j is even:

$$\langle\ \vec{y}\ \rangle \in \mathrm{BL}_\ell \ \Rightarrow\ \mathrm{Prob}_z[\pi_{(1,\ell)} \circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z) \in \text{co-}\mathrm{BL}_\ell] \geq 1 - \frac{1}{k} + \varepsilon.$$

Proof: (by induction on j)

<u>Base cases $\mathcal{P}_0$ & $\mathcal{P}_1$:</u> These follow directly from the hypothesis of Lemma 3.7 and the definitions of BL_{2k} and $\text{co-}\mathrm{BL}_{2k}$.

<u>Assume $\mathcal{P}_j$, Induction case $\mathcal{P}_{j+2}$:</u> We treat the odd and even cases separately.

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

$$\langle\ \vec{y}\ \rangle \in \text{co-BL}_\ell \ \Rightarrow\ \mathrm{Prob}_z[\pi_{(1,\ell)} \circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z) \in \mathrm{BL}_\ell] = 1.$$

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

$$\langle\ \vec{y},\ x_{j+2},\ x_{j+1}\ \rangle \in \text{co-BL}_\ell \ \Rightarrow\ \mathrm{Prob}_z[\pi_{(1,\ell)} \circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z) \in \mathrm{BL}_\ell] = 1.$$

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

$$\langle\ \vec{y},\ x_{j+2},\ x_{j+1}\ \rangle \in \text{co-BL}_\ell \iff \langle\ \vec{y}\ \rangle \in \text{co-BL}_{\ell-2}$$

and thus we have that

$$\langle\ \vec{y}\ \rangle \in \text{co-BL}_{\ell-2} \Rightarrow \mathrm{Prob}_z[\pi_{(1,\ell)} \circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z) \in \mathrm{BL}_\ell] = 1. \tag{3.1}$$

We claim that

$$\pi_{(1,\ell)} \circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z) \in \mathrm{BL}_\ell \ \Rightarrow\ \pi_{(1,\ell-2)} \circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z) \in \mathrm{BL}_{\ell-2}. \tag{3.2}$$

Proof: Let us denote the ℓ-tuple $\pi_{(1,\ell)} \circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z)$ as $\langle\ h_1, \ldots, h_\ell\ \rangle$. To prove equation 3.2, let us assume that $\langle\ h_1, \ldots, h_\ell\ \rangle \in \mathrm{BL}_\ell$. We show that this means that $\langle\ h_1, \ldots, h_{\ell-2}\ \rangle \in \mathrm{BL}_{\ell-2}$. By the definition of BL$_\ell$ for odd ℓ we know that since $\langle\ h_1, \ldots, h_\ell\ \rangle \in \mathrm{BL}_\ell$,

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

Now if $h_\ell \in \overline{\mathrm{SAT}}$ then it must be the case that

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

which implies $\langle\ h_1, \ldots, h_{\ell-2}\ \rangle \in \mathrm{BL}_{\ell-2}$. 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 $j+2 < 2k \Rightarrow (\ell-2) \geq 1$. If $(\ell-2) = 1$ then that means that

$$\langle\ h_1, \ldots, h_{\ell-2}\ \rangle = \langle\ h_1\ \rangle \in \mathrm{BL}_{\ell-2} = \mathrm{BL}_1,$$

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

$$\langle\ h_1, \ldots, h_{\ell-2}\ \rangle \in \mathrm{BL}_{\ell-2} \iff \langle\ h_1, \ldots, h_{\ell-3}\ \rangle \in \mathrm{BL}_{\ell-3} \text{ or } h_{\ell-2} \in \mathrm{SAT}$$

and since $h_{\ell-2} \in \mathrm{SAT}$, $\langle\ h_1, \ldots, h_{\ell-2}\ \rangle$ must be in $\mathrm{BL}_{\ell-2}$.

This proves equation 3.2. Therefore, by equations 3.1 and 3.2 we establish $\mathcal{P}_{j+2}$ for odd j, i.e.,

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

$$\langle\ \vec{y}\ \rangle \in \text{co-BL}_{\ell-2} \Rightarrow \mathrm{Prob}_z[\pi_{(1,\ell-2)} \circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z) \in \mathrm{BL}_{\ell-2}] = 1.$$

Case 2: j is even. The proof of this case is very similar to that of Case 1, and hence omitted.

Thus, we have proved the lemma by induction. □

We now use a variant of the hard/easy proof technique [Kad88] to prove the main theorem. We first define a *hard* sequence of formulas.

Definition Suppose $\mathrm{BL}_{2k} \leq^{\mathrm{rp}}_{\mathrm{m}} \text{co-BL}_{2k}$ via h. Then, we call $\langle\ 1^m, x_j, \ldots, x_1\ \rangle = \langle\ 1^m, \vec{x}\ \rangle$ 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. $\langle\ 1^m, x_{j-1}, \ldots, x_1\ \rangle$ is a hard sequence with respect to h.

5. For all $\vec{y} = y_1, \ldots, y_\ell \in \{0,1\}^{m \times \ell}$ (where $\ell = 2k - j$),

$$\text{Prob}_z[\ \pi_{\ell+1} \circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z) \in \text{SAT}\ ]\ \leq\ \frac{\lfloor j/2 \rfloor}{k}\ +\ (j \bmod 2) * \frac{\varepsilon}{2}.$$

If $\langle\ 1^m, x_j, \ldots, x_1\ \rangle$ 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* $\leq_{\text{m}}^{\text{rp}}$-reduction from BL_{2k} to co-BL_{2k}, a hard sequence of order j for length m induces an asymmetric probabilistic reduction from BL_{2k-j} to co-BL_{2k-j} for tuples of strings of length m.

Lemma 3.8 Suppose $\text{BL}_{2k} \leq_{\text{m}}^{\text{rp}}$ co-BL_{2k} via h. Then, the following proposition $\mathcal{Q}_j$ holds for all j, $0 \leq j \leq 2k - 1$:

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

$$\langle\ \vec{y}\ \rangle \in \text{BL}_\ell\ \Rightarrow\ \text{Prob}_z[\ \pi_{(1,\ell)} \circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z) \in \text{co-BL}_\ell\ ] \geq 1 - \tfrac{1}{k} + \varepsilon,$$

$$\langle\ \vec{y}\ \rangle \in \text{co-BL}_\ell\ \Rightarrow\ \text{Prob}_z[\ \pi_{(1,\ell)} \circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z) \in \text{BL}_\ell\ ] \geq 1 - \tfrac{j}{2k}.$$

If ℓ is odd:

$$\langle\ \vec{y}\ \rangle \in \text{co-BL}_\ell\ \Rightarrow\ \text{Prob}_z[\ \pi_{(1,\ell)} \circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z) \in \text{BL}_\ell\ ] = 1,$$

$$\langle\ \vec{y}\ \rangle \in \text{BL}_\ell\ \Rightarrow\ \text{Prob}_z[\ \pi_{(1,\ell)} \circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z) \in \text{co-BL}_\ell\ ] \geq 1 - \tfrac{j+1}{2k} + \tfrac{\varepsilon}{2}.$$

Proof: (by induction on j)

Base case $\mathcal{Q}_0$: This follows trivially from the hypothesis of the lemma.

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

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

$$\langle\ \vec{y}\ \rangle \in \mathrm{BL}_\ell \ \Rightarrow\ \mathrm{Prob}_z[\ \pi_{(1,\ell)}\circ h(\langle\ \vec{y},\ \vec{x}\ \rangle, z) \in \text{co-BL}_\ell\ ] \geq 1 - \frac{1}{k} + \varepsilon.$$

In particular, for $y_\ell = x_{j+1}$ and $\vec{y'} = y_1, \ldots, y_{\ell-1}$ we have

$$\langle\ \vec{y'},\ x_{j+1}\ \rangle \in \mathrm{BL}_\ell \ \Rightarrow\ \mathrm{Prob}_z[\pi_{(1,\ell)}\circ h(\langle\ \vec{y'},\ \vec{x'}\ \rangle, z) \in \text{co-BL}_\ell] \geq 1 - \frac{1}{k} + \varepsilon.$$

Using the definitions of BL_ℓ and co-BL_ℓ for even ℓ, we have that for all $\vec{y'} = y_1, \ldots, y_{\ell-1} \in \{0,1\}^{m\times(\ell-1)}$,

$$\langle\ \vec{y'}\ \rangle \in \mathrm{BL}_{\ell-1} \text{ and } x_{j+1} \in \overline{\mathrm{SAT}} \ \Rightarrow \tag{3.3}$$

$$\mathrm{Prob}_z[\ \pi_{(1,\ell-1)}\circ h(\langle\ \vec{y'},\ \vec{x'}\ \rangle, z) \in \text{co-BL}_{\ell-1} \text{ or } \pi_\ell\circ h(\langle\ \vec{y'},\ \vec{x'}\ \rangle, z) \in \mathrm{SAT}\ ] \geq 1 - \frac{1}{k} + \varepsilon.$$

Since $\langle\ 1^m, x_{j+1}, \ldots, x_1\ \rangle$ is a hard sequence, we know conditions 1 and 5 of the definition hold. That is, $x_{j+1} \in \overline{\mathrm{SAT}}$ and for all $\vec{y'} = y_1, \ldots, y_{\ell-1} \in \{0,1\}^{m\times(\ell-1)}$,

$$\mathrm{Prob}_z[\ \pi_\ell\circ h(\langle\ \vec{y'},\ \vec{x'}\ \rangle, z) \in \mathrm{SAT}\ ] \leq \frac{\lfloor (j+1)/2 \rfloor}{k} + \frac{\varepsilon}{2} \tag{3.4}$$

(since j is even, $(j+1) \bmod 2$ is 1).

Now we can establish Q_{j+1}. If $\langle\ y_1, \ldots, y_{\ell-1}\ \rangle \in \mathrm{BL}_{\ell-1}$, then by equations 3.3, 3.4 and the fact that $x_{j+1} \in \overline{\mathrm{SAT}}$, we have

$$\begin{aligned} &\mathrm{Prob}_z[\ \pi_{(1,\ell-1)}\circ h(\langle\ \vec{y'},\ \vec{x'}\ \rangle, z) \in \text{co-BL}_{\ell-1}\ ] \\ &\qquad \geq 1 - \frac{1}{k} + \varepsilon - \frac{\lfloor (j+1)/2 \rfloor}{k} - \frac{\varepsilon}{2} = 1 - \frac{j+2}{2k} + \frac{\varepsilon}{2} \end{aligned}$$

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

If on the other hand, $\langle\ y_1, \ldots, y_{\ell-1}\ \rangle \in \text{co-BL}_{\ell-1}$ then by conditions 3 and 4 in the definition of a *hard* sequence, we know that $x_1, \ldots, x_{j+1} \in \overline{\mathrm{SAT}}$. Since h is a *nested* $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction, by a direct application of the Probability Recovery Lemma we know that in this case

$$\mathrm{Prob}_z[\ \pi_{(1,\ell-1)}\circ h(\langle\ \vec{y'},\ \vec{x'}\ \rangle, z) \in \mathrm{BL}_{\ell-1}\ ] = 1.$$

Thus, we have proved $\mathcal{Q}_{j+1}$ for the case when $\ell = 2k - j$ is even.

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

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

Lemma 3.9 Suppose $\mathrm{BL}_{2k} \leq_{\mathrm{m}}^{\mathrm{rp}} \text{co-}\mathrm{BL}_{2k}$ via h. Let $\langle\, 1^m, x_j, \ldots, x_1 \,\rangle = \langle\, 1^m, \vec{x} \,\rangle$ be a maximal hard sequence with respect to h. Define $\ell = 2k - j$ and let $\vec{y'} = y_1, \ldots, y_{\ell-1}$ denote an arbitrary $(\ell-1)$-tuple of strings of size m. Then,

$$\begin{aligned} y \;\in\; \overline{\mathrm{SAT}} \;\Rightarrow\; & \exists\, \vec{y'}, \\ & \mathrm{Prob}_z[\; \pi_\ell \circ h(\langle\, \vec{y'}, y, \vec{x} \,\rangle, z) \in \mathrm{SAT} \;] \;\geq\; \frac{\lfloor (j+1)/2 \rfloor}{k} + ((j+1) \bmod 2) * \frac{\varepsilon}{2} \end{aligned}$$

and

$$\begin{aligned} y \;\in\; \mathrm{SAT} \;\Rightarrow\; & \forall\, \vec{y'}, \\ & \mathrm{Prob}_z[\; \pi_\ell \circ h(\langle\, \vec{y'},\; y,\; \vec{x} \,\rangle, z) \in \mathrm{SAT} \;] \;\leq\; \frac{\lfloor (j+1)/2 \rfloor}{k} - (j \bmod 2) * \frac{\varepsilon}{2}. \end{aligned}$$

Proof: If $j = 2k - 1$ ($\vec{y'}$ is the empty sequence), then, by Lemma 3.8, for all $y \in \{0,1\}^m$

$$y \in \overline{\mathrm{SAT}} \;\Rightarrow\; \mathrm{Prob}_z[\; \pi_1 \circ h(\langle\, y,\; \vec{x} \,\rangle, z) \in \mathrm{SAT} \;] = 1$$

and

$$y \in \mathrm{SAT} \;\Rightarrow\; \mathrm{Prob}_z[\; \pi_1 \circ h(\langle\, y,\; \vec{x} \,\rangle, z) \in \overline{\mathrm{SAT}} \;] \geq \frac{\varepsilon}{2}.$$

Thus, the lemma holds when $j = 2k - 1$. Now consider the case when $j < 2k - 1$. Let $\ell = 2k - j$.

