Generalized Lowness and Highness and Probabilistic Complexity Classes

Andrew Klapper
University of Manitoba

Abstract

We introduce generalized notions of low and high complexity classes and study their relation to structural questions concerning bounded probabilistic polynomial time complexity classes. We show, for example, that for a bounded probabilistic polynomial time complexity class $\mathcal{C} = BP\Sigma_k^P$, $L\mathcal{C} = H\mathcal{C}$ implies that the polynomial hierarchy collapses to $\mathcal{C}$. This extends Schöning's result for $\mathcal{C} = \Sigma_k^P$ ($L\mathcal{C}$ and $H\mathcal{C}$ are the low and high sets defined by $\mathcal{C}$.) We also show, with one exception, that containment relations between the bounded probabilistic classes and the polynomial hierarchy are preserved by their low and high counterparts. $LBPP$ and $LBPNP$ are characterized as $NP \cap BPP$ and $NP\cap$co-$BPNP$, respectively. These characterizations are then used to recover of Boppana, Hastad, and Zachos's result that if co-$NP \subset BPNP$, then the polynomial hierarchy collapses to $BPNP$, and Ko's result that if $NP \subset BPP$, then the polynomial hierarchy collapses to BPP.

Keywords: lowness, bounded probabilistic class, structural complexity theory, polynomial hierarchy

1 Introduction

Recently, increasing attention has been given to classes of problems for which fast solutions exist that involve probabilistic computation. Such problems lie in various probabilistic complexity classes, which can be defined in terms of sequences of existential, probabilistic, and universal quantifiers, in much the same way as the complexity classes in the polynomial hierarchy are defined by sequences of existential and universal quantifiers. Examples of such classes include BPP, originally defined by Gill [5], and further studied, e.g., by Sipser [16], and Schöning [13], the various forms of Arthur-Merlin combinatorial game classes, defined by

Babai [2], the interactive proof systems, defined by Goldwasser, Micali, and Rackoff [7], and further studied, e.g., in [8], [1], [6], [4], and others, $BPNP$, defined by Schöning [14], and, more generally, the probabilistic complexity classes defined by sensible pairs of sequences of quantifiers, defined by Hinman and Zachos [9], and studied in [17], [18], and [3]. This list is not intended to be exhaustive, but rather to give an outline of some of the major papers in the growing body of work in this field.

A great many identities have been proved among these classes. For example, $BPNP = AM$ (see [14]) $= IP$ (see [8]) $= (\exists^+\exists/\exists^+\forall)$ (see [9]). Much of what is known about identities and inclusions between these classes is summarized in section 2. It turns out that a large number of these classes are equal to classes in what can be called the bounded probabilistic polynomial hierarchy, $\{BP\Sigma_k^P, BP\Pi_k^P, k = 0, 1, \ldots\}$. It has been shown in [9] that these classes in fact lie in the polynomial hierarchy. In fact, for $k \geq 1$, $BP\Sigma_k^P \subset \Pi_{k+1}^P$. A basic question is what containment relationship exists between BPP and NP. That is, which (if either) is stronger, randomness or nondeterminism?

A major outstanding problem of complexity theory is whether the polynomial hierarchy collapses, and, if so, to what level. Much work has been done attempting to relate the collapse of the polynomial hierarchy to other structural phenomena. Schöning [12] defined two hierarchies of complexity classes - the low and high hierarchies $\{L_k^P\}$ and $\{H_k^P\}$ - lying inside NP. He went on to show that the polynomial hierarchy collapses to the kth level if and only if $L_k^P \cap H_k^P \neq \emptyset$ if and only if $L_k^P = H_k^P$. In [13] Schöning showed that (1) $NP \cap$co-$BPNP \subset L_2^P$, and graph isomorphism is in co-$BPNP$. It follows that if graph isomorphism is NP-complete, then the polynomial hierarchy collapses to Σ_2^P. This can also be proved using the result of Boppana, Hastad, and Zachos [3] that if co-NP is contained in $BPNP$, then the polynomial hierarchy collapses to $BPNP$. We note also that Ko [10] showed that if NP is contained in BPP, then the polynomial hierarchy collapses to BPP.

The purpose of this paper is to extend Schöning's lowness and highness techniques to the bounded probabilistic polynomial complexity classes. In particular, in section 3 we extend the definitions of lowness and highness to the bounded probabilistic polynomial classes (these definitions, in fact, apply to any complexity class for which we have a reasonable notion of extension by oracles). For each $k \geq 0$, this gives classes $L\mathcal{C}$ and $H\mathcal{C}$ for each class $\mathcal{C}$ in the bounded probabilistic polynomial hierarchy. We then show (with two exceptions) that the inclusions among the bounded probabilistic polynomial classes carry over to their low and high counterparts. Section 4 contains our main results. We first show that for each such $\mathcal{C}$, the polynomial hierarchy collapses to $BP\mathcal{C}$ if and only if $LBP\mathcal{C} \cap HBP\mathcal{C} \neq \emptyset$ if and only if $LBP\mathcal{C} = HBP\mathcal{C}$. We then characterize $LBPP$ as $NP \cap BPP$, and characterize $LBPNP$ as $NP\cap$ co-$BPNP$. These facts are then used to recover the hierarchy collapsing results mentioned above.