Suppose $y \in \overline{\mathrm{SAT}}$. Since $\langle\, 1^m, x_j, \ldots, x_1 \,\rangle$ is maximal, $\langle\, 1^m, y, x_j, \ldots, x_1 \,\rangle$ is not a hard sequence. However, $j + 1 \leq 2k - 1$, $|y| = m$, $y \in \overline{\mathrm{SAT}}$ and

$\langle 1^m, x_j, \ldots, x_1 \rangle$ is a hard sequence. So, $\langle 1^m, y, x_j, \ldots, x_1 \rangle$ must fail to be a hard sequence by failing to satisfy condition 5 of the definition of hard sequences. Thus, we get the required result.

$$\exists \quad \vec{y'} \in \{0,1\}^{m\times(\ell-1)},$$
$$\mathrm{Prob}_z[\ \pi_\ell \circ h(\langle\ \vec{y'}, y, \vec{x}\ \rangle, z) \in \mathrm{SAT}\ ] > \frac{\lfloor (j+1)/2 \rfloor}{k} + ((j+1) \bmod 2) * \frac{\varepsilon}{2}.$$

If on the other hand, $y \in$ SAT then by by Lemma 3.8 we know that for all $\vec{y'} = y_1, \ldots, y_{\ell-1} \in \{0,1\}^{m\times\ell-1}$:
If ℓ is even:

$$\langle\ \vec{y'}, y\ \rangle \in \text{co-BL}_\ell \ \Rightarrow\ \mathrm{Prob}_z[\ \pi_{(1,\ell)} \circ h(\langle\ \vec{y'},\ y,\ \vec{x}\ \rangle, z) \in \mathrm{BL}_\ell\ ] \geq 1 - \frac{j}{2k}$$

and if ℓ is odd:

$$\langle\ \vec{y'}, y\ \rangle \in \mathrm{BL}_\ell \ \Rightarrow\ \mathrm{Prob}_z[\ \pi_{(1,\ell)} \circ h(\langle\ \vec{y'},\ y,\ \vec{x}\ \rangle, z) \in \text{co-BL}_\ell\ ] \geq 1 - \frac{j+1}{2k} + \frac{\varepsilon}{2}.$$

Now let us consider the even and odd cases separately. If ℓ is even, then by the definition of co-BL_ℓ, we know that $y \in$ SAT implies that for all $\vec{y'} = y_1, \ldots, y_{\ell-1}$, $\langle\ \vec{y'}, y\ \rangle \in$ co-BL_ℓ. Therefore, we have that

$$\forall\ \vec{y'}\ \mathrm{Prob}_z[\pi_{(1,\ell)} \circ h(\langle\ \vec{y'},\ y,\ \vec{x}\ \rangle, z) \in \mathrm{BL}_\ell] \geq 1 - \frac{j}{2k}.$$

By the definition of BL_ℓ for even ℓ, we know that $\langle\ u_1, \ldots, u_\ell\ \rangle \in \mathrm{BL}_\ell \Rightarrow u_\ell \in \overline{\mathrm{SAT}}$. Thus, we get the desired result

$$\forall\ \vec{y'},\ \mathrm{Prob}_z[\pi_\ell \circ h(\langle\ \vec{y'},\ y,\ \vec{x}\ \rangle, z) \in \overline{\mathrm{SAT}}\ ] \geq 1 - \frac{j}{2k},$$

i.e.,

$$\forall\ \vec{y'},\ \mathrm{Prob}_z[\pi_\ell \circ h(\langle\ \vec{y'},\ y,\ \vec{x}\ \rangle, z) \in \mathrm{SAT}\ ]\ \leq\ \frac{j}{2k}\ =\ \frac{\lfloor (j+1)/2 \rfloor}{k} + (j \bmod 2) * \frac{\varepsilon}{2}.$$

If ℓ is odd then using a very similar reasoning as in the even case we can get the desired result, i.e.,

$$\forall\ \vec{y'}\ \mathrm{Prob}_z[\ \pi_\ell \circ h(\langle\ \vec{y'},\ y,\ \vec{x}\ \rangle, z) \in \mathrm{SAT}\ ] \leq \frac{j+1}{2k} + \frac{\varepsilon}{2}\ =\ \frac{\lfloor (j+1)/2 \rfloor}{k} + (j \bmod 2) * \frac{\varepsilon}{2}.$$

This completes the proof of the Lemma. □

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

Theorem 3.2 Let L be any language in co-BH($2k$). If $\mathrm{BL}_{2k} \leq_{\mathrm{m}}^{\mathrm{rp}} L$ with probability $1 - 1/k + 1/p(n)$, for some polynomial p, then the PH collapses to Σ_3^{P}.

Proof: By Lemma 3.5 we know that $\mathrm{BL}_{2k} \leq_{\mathrm{m}}^{\mathrm{rp}}$ co-BL_{2k} with probability $1 - 1/k + 1/p(n)$ via the *nested* $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction h. Thus, Lemma 3.9 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 $\langle\, F, a, y, z \,\rangle$ does the following. Suppose a is of the form $\langle\, 1^m, x_j, \ldots, x_1 \,\rangle$ 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 3.9,

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

and

$$F \in \mathrm{SAT} \Rightarrow \forall y,\ \mathrm{Prob}_z[\, N \text{ accepts } \langle\, F, a, y, z \,\rangle \,] \leq \frac{\lfloor (j+1)/2 \rfloor}{k} - (j \bmod 2) * \frac{\varepsilon}{2}.$$

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,Cha91]. □

3.4.3 Other cases

In the previous section we saw a complete proof of the optimality of the probability $1 - 1/k$ $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction from BL_{2k} to co-BL_{2k}. With minor modifications, the same proof technique can also be used to establish the optimality of the $\leq_{\mathrm{m}}^{\mathrm{rp}}$ and $\leq_{\mathrm{m}}^{\mathrm{bpp}}$

reductions between complementary classes in the Boolean Hierarchy. We therefore state these theorems without proof.

Theorem 3.3 Let L be any language in BH($2k$). If co-$\mathrm{BL}_{2k} \leq_{\mathrm{m}}^{\mathrm{rp}} L$ with probability $1 - 1/(k+1) + 1/p(n)$, for some polynomial p, then the PH collapses to Σ_3^{P}.

Theorem 3.4 Let L be any language in co-BH($2k+1$). If $\mathrm{BL}_{2k+1} \leq_{\mathrm{m}}^{\mathrm{rp}} L$ with probability $1 - 1/(k+1) + 1/p(n)$, for some polynomial p, then the PH collapses to Σ_3^{P}.

Theorem 3.5 Let L be any language in BH($2k+1$). If co-$\mathrm{BL}_{2k+1} \leq_{\mathrm{m}}^{\mathrm{rp}} L$ with probability $1 - 1/(k+1) + 1/p(n)$, for some polynomial p, then the PH collapses to Σ_3^{P}.

Theorem 3.6 Let L be any language in BH(k). If co-$\mathrm{BL}_k \leq_{\mathrm{m}}^{\mathrm{bpp}} L$ with probability $1 - 1/(k+1) + 1/p(n)$, for some polynomial p, then the PH collapses to Σ_3^{P}.

The optimality of all other reductions in Figure 3.2 follow as corollaries to these basic theorems on the optimality of the random reductions between complementary classes in the Boolean Hierarchy. For example, the probability $1 - 1/(k+1)$ $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction from $\mathrm{BL}_{2k+1} \oplus$co-$\mathrm{BL}_{2k+1}$ to $\mathrm{BL}_{2k} \oplus$co-BL_{2k} must be almost optimal (i.e., its success probability cannot be improved by even $1/poly$) otherwise we would get a probability $1 - 1/(k+1) + 1/poly$ $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction from BL_{2k+1} to co-BL_{2k+1} and that would cause the PH to collapse according to Theorem 3.4.

3.5 Randomization and Bounded Query Functions

In view of the results presented earlier, a natural question that arises is whether similar results also hold for *functions* computable using bounded queries to SAT. In this section we address this issue and provide strong evidence that randomness is not very useful in saving queries in this case.

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

Definition [AG88,Bei91] 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, 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 Turing machine 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 there is a function f such that 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 to SAT. Note that k need not be a constant, it could be any function of the input.

Lemma 3.10 Any function f in $\mathrm{PF}^{\mathrm{SAT}}_{k-T}$ ($\mathrm{PF}^{\mathrm{SAT}}_{k-tt}$) is also in the class $\mathrm{RPF}^{\mathrm{SAT}}_{(k-1)-T}[1/2 + 1/2^{p(n)}]$ (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 an inverse exponential 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. $\square$

The next theorem provides strong evidence that this is the best that randomness can achieve.

Theorem 3.7 Let g be an increasing integer valued function such that $g(n) = O(\log n)$ 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

$$\mathrm{F}^{\mathrm{A}}_{k}(x_1, \ldots, x_k) = \mathrm{A}(x_1).\mathrm{A}(x_2)\ldots\mathrm{A}(x_k)$$

where ... denotes the concatenation of bits and $\mathrm{A}(x)$ is the characteristic function of A, i.e., $\mathrm{A}(x) = 1$ if $x \in \mathrm{A}$, 0 otherwise.

Our proof of Theorem 3.7 uses the following version of a theorem proved in [ABG90].

Lemma 3.11 [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 $A \in \mathrm{NP}/poly \cap$ co-NP$/poly$.

We now proceed with the proof of Theorem 3.7 . Suppose, for some $g(n) = \mathrm{O}(\log n)$, 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 $pad(x_1, x_2, \ldots, x_{h(m)})$ denote a padded collection of $h(m)$ formulas $\langle x_1, \ldots, x_{h(m)} \rangle$ of size m each, such that the size of the padded collection is exactly m^2. Define a function $\mathcal{T}$, which on input of the form $pad(x_1, x_2, \ldots, x_{h(m)})$ 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 $I = pad(x_1, x_2, \ldots, x_{h(m)})$ of size $n = m^2$, 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 $h(m) = \mathrm{O}(\log n)$, it is possible to completely explore the entire query/computational tree of M on I and random string r in polynomial time. Since there are $h(m) - 1$ oracle queries in this tree, it has $2^{h(m)-1}$ leaves and the value output by M on I and r 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 [Sch89], 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 $\mathcal{Z}$, which outputs $R_1, \ldots, R_{c(n)}$ on input 0^n.

On input $J = \text{pad}(x_1,\ x_2,\ \ldots\ ,\ x_{h(m)})$ of size n and given advice $\mathcal{Z}(0^n) = R_1, \ldots, R_{c(n)}$, 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

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

but that cannot happen because 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 3.11. 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,

$w(c) \neq \mathrm{F}^{\mathrm{SAT}}_{h(m)}(x_1, x_2, \ldots, x_{h(m)})$. Therefore, from the lemma it follows that SAT $\in$ NP/*poly* $\cap$ co-NP/*poly* and PH collapses to Σ_3^{P} [Yap83].

3.6 Open Problems

In this chapter, we obtained tight performance bounds on the success probabilities of random reductions between nearby classes in the Boolean and Query Hierarchies. For example, we have tight bounds for random reductions which save one or two nonadaptive queries to SAT. However we do not have tight bounds for reductions between classes which are further apart. For example, we can construct a probability $1/3$ $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction from BL_6 (a union of 3 $\mathrm{D^P}$ languages) to BL_2 (the complete language for $\mathrm{D^P}$) but we cannot prove that it is optimal. One upper bound for this case is $1 - 1/\sqrt{3} + 1/poly$ ($\approx 0.42265 + 1/poly$) because by taking the OR of two trials of a reduction with this success probability one could obtain an $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduction from BL_6 to BL_4 with success probability above $2/3 + 1/poly$ and collapse the PH. We conjecture that tight performance bounds exist for such reductions as well. Our proof technique is based on the hard/easy argument which is very suitable for analyzing reductions between complementary classes at the same level of the Boolean Hierarchy but does not seem to be effective for analyzing random reductions between classes which are far apart. More powerful proof techniques may be needed to resolve such questions.

Chapter 4

Threshold Behavior [1]

Traditionally, researchers defined random reductions without giving much consideration to the success probability that the reductions had to achieve. Typically, reductions are required to behave correctly with some constant probability but sometimes the success probability is as low as inverse polynomial. In the previous chapter however, we saw several results in which the exact value of the success probability played a crucial role. For instance, we saw that, unlike deterministic $\leq_{\mathrm{m}}^{\mathrm{P}}$-reductions, random reductions were able to reduce *harder* languages in the Boolean and Query hierarchies to *simpler* languages with varying degree of success. We also saw that there were limits on the success probability of such reductions depending on the type of reduction and the classes involved.

In this chapter, we first show that the traditional approach that does place much emphasis on the success probability of random reductions, results in several anomalies especially when dealing with the notion of completeness under random reductions. As an example, we will consider USAT (the language of uniquely satisfiable formulas) and show that traditional approach provides ambiguous and misleading information about its complexity.

We then examine random reductions under a new light and propose that there

[1]The results presented in this chapter have been drawn from [CKR91] and [Roh92]

is natural and "correct" definition for the success probability — especially when one considering *completeness* under random reductions. As we saw in Chapter 3, the power of randomness allows us to reduce harder languages to simpler ones with certain probability of success. Thus, if we set the correctness probability to be very low then much simpler sets can be complete for a class under random reductions. Conversely, if the probability of correctness is required to be very high, then as we saw in Chapter 3, due to the limitations of randomness, such reductions could not substantially reduce the complexity of a problem. In fact, in the limiting case if the success probability is made to be 1, then the class of sets complete under random reductions is identical to the class of $\leq_m^P$-complete sets. As expected, sets complete under extremely high probability random reductions cannot be substantially simpler than the $\leq_m^P$-complete sets. We propose the thesis that as we vary the success probability, at some exact point, the probability of correctness is just high enough to make the definition right. We call this point the *threshold* and we believe that it is a fundamental attribute of every complexity class which precisely characterizes the tradeoff involved in the use of randomness to reduce the complexity of problems within it, i.e., it quantifies the minimal amount of error that must be introduced in order to reduce the complexity of the hardest problems in the class using randomness. We then defend this thesis by proving the existence of such thresholds for several complexity classes such as NP, co-NP, D^P, co-$D^P$and the Boolean and Query hierarchies.

4.1 USAT and Random Reductions

From the beginning, the study of the complexity of USAT has been tied to the class D^P and to random reductions. The class D^P was defined by Papadimitriou and Yannakakis [PY84] to characterize the complexity of the facets of polytopes and the complexity of optimization problems such as CLIQUE(1) (defined in chapter 3).

Definition: We define D^P, co-$D^P$and their $\leq_m^P$-complete languages $\mathrm{SAT}\wedge\overline{\mathrm{SAT}}$

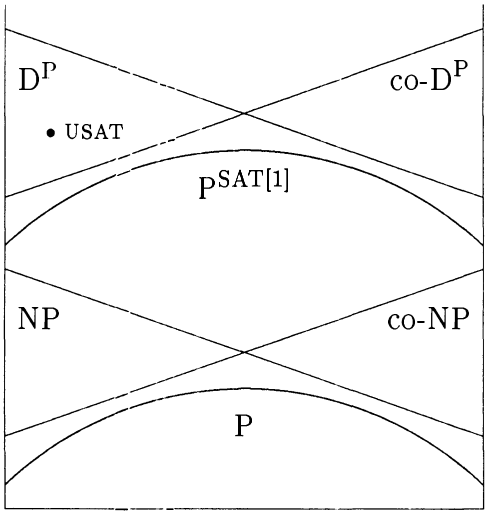


Figure 4.1: USAT and related complexity classes.

and $\overline{\mathrm{SAT}} \vee \mathrm{SAT}$.

$$\begin{aligned} \mathrm{D^P} &= \{\ L_1 \cap \overline{L_2} \mid L_1, L_2 \in \mathrm{NP}\ \}, \\ \text{co-}\mathrm{D^P} &= \{\ \overline{L_1} \cup L_2 \mid L_1, L_2 \in \mathrm{NP}\ \}, \end{aligned}$$

$$\mathrm{SAT} \wedge \overline{\mathrm{SAT}} = \{\ \langle\ F_1, F_2\ \rangle \mid F_1 \in \mathrm{SAT} \text{ and } F_2 \in \overline{\mathrm{SAT}}\ \},$$

$$\overline{\mathrm{SAT}} \vee \mathrm{SAT} = \{\ \langle\ F_1, F_2\ \rangle \mid F_1 \in \overline{\mathrm{SAT}} \text{ or } F_2 \in \mathrm{SAT}\ \}.$$

The set of uniquely satisfiable Boolean formulas, USAT, is easily seen to be in $\mathrm{D^P}$. So, the natural question that arises is: Is USAT $\mathrm{D^P}$-compete ? Blass and Gurevich [BG82] answered this question partially. They noticed that

$$\begin{aligned} \text{USAT is } \leq_m^P\text{-complete for } \mathrm{D^P} &\iff \mathrm{SAT} \wedge \overline{\mathrm{SAT}} \leq_m^P \mathrm{USAT} \\ &\iff \mathrm{SAT} \leq_m^P \mathrm{USAT}. \end{aligned}$$

So, the question of whether USAT can be $\leq_m^P$-complete for $\mathrm{D^P}$ hinges on whether there is a $\leq_m^P$-reduction from SAT to USAT. Then, they showed that there are oracle worlds where no such reduction can exist. This meant a non-relativizing proof technique would be needed to answer the question—a fairly difficult obstacle, indeed.