It is a pleasure to thank Prof. S. Zachos for advice which helped lead to these results. I am also grateful to Prof. U. Schöning and Prof. L. Longpré for their many suggestions, and to the participants in the joint Northeastern-BU complexity seminar for support and useful discussions.

2 Bounded Probabilistic Classes

In this section the definition of bounded probabilistic complexity classes is recalled, as defined for example by Schöning [15], along with that of Hinman and Zachos' [9] more general classes defined by sensible pairs of quantifier sequences. The known structural relations among these classes are summarized.

Definition 1 *Let $\mathcal{C}$ be any complexity class. Then $L \in BP\mathcal{C}$, if and only if there is a language $K \in \mathcal{C}$ and polynomials $p(n)$ and $q(n) > 1$ such that $x \in L$ implies $Prob\{y \mid |y| = p(|x|) \wedge (x,y) \in K\} > 1 - 2^{-q(|x|)}$, and $x \notin L$ implies $Prob\{y \mid |y| = p(|x|) \wedge (x,y) \in K\} < 2^{-q(|x|)}$.*

More generally, we can consider complexity classes defined by sequences of quantifiers. We use a notation due to Hinman and Zachos [9]. All quantifiers in this paper will be $\exists$, $\forall$, or $\exists^+$, where $\exists^+ y|P(y)$ means $Prob\{y \mid P(y)\} > 2/3$ (or, more generally, $\exists^+ P(x,y)$ means there are polynomials $p(n)$, $q(n)$ such that, given input x, $Prob\{y, |y| = p(|x|) \mid P(x,y)\} > 2^{-q(|x|)}$). In general, quantifiers are assumed to have range bounded polynomially in the length of the input. Note that many of the definitions here also make sense for the quantifier $\Re$, meaning "at least fifty percent", but we will not discuss classes defined by $\Re$ here.

We can now define quantifier classes as follows

Definition 2 (Zachos)
1. *A pair (Q/Q') of finite sequences of quantifiers is sensible if they are of the same length and for any predicate $P(x,y)$, $(Q'y : \neg P(x,y) \Rightarrow \neg Qy : P(x,y))$.*
2. *Let $P(x,y)$ be a predicate and (Q/Q') be a sensible pair. If, for every x, either $(Qy : P(x,y))$ or $(Q'y : \neg P(x,y))$, (these sets are disjoint by sensibility) then the language $\{x \mid Qy : P(x,y)\}$ will be denoted by $(Q/Q')P$. The complement of L is $\{x \mid Q'y : \neg P(x,y)\}$. If K is a language, we will write $(Q/Q')K$ for $(Q/Q')((x,y) \in K)$, and, if $\mathcal{C}$ is a complexity class, we will write $(Q/Q')\mathcal{C}$ for the class of all languages of the form $(Q/Q')K$, $K \in \mathcal{C}$.*

We also write (Q/Q') for $(Q/Q')P$ (where P is the class of polynomial time computable languages). Note that co-$(Q/Q')K = (Q'/Q)$co-K. Many of the classes defined by sensible quantifier pairs are known by other names. For example,

1. $(\exists/\forall) = NP$;

2. $(\exists\forall\ldots Q_k/\forall\exists\ldots Q'_k) = \Sigma_k^P$; $(\forall\exists\ldots Q_k/\exists\forall\ldots Q'_k) = \Pi_k^P$;

3. $(\exists^+/\forall) = R$;

4. $(\exists^+/\exists^+) = BPP$;

5. $(\exists^+\exists/\exists^+\forall) = (\forall\exists/\exists^+\forall) = BPNP = AM$ (the Arthur-Merlin games, defined originally by Babai [2]. This equality was proved by Hinman and Zachos [9]);

6. $(\exists\exists^+/\forall\exists^+) = (\exists\forall/\forall\exists^+) = MA$ (the Merlin-Arthur games, also defined in [2], the equality proved in [9]);

7. $(\exists^+\exists\forall\ldots Q_k/\exists^+\forall\exists\ldots Q'_k) = (\forall\exists\forall\ldots Q_k/\exists^+\forall\exists\ldots Q'_k) = BP\Sigma_{k-1}^P$.

It has been shown that all sensible quantifier pairs (either when considered as defining complexity classes, or as operators on classes which are closed downward under polynomial time many-one reducibility) reduce to the pairs in the above list, or their complements. We can think of sensible quantifier pairs as operators on complexity classes.

The composition of two quantifier pairs is not, a priori, a quantifier pair. We do, however, have an inclusion in one direction: $(Q/Q')(R/R')\mathcal{C} \subset (QR/Q'R')\mathcal{C}$. The reverse inclusion is only known to hold in special cases. To see the difference between these classes, note that L is in $(QR/Q'R')\mathcal{C}$ if and only if there is a $K \in \mathcal{C}$ such that $x \in L \Rightarrow (QyRz : (x,y,z) \in K)$ and $x \notin L \Rightarrow (Q'yR'z : (x,y,z) \notin K)$. L is in $(Q/Q')(R/R')\mathcal{C}$ if and only if, there is a K such that, in addition, for every (x,y), either $(Rz : (x,y,z) \in K)$ or $(R'z : (x,y,z) \notin K)$ (that is, $\{(x,y) | Rz : (x,y,z) \in K\}$ is in $(R/R')\mathcal{C}$.)

Definition 3 *A pair (R/R') of quantifier sequences is complimentary if for every predicate P, $(\neg Ry : P(x,y)) \Leftrightarrow (Ry : \neg P(x,y))$.*

Note that a complimentary quantifier pair is necessarily sensible. Of the pairs built from $\exists$, $\forall$, and $\exists^+$, only $(\exists/\forall)$, $(\forall/\exists)$, and the various compositions of these two pairs are complimentary.

Lemma 1 *Let (R/R') be a complimentary quantifier pair, and let (Q/Q') be a sensible quantifier pair. Then for any class $\mathcal{C}$, $(Q/Q')(R/R')\mathcal{C} = (QR/Q'R')\mathcal{C}$.*

Proof: The inclusion $\subset$ holds in general. To see the reverse inclusion, let $L \in (QR/Q'R')\mathcal{C}$. Then there is a $J \in \mathcal{C}$ such that

$$\begin{aligned} x \in L &\Rightarrow QyRz : (x,y,z) \in J \\ x \notin L &\Rightarrow Q'yR'z : (x,y,z) \notin J. \end{aligned}$$

Let K be the set of pairs (x,y) such that $Rz : (x,y,z) \in J$. We will show that $K \in (R/R')\mathcal{C}$, and that $L = (Q/Q')K$, proving the lemma.

To see the first assertion, note that

$$\begin{aligned} (x,y) \in K &\Rightarrow Rz : (x,y,z) \in J \\ (x,y) \notin K &\Rightarrow \neg Rz : (x,y,z) \in J \\ &\Rightarrow R'z : (x,y,z) \notin J \end{aligned}$$

To see the second assertion, note that

$$\begin{aligned} x \in L &\Rightarrow QyRz : (x,y,z) \in J \\ &\Rightarrow Qy : (x,y) \in K \\ x \notin L &\Rightarrow Q'yR'z : (x,y,z) \notin J' \\ &\Rightarrow Q'y\neg Rz : (x,y,z) \in J' \\ &\Rightarrow Q'y : (x,y) \notin K \end{aligned}$$

$\square$

The following identities and inclusions are useful in proving the identifications above and in computing compositions of sensible pairs of quantifier sequences. They were proved by Hinman and Zachos [9].

Theorem 1 (Hinman and Zachos) *1.* $(\exists^+/\exists^+) = (\exists^+\forall/\forall\exists^+) = (\forall\exists^+/\exists^+\forall)$*;*

2. $(\exists\forall/\exists\exists^+) \subset (\forall\exists/\exists^+\forall)$*;*

3. $(\exists^+\exists/\exists^+\forall) = (\forall\exists/\exists^+\forall)$*;*

4. $(\exists\exists^+/\forall\exists^+) = (\exists\forall/\forall\exists^+)$*;*

5. For any quantifier Q*,* $(\forall/Q) \subset (\exists^+/Q) \subset (\exists/Q)$ *(where we treat pairs that are not sensible as defining empty classes);*

In order to identify classes defined by oracles, it is desirable to be able to reduce such classes to classes defined by the composition of quantifier pairs. The following lemma allows us to do so in several important cases. First we need a definition.

Definition 4 *Let $\mathcal{C}$ be a complexity class. $\mathcal{C}$ will be said to be robust if for every language K in $\mathcal{C}$ the following two languages, K_1 and K_2, are in $\mathcal{C}$ as well.*

1. *Let # be an additional symbol. Define $K_1 = K\#^*$.*
2. *Let $p(n)$ be a polynomial. Define $K_2 = \{< x_1, \ldots, x_m >\mid x_1, \ldots, x_m \in K, |x_1| = \ldots = |x_m| = n, m \leq p(n)\}$, where $<,>$ is a pairing function.*

Note that in this definition K_1 is many-one equivalent to K, K_2 is d-reducible to K, and K is many-one reducible to K_2. Thus any complexity class which is closed under d-reducibility is robust. In particular, all the classes in the polynomial hierarchy, as well as the bounded probabilistic classes derived from the classes in the polynomial hierarchy are robust.