Valiant and Vazirani [VV86] did not surmount this obstacle, but they did manage to circumvent it. They were able to construct a *random reduction* from SAT to USAT. More precisely, they showed that SAT $\leq_{\mathrm{m}}^{\mathrm{rp}}$ USAT with probability $1/(4n)$ (as a special case we will henceforth write $A \leq_{\mathrm{m}}^{\mathrm{vv}} B$ if $A \leq_{\mathrm{m}}^{\mathrm{rp}} B$ with probability $1/p(n)$ for some polynomial p). Thus, based on the observation of Blass and Gurevich, USAT became complete for $\mathrm{D^P}$ under random reductions[2]. However, this type of reduction is not quite satisfying, because the probability of the reduction being correct approaches zero as the length increases. One would have expected a probability bound of 1/2 (in keeping with the Adleman-Manders [AM77] definition). The justification for the Valiant-Vazirani definition is that in many situations the probability bound can be amplified, in which case, the definitions would be equivalent.

Definition: For any language B, we define the following classes.

$$\begin{aligned}
\mathrm{OR}_2(B) &= \{\ \langle\ x, y\ \rangle \mid x \in B \text{ or } y \in B\ \} \\
\mathrm{OR}_\omega(B) &= \{\ \langle\ x_1, \ldots, x_n\ \rangle \mid \text{for some } i, 1 \leq i \leq n,\ x_i \in B\ \} \\
\mathrm{AND}_2(B) &= \{\ \langle\ x, y\ \rangle \mid x \in B \text{ and } y \in B\ \} \\
\mathrm{AND}_\omega(B) &= \{\ \langle\ x_1, \ldots, x_n\ \rangle \mid \text{for all } i, 1 \leq i \leq n,\ x_i \in B\ \}.
\end{aligned}$$

If $\mathrm{OR}_2(B) \leq_{\mathrm{m}}^{\mathrm{P}} B$ via some polynomial-time function, then we say that B has an OR_2 function, or simply that B has OR_2. We use the same terminology for OR_ω, AND_2 and AND_ω. If a language B has OR_ω, then it is possible to amplify the probability bound of a $\leq_{\mathrm{m}}^{\mathrm{vv}}$-reduction to B. Thus, when B has OR_ω, $\leq_{\mathrm{m}}^{\mathrm{vv}}$-reductions and $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions with high probability are equivalent.

Fact 4.1 If $A \leq_{\mathrm{m}}^{\mathrm{vv}} B$ and B has OR_ω, then $A \leq_{\mathrm{m}}^{\mathrm{rp}} B$ with probability $1 - 1/exp$.

[2] Valiant and Vazirani credit Alan Selman for this application of their reduction.

So, in cases where one randomly reduces to a language which has OR_ω, it does not matter which definition is used. Languages such as SAT and $\overline{\mathrm{SAT}}$ are robust and have both OR_ω and AND_ω. Also, the complete languages for complexity classes such as Σ_k^{P}, Π_k^{P}, PP, $\oplus$P, MOD_kP and PSPACE all have both OR_ω and AND_ω. Indeed, it is difficult to find natural languages which do not have OR_ω or AND_ω. However, there are some good reasons to believe that USAT does not have OR_ω (see Corollary 4.3). Thus, there is no obvious way to amplify the Valiant-Vazirani reduction from SAT to USAT. In the next section, we investigate some anomalies created by the non-robustness of USAT.

4.2 Anomalous Behavior

The first and most obvious problem with the statement "USAT is complete for $\mathrm{D^P}$ under random reductions" is that it fails to consider that USAT can be complete for larger classes as well. In fact, a simple observation will show that USAT is $\leq_{\mathrm{m}}^{\mathrm{vv}}$-complete for a seemingly much larger class, $\mathrm{P}^{\mathrm{SAT}[\log n]}$. $\mathrm{P}^{\mathrm{SAT}[\log n]}$ is the class of languages accepted by polynomial-time Turing machines which ask at most $O(\log n)$ queries to the SAT oracle. Introduced by Papadimitriou and Zachos [PZ83], $\mathrm{P}^{\mathrm{SAT}[\log n]}$ captures problems such as Sat-Mod-k, Clique-Mod-k, Unique Optimal Clause Satisfiability, Unique Optimal Clique and other problems related to optimal solution sizes of many NP optimization problems. [Kad89,Kre88].

Lemma 4.1 USAT is $\leq_{\mathrm{m}}^{\mathrm{vv}}$-complete for $\mathrm{P}^{\mathrm{SAT}[\log n]}$.

Proof: Using the fact that $\mathrm{OR}_\omega(\mathrm{SAT}\wedge\overline{\mathrm{SAT}})$ is $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete for $\mathrm{P}^{\mathrm{SAT}[\log n]}$, observe that there is a trivial $\leq_{\mathrm{m}}^{\mathrm{vv}}$-reduction from $\mathrm{OR}_\omega(\mathrm{SAT}\wedge\overline{\mathrm{SAT}})$ to $\mathrm{SAT}\wedge\overline{\mathrm{SAT}}$. On input $\langle\, x_1, \ldots, x_n \,\rangle$, the reduction chooses $1 \leq i \leq n$ at random and prints out x_i. This random reduction will succeed with probability $1/n$. So, combined with the Valiant-Vazirani reduction, we get a $\leq_{\mathrm{m}}^{\mathrm{vv}}$-reduction from $\mathrm{OR}_\omega(\mathrm{SAT}\wedge\overline{\mathrm{SAT}})$ to USAT with probability $1/(4n^2)$. $\square$

There is compelling evidence that the classes $\mathrm{P}^{\mathrm{SAT}[\log n]}$ and D^{P} have very different structural properties. For example, $\mathrm{P}^{\mathrm{SAT}[\log n]}$ is closed under complementation, but D^{P} cannot equal co-D^{P} unless PH collapses [CK90a,Kad88]. Also, all $\mathrm{P}^{\mathrm{SAT}[\log n]}$ $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete languages have OR_ω, but D^{P} $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete languages cannot have OR_2 unless PH collapses [CK90b]. Since we expect complete sets to inherit structural properties of the classes they represent, USAT being $\leq_{\mathrm{m}}^{\mathrm{vv}}$-complete for both these classes raises doubts as to whether completeness under $\leq_{\mathrm{m}}^{\mathrm{vv}}$ reductions makes sense. Lemma 4.1 can also give misleading results about the complexity of certain optimization problems. For instance, it implies that all the Unique Optimization problems in $\mathrm{P}^{\mathrm{SAT}[\log n]}$ can be reduced to Unique Satisfiability. At first glance, this appears to say that we can solve optimization problems using randomness without having to do any optimization. However, the probability of a correct reduction taking place is only $1/(4n^2)$. Moreover, there is no known way to improve this probability bound (see Corollary 4.3). In fact, this lemma really demonstrates the anomalies created by random reductions that allow very low probability bounds.

One might hope that these anomalies would disappear if we used only the Adleman-Manders definition of random reductions. However, in the next section, we will show that even if we restrict our attention to $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions with probability $1/2$ or any other constant, such anomalies will still occur. First, we must explain what *threshold behavior* means.

4.3 Threshold Behavior

Considering the anomalous behavior of random reductions described above, one may be tempted to dismiss completeness under random reductions as meaningless. However, the $\leq_{\mathrm{m}}^{\mathrm{vv}}$-reduction from SAT to USAT proved to be useful in many areas of research. For example, Richard Beigel [Bei88] used it to show that SAT is superterse unless RP = NP. Also, Toda [Tod89] used a similar reduction in his proof that $\mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$. This result, in turn, led to the Lund, Fortnow, Karloff and Nisan [LFKN90] result: $\mathrm{PH} \subseteq \mathrm{IP}$. So, there should be little doubt in the reader's mind regarding the usefulness of the Valiant-Vazirani reduction. The more pertinent questions are: What does this reduction mean? Does USAT being $\leq_{\mathrm{m}}^{\mathrm{vv}}$-complete for D^{P} mean that it is somehow representative of the whole class? How does the complexity of $\leq_{\mathrm{m}}^{\mathrm{vv}}$-complete sets compare with the $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete languages? We answer these questions in terms of *thresholds* of probability bounds.

4.3.1 Examples of Threshold Behavior

To illustrate what we mean by threshold behavior, we turn to a more familiar setting—namely that of NP and co-NP. Let A be any $\leq_{\mathrm{m}}^{\mathrm{vv}}$-complete set for NP. Then, we know that A has the following properties [AM77,ZH86,BHZ87,Sch89].

- $A \notin \mathrm{P}$, unless RP = NP and PH collapses.
- $A \notin \mathrm{RP}$, unless RP = NP and PH collapses.
- $A \notin$ co-NP, unless PH collapses.

This compares favorably with the many-one complete set SAT:

- SAT $\notin$ P, unless P = NP.
- SAT $\notin$ RP, unless RP = NP and PH collapses.
- SAT $\notin$ co-NP, unless PH collapses.

From this comparison, we can draw the conclusion that the $\leq_{\mathrm{m}}^{\mathrm{vv}}$-complete set, A, behaves like the $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete set, SAT, and that the complexity of the set A is representative of the complexity of the entire class NP.

In contrast, the trivial singleton set $B = \{1\}$ is complete for NP under random reductions with probability 2^{-n}. Clearly, B is not representative of the complexity of NP. So, completeness under $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions with probability 2^{-n} does not make sense for NP. However, they *do* make sense for co-NP. Let C be a set complete for co-NP under $\leq_m^{rp}$-reductions with probability 2^{-n}. Then, $C \notin$ NP, unless NP = co-NP.

These results show that completeness for random reductions make sense only if the random reductions have a probability bound above a certain threshold. Moreover, the threshold is different for different complexity classes. For NP, the threshold is $1/poly$; for co-NP, $1/exp$. Alternatively, we can look at these results in terms of random reductions from a set to its complement.

- SAT $\leq_m^{rp} \overline{\text{SAT}}$ with probability 2^{-n}.
- SAT $\not\leq_m^{rp} \overline{\text{SAT}}$ with probability $1/p(n)$, for any polynomial $p(n)$, unless PH collapses.

These two statements show that when we consider random functions which reduce SAT to $\overline{\text{SAT}}$, a probability threshold occurs at $1/poly$. Similarly, for $\overline{\text{SAT}}$, the probability threshold occurs at $1/exp$.

- There is no known $\leq_m^{rp}$-reduction from $\overline{\text{SAT}}$ to SAT with probability greater than 0.
- $\overline{\text{SAT}} \not\leq_m^{rp}$ SAT with probability $2^{-p(n)}$, for any polynomial $p(n)$, unless NP = co-NP.

4.3.2 Threshold Behavior in the classes D^P and co-D^P

In this section, we show how probability thresholds can explain the anomaly presented in Section 4.2. Recall that SAT$\wedge\overline{\text{SAT}}$ and $\overline{\text{SAT}}\vee$SAT are the $\leq_m^{\text{P}}$-complete sets for D^P and co-D^P, respectively. We will show that beyond a certain probability threshold, the $\leq_m^{rp}$-complete sets for D^P and for co-D^P have many of the properties of the $\leq_m^{\text{P}}$-complete sets. Some of these properties are [CK90a,CK90b, Kad88]:

- SAT$\wedge\overline{\text{SAT}} \not\equiv_m^{\text{P}} \overline{\text{SAT}}\vee$SAT, unless D^P = co-D^P and PH collapses.

- $\mathrm{SAT}\wedge\overline{\mathrm{SAT}} \not\in \text{co-D}^{\mathrm{P}}$ and $\overline{\mathrm{SAT}}\vee\mathrm{SAT} \not\in \mathrm{D}^{\mathrm{P}}$, unless $\mathrm{D}^{\mathrm{P}} = \text{co-D}^{\mathrm{P}}$ and PH collapses.
- $\mathrm{SAT}\wedge\overline{\mathrm{SAT}}$ does not have OR_2, unless $\mathrm{D}^{\mathrm{P}} = \text{co-D}^{\mathrm{P}}$ and PH collapses.
- $\overline{\mathrm{SAT}}\vee\mathrm{SAT}$ does not have AND_2, unless $\mathrm{D}^{\mathrm{P}} = \text{co-D}^{\mathrm{P}}$ and PH collapses.

We will use these properties as a benchmark to test if a particular concept of completeness for D^{P} and co-D^{P} makes sense. First, we show that for completeness under $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions, the probability threshold for D^{P} is bounded below by 2^{-n}.

Lemma 4.2 $\overline{\mathrm{SAT}}$ is complete for D^{P} under $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions with probability 2^{-n}.

Proof: The following is the required reduction from $\mathrm{SAT}\wedge\overline{\mathrm{SAT}}$ to $\overline{\mathrm{SAT}}$. On input $\langle F_1, F_2 \rangle$, try to randomly guess a satisfying assignment for F_1. If F_1 is satisfiable, the reduction will find a satisfying assignment with probability at least 2^{-n}. If a satisfying assignment is found, the reduction simply prints out F_2. Otherwise, it prints out a fixed satisfiable formula. □

Corollary 4.1 $\mathrm{SAT}\wedge\overline{\mathrm{SAT}} \leq_{\mathrm{m}}^{\mathrm{rp}} \overline{\mathrm{SAT}}\vee\mathrm{SAT}$ with probability 2^{-n}.

Since $\overline{\mathrm{SAT}} \in$ co-D^{P} and since $\overline{\mathrm{SAT}}$ has OR_ω and AND_ω, we can safely say that completeness under $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions with probability $1/exp$ does not make sense for D^{P}. The following theorems, show that completeness under $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions starts making sense when the probability bound is $1/poly$. Hence, the probability threshold for D^{P} is bounded above by $1/poly$.

Theorem 4.1 $\mathrm{SAT}\wedge\overline{\mathrm{SAT}} \not\leq_{\mathrm{m}}^{\mathrm{vv}} \overline{\mathrm{SAT}}\vee\mathrm{SAT}$, unless $\mathrm{PH} \subseteq \Sigma_3^{\mathrm{P}}$.

Proof: Suppose $\mathrm{SAT}\wedge\overline{\mathrm{SAT}} \leq_{\mathrm{m}}^{\mathrm{vv}} \overline{\mathrm{SAT}}\vee\mathrm{SAT}$. Then that means that all languages in BH(2) randomly reduced to a language in co-BH(2) under probability $1/poly$ $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions. But as shown in Chapter 3, (Theorem 3.2, see also Figure 3.2) this immediately implies that the PH collapses to Σ_3^{P}. □

As an immediate corollary of Theorem 4.1 we can conclude that completeness under $\leq_m^{vv}$-reductions *does* make sense for D^P. Moreover, we can show that languages complete for D^P under $\leq_m^{vv}$-reductions have properties very similar to the properties of the $\leq_m^P$-complete language $\text{SAT}\wedge\overline{\text{SAT}}$.

Theorem 4.2 Let A be complete for D^P under $\leq_m^{vv}$-reductions. Then,

1. $A \not\equiv_m^P \overline{A}$, unless PH collapses.
2. $A \notin$ co-D^P, unless PH collapses.
3. A does not have OR_ω, unless PH collapses.

Proof (Sketch): Parts 1 and 2 follow from Theorem 4.1. The proof of part 3 is similar to the proof of Theorem 4.1. Simply note that if A has OR_ω, then there is an $\leq_m^{rp}$-reduction from $\overline{\text{SAT}}\vee\text{SAT}$ to $\text{SAT}\wedge\overline{\text{SAT}}$ with very high probability. This condition is sufficient to mimic the proof that $\overline{\text{SAT}}\vee\text{SAT}\leq_m^P \text{SAT}\wedge\overline{\text{SAT}} \Rightarrow$ PH collapses. See Theorem 4.4. □

As a special case of a language $\leq_m^{vv}$-complete for D^P, USAT has all the properties listed above. In fact, since we also know that $\overline{\text{SAT}}\leq_m^P \text{USAT}$ we can get even stronger results.

Corollary 4.2 USAT $\notin$ co-D^P, unless PH collapses.

Corollary 4.3 USAT does not have OR_ω, unless PH collapses.

Theorem 4.3 $\text{USAT} \not\equiv_m^P \overline{\text{USAT}}$, unless D^P = co-D^P and PH collapses to Δ_3^P.

Proof: Since $\overline{\text{SAT}}\leq_m^P \text{USAT}$ it follows that $\text{SAT}\leq_m^P \overline{\text{USAT}}$. Hence, if we assume that $\text{USAT}\equiv_m^P \overline{\text{USAT}}$, then SAT $\leq_m^P$-reduces to USAT as well. Thus, USAT becomes $\leq_m^P$-complete for D^P, so D^P = co-D^P and the Polynomial Hierarchy collapses to Δ_3^P by Kadin [Kad88]. □

Thus, we can conclude that Valiant and Vazirani made right decision when they used $\leq_{\mathrm{m}}^{\mathrm{vv}}$-reductions to talk about completeness for $\mathrm{D^P}$ under random reductions. However, completeness under $\leq_{\mathrm{m}}^{\mathrm{vv}}$-reductions may not make sense for other complexity classes. In fact, the following theorems show that the threshold for co-$\mathrm{D^P}$ is $1/2 + 1/poly$.

Lemma 4.3 $\mathrm{SAT}\oplus\overline{\mathrm{SAT}}$ is complete for co-$\mathrm{D^P}$ under $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions with probability $1/2 + 2^{-n^2}$.

Proof (Sketch): Recall that $\overline{\mathrm{SAT}}\vee\mathrm{SAT}$ is $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete for co-$\mathrm{D^P}$and that

$$\mathrm{SAT}\oplus\overline{\mathrm{SAT}} = \{\ 0F \mid F \in \mathrm{SAT}\ \} \cup \{\ 1F \mid F \in \overline{\mathrm{SAT}}\ \}.$$

Also, note that $\mathrm{SAT}\oplus\overline{\mathrm{SAT}} \in \mathrm{D^P} \cap \text{co-}\mathrm{D^P}$. It is simple to construct a random reduction with probability greater than or equal to 1/2, because

$$\langle\ F_1, F_2\ \rangle \in \overline{\mathrm{SAT}}\vee\mathrm{SAT} \iff F_1 \in \overline{\mathrm{SAT}} \text{ or } F_2 \in \mathrm{SAT}.$$

Thus, a randomized function can choose F_1 or F_2 with equal probability, then output $1F_1$ or $0F_2$. If $\langle\ F_1, F_2\ \rangle$ is indeed an element of $\overline{\mathrm{SAT}}\vee\mathrm{SAT}$, then it follows that $\mathrm{Prob}_{i\in\{0,1\}}[\ iF_{2-i} \in \mathrm{SAT}\oplus\overline{\mathrm{SAT}}\ ] \geq 1/2$. On the other hand, if $\langle\ F_1, F_2\ \rangle \notin \overline{\mathrm{SAT}}\vee\mathrm{SAT}$, then $\mathrm{Prob}_{i\in\{0,1\}}[\ iF_{2-i} \in \mathrm{SAT}\oplus\overline{\mathrm{SAT}}\ ] = 0$. To improve the probability beyond 1/2, simply observe that if $F_2 \in \mathrm{SAT}$, then there is a small probability of finding a satisfying assignment through random guessing. This fact can be used to improve the probability to $1/2 + 2^{-n^2}$. □

Using the fact that $\mathrm{SAT}\oplus\overline{\mathrm{SAT}} \in \mathrm{D^P}\cap\text{co-}\mathrm{D^P}$, we also obtain the following corollary:

Corollary 4.4 $\overline{\mathrm{SAT}}\vee\mathrm{SAT} \leq_{\mathrm{m}}^{\mathrm{rp}} \mathrm{SAT}\wedge\overline{\mathrm{SAT}}$ with probability $1/2 + 2^{-n^2}$.

Since $\mathrm{SAT}\oplus\overline{\mathrm{SAT}} \in \mathrm{D^P}$, we can conclude that this set does not meet our criteria for a sensible complete language for co-$\mathrm{D^P}$. However, if we require the reduction to have a probability bound above $1/2 + 1/poly$, then completeness makes sense.

Theorem 4.4 $\overline{\mathrm{SAT}} \vee \mathrm{SAT} \not\leq_{\mathrm{m}}^{\mathrm{rp}} \mathrm{SAT} \wedge \overline{\mathrm{SAT}}$ with probability $1/2 + 1/p(n)$, for any polynomial $p(n)$, unless $\mathrm{PH} \subseteq \Sigma_3^{\mathrm{P}}$.

Proof: Suppose $\overline{\mathrm{SAT}} \vee \mathrm{SAT} \leq_{\mathrm{m}}^{\mathrm{rp}} \mathrm{SAT} \wedge \overline{\mathrm{SAT}}$ with probability $1/2 + 1/p(n)$. Then that means that all languages in co-BH(2) randomly reduced to a language in BH(2) under probability $1/2 + 1/poly$ $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions. But as shown in Chapter 3, (Theorem 3.3, see also Figure 3.2) this immediately implies that the PH collapses to Σ_3^{P}. □

Now we can show that the languages complete for co-D^{P} under $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions with probability beyond the threshold of $1/2+1/poly$ have many properties enjoyed by the $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete languages for co-D^{P}.

Theorem 4.5 Let A be complete for co-D^{P} under $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions with probability $1/2 + 1/p(n)$, for some polynomial p. Then,

1. $A \not\equiv_{\mathrm{m}}^{\mathrm{P}} \overline{A}$, unless $\mathrm{PH} \subseteq \Delta_3^{\mathrm{P}}$.
2. $A \notin \mathrm{D}^{\mathrm{P}}$, unless $\mathrm{PH} \subseteq \Sigma_3^{\mathrm{P}}$.
3. A does not have AND_2, unless $\mathrm{PH} \subseteq \Delta_3^{\mathrm{P}}$.

Proof: Parts 1 and 3 follow from Theorem 4.1 and Part 2 follows from Theorem 4.4. □

4.3.3 Threshold behavior in the Boolean and Query Hierarchies

Next we turn to the classes in the Boolean and Bounded Query Hierarchies. In Chapter 3, we proved *tight* bounds on the success probabilities of random reductions between nearby classes in the Boolean and Query Hierarchies. These *tight* bounds immediately provide us with the thresholds for these classes as well. For instance, we showed that the BL_{2k} the complete language for $\mathrm{BH}(2k)$ $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reduces to the complementary language co-BL_{2k} with probability $1 - 1/k$. Thus, $1 - 1/k$ is a lower bound on the threshold for $\leq_m^{rp}$-completeness in BH($2k$). Moreover, Figure 3.2 also shows that $1 - 1/k + 1/poly$ is an upper bound on the threshold for BH($2k$).

Theorem 4.6 For the class BH($2k$), the threshold probability for completeness under $\leq_m^{rp}$-reductions is $1 - 1/k + 1/poly$.

Based on this theorem we can easily show that any language L which is $\leq_m^{rp}$-complete for BH($2k$) with a probability above the threshold inherits the following hardness properties (assuming PH infinite) usually associated with the $\leq_m^P$-complete languages:

- L is not in a simpler class such as $P^{SAT||[2k-1]}$ or in the complementary class co-BH($2k$).
- L is not closed under OR_ω (or even OR_2 if $k > 1$).

Also, for every $k \geq 1$, we can exhibit a language in BH($2k$) which is $\leq_m^{rp}$-complete with probability above the threshold. Define the languages $BL_{2k-2} \vee$ USAT as

$$\{ \langle x_1, \ldots, x_{2k-2}, y \rangle \mid \langle x_1, \ldots, x_{2k-2} \rangle \in BL_{2k-2} \text{ or } y \in \text{USAT} \}.$$

Lemma 4.4 For any $k \geq 1$, the language $BL_{2k-2} \vee$ USAT is $\leq_m^{rp}$-complete for BH($2k$) via a probability $1 - 1/k + 1/(4kn)$ reduction.

Proof: $BL_{2k-2} \vee \text{USAT} \in \text{BH}(2k)$ because USAT $\in D^P$. The $\leq_m^{rp}$-reduction from BL_{2k} to $BL_{2k-2} \vee$ USAT can be obtained by combining the $\leq_m^{rp}$-reduction used in Lemma 3.1 with the probability $1/4n$ Valiant-Vazirani [VV86] reduction from the SAT$\wedge\overline{\text{SAT}}$ to USAT. □

Thus, the languages $BL_{2k-2} \vee$ 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 4.7 There is a recursive oracle O such that relative to O, $\forall k$, the relativized language corresponding to $\mathrm{BL}_{2k-2} \vee \mathrm{USAT}$ 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].

Chapter 5

Completeness under Random Reductions: Absolute Separations [1]

In the preceding chapters we saw that the correctness probability of random reduction is crucial as far as the notion of completeness under random reductions is concerned. In fact, we saw that for many complexity classes $\mathcal{C}$, the behavior of sets complete under random reductions is determined by a *probability threshold* $\mathcal{C}_T$. Sets which are complete for $\mathcal{C}$ under random reductions with correctness probability below $\mathcal{C}_T$ may be far simpler than the $\leq_m^p$-complete sets of $\mathcal{C}$, whereas all sets complete under reductions with correctness probability above $\mathcal{C}_T$ inherit many of the hardness properties typical of $\leq_m^{\mathrm{P}}$-complete sets. For many classes the value of this threshold is non-trivial, for instance the *threshold* probability for two-sided error random reductions in $\mathrm{P}^{\mathrm{SAT}||[k]}$ is $1 - 1/(k+1)$, i.e., sets which are complete for these classes under probability $1 - 1/(k+1) + 1/poly(n)$ reductions are harder than certain sets which are complete under probability $1 - 1/(k+1)$

[1] The results presented in this chapter have been drawn from [CR93]

reductions unless the Polynomial Hierarchy (PH) collapses. This established that, if the PH is infinite, the notion of completeness under probability $1 - 1/(k+1)$ random reductions must differ from that under probability $1-1/(k+1)+1/poly(n)$ random reductions, thus highlighting the sensitivity of the notion of completeness under random reductions to small changes in the success probability.

However, all these results rely on the assumption that the PH is an infinite hierarchy (actually, we only need to assume that it does not collapse to its third level). Although this is a standard assumption in complexity theory, it is still not a conclusive proof that random reductions have the kind of behavior that we claimed. In this chapter, we provide further evidence that such behavior is inherent to random reductions by proving that it is present in very high complexity classes *without making any structural assumptions*. In doing so, we systematically explore the notion of completeness under random reductions with different probabilities. We compare completeness under various random reductions to each other and to completeness under deterministic reductions. We obtain optimal separations between completeness notions under different random reductions and completeness under deterministic reductions. Our results also show that even a vanishingly small difference of $1/poly(n)$ in the success probability of the reduction yields a different class of complete sets.

Separating various deterministic polynomial time reductions and notions of completeness under such reductions has been an active area of research. The study of polynomial time reducibilities originated with the seminal work of Ladner, Lynch and Selman [LLS75] who defined several truth table reductions and showed that all these reductions along with $\leq_m^p$ and $\leq_T^p$ reductions are distinct in E. They left open the question of whether the complete degrees for these reductions were distinct in E. This was partly resolved by Ko and Moore [KM81], and finally Watanabe [Wat87] developed a powerful proof methodology to show that in E most notions of completeness were indeed different. Subsequently, similar results were obtained by Buhrman, Homer and Torenvliet [BHT90] for nondeterministic classes such as NE. An interesting question left open by Watanabe's work was whether in E, the notion of $\leq_m^p$-completeness was different from the notion of $\leq_{1-tt}^p$-completeness. In a surprising development these were shown to be the same for E by Homer, Kurtz and Royer and a similar result for NE was obtained by Buhrman (cf. [Hom90]).

These absolute separation results are only known for highly intractable complexity classes such as E and beyond. Such results have not been forthcoming for easier classes such as PSPACE since this would imply that P $\neq$ PSPACE. Moreover, due to contradictory relativizations of the P = PSPACE question such separations can not be obtained in PSPACE using relativizing techniques. For separating completeness under different random reductions the situation is much worse: there exists a relativized world where RP = E [Wil85] and another relativized world where BPP = E^{Δ_2} [Hel86]! Therefore, in order to obtain strong separation results we have to go to higher classes.

First we show that in E there are sets which are complete under very high probability random reductions but which are not complete under any deterministic reductions. The construction is a straight forward diagonalization which ensures that the resulting set does not a satisfy a property possessed by all $\leq_T^p$-complete sets for E [Wat87]. Next, we examine whether there are sets which are complete under deterministic reductions but not under certain random reductions. As noted earlier, relativizing techniques can not yield such results in classes like E or E^{Δ_2}. However, we do obtain very sharp separation results in slightly higher classes like E^{Σ_2} or $\tilde{E}$ (superexponential time). The crucial difference between the classes E^{Σ_2} and E^{Δ_2} that allows our techniques to work is that in E^{Σ_2} we can approximate the the number of accepting paths of a nondeterministic superpolynomial time machine to a high degree of accuracy based on Stockmeyer's result on approximate counting [Sto83].

Since our techniques for one sided error ($\leq_m^{rp}$) reductions differ from those for two sided error ($\leq_m^{\mathrm{bpp}}$) reductions, we treat these reductions separately. For $\leq_m^{\mathrm{rp}}$ reductions we prove that for every $k > 1$ there is a set in E^{Σ_2} which is $\leq_{k-tt}^{p}$-complete (actually $\leq_{k-\mathrm{disj}}^{p}$ complete) but not $\leq_m^{\mathrm{rp}}$-complete with probability $1/k+1/poly(n)$. This is an optimal separation result because sets which are $\leq_{k-\mathrm{disj}}^{p}$ complete are also $\leq_m^{\mathrm{rp}}$ complete with probability $1/k$. Next we show that for any p there exists a set in E^{Σ_2} which is $\leq_m^{\mathrm{rp}}$ complete with probability p but not $\leq_m^{\mathrm{rp}}$ complete with probability $p + 1/poly(n)$. This supports the thesis that the notion of completeness is very sensitive to small changes (even a vanishingly small change of $1/poly(n)$) in success probability of the random reduction. Our approach to $\leq_m^{\mathrm{rp}}$ reductions is based on the methodology proposed by Watanabe. We first identify a property that is possessed by all sets which are complete for E^{Σ_2} under probability $p+1/poly(n)$ $\leq_m^{\mathrm{rp}}$ reductions. We then construct a set by diagonalizing against this property while ensuring that the set constructed is $\leq_m^{\mathrm{rp}}$-complete under a probability p reduction.