Lemma 2 *Let $\mathcal{C}$ be a robust complexity class. Let K be any language in $\mathcal{C}$, M a deterministic polynomial time oracle Turing machine, and let $L = L(M, K)$. Then there is a language $K' \in \mathcal{C}$ and a polynomial time nondeterministic oracle Turing machine M' which, after its initial guess, computes deterministically, generating a pair of query strings y_1 and y_2, and accepts if and only if y_1 is accepted and y_2 is rejected by the oracle, such that $L = L(M', K')$.*

Proof: By the first robustness condition, we may assume that all queries made by M are for strings of the same length. More precisely, we may assume that there is a polynomial $p(n)$ such that, on input x, M produces only query strings of length $p(|x|)$. We may also assume there is a polynomial $q(n)$ such that on input x, M makes precisely $q(|x|)$ queries, and that $q(n)$ is of the form $r(p(n))$ for some polynomial $r(n)$. Let K' be produced from K by the second robustness condition, using $r(n)$ as the polynomial. Now we can define M' as follows: on input x, guess a binary string y of length $q(|x|)$. M' maintains a pair of strings, u and v, initially null. M' simulates the behavior of M, but when M queries K with the ith query string z, M' examines the ith bit of y. If this bit is a 1, then M' concatenates z onto u and continues as if the query to K accepted. Otherwise, M' concatenates z onto v and continues as if the query had rejected. If, at the end of the simulation, M rejects, then M' rejects. If, however, M accepts, then M' accepts if and only if u is in K' and v is not in K'. This machine has the desired properties. $\Box$

We refer to the two oracle queries y_1 and y_2 as positive and negative queries, respectively. We can apply this result to Zachos' quantifier classes.

Corollary 1 *Let $\mathcal{C}$ be a robust complexity class. Let M be a deterministic polynomial time oracle Turing machine, let $K \in \mathcal{C}$, and let $(Q\exists/Q'\forall)$ be a sensible pair of quantifier sequences. Suppose $L = (Q\exists/Q'\forall)L(M, K)$ is a well defined language, as in definition 2.2. Then there is a deterministic polynomial time oracle Turing machine M'' and a language $K' \in \mathcal{C}$, with $L = (Q\exists/Q'\forall)L(M'', K')$, such that M'' computes deterministically, then makes two oracle queries y_1 and y_2, accepting if and only if y_1 is accepted and y_2 is rejected by the oracle.*

Proof: By lemma 1, the guessing stage of the machine M' in lemma 2 can be combined with the inner quantifier pair. □

If $\mathcal{C}$ is closed under many-one reducibility, (i.e., $L \leq_m^P K \in \mathcal{C} \Rightarrow L \in \mathcal{C}$) then so is $(Q\exists/Q'\forall)\mathcal{C}$. Therefore, if $\mathcal{C}$ is closed under many-one reducibility, and the negative query can be eliminated, then, in fact, L (in the corollary) is in $(Q\exists/Q'\forall)\mathcal{C}$.

3 Low and High Classes

In this section we generalize the definitions of low and high complexity classes given by Schöning [12]. The purpose of defining these classes is to provide a tool for studying the relationships among complexity classes in the polynomial hierarchy (PH). Schöning [12] has shown that the separation properties of certain classes in PH are reflected in separation properties of the low and high classes in NP.

We will define notions of lowness and highness relative to complexity classes for which there is a reasonable notion of extendibility by adding oracles. This includes, for example, all complexity classes defined by Turing machines with some restrictions on computation paths (e.g., certain numbers of paths accept for input in the given language, or paths below a certain point in the computation tree are polynomial time computable.) We do not have an axiomatic notion of extention of complexity classes by oracles. This is natural since oracles really extend the machines models underlying complexity classes. Strictly speaking, when discussing extensions of complexity classes by oracles, we should define the extension of machine models by oracles. Different machine models for a given class may give different notions of oracle. For example, if $P = NP$, then deterministic and nondeterministic polynomial time bounded Turing machines are machine models for the same class. But it is well known that there is an oracle A such that $P^A \neq NP^A$, that is, the two machine models give different notions of oracle.

However, at the very least, for a complexity class $\mathcal{C}$ containing P to be extendable by oracles we should have, for every language L, a complexity class $\mathcal{C}^L$ such that

1. If $L \in P$ then $\mathcal{C}^L = \mathcal{C}$.

2. For every K, if $L \leq_T^P K$, then $\mathcal{C}^L \subset \mathcal{C}^K$.

3. $\mathcal{C} \subset \mathcal{C}^L$.

We will not explore further here the question of axioms for oracles.

Definition 5 *Let $\mathcal{C}$ be a complexity class with a notion of extension by oracle. Let L be a language in NP. Then L will be said to be low-$\mathcal{C}$ (respectively, high-$\mathcal{C}$) if $\mathcal{C}^L = \mathcal{C}$ (resp., $\mathcal{C}^L = \mathcal{C}^{SAT}$). We denote by $L\mathcal{C}$ (respectively, $H\mathcal{C}$) the set of languages which are low-$\mathcal{C}$ (resp., high-$\mathcal{C}$.)*

For $\mathcal{C} = \Sigma_i^P$ or $\Pi_i^P, i = 0, 1, \ldots$ this definition gives the low and high classes L_i^P and H_i^P as defined by Schöning [12]. We will be particularly interested in the classes $LBP\Sigma_i^P$ and $HBP\Sigma_i^P, i = 0, 1, \ldots$ defined by the bounded probabilistic complexity classess $BP\Sigma_i^P$. For any class $\mathcal{C}$ for which $\mathcal{C}^L$ is defined for any language L, we define $(BP\mathcal{C})^L = BP(\mathcal{C}^L)$. More generally, if (Q/Q') is a sensible pair of quantifier sequences, we define $((Q/Q')\mathcal{C})^L = (Q/Q')(\mathcal{C}^L)$. In all cases in which these classes correspond to classical complexity classes, these definitions are consistent with the traditional definitions of oracles.

Our first goal is to show that the containment relationships that hold for the bounded probabilistic classes and PH hold for the low and high classes derived from them. Recall from [12] that for $k \geq 0$, $L_k^P \subset L_{k+1}^P$, and $H_k^P \subset H_{k+1}^P$. We extend this result to

Theorem 2 *1. For $i \geq 0$, $L_i^P \subset LBP\Sigma_i^P$, and $H_i^P \subset HBP\Sigma_i^P$.*

2. For $i \geq 1$, $LBP\Sigma_i^P \subset L_{i+1}^P$, and $HBP\Sigma_i^P \subset H_{i+1}^P$.

3. $LBPP \subset LMA \subset LBPNP$, and $HBPP \subset HMA \subset HBPNP$.

Proof:

1. Let $A \in L_i^P$. Then $(\Sigma_i^P)^A = \Sigma_i^P$. By lemma 1, $(BP\Sigma_i^P)^A = (\exists^+/\exists^+)(\Sigma_i^P)^A = (\exists^+/\exists^+)(\Sigma_i^P) = BP\Sigma_i^P$, so $A \in LBP\Sigma_i^P$.

   Let $A \in H_i^P$. Then $(\Sigma_i^P)^A = (\Sigma_i^P)^{SAT}$. Consequently, $(BP\Sigma_i^P)^A = (\exists^+/\exists^+)(\Sigma_i^P)^A = (\exists^+/\exists^+)(\Sigma_i^P)^{SAT} = (BP\Sigma_i^P)^{SAT}$, so $A \in HBP\Sigma_i^P$.

2. First note that for $i \geq 1$

$$\begin{array}{rcll} \Pi_{i+1}^P & = & (\forall/\exists)\Sigma_i^P & \\ & \subset & (\forall/\exists)(\exists^+/\exists^+)\Sigma_i^P & \\ & = & (\forall/\exists)BP\Sigma_i^P & \\ & \subset & (\forall/\exists)\Pi_{i+1}^P & \text{by extending Lautemann [11]} \\ & = & \Pi_{i+1}^P & \text{by lemma 1.} \end{array}$$

Moreover, all these relations hold relative to an oracle. It follows that for any language A, $(\Pi_{i+1}^P)^A = ((\forall/\exists)BP\Sigma_i^P)^A = (\forall/\exists)(BP\Sigma_i^P)^A$.

Now let $A \in LBP\Sigma_i^P$. Then $(BP\Sigma_i^P)^A = BP\Sigma_i^P$. We will show that $(\Pi_{i+1}^P)^A = \Pi_{i+1}^P$, and it will follow that $A \in L_{i+1}^P$. We have

$$\begin{array}{rcl} (\Pi_{i+1}^P)^A & = & (\forall/\exists)(BP\Sigma_i^P)^A \\ & = & (\forall/\exists)BP\Sigma_i^P \\ & = & \Pi_{i+1}^P \end{array}$$

The second inclusion is proved similarly.

3. Let $A \in LBPP$, so $BPP^A = BPP$. We have

$$\begin{array}{rcl} MA^A & = & (\exists\exists^+/\forall\exists^+)^A \\ & \subset & (\exists\exists^+/\forall\exists^+)(\exists^+/\exists^+)^A \\ & = & (\exists\exists^+/\forall\exists^+)(\exists^+/\exists^+) \\ & \subset & (\exists\exists^+/\forall\exists^+) \\ & = & MA \end{array}$$

So $A \in LMA$.

Next let $A \in LMA$, so $MA^A = MA$. We have

$$\begin{array}{rcll} BPNP^A & = & (\exists^+\exists/\exists^+\forall)^A & \\ & = & (\exists^+/\exists^+)(\exists/\forall)^A & \text{by lemma 1} \\ & \subset & (\exists^+/\exists^+)(\exists\exists^+/\forall\exists^+)^A & \\ & = & (\exists^+/\exists^+)MA^A & \\ & = & (\exists^+/\exists^+)MA & \\ & \subset & (\exists^+\exists\exists^+/\exists^+\forall\exists^+) & \\ & \subset & (\exists^+\exists^+\exists/\exists^+\exists^+\forall) & \text{by theorem 1} \\ & = & (\exists^+\exists/\exists^+\forall) & \\ & = & BPNP & \end{array}$$

So $A \in LBPNP$.