To obtain *optimal* separations for $\leq_m^{\mathrm{bpp}}$-reductions we employ a new technique. For every $k > 1$ we construct a set in E^{Σ_2} which is $\leq_{k-tt}^{p}$-complete (actually $\leq_{k-\mathrm{disj}}^{p}$ as well as $\leq_{k-\mathrm{conj}}^{p}$ complete) but not $\leq_m^{\mathrm{bpp}}$-complete with probability $1/2 + 1/2k + 1/poly(n)$. If a $\leq_{k-tt}^{p}$ complete set is both $\leq_{k-\mathrm{disj}}^{p}$ and $\leq_{k-\mathrm{conj}}^{p}$ complete then it is actually complete under probability $1/2 + 1/2k$ $\leq_m^{\mathrm{bpp}}$-reductions. Hence this is an optimal separation. We extend this technique to show that for any p, there exists a set in E^{Σ_2} which is complete under probability p $\leq_m^{\mathrm{bpp}}$ reductions but not probability $p + 1/poly(n)$ reductions. The technique used to prove these results is to first construct a highly structured set which satisfies certain diagonalization requirements and then to explicitly construct a language in E^{Σ_2} which cannot reduce to the constructed set with high probability $\leq_m^{\mathrm{bpp}}$-reductions.

The chapter is organized as follows: In Section 5.1 we provide the necessary background. In Section 5.2 we outline the construction of sets which are complete under high probability random reduction but not complete under deterministic reductions. Section 5.3 describes the separation results for $\leq_m^{\mathrm{rp}}$-reductions and in Section 5.4 we present results for $\leq_m^{\mathrm{bpp}}$-reductions.

5.1 Preliminaries

We will use E to denote the class $\cup_{c>0}\mathrm{DTIME}(2^{cn})$. In this chapter we will also deal with the classes E^{Δ_2} and E^{Σ_2} which are members of the weak exponential hierarchy [Hem87]. The class E^{Δ_2} consists of languages that are accepted by exponential time (2^{cn}) oracle machines which can query a Δ_2^p complete set. Similarly the class E^{Σ_2} consists of languages that are accepted by exponential time oracle machines which can query a Σ_2^p complete set.

We assume a standard enumeration of deterministic polynomial time transducers, $M_1, M_2, \ldots$, and randomized polynomial time transducers, $R_1, R_2 \ldots$ in which each machine appears infinitely often. Also the running times of M_i and R_i are bounded by n^i.

5.2 Separation results in exponential time

In this section we construct sets in E which are complete under very high probability $(1 - \frac{1}{2^{cn}})$ $\leq_m^{\mathrm{rp}}$ and $\leq_m^{\mathrm{bpp}}$ reductions which are not complete under deterministic reductions.

We use the following result of Watanabe [Wat87] which establishes a property of sets which are complete under deterministic reductions.

Lemma 5.1 [Wat87] Let A be any $\leq_{\mathrm{T}}^p$-complete set for E. Then there exists a polynomial time oracle machine M such that for all but finitely many n, the set, Q, of strings queried by $M^{A^{\leq n}}(0^n)$ contains a string in A of length greater than n i.e. $Q \cap A^{>n} \neq \emptyset$.

Since we will be using generalizations of this lemma later on in some of our proofs, we shall provide its proof here.

Proof: [Wat87] Given a set A which is $\leq_{\mathrm{T}}^{p}$-complete for E, define a language L_A based on A as follows. On input $\langle\ i, 0^n\ \rangle$:

- Compute the set $A^{\leq n}$. This can be done in E.
- Let M_i be the i'th polynomial time oracle machine in the standard enumeration of such machines. Simulate $M_i^{A^{\leq n}}(\langle\ i, 0^n\ \rangle)$ for 2^n steps.
- If the computation is not complete in 2^n steps or if the final configuration is *accepting* then **reject** the input. Otherwise (the final configuration is *rejecting*), **accept** the input. .

From the definition, it follows that L_A constructed as above is in E and since A is $\leq_{\mathrm{T}}^{p}$-complete, it follows that $L_A \leq_{\mathrm{T}}^{p} A$ via some polynomial time oracle machine M_k. For sufficiently large n the running time of M_k on $\langle\ k, 0^n\ \rangle$ will always be less than 2^n. For any such n, let us compare the behavior of $M_k^{A^{\leq n}}(\langle\ k, 0^n\ \rangle)$ and $M_k^{A}(\langle\ k, 0^n\ \rangle)$. If $\langle\ k, 0^n\ \rangle \in L_A$ then by definition of L_A we must have that $M_k^{A^{\leq n}}(\langle\ k, 0^n\ \rangle)$ must reject whereas $M_k^{A}(\langle\ k, 0^n\ \rangle)$ should accept in order to carry out a correct reduction. Similarly, if $\langle\ k, 0^n\ \rangle \notin L_A$ then we must have that $M_k^{A^{\leq n}}(\langle\ k, 0^n\ \rangle)$ accepts whereas $M_k^{A}(\langle\ k, 0^n\ \rangle)$ rejects. Thus, in either case, for large n, the output of $M_k^{A^{\leq n}}(\langle\ k, 0^n\ \rangle)$ is different from the output of $M_k^{A}(\langle\ k, 0^n\ \rangle)$ which can only happen if for all inputs of the type $\langle\ k, 0^n\ \rangle$ (n large), the machine M_k queries at least one string in $A^{>n}$. Thus, an oracle machine M which on input 0^n simulates the oracle machine M_k on input $\langle\ k, 0^n\ \rangle$ satisfies the requirements of Lemma 5.1. □

We use this characterization to construct, via diagonalization, a set A which is $\leq_{m}^{\mathrm{rp}}$-complete under high probability reductions, but not $\leq_{\mathrm{T}}^{p}$-complete.

Theorem 5.1 *There exists a set A, which is $\leq_{m}^{\mathrm{rp}}$-complete for E with probability $1 - \frac{1}{2^{cn}}$ but not $\leq_{\mathrm{T}}^{p}$-complete.*

Proof: Let U be a standard $\le_m^p$-complete set for E. The elements of A will be of the form $u\#x$ where $u \in U$ and x is a string in Σ^* of length $|u|$. We maintain the following property:

$\boldsymbol{\mathcal{P}}$: For each $u \in U$ almost all strings of the form $u\#x$, with $|u| = |x|$, will be in A.

This immediately gives an obvious high probability $\le_m^{\mathrm{rp}}$-reduction from U to A. The construction maintains $\boldsymbol{\mathcal{P}}$ while ensuring that the resulting set is not $\le_{\mathrm{T}}^p$-complete for E.

Construction of $\boldsymbol{A}$: The set A is built in stages. We ensure that for every oracle machine M_i, for infinitely many n, none of the strings, of length greater than n, queried by $M^{A^{\le n}}(0^n)$ is in A. By Lemma 5.1, A will not be $\le_{\mathrm{T}}^p$-complete for E. This is a straightforward diagonalization argument where we balance this negative requirement with the positive requirement $\boldsymbol{\mathcal{P}}$.

Let $n_1 = 0$ and $A_0 = \emptyset$. Each machine appears infinitely often in the enumeration of oracle machines and after a certain stage, every time a machine M appears it gets diagonalized against.

Stage i: Let $n = n_i$. Run $M_i^{A_{i-1}}(0^n)$ for $2^{\frac{n}{4}}$ steps. If it does not halt let $n_{i+1} = n_i + 1$ and put into A all strings of the form $u\#x$ of length $n+1$ and go to stage $i+1$. If the simulation succeeds within $2^{\frac{n}{4}}$ steps let Q be the set of strings queried by by $M_i^{A_{i-1}}(0^n)$. Let Q_1 be the subset of $Q^{>n}$ consisting of strings of the form $x\#y$, where $|x| = |y|$. Let m be the size of the largest string of Q.

$$A_i = A_{i-1} \cup \{u\#x \mid |u| = |x|, u \in U, n < |u\#x| \le m\} - Q_1$$

Let $n_{i+1} = m$ and go to stage $i+1$.

Since the entire construction up to any length n can be carried out in time 2^{cn}, for some constant c, the set A is in E. Each oracle machine M gets diagonalized infinitely often and, by Lemma 5.1, A is not $\le_{\mathrm{T}}^p$-complete. The property $\boldsymbol{\mathcal{P}}$ is maintained, since for all $u \in U^{=n}$, only one machine M_i can rule out strings of the form $u\#x$ from A. This occurs at a stage when the input to M_i is 0^k with

$k < |u\#x|$ which is at most $2n+1$. From the construction, $M_i(0^k)$ runs for at most $2^{\frac{2n+1}{4}}$ steps and hence $|Q_1| \leq |Q| \leq 2^{\frac{2n+1}{4}}$ in that stage. Hence at most $2^{\frac{2n+1}{4}}$ strings of the form $u\#x$ can be in $\overline{A}$ for any $u \in U$. The fraction of strings of the form $u\#x$ which are in A is at least $\frac{2^n - 2^{(2n+1)/4}}{2^n} \geq 1 - \frac{1}{2^{n/4}}$. Thus, the construction maintains the property $\mathcal{P}$. □

Corollary 5.1 There exists a set which is complete for E under probability $1 - 1/2^{cn}$ $\{\leq_m^{\mathrm{rp}}, \leq_m^{\mathrm{bpp}}\}$ reductions but which is not $\{\leq_m^p, \leq_{k-tt}^p, \leq_{tt}^p, \leq_{\mathrm{T}}^p\}$-complete.

Note that the set constructed above is $\leq_m^{\mathrm{rp}}$-complete for E with inverse exponential error probability. Since an $\leq_m^p$-complete set is $\leq_m^{\mathrm{rp}}$-complete with probability 1, in this special case we have a separation between completeness notions for reductions with probabilities that are very close.

5.3 Separation results for one sided random reductions

In this section we construct sets which are complete under deterministic reductions but not complete under one sided error random reductions. Since the question RP $\stackrel{?}{=}$ E is unresolved we can not hope to find such a separation in E because if RP=E then all non trivial sets in E are $\leq_m^{\mathrm{rp}}$-complete with high probability. Since these techniques are also applicable to certain $\leq_m^{\mathrm{bpp}}$-reduction we have to go to the class E^{Σ_2} to avoid the relativized BPP $= \mathrm{E}^{\Delta_2}$ result [Hel86].

In section 5.3.1, we construct a set which is $\leq_{k-\mathrm{disj}}^p$-complete (and hence $\leq_m^{\mathrm{rp}}$-complete with probability $\frac{1}{k}$) but which is not $\leq_m^{\mathrm{rp}}$-complete with probability $\frac{1}{k} + \frac{1}{poly(n)}$. When constructing sets which are $\leq_{k-tt}^p$-complete but not $\leq_m^{\mathrm{rp}}$-complete we have to contend with $\leq_{k-\mathrm{disj}}^p$- complete sets which are $\leq_m^{\mathrm{rp}}$-complete with probability $\frac{1}{k}$. Thus $p = \frac{1}{k} + \frac{1}{poly(n)}$ is the smallest probability[2] at which we can find a set

[2]Actually a smaller difference of $\frac{1}{2^{cn}}$ sufices.

which is not $\leq_m^{rp}$-complete under probability p reductions. This directly shows that the class of complete sets for $\leq_m^{rp}$-reductions with probabilities $\frac{1}{k}$ and $\frac{1}{k}+\frac{1}{poly(n)}$ differ. In section 5.3.2 we extend this to show that for any p, the classes of complete sets under probability p $\leq_m^{rp}$-reductions and probability $p+\frac{1}{poly(n)}$ reductions differ.

5.3.1 Deterministic but not $\leq_m^{rp}$-complete sets

Motivated by the approach used by Watanabe [Wat87], we first establish a property of sets $\leq_m^{rp}$-complete for E^{Σ_2} with probability p and then diagonalize to produce sets which are not $\leq_m^{rp}$-complete with probability p.

Lemma 5.2 Let A be $\leq_m^{rp}$-complete for E^{Σ_2} with probability p. Then there exists a randomized polynomial time transducer, R, such that for all but finitely many n, $R(0^n)$ outputs an element of $A^{>n}$ with probability at least p.

Proof: Extending the approach of [Wat87] we define a language L_A as follows. On input $< i, 0^n >$:

- Construct the set $A^{\leq n}$. Since there are at most $|\Sigma|^{n+1}$ elements to be tried this can be done in E^{Σ_2}. Let $\vec{b}$ be the exponential sized characteristic vector for $A^{\leq n}$.

- Query the Σ_2^p oracle if R_i on input $< i, 0^n >$ outputs an x_i with $\vec{b}(x_i) = 1$ on some path in at most 2^n steps. The input to the oracle is $(< i, 0^n >, \vec{b})$ whose size is at least 2^n, and an NP machine can simulate $R_i(< i, 0^n >)$ for at most $|(< i, 0^n >, \vec{b})|$ steps and detect if such an x_i exists. If no such x_i exists then **accept** else **reject**.

Clearly L_A is in E^{Σ_2}. Since A is $\leq_m^{rp}$-complete for E^{Σ_2} with probability p, there exists a probabilistic transducer, R_j, witnessing the reduction from L_A to A. For sufficiently large n, the running time of $R_j(< j, 0^n >)$ is less than 2^n. For such n,

$< j, 0^n >$ must be in L_A, or else $R_j(< j, 0^n >)$ would only output elements of $\overline{A}$ and by definition of L_A, we would get $< j, 0^n > \in L_A$.

Thus, for all but finitely many n, $R_j(< j, 0^n >)$ outputs an element of A with probability at least p. It can not output any element in $A^{\leq n}$, otherwise, by construction, $< j, 0^n >$ would not be in L_A. The required machine R, on input 0^n, simulates $R_j(< j, 0^n >)$. □

We now construct a set which is $\leq^p_{2-tt}$ complete but not $\leq^{\mathrm{rp}}_m$-complete with probability $p \geq \frac{1}{2} + \epsilon$. The $\leq^p_{2-tt}$ condition is actually a disjunction and hence the set is $\leq^{\mathrm{rp}}_m$-complete with probability $\frac{1}{2}$.

Theorem 5.2 *There exists a set B which is $\leq^p_{2-tt}$-complete for E^{Σ_2} but which is not complete under $\leq^{\mathrm{rp}}_m$-reductions with probability $p \geq \frac{1}{2} + \epsilon$, where $\epsilon = \frac{1}{poly(n)}$.*

Proof: The set B is constructed in stages. In the construction we ensure that for every probabilistic transducer R there are infinitely many n, such that $R(0^n)$ does not output elements of $B^{>n}$ with probability p and thus by Lemma 5.2 B cannot be $\leq^{\mathrm{rp}}_m$-complete with probability p.

The set B consists of strings of the form $u\#0$ or $u\#1$ where $u \in U$, the standard complete set for E^{Σ_2}. For each $u \in U$ at least one of the strings $u\#0$ or $u\#1$ will be in B. This ensures that B is $\leq^p_{2-tt}$ hard for E^{Σ_2}. Let $n_0 = 1$ and $B_0 = \emptyset$.

Stage *i*: Let $n = n_i$. By assumption the running time of $R_i(0^n)$ is at most n^i. If this is larger than $n^{\log(n)}$ then set $n_{i+1} = 2^{n_i}$ and add to B all strings of the form $u\#0$, $u \in U$, of length between $n_i + 1$ and n_{i+1} and go to stage $i + 1$.