The remaining inclusions are proved similarly. □

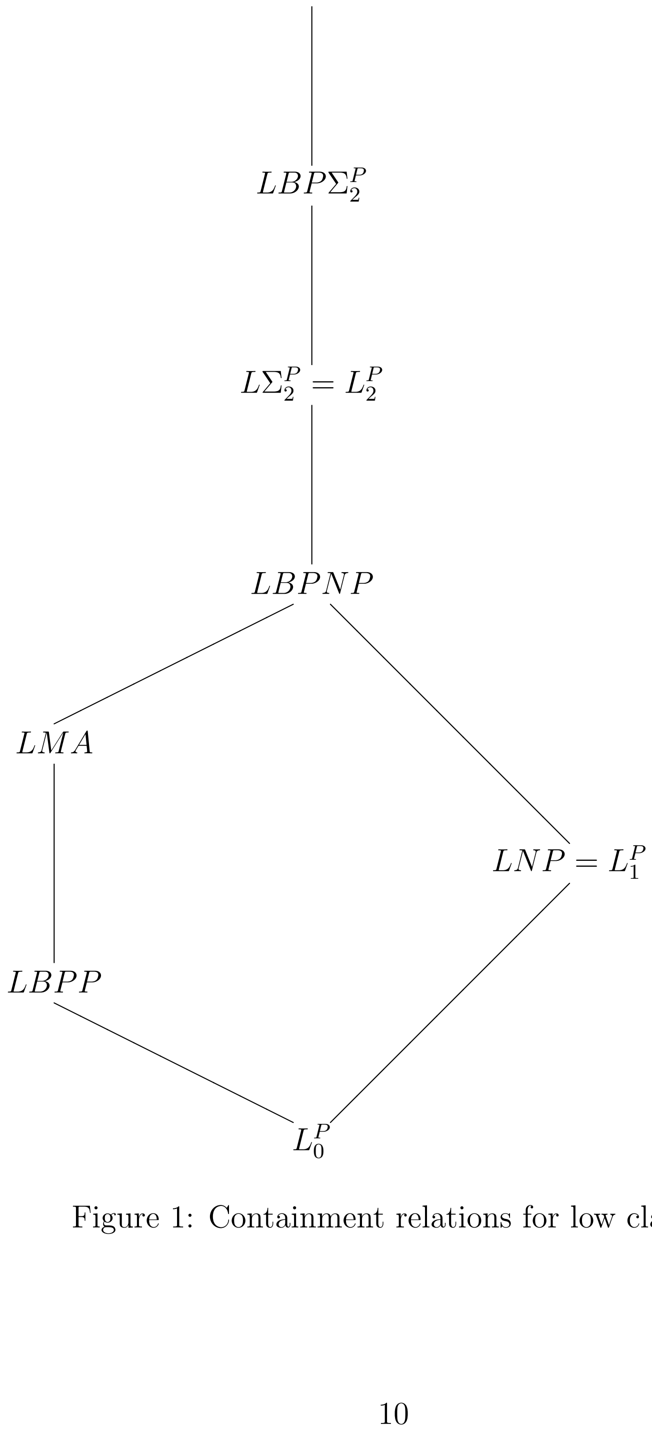

Figure 1: Containment relations for low classes

Figure 3 describes the inclusions known among the low classes. Replacing "L" by "H" in this figure gives the known inclusions among the high classes. Note that while $NP \subset MA$, we have not been able to show that $L_1 \subset LMA$. On the other hand, we view the inability to prove a containment between $LBPP$ and L_1 as a reflection of the inability to prove a containment between BPP and NP. More generally, given complexity classes $\mathcal{C} \subset \mathcal{D}$, both extendable by oracles, the question arises whether $L\mathcal{C} \subset L\mathcal{D}$ and $H\mathcal{C} \subset H\mathcal{D}$. The dependence of the above proofs on characteristics of the specific complexity classes involved makes these general inclusions seem unlikely. The possibility remains that axioms can be found that make these inclusions true, while still holding for all generally accepted notions of oracle.

4 Collapsing the Polynomial Hierarchy

Schöning [12] has shown that, for every $k \geq 0$, if the polynomial hierarchy does not collapse to Σ_k^P, then L_k^P is disjoint from H_k^P, and if the polynomial hierarchy collapses to Σ_k^P, then $L_k^P = H_k^P$. We next extend these results to bounded probabilistic classes. We first need to prove that if one level of the bounded probabilistic polynomial hierarchy collapses, then the entire hierarchy (and hence the polynomial hierarchy) collapses.

Lemma 3 *For all $k \geq 0$, if $BP\Sigma_k^P = BP\Sigma_{k+1}^P$, then the polynomial hierarchy collapses to $BP\Sigma_k^P$.*

Proof: It suffices to show inductively that for all $m \geq k$, $BP\Sigma_m^P = BP\Sigma_{m+1}^P$, the initial case being true by hypothesis. If $BP\Sigma_m^P = BP\Sigma_{m+1}^P$, then $BP\Pi_m^P = BP\Pi_{m+1}^P$, so

$$\begin{array}{rcll} BP\Sigma_{m+2}^P & = & (\exists^+\exists/\exists^+\forall)\Pi_{m+1}^P & \\ & \subset & (\exists^+\exists/\exists^+\forall)BP\Pi_{m+1}^P & \\ & = & (\exists^+\exists/\exists^+\forall)BP\Pi_m^P & \text{by hypothesis} \\ & \subset & (\exists^+\exists\exists^+/\exists^+\forall\exists^+)\Pi_m^P & \\ & \subset & (\exists^+\exists^+\exists/\exists^+\exists^+\exists)\Pi_m^P & \text{by theorem 1} \\ & = & (\exists^+\exists/\exists^+\forall)\Pi_m^P & \\ & = & BP\Sigma_{m+1}^P. & \end{array}$$

□

It can be shown similarly that if $\Sigma_{k+1}^P \subset BP\Sigma_k^P$, then the polynomial hierarchy collapses to $BP\Sigma_k^P$, and that if $\Sigma_k^P = BP\Sigma_k^P$, then the polynomial hierarchy collapses to Σ_k^P.

Theorem 3 *For each $k \geq 0$,*

1. *If the polynomial hierarchy does not collapse to $BP\Sigma_k^P$, then $LBP\Sigma_k^P \cap HBP\Sigma_k^P = \emptyset$.*

2. *If the polynomial hierarchy collapses to $BP\Sigma_k^P$, then $LBP\Sigma_k^P = HBP\Sigma_k^P = NP$.*

Proof: If $A \in LBP\Sigma_k^P \cap HBP\Sigma_k^P$, then $BP\Sigma_{k+1}^P \subset BP(\Sigma_k^P)^A \subset BP\Sigma_k^P$, hence $BP\Sigma_{k+1}^P = BP\Sigma_k^P$. The first assertion then follows from lemma 3.

Let $A \in NP$. Then $BP\Sigma_k^P \subset BP(\Sigma_k^P)^A \subset BP\Sigma_{k+1}^P$. If the polynomial hierarchy collapses to $BP\Sigma_k^P$, then these three sets are equal, implying A is in $LBP\Sigma_k^P$ and in $HBP\Sigma_k^P$. □

It is well known that $L_0 = P$ and $L_1 = NP \cap \text{co-}NP$. The next result characterizes $LBPNP$ and $LBPP$.

Proposition 1 *1. $BPNP^{BPNP \cap \text{co-}BPNP} = BPNP$.*

2. $LBPNP = NP \cap co\text{-}BPNP = NP \cap co\text{-}BPNP \cap BPNP$.

3. (Zachos) $BPP^{BPP} = BPP$.

4. $LBPP = NP \cap BPP$.

Proof: To see the assertion, let L be a language in $BPNP^{BPNP \cap \text{co-}BPNP}$. By corollary 1, we may assume that L is recognized by a $BPNP$ oracle machine which, for each computation path, makes a single positive query to a language in $BPNP$, and a single negative query to a language in co-$BPNP$. A negative query to a language in co-$BPNP$ is the same as a positive query to a language in $BPNP$, and, by robustness, two positive queries to $BPNP$ can be replaced by one such query, so L is in $(\exists^+\exists/\exists^+\forall)(\exists^+\exists/\exists^+\forall) = (\exists^+\exists/\exists^+\forall) = BPNP$. This gives the first assertion.

Next we prove the second assertion. By the first assertion, $NP \cap \text{co-}BPNP \subset LBPNP$, since $NP \subset BPNP$. $LBPNP \subset NP$ by definition, so next we show that $LBPNP \subset$co-$BPNP \cap BPNP$. But $L \in LBPNP \Rightarrow BPNP^L = BPNP \Rightarrow L \in BPNP$. Moreover, any machine that queries L can be replaced by one that recognizes the same language, but queries $\overline{L}$. Thus $\overline{L} \in BPNP$, i.e., $L \in$co-$BPNP$.

On the other hand, $NP \cap \text{co-}BPNP \cap BPNP \subset NP \cap \text{co-}BPNP$, hence we get a series of inclusions $NP \cap \text{co-}BPNP \subset LBPNP \subset NP \cap \text{co-}BPNP \cap BPNP \subset NP \cap \text{co-}BPNP$. These inclusions collapse to give equality.

The last assertion follows from the third by a similar argument. □

Finally, we can use proposition 1 to recover a result of Ko [10] and a result of Boppana, Hastad, and Zachos [3] on the collapse of the polynomial hierarchy.

Theorem 4 1. *(Ko) If NP is contained in BPP, then the polynomial hierarchy collapses to BPP.*

2. *(Boppana, Hastad, Zachos) If co-NP is contained in $BPNP$ then the polynomial hierarchy collapses to $BPNP$.*

Proof: If $NP \subset BPP$, then

$$\begin{aligned} LBPP &= NP \cap BPP, \text{by proposition 1} \\ &= NP \end{aligned}$$

Hence $LBPP \cap HBPP$ is nonempty. By theorem 3, the polynomial hierarchy collapses to BPP.

If co-$NP \subset BPNP$ then $NP \subset$co-$BPNP$, so

$$\begin{aligned} LBPNP &= NP \cap \text{co-}BPNP, \text{by proposition 1} \\ &= NP \end{aligned}$$

Hence $LBPNP \cap HBPNP$ is nonempty. By theorem 3, the polynomial hierarchy collapses to $BPNP$. □

For a concrete example of the significance of this problem, recall the problem GRAPH ISO of determining whether two graphs are isomorphic. This problem is known to be in NP, but it is anopen problem whether it is NP-complete. It has been shown ([13]) that GRAPH ISO $\in co - BPNP$. It follows from proposition 1 and theorem 3 that if GRAPH ISO is in $HBPNP$ (which holds, for example, if GRAPH ISO is NP-complete) then the polynomial hierarchy collapses to $BPNP$.

5 Conclusions

We have extended Schöning's lowness and highness techniques to bounded probabilistic polynomial time classes, and proved some basic properties. We have also showed how these techniques can be used to recover known results on the collapse of the polynomial hierarchy. Several questions remain unanswered. We have characterized $LBPP$ and $LBPNP$, in terms similar to the characterizations of L_1^P and L_0^P - in all these cases, $L\mathcal{C}$ =co-$\mathcal{C} \cap NP$. Unless the polynomial hierarchy collapses, this pattern will not persist at or above $\mathcal{C} = \Sigma_2^P$, for then the intersection would give NP, which includes $H\mathcal{C}$. It is not known, however, whether

LMA =co-$MA \cap NP$. The reduction of oracle queries to a single positive and a single negative query is not adequate here, since the inner quantifier pair in MA is $(\exists^+/\exists^+)$.

More generally, we would like characterizations of the remaining low and high sets. High sets appear harder to characterize in general. The known results depend on the existence of complete sets of various types. It may be necessary to invent probabilistic reducibilities in order to characterize high bounded probabilistic sets.

With or without such a characterization, we conjecture that if co-NP is a subset of MA, then the polynomial hierarchy collapses to MA. We would like to see a more universal result that implies all such collapsing conditions.

Finally, we would like to apply these techniques to other classes in the poynomial hierarchy, such as D^P, Δ_2^P, and ZPP^{NP}. By so doing, we would hope to gain insight into several questions raised by Zachos and Fürer [18] concerning containment between these classes and the bounded probabilistic classes.

References

[1] W. Aiello, S. Goldwasser, J. Hastad. "On the power of interaction", *Proceedings of 27th Annual IEEE Symposium on Foundations of Computer Science*, 1986, 368-379.

[2] L. Babai. "Trading group theory for randomness", *Proceedings of 17th Symposium on the Theory of Computation*, 1985, 421-429.

[3] R. Boppana, J. Hastad, S. Zachos. *Does co-NP have short interactive proofs?*, Manuscript, 1986.

[4] A. Condon, R. Ladner. "Probabilistic game automata", *Proceedings of 1st Annual Conference on Structure in Complexity Theory, 1986, Lecture Notes in Computer Science*, **223**, Springer-Verlag, 1986, 144-162.

[5] J. Gill. Computational complexity of probabilistic Turing machines, *SIAM Journal on Computing*, **6**, 1977, 675-695.

[6] O. Goldreich, S. Micali, A. Wigderson. "Proofs that yield nothing but the validity of the assertion and the methodology of cryptographic protocol design", *Proceedings of 27th Annual IEEE Symposium on Foundations of Computer Science*, 1986, 174-187.

[7] S. Goldwasser, S. Micali, C. Rackoff. "The knowledge complexity of interactive proof-systems", *Proceedings of the 17th Annual Symposium on the Theory of Computation*, 1985, 291-304.

[8] S. Goldwasser, M. Sipser. “Private coins versus public coins in interactive proof systems”, *Proceedings of the 18th Annual Symposium on the Theory of Computation*, 1986, 59-68.

[9] P. Hinman, S. Zachos. “Probabilistic machines, oracles, and quantifiers”, *Proceedings of the Oberwolfach Recursion-theoretic Week, Lecture Notes in Mathematics*, **114**, Springer-Verlag, 1984, 159-192.

[10] K. Ko. Some observations on the probabilistic algorithms and NP-hard problems, *Information Processing Letters*, **14**, 39-43.

[11] C. Lautemann. BPP and the polynomial hierarchy, *Information Processing Letters*, **17**, 1983, 215-217.

[12] U. Schöning. A low and high hierarchy within NP, *Journal of Computer and System Sciences*, **27**, 1983, 14-28.

[13] U. Schöning. “Graph isomorphism is in the low hierarchy”, *Proceedings of the 4th STACS, Lecture Notes in Computer Science*, **247**, Springer-Verlag, 1986, 114-124.

[14] U. Schöning. “Complexity and Structure”, *Lecture Notes in Computer Science*, **211**, Springer-Verlag, 1986.

[15] U. Schöning. “Probabilistic complexity classes and lowness”, *Proceedings of Second Annual Conference on Structure in Complexity Theory*, 1987, 2-8.

[16] M. Sipser. “A complexity theoretic approach to randomness”, *Proceedings of the 15th Annual ACM Symposium on the Theory of Computation*, 1983, 330-335.

[17] S. Zachos. “Probabilistic quantifiers, adversaries, and complexity classes”, *Proceedings of the First Annual Conference on Structure in Complexity Theory, Lecture Notes in Computer Science*, **