Use the Σ^p_2 oracle, using sufficient padding, to find m, the size of the largest string that $R_i(0^n)$ outputs. To meet the diagonalization requirements we must ensure that the probability that $R_i(0^n)$ outputs any of the strings that we add to B is less than p. On the other hand, for each $u \in U$ we must add one of $u\#0$ or $u\#1$ to B. If among $u\#0$ and $u\#1$, the string that $R_i(0^n)$ outputs with lower probability is added to B, then the probability that $R_i(0^n)$ outputs a string in $B^{>n}$ is at most $\frac{1}{2}$. However we can not naively simulate all paths of R_i to determine which string is output with lower probability since that may require $2^{n^{\log(n)}}$ time. Instead we adopt the approach of approximating these probabilities using the following result due to Stockmeyer.

Lemma 5.3 [Sto83] Let M be an NP machine. Then in Δ_3 we can, on any input x, estimate #ACC(M,x), the number of accepting paths of M on x to within a multiplicative factor of $1 \pm \delta$ where δ is inverse polynomial in the length of the input.

Consider strings of the form $u\#0$ or $u\#1$, with $u \in U$, of length l, where $n_i < l \leq m$. To decide which of $u\#0$ or $u\#1$ to put into B we first approximate the probability that $R_i(0^n)$ outputs these strings. Since $R_i(0^n)$ runs for at most $n^{\log(n)}$ steps on any path and we have the resources to produce an exponential sized padding we can use Lemma 5.3 to approximate these probabilities to within a multiplicative factor of $(1 \pm \delta)$, where δ is as low as $\frac{1}{2^{cn}}$.

Let $\tilde{p}_u(0)$ and $\tilde{p}_u(1)$ be the computed estimates of the probability that $R_i(0^n)$ outputs $u\#0$ and $u\#1$ respectively and let $p_u(0)$ and $p_u(1)$ be the actual probabilities. We put in B the string which has the lower estimated probability. The use of estimates may cause a string with larger actual probability to be put into B but we now show that this does not introduce a large error.

For each $u \in U$ if we had put the string with lower actual probability then

$$\text{Prob}[\ R_i(0^n) \text{ outputs a string in } B^{>n}\ ] \leq \sum_u min(p_u(0), p_u(1)) \leq \frac{1}{2}$$

where the summation is over all u such that $|u\#\{0,1\}| > n$. When estimates are used, the probability that $R_i(0^n)$ outputs a string in $B^{>n}$ is $\sum_u min(\tilde{p}_u(0), \tilde{p}_u(1))$. Notice that for all u,

$$min(\tilde{p}_u(0), \tilde{p}_u(1)) \leq min(p_u(0), p_u(1))(1+\delta)/(1-\delta),$$

where δ is at most $\frac{1}{2^{cn}}$. Thus, the probability that $R_i(0^n)$ outputs a string in $B^{>n}$ is at most $\frac{1}{2}(1 + \frac{1}{2^{c'n}})$. For lengths $m+1$ to $n_{i+1} = 2^{n_i}$, put $u\#0$ into A, where

$u \in U$. □

$B \in \mathrm{E}^{\Sigma_2}$ since the above construction can be carried out up to any length in E^{Σ_2}. By construction, we ensured that B is $\leq^p_{2-tt}$-hard for E^{Σ_2}. Hence B is $\leq^p_{2-tt}$ complete for E^{Σ_2}. Since each R_j is diagonalized infinitely often, by Lemma 5.2, B is not $\leq^{\mathrm{rp}}_m$-complete with probability $p \geq \frac{1}{2} + \epsilon$. □

Corollary 5.2 For any k, there exists a set C_k which is $\leq^p_{k-tt}$-complete for E^{Σ_2} but not $\leq^{\mathrm{rp}}_m$-complete with probability for $\frac{1}{k} + \frac{1}{poly(n)}$. There exists sets which are $p(n)$-$\{\leq^p_{tt}, \leq^p_{\mathrm{T}}\}$ complete but which are not $\leq^{\mathrm{rp}}_m$ complete with probability greater than $\frac{1}{p(n)} + \frac{1}{poly}$.

The big step in the above construction was the ability to estimate probabilities in E^{Σ_2}. In exponential time we can not simulate all paths of an arbitrary R_i to compute actual probabilities. However in classes such as $\bar{E} = \mathrm{DTIME}(2^{n^{\log(n)}})$, and beyond, the probabilities can be computed exactly. The following can be shown along the same lines.

Theorem 5.3 *For any k, there exists a set C_k which is $\leq^p_{k-tt}$-complete for $\bar{E}$ but not $\leq^{\mathrm{rp}}_m$-complete with probability for $\frac{1}{k} + \frac{1}{poly(n)}$. There exists sets which are $p(n)$-$\{\leq^p_{tt}, \leq^p_{\mathrm{T}}\}$ complete but which are not $\leq^{\mathrm{rp}}_m$ complete with probability greater than $\frac{1}{p(n)} + \frac{1}{poly}$.*

5.3.2 Separations of different probability $\leq^{\mathrm{rp}}_m$-reductions

Let us now consider the problem of separating the class of complete sets under different $\leq^{\mathrm{rp}}_m$-reductions in the class E^{Σ_2}. We note again that if RP=E then every non trivial set is complete with probability $1 - \frac{1}{exp}$. However at classes such as E^{Σ_2} which are higher than E we show that there is a very fine distinction in the completeness notions for $\leq^{\mathrm{rp}}_m$ reductions with different probabilities.

In Corollary 5.2 we obtained sets which were $\leq^p_{k-tt}$ complete but not $\leq^{\text{rp}}_m$-complete with probability $\frac{1}{k}+\frac{1}{poly(n)}$. In our construction the truth table condition was a disjunction and hence the set was also $\leq^{\text{rp}}_m$-complete with probability $\frac{1}{k}$. Therefore, we have separated the notions of $\leq^{\text{rp}}_m$-completeness for probabilities $\frac{1}{k}$ and $\frac{1}{k}+\frac{1}{poly(n)}$ for integral k. We will use similar ideas to prove the following theorem which separates reductions with probabilities p and $p+\frac{1}{poly(n)}$ for any arbitrary value of p.

Theorem 5.4 *For any p, there exists a set A in E^{Σ_2} which is $\leq^{\text{rp}}_m$-complete with probability p but not $\leq^{\text{rp}}_m$-complete with probability $p+\frac{1}{poly(n)}$.*

Proof: The construction uses ideas from the proofs of Theorems 5.1 and 5.3. The constructed set A will consist of strings of the form $u\#x$ where $u \in U$, the standard complete set for E^{Σ_2}, with $|x| = |u|$. For each u in U we will ensure that at least a p fraction of the strings of the form $u\#x$ will be in A. Hence the set A will be $\leq^{\text{rp}}_m$-complete with probability p. In order to ensure that it is not $\leq^{\text{rp}}_m$-complete with probability $p+\frac{1}{poly(n)}$ we ensure that for each randomized polynomial time transducer, R, for infinitely many n, the probability that $R(0^n)$ outputs an element of $A^{>n}$ is less than $p+\frac{1}{poly(n)}$. By Lemma 5.2 the set A can not be $\leq^{\text{rp}}_m$-complete with probability $p+\frac{1}{poly(n)}$

The set A is constructed in stages: Let $n_0 = 1$ and $A_0 = \emptyset$. Stage i tries to diagonalize against the i^{th} probabilistic transducer R_i.

Stage i: Let $n = n_i$ and consider $R_i(0^n)$. By assumption $R_i(0^n)$ does not run for more than n^i steps . If this is larger than $n^{\log(n)}$ then we set $n_{i+1} = 2^{n_i}$ and add to B all strings of the form $u\#x$, $u \in U$ and $|x| = |u|$, of length between n_i+1 and n_{i+1} and go to the next stage.

Query the Σ^p_2 oracle using sufficient padding, to find m, the size of the largest string output by $R_i(0^n)$ on any path. To diagonalize against this machine we must make sure that the strings we put in A of length between $n+1$ and m are such that the probability that the machine outputs any of these strings is less than $p+\frac{1}{poly(n)}$.

A simple way to ensure this is as follows: For every $u \in U$, of appropriate size, consider all strings of the form $u\#x$ where $|x| = |u|$. Sort them on the basis of the probability with which $R_i(0^n)$ outputs them. Add to A the p fraction of these strings with the lowest probability. This will ensure that $R_i(0^n)$ outputs a string in A with probability[3] no more than p and we will have successfully diagonalized against R_i. However, since we cannot compute these probabilities exactly we will settle for approximations of these with error approximately $\frac{1}{2^{cn}}$ for some $c > 0$ and perform the above step based on the estimated probabilities.

For lengths m to $n_{i+1} = 2^{n_i}$ we put into A all strings of the form $u\#x$ with $|u| = |x|$. $\square$

All that we need to show to complete the proof is to show that the error in using estimated probabilities does not affect the diagonalization. This is easily seen as follows: For any $u \in U$, let q_u be the probability that $R_i(0^n)$ outputs a string of the form $u\#x$ where x is any string of length $|u|$. Clearly $\sum_{u \in U} q_u \leq 1$.If we use exact probabilities and put into A the lowest p fraction then the probability that $R_i(0^n)$ outputs a string of the form $u\#x$ which are in A would at most be pq_u. Thus, the probability that $R_i(0^n)$ outputs a string in $A^{>n}$ would at most $\sum_{u \in U} pq_u \leq p$. When we use estimated probabilities then the estimate of q_u is at most $q_u(1 + \epsilon)$ where ϵ is at most $\frac{1}{2^{cn}}$ by Lemma 5.3. If we take the lowest p fraction based on estimates then the estimated probability that $R_i(0^n)$ outputs one of these strings is at most $pq_u(1 + \epsilon)$. The actual probability of this can at most be $pq_u\frac{1+\epsilon}{1-\epsilon}$. Thus, the probability that $R_i(0^n)$ outputs a strings that is in $A^{>n}$ is at most $p\frac{1+\epsilon}{1-\epsilon}$ which is much lower than $p + \frac{1}{poly(n)}$ and hence the diagonalization is successful. $\square$

We note that similar results hold for deterministic classes like $\tilde{E}$ or nice classes containing $\tilde{E}$.

[3]The small error of $\frac{1}{2^{\frac{n}{2}}}$ in approximating a p fraction can be ignored.

5.4 Separations for $\leq_m^{\mathrm{bpp}}$-reductions

In this section we consider reductions with two sided error. We construct sets which are complete under deterministic reductions but not under $\leq_m^{\mathrm{bpp}}$-reductions. We also obtain separation results for completeness under $\leq_m^{\mathrm{bpp}}$-reductions with different probabilities. To obtain *optimal* separation results for $\leq_m^{\mathrm{bpp}}$-reductions we employ a technique that is different from that of Watanabe [Wat87]: We construct highly structured sets which meet certain diagonalization requirements and then explicitly find sets which do not $\leq_m^{\mathrm{bpp}}$ reduce to the constructed sets beyond a certain probability.

We illustrate our results on $\leq_m^{\mathrm{bpp}}$-reductions by constructing a set A which is $\leq_m^{\mathrm{bpp}}$-complete for E^{Σ_2} with probability $\frac{2}{3}$ but not with probability $\frac{2}{3}+\frac{1}{poly(n)}$. In fact the set we construct is complete under $\leq_{3-\mathrm{disj}}^{p}$-reductions and $\leq_{3-\mathrm{conj}}^{p}$-reductions and hence we have an example of a set which is $\leq_{3-tt}^{p}$-complete but not $\leq_m^{\mathrm{bpp}}$-complete with probability $\frac{2}{3}+\frac{1}{poly(n)}$. This separation is also optimal in the sense that a $\leq_{3-tt}^{p}$-complete set may be both $\leq_{3-\mathrm{conj}}^{p}$-complete and $\leq_{3-\mathrm{disj}}^{p}$-complete and hence $\leq_m^{\mathrm{bpp}}$-complete with probability $\frac{2}{3}$. Since the set is $\leq_{3-\mathrm{disj}}^{p}$-complete it is also $\leq_m^{\mathrm{rp}}$-complete with probability $\frac{1}{3}$ and similarly it is also $\leq_m^{\mathrm{co-rp}}$-complete with probability $\frac{1}{3}$. This is also an optimal separation between $\leq_m^{\mathrm{rp}}$-completeness and $\leq_m^{\mathrm{bpp}}$-completeness since a set which is $\leq_m^{\mathrm{rp}}$-complete with probability $\frac{1}{3}$ could also be $\leq_m^{\mathrm{co-rp}}$-complete with probability $\frac{1}{3}$ and hence $\leq_m^{\mathrm{bpp}}$-complete with probability $\frac{2}{3}$. The proof generalizes to yield similar optimal separations between $\leq_m^{\mathrm{bpp}}$-completeness and $\leq_{k-tt}^{p}$-completeness and between $\leq_m^{\mathrm{bpp}}$-reductions with different probabilities.

The following analogy illustrates the intuition behind the construction of A: Suppose we put in A 'few' strings of the form $0x\{1,2,3\}$ and 'most' strings of the form $1x\{1,2,3\}$ while maintaining $\leq_{3-\mathrm{conj}}^{p}$-completeness and $\leq_{3-\mathrm{disj}}^{p}$-completeness. If we can then construct L_A such that any high probability reduction from L_A to A must either produce 'lots' of strings in A of the form $0x\{1,2,3\}$ or produce 'lots' of strings in $\overline{A}$ of the form $1x\{1,2,3\}$ then the properties of A would ensure that L_A can not reduce to A via high probability reductions. In the actual construction the terms 'few', 'most' etc. would refer to probabilities and not numbers.

Theorem 5.5 *There exists a set A which is complete for E^{Σ_2} under $\leq^p_{3-\mathrm{disj}}$ and $\leq^p_{3-\mathrm{conj}}$ reductions (and hence $\leq^{\mathrm{bpp}}_m$-complete with probability $\frac{2}{3}$) but not $\leq^{\mathrm{bpp}}_m$-complete with probability $\frac{2}{3}+\frac{1}{poly(n)}$.*

Proof: Let U be the standard complete set for E^{Σ_2}. The set A will be a subset of $(0+1)\Sigma^*(1+2+3)$ and have the following form:

1. The strings $1u1$, $1u2$, $1u3$ will be in A for all $u \in U$.
2. Exactly two of the strings $1\overline{u}1$, $1\overline{u}2$, $1\overline{u}3$ will be in A for each $\overline{u} \in \overline{U}$.
3. Exactly one of the strings $0u1$, $0u2$, $0u3$ will be in A for each $u \in U$.
4. None of the strings $0\overline{u}1$, $0\overline{u}2$, $0\overline{u}3$ will be in A for $\overline{u} \in \overline{U}$.

It is easy to check that conditions 1 and 2 ensure that A is hard for E^{Σ_2} under $\leq^p_{3-\mathrm{conj}}$-reductions and under probability $\frac{1}{3}$ $\leq^{\mathrm{co-rp}}_m$-reductions. Also conditions 3 and 4 ensure that A is hard for E^{Σ_2} under $\leq^p_{3-\mathrm{disj}}$reductions and under probability $\frac{1}{3}$ $\leq^{\mathrm{rp}}_m$-reductions. The following lemma shows that A is also hard for E^{Σ_2} under probability $\frac{2}{3}$ $\leq^{\mathrm{bpp}}_m$-reductions.

Lemma 5.4 $U \leq^{\mathrm{bpp}}_m A$ with probability $\frac{2}{3}$.

Proof of Lemma: Consider the following random reduction from U to A:

- On input x, choose i uniformly from {0,1} and j uniformly from {1,2,3} and output ixj.

If $x \in U$ then the only cases when $ixj \notin A$ are when $i = 0$ and $0xj$ is one of the two such strings not in A (see condition 3). Thus the error probability is $\frac{1}{3}$. Similarly if x is not in U then an error occurs only when $i = 1$ and ixj is one of the 2 such strings not in $\overline{A}$ (condition 2). Thus, the reduction is a probability $\frac{2}{3}$ $\leq_m^{\mathrm{bpp}}$-reduction. □

A is constructed by diagonalizing against all probabilistic polynomial time transducers R_i to ensure that for each R_i, for infinitely many n, the following two conditions are satisfied:

(a) For each $u \in U$, the probability that $R_i(0^n)$ outputs a string of the form $0uj$ *in* A is at most $\frac{1}{3} + \delta$ of the probability that it outputs *any* string of the form $0uj$, where $j \in \{1, 2, 3\}$. Here δ is a very small ($\frac{1}{2^{cn}}$ for some $c > 0$) error parameter.

(b) For each $\overline{u} \in \overline{U}$, the probability that $R_i(0^n)$ outputs a string of the form $1\overline{u}j$ *in* A is at least $\frac{2}{3} - \delta$ of the probability that it outputs *any* string of the form $1\overline{u}j$.

A is constructed in stages. Let $n_0 = 1$ and $A_0 = \emptyset$. Stage i tries to diagonalize against R_i.

Stage i: Let $n = n_i$. By assumption, the running time of $R_i(0^n)$ is at most n^i. If this exceeds $n^{\log(n)}$ then set $n_{i+1} = 2^{n_i}$, extend A by adding all strings of the form $0u1$, $1u1$, $1u2$, $1u3$, $1\overline{u}1$ and $1\overline{u}2$ of lengths between $n_i + 1$ and n_{i+1}, where $u \in U$ and $\overline{u} \in \overline{U}$ and go to the next stage.

Otherwise, use the Σ_2^p oracle to find m, the length of the largest string output by $R_i(0^n)$ on some path. Add to A strings of the form $1u1$, $1u2$ and $1u3$ where $u \in U$ and $n + 1 \leq |1uj| \leq m$. To meet diagonalization requirement (**a**), for each $u \in U$ such that $n+1 \leq |1uj| \leq m$, estimate the probabilities that $R_i(0^n)$ outputs $0uj$, where $j \in \{1, 2, 3\}$. Among the three strings $0u1$, $0u2$ and $0u3$ put into A, the string with the *lowest* estimated probability. To meet requirement (**b**), for each $\overline{u} \in \overline{U}$ such that $n+1 \leq |1\overline{u}j| \leq m$, estimate the probabilities that $R_i(0^n)$ outputs $1\overline{u}j$, and put into A the two strings with the *largest* and *second largest* estimated probability.

It is clear that if exact probabilities were used in this step then both the conditions would be satisfied with $\delta = 0$. As we have done before we can argue that the use of estimates only adds a small (inverse exponential) error parameter δ. Thus, both conditions are satisfied and we say that R_i has been diagonalized at length n. For lengths between $m+1$ and $n_{i+1} = 2^{n_i}$ add to A all strings of the form $0u1$, $1u1$, $1u2$, $1u3$, $1\overline{u}1$ and $1\overline{u}2$ where $u \in U$ and $\overline{u} \in \overline{U}$. □

By construction A is in E^{Σ_2}. We now explicitly construct a set L_A in E^{Σ_2} which does not reduce to A by a probability $\frac{2}{3} + \frac{1}{poly(n)}$ reduction.

Lemma 5.5 There exists a set L_A in E^{Σ_2} which is not reducible to A by a probability $\frac{2}{3} + \frac{1}{poly(n)}$ $\leq_m^{\mathrm{bpp}}$-reduction.

Proof: The language L_A is defined as follows: On input $< i, 0^n >$:

- Construct $A^{\leq n}$. Since there are at most $|\Sigma|^{n+1}$ elements to be tried, this can be done in E^{Σ_2}. If on some path $R_i(0^n)$ runs for more than $n^{\log(n)}$ steps then we **reject**.

- Define the following probabilities:

$$p \stackrel{\text{def}}{=} \mathrm{Prob}[R_i(< i, 0^n >) \text{ outputs a string in } A^{\leq n}].$$

$$q \stackrel{\text{def}}{=} \mathrm{Prob}[R_i(< i, 0^n >) \text{ outputs a string in } \overline{A}^{\leq n} \text{ or a string not in } (0+1)\Sigma^*(1+2+3)].$$

$$r \stackrel{\text{def}}{=} \mathrm{Prob}[R_i(< i, 0^n >) \text{ outputs a string in } 0\Sigma^*(1+2+3) \text{ of length greater than } n].$$

$$s \stackrel{\text{def}}{=} \mathrm{Prob}[R_i(< i, 0^n >) \text{ outputs a string in } 1\Sigma^*(1+2+3) \text{ of length greater than } n].$$

Note that $p+q+r+s = 1$. Estimate the probabilities p and s with high accuracy (within a multiplicative factor of $1+\delta$ where δ is inverse exponential). If the estimated probabilities $\tilde{p}$ and $\tilde{s}$ sum to less than $\frac{1}{2}$ then **accept** else **reject**. □

The intuition why L_A can not reduce to A with 'high' probability is as follows: If $< i, 0^n > \in L_A$ then a reduction from L_A to A must produce of strings in A with high probability. Since s is 'small' in this case that means a 'large' probability must be concentrated on strings the form $0xj$ *in* A. But condition **(a)** in the construction of A ensures that this can not happen. A similar conflict occurs under the assumption that $< i, 0^n > \notin L_A$ due to condition **(b)** and the definition of L_A.

We now carefully formalize this argument to show that there is no probability $\frac{2}{3} + \frac{1}{poly(n)} \leq_m^{\text{bpp}}$-reduction from L_A to A. Assume R_{i_0} carries out such a reduction L_A to A. Let R_j be the machine which, on input 0^n, simulates $R_{i_0}(< i_0, 0^n >)$. By the construction of A, R_j is diagonalized at infinitely many lengths. Consider a large n such that $R_j(0^n)$ runs for at most $n^{\log(n)}$ steps and such that R_j is diagonalized at this length in the construction of A.

Suppose $< i_0, 0^n >$ belongs to L_A. Then since R_{i_0} reduces L_A to A, R_j outputs a string in $\overline{A}$ with probability less than $\frac{1}{3} - \frac{1}{poly(n)}$. By definition of L_A we know that $\tilde{p} + \tilde{s} < \frac{1}{2}$ and hence $q + r \geq \frac{1}{2} - \frac{1}{exp}$. The probability $q + r$ can be apportioned into the following disjoint parts:

1. Probability q on strings in $\overline{A}^{\leq n}$ and strings not in the set $(0+1)\Sigma^*(1+2+3)$.

2. Probability r_1 on strings of the form $0\overline{u}j$.

3. Probability r_{21} on strings of the form $0uj$, $j \in \{1, 2, 3\}$, which are in A.

4. Probability r_{22} is on strings of the form $0uj$, $j \in \{1, 2, 3\}$, which are not in A.

Note that only the strings which account for r_{21} are in A. Thus, we have that $q + r_1 + r_{22} < \frac{1}{3} - \frac{1}{poly(n)}$, since this a bound on the probability that R_j outputs a string in $\overline{A}$. Therefore,

$$r_{21} \geq \frac{1}{2} - \frac{1}{exp} - (\frac{1}{3} - \frac{1}{poly(n)}) = \frac{1}{6} - \frac{1}{exp} + \frac{1}{poly(n)}.$$

But by construction of A, since R_j is diagonalized at this length we have (from condition **(a)** of the construction of A) that $r_{21} \leq (\frac{1}{3} + \delta)(r_{21} + r_{22})$. Since r_{22} is less than $\frac{1}{3} - \frac{1}{poly(n)}$ we can derive that

$$r_{21} < \frac{1}{6} + \frac{1}{exp} - \frac{1}{poly(n)},$$

which contradicts the above lower bound for r_{21}. Thus, we can not have $< i_0, 0^n >\in L_A$.

So $< i_0, 0^n >\notin L_A$. Then by definition, R_j outputs a string in A with probability less than $\frac{1}{3} - \frac{1}{poly(n)}$. By construction of L_A we know that $\tilde{p} + \tilde{s} \geq \frac{1}{2}$ and hence $p + s \geq \frac{1}{2} - \frac{1}{exp}$. The probability $p + s$ is as before apportioned into four disjoint parts as follows:

1. Probability p is on strings in $A^{\leq n}$.
2. Probability s_1 is on strings of the form $1uj$, $j \in \{1, 2, 3\}$.
3. Probability s_{21} is on strings of the form $1\overline{u}j$, $j \in \{1, 2, 3\}$, which are in A.
4. Probability s_{22} is on strings of the form $1\overline{u}j$, $j \in \{1, 2, 3\}$, which are not in A.

Only strings which contribute to s_{22} are in $\overline{A}$. Thus, $p + s_1 + s_{21} < \frac{1}{3} - \frac{1}{poly(n)}$ and hence

$$s_{22} \geq \frac{1}{2} - \frac{1}{exp} - (\frac{1}{3} - \frac{1}{poly(n)}) = \frac{1}{6} - \frac{1}{exp} + \frac{1}{poly(n)}.$$

But by the construction of A since we diagonalize the machine R_j at this stage we have $s_{21} \geq (\frac{2}{3} - \delta)(s_{21} + s_{22})$. Since s_{21} is less than $\frac{1}{3} - \frac{1}{poly(n)}$ we have $s_{22} < \frac{1}{6} + \frac{1}{exp} - \frac{1}{poly(n)}$ which contradicts the above lower bound for s_{22}. Thus, we can not have $< i_0, 0^n >\notin L_A$.

Since in either case ($< i_0, 0^n >\in L_A$ and $< i_0, 0^n >\notin L_A$), we obtained a contradiction, it must be the case L_A does not $\leq_m^{\mathrm{bpp}}$-reduce to A with probability $\frac{2}{3} + \frac{1}{poly(n)}$. □

Using the same technique we can prove the following:

Theorem 5.6 *For any k there is a set which is $\leq^p_{k-\mathrm{disj}}$ complete and $\leq^p_{k-\mathrm{conj}}$ complete for E^{Σ_2} (and hence $\leq^{\mathrm{bpp}}_m$-complete with probability $\frac{1}{2}+\frac{1}{2k}$) but not $\leq^{\mathrm{bpp}}_m$-complete with probability $\frac{1}{2}+\frac{1}{2k}+\frac{1}{poly(n)}$.*

We can combine the techniques from the above proof and the proof of Theorem 5.6 to obtain fine separations between $\leq^{\mathrm{bpp}}_m$-reductions as in the case of $\leq^{\mathrm{rp}}_m$-reductions. We state the following theorem without the proof.

Theorem 5.7 *For any p, there is a set A which is $\leq^{\mathrm{rp}}_m$-complete for E^{Σ_2} with probability $2p-1$ and also $\leq^{\mathrm{co-rp}}_m$-complete with probability $2p-1$ (and hence $\leq^{\mathrm{bpp}}_m$-complete with probability p) but which is not $\leq^{\mathrm{bpp}}_m$-complete with probability $p+\frac{1}{poly(n)}$.*

As before we note similar theorems will hold for the class $\tilde{E}$ and nice class which contain $\tilde{E}$.

5.5 Discussion

We have shown that the class of complete sets under random reductions is very sensitive to very small changes in the success probability of reductions. We can extend the arguments to show that even smaller differences ($\frac{1}{2^{\epsilon n}}$) are enough to yield a different class of complete sets.

Most of the separation results obtained in the previous sections rely crucially on the fact that the class in question is closed under complement. However by directly extending the techniques of Homer [Hom90] one can show that some of the results of Section 5.2 hold for nondeterministic exponential time classes as well. Since there are relativized worlds where BPP=NEXP we can not obtain separation results for two sided random reductions in NEXP without first separating the base classes. However, even though we know RP≠NEXP we are unable to obtain separations of completeness under one sided error random reductions in NEXP and we leave this as an open question.

Chapter 6

Random Reductions and Sparse Sets [1]

The class NP plays a central role in structural complexity theory. Great effort has gone into understanding the structure of this class both for theoretical and practical reasons. In 1977, drawing from the evidence available, Berman and Hartmanis postulated their famous conjecture that all NP-complete sets are p-isomorphic [BH77]. This drew a lot of attention from the complexity theory community. Since proving the isomorphism conjecture is at least as hard as proving P $\neq$ NP, efforts were concentrated on proving that it was false. One way to do so is to show that there is a sparse NP-complete set. But, Karp and Lipton showed that even this may be quite difficult by proving that if there is a sparse NP-hard set then PH collapses [KL80]. Then, in a much celebrated result, Mahaney showed that existence of a sparse many-one NP-hard set implies that P = NP and hence, contrary to belief, that NP is feasibly recognizable [Mah82]. This led researchers to investigate the more general problem : For what other types of reductions r, does the existence of a sparse r-hard set for NP imply that P = NP. In a beautiful result, Ogiwara and Watanabe show that this is true for bounded truth-table reductions which are

[1] The results presented in this chapter have been drawn from [RR92]

much more general than many-one reductions [OW90].

In this chapter we investigate random reductions from NP-complete sets to sparse sets. We will show that the existence of a sparse set which is NP-hard under $\leq_{m}^{co-rp}$-reductions with non-negligible probability, would imply that NP=RP and hence all languages in the Polynomial-time Hierarchy can be recognized feasibly. This provides a strong evidence for the non-existence of such sets. Random reductions from NP-complete sets to sparse sets have been studied earlier by Mahaney and Simon [MS81], but the random reductions they considered did not make any errors but instead could fail to produce any output with small probability. Moreover, they showed that existence of sparse NP-hard sets under these reductions would imply NP=co-NP. We strengthen these results in two ways. Firstly, we consider reductions which are allowed to err in one direction. Secondly we show that existence of sparse NP-hard sets under these reductions implies NP=RP which immediately provides efficient randomized algorithms for the PH. This sort of feasibility is not known to follow from NP=co-NP. Since Mahaney's technique is not easily extendible to random reductions, we use the properties of the left set of SAT to prove our results. The left set of SAT also played a key role in the Ogiwara-Watanabe result.

6.1 Definitions

Convention Since we will be considering random reductions to NP which is a robust class closed under OR_2, OR_ω and majority, the the success probability of the reductions is not of much concern. In this chapter we will use the symbols $\leq_{m}^{rp}$ and $\leq_{m}^{co-rp}$ to denote $\leq_{m}^{rp}$ and $\leq_{m}^{co-rp}$ reductions will success probability 3/4. Due to the robustness of NP all our results hold even if the success probability was defined to be as low as $1/poly$.

We now introduce the language LSAT, which is the left set of the language SAT. The language LSAT played a major role in the Ogiwara-Watanabe proof.

We will be using it extensively in this chapter.

Definition LSAT $= \{ < x, w > \mid \exists w' \; w \leq w'$ and w' satisfies $x \}$. Here x is a boolean formula and $w, w' \in \{0,1\}^n$ are truth assignments to its n variables, represented as bit strings.

We shall often be talking about intervals. By the interval (w_i, w_j) we mean the set $\{ w \mid w_i \leq w \leq w_j \}$. We assume that all of $w, w_i, w_j \in \{0,1\}^t$ for some t. In this chapter whenever we talk of ordering of strings, we mean the lexicographical ordering. When we talk of ordering of intervals, we mean the lexicographical ordering according to their left end-point. An interesting and very useful property of LSAT is that, if $< x, w > \in$ LSAT and $w' \leq w$ then $< x, w' > \in$ LSAT.

6.2 Main Result

Theorem 6.1 *For any sparse set S, if* SAT $\leq_{\mathrm{m}}^{\mathrm{co-rp}} S$ *then* NP=RP.

Proof: Let S be a sparse set such that SAT $\leq_{\mathrm{m}}^{\mathrm{co-rp}} S$. Let l be a polynomial such that $|S^{\leq n}| \leq l(n)$. Our aim is to exhibit an RP algorithm for SAT. Recall that this means that given a formula x, the algorithm should accept x with high probability if x is satisfiable and reject x otherwise. To do so, we shall use the language LSAT defined previously. As LSAT $\leq_{\mathrm{m}}^{\mathrm{P}}$ SAT, by composing reductions LSAT $\leq_{\mathrm{m}}^{\mathrm{co-rp}} S$. Let g be this reduction. Hence we have that,

- $< x, w > \in \text{LSAT} \Rightarrow g(< x, w >, z) \in S$ for all z,
- $< x, w > \notin \text{LSAT} \Rightarrow \Pr_z[g(< x, w >, z) \notin S] \geq \frac{3}{4}$,

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

We use the interval search technique developed in [OW90] combined with a new idea to device an RP algorithm for SAT. The algorithm works as follows :

Algorithm RCHECKSAT

Input : A boolean formula x having n variables.

- Let p denote a polynomial having the property that $|g(<x,w>,z)| \leq p(|x|)$. Let $N = l(p(|x|))$ and consider the interval $(0^n, 1^n)$.

- Partition this interval into N smaller intervals of roughly equal size, that is, no two interval sizes differ by more than 1.

- While there are intervals of size > 1 do

  - Split each interval into two smaller intervals of roughly equal size .

  - $(*)$Eliminate intervals from the set of resulting intervals using the interval elimination procedure described below. With very high probability, this procedure retains the interval containing w_{max}, the lexicographically largest satisfying assignment of x. No more than N intervals will remain after this procedure terminates.

- If any of the remaining w's satisfy x then **accept** else **reject**.

end RCHECKSAT

The main idea in this proof is accomplishing interval elimination in step $(*)$ of the algorithm, that is, eliminating enough intervals efficiently in a way such that with high probability the interval containing w_{max}, is not dropped. We call this the subset technique. The interval elimination procedure is carried out as follows :

Procedure INTERVAL-ELIMINATION

Input : $\{t_1, t_2 \ldots\}$, a collection of intervals ordered from left to right.

- Let w_i denote the leftmost point of the interval t_i.

- For each interval t_i, pick $z_1 \ldots z_m$ at random (m to be specified later) and let $W_i = \{\ g(x, w_i, z_k) \mid 1 \leq k \leq m\ \}$.

- $(\dagger)$Find the least i (if it exists) such that $|\bigcup_{j \leq i} W_j| > N$. Let i_0 denote this index. Eliminate all intervals $t_i, i \geq i_0$.

- Repeat the following until no more changes occur
  - For simplicity, call the remaining intervals consecutively from left to right $t_1, t_2 \ldots$ and rename the W_i's correspondingly.
  - Find the first i such that $W_i \subseteq \bigcup_{j<i} W_j$. Remove t_{i-1}.

end INTERVAL-ELIMINATION

We now show that this interval elimination procedure retains the interval containing w_{max} with very high probability.

Intervals are eliminated by the elimination procedure in two cases. In the first case, an interval t_i is dropped if $i \geq i_0$ (see †). We claim that this interval cannot contain w_{max}, for if it did, then each of $< x, w_1 >, \ldots, < x, w_{i_0} >$ would be in LSAT. Hence $W_1, \ldots, W_{i_0}$ would all be subsets of S by the definition of the reduction g. Hence, $\bigcup_{i \leq i_0} W_i$ is a subset of S. Since there are at most N elements of S which g can map onto, and $| \bigcup_{i \leq i_0} W_i| > N$, we get a contradiction.

For the second case note the following : Let t_{max} be the the interval that contains w_{max} (if x is satisfiable). By definition of g, for all $i \leq max$, $W_i \subseteq S$. Also, for all $i > max$, the probability that $W_i \subseteq S$ is less than $1/4^m$. Hence, the probability that any of the W_i, $i > max$, is a subset of S is less than $2N \times 1/4^m$. Note that, t_{max} cannot be eliminated unless there is an interval t_k to its right such that $W_k \subseteq \bigcup_{j \leq max} W_j \subseteq S$. Hence the probability that t_{max} is eliminated in any single execution of the interval elimination procedure is less than $2N/4^m$. Since this procedure is executed at most n times by the algorithm, the probability that t_{max} is ever eliminated is less than $n \times 2N \times 1/4^m$ which can be made arbitrarily small by the choice of a proper polynomial for m.

We now prove that after the elimination procedure terminates, no more than N intervals remain. The elimination of the intervals i_0 and beyond (see †) ensures that there are at most N elements in the remaining intervals. Suppose at the end more than N intervals remain. Call the first $N+1$ intervals $t_1 \ldots t_{N+1}$ and the corresponding sets $W_1 \dots W_{N+1}$. Since for all $i, W_i \not\subseteq \bigcup_{j<i} W_j$, by induction it can be shown that the cardinality of $\bigcup_{j\leq k} W_j$ is at least k. Therefore, $|\bigcup_{j\leq(N+1)} W_j| \geq N+1$ which is a contradiction.

It is easy to see that algorithm RCHECKSAT runs in polynomial time. It never accepts unsatisfiable formula and accepts satisfiable formulas with high probability. Hence, it is an RP algorithm for SAT. □

Using the same idea, we can establish the following:

Theorem 6.2 *For any sparse set S, if* $\overline{\mathrm{SAT}} \leq_{\mathrm{m}}^{\mathrm{co-rp}} S$ *then* NP=RP.

Theorem 6.3 *For any sparse set S, if* SAT $\leq_{\mathrm{conj}}^{\mathrm{p}} S$ *then* P=NP.

This result has been obtained independently by Arvind *et al* [AHHK92]. Our technique also proves the result for $\overline{\mathrm{SAT}}$ but this was already known [Ukk83,Yap83]. From the above theorems we also get the following corollaries:

Corollary 6.1 For any tally set T, if SAT(or $\overline{\mathrm{SAT}}$) $\leq_{\mathrm{m}}^{\mathrm{rp}}$ T then NP=RP.

Corollary 6.2 For any tally set T, if SAT(or $\overline{\mathrm{SAT}}$) $\leq_{\mathrm{disj}}^{\mathrm{p}}$ T then P=NP.

The result for SAT in Corollary 6.2 also follows from [Ukk83,Yap83] and the result for $\overline{\mathrm{SAT}}$ also follows from the independent work of Arvind *et al* [AHHK92].

6.3 Open Problems

1. Can the similar results be obtained for $\leq_{\mathrm{m}}^{\mathrm{rp}}$-reductions to sparse sets ?
2. Can similar results be obtained for $\leq_{\mathrm{disj}}^{\mathrm{p}}$-reductions to sparse sets ? This may be difficult in light of the fact that $\leq_{\mathrm{disj}}^{\mathrm{p}}$-reductions to sparse sets are at least as powerful as bounded truth-table reductions to sparse sets [EHOW91].

Bibliography

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

[AHHK92] V. Arvind, Y. Han, L. Hemachandra, and J. Köbler. Reductions to sets of low information content. In *Proceedings of the 19th International Colloquium on Automata, Languages, and Programming*, Springer Verlag *Lecture Notes in Computer Science #650*, pages 162–173, 1992.

[AKL+79] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovasz, and C. Rackoff. Random walks, universal sequences and the complexity of maze problems. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 218–223, 1979.

[ALM+92] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 14–23, 1992.

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

[AS92] S. Arora and S. Safra. Probabilistic checking of proofs; a new characterization of np. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 2–13, 1992.

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

[BDG90] J. L. Balcazar, J. Diaz, and J. Gabarro. *Structural Complexity I.* volume 11 of *EATCS Monographs on Theoretical Computer Science.* Springer-Verlag, 1990.

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

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

[Bei91] R. Beigel. Bounded queries to sat and the boolean hierarchy. *Theoretical Computer Science*, 84(3):199–223, 1991.

[BF90] L. Babai and L. Fortnow. A characterization of #P by arithmetic straight line programs. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 26–34, 1990.

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

[BH77] L. Berman and J. Hartmanis. On isomorphism and density of NP and other complete sets. *SIAM Journal on Computing*, 6:305–322, June 1977.

[BHT90] H. Buhrman, S. Homer, and L. Torenvliet. On complete sets for nondeterministic classes. Technical Report 90-013, Department of Computer Science, Boston University, 1990.

[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.

[CH86] J. Cai and L. A. Hemachandra. The Boolean hierarchy: Hardware over NP. In *Structure in Complexity Theory*, Springer-Verlag *Lecture Notes in Computer Science #223*, pages 105–124, 1986.

[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.

[Cha91] R. Chang. *On the Structure of NP Computations under Boolean Operators.* Ph.D. dissertation, Cornell University, 1991.

[CK90a] R. Chang and J. Kadin. The Boolean hierarchy and the polynomial hierarchy: a closer connection. In *Proceedings of the 5th Structure in Complexity Theory Conference*, pages 169–178, July 1990.

[CK90b] R. Chang and J. Kadin. On computing Boolean connectives of characteristic functions. Technical Report 90-1118, Department of Computer Science, Cornell University, May 1990.

[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. To appear in the *Journal of Computer and System Sciences.*

[CR93] S. Chari and P. Rohatgi. On completeness under random reductions. In *Proceedings of the 8th Structure in Complexity Theory Conference*, pages 176–184, May 1993.

[CRS93] S. Chari, P. Rohatgi, and A. Srinivasan. Randomness optimal unique element isolation with applications to perfect matching and related problems. In *ACM Symposium on Theory of Computing*, pages 458–467, 1993.

[EHOW91] E.Allender, L. Hemachandra, M. Ogiwara, and O. Watanabe. Relating equivalence and reducibility to sparse sets. In *Proceedings of the 6th Structure in Complexity Theory Conference*, pages 220–229, 1991.

[FGL+91] U. Feige, S. Goldwasser, L. Lovasz, S. Safra, and M. Szegedy. Approximating clique is almost np complete. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 2–12, 1991.

[GJY87] J. Goldsmith, D. Joseph, and P. 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.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GY89] R. L. Graham and A. C. Yao. On the improbability of reaching byzantine agreements. In *ACM Symposium on Theory of Computing*, pages 467–478, 1989.

[Hau14] F. Hausdorff. Grundzüge der mengenlehre. Leipzig, 1914.

[Hel86] H. Heller. On relativized exponential and probabilistic complexity classes. *Information and Control*, 71:231–243, 1986.

[Hem87] L. Hemachandra. *Counting in Structural Complexity Theory*. Ph.D. dissertation, Cornell University, 1987.

[Hom90] S. Homer. Structural properties of nondeterministic complete sets. In *Proceedings of the 5th Structure in Complexity Theory Conference*, pages 3–10, July 1990.

[HU79] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

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

[Kad89] J. Kadin. $P^{NP[\log n]}$ and sparse Turing complete sets for NP. *Journal of Computer and System Sciences*, 39:282–298, December 1989.

[KL80] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *ACM Symposium on Theory of Computing*, pages 302–309, 1980.

[KM81] K. Ko and D. Moore. Completeness, approximation and density. *SIAM Journal on Computing*, 10:787–796, 1981.

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

[KUW86] R. M. Karp, E. Upfal, and A. Wigderson. Constructing a perfect matching is in Random *NC*. *Combinatorica*, 6(1):35–48, 1986.

[LFKN90] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 2–10, 1990.

[LLS75] R. E. Ladner, N. A. Lynch, and A. L. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1(2):103–123, 1975.

[LP81] H. R. Lewis and C. H. Papadimitriou. *Elements of the Theory of Computation*. Prentice-Hall, 1981.

[LY93] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. In *ACM Symposium on Theory of Computing*, pages 286–293, 1993.

[Mah82] S. Mahaney. Sparse complete sets for NP: Solution of a conjecture of Berman and Hartmanis. *Journal of Computer and System Sciences*, 25(2):130–143, 1982.

[MS81] S. Mahaney and J. Simon. Polynomial self-reducibility and sparseness. Manuscript, 1981.

[MVV87] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[OW90] M. Ogiwara and O. Watanabe. On polynomial time bounded truth-table reducibility of np sets to sparse sets. In *ACM Symposium on Theory of Computing*, pages 457–467, 1990.

[PY84] C. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). *Journal of Computer and System Sciences*, 28(2):244–259, April 1984.

[PZ83] C. Papadimitriou and S. Zachos. Two remarks on the power of counting. In *Sixth GI Conference on Theoretical Computer Science*, pages 269–276. Springer-Verlag *Lecture Notes in Computer Science #145*, 1983.

[Rab80] M. O. Rabin. Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12:128–138, 1980.

[Roh92] P. Rohatgi. Saving queries with randomness. In *Proceedings of the 7th Structure in Complexity Theory Conference*, pages 71–83, June 1992. To appear in the *Journal of Computer and System Sciences*.

[RR92] D. Ranjan and P. Rohatgi. On randomized reductions to sparse sets. In *Proceedings of the 7th Structure in Complexity Theory Conference*, pages 239–242, June 1992.

[Sch89] U. Schöning. Probabilistic complexity classes and lowness. *Journal of Computer and System Sciences*, 39(1):84–100, 1989.

[Sha90] A. Shamir. IP = PSPACE. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 11–15, 1990.

[Sto83] L. J. Stockmeyer. The complexity of approximate counting. In *ACM Symposium on Theory of Computing*, pages 118–126, 1983.

[TO91] S. Toda and M. Ogiwara. Counting classes are at least as hard as the polynomial-time hierarchy. In *Proceedings of the 6th Structure in Complexity Theory Conference*, pages 2–12, July 1991.

[Tod89] S. Toda. On the computational power of PP and $\oplus$P. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 514–519, 1989.

[Ukk83] E. Ukkonen. Two results on polynomial time truth-table reductions to sparse sets. *SIAM Journal on Computing*, 12(3):580–587, 1983.

[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] K. 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.*

[Wat87] O. Watanabe. A comparison of polynomial time completeness notions. *Theoretical Computer Science*, 54:249–265, 1987.

[Wil85] C. Wilson. Relativized circuit complexity. *Journal of Computer and System Sciences*, 31(2):169–181, 1985.

[WW85] K. Wagner and G. Wechsung. On the Boolean closure of NP. In *Proceedings of the 1985 International Conference on Fundamentals of Computation Theory*, Springer-Verlag *Lecture Notes in Computer Science #199.* pages 485–493, 1985.

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

[ZH86] S. Zachos and H. Heller. A decisive characterization of BPP. *Information and Control*, 69(1–3):125–135, 1986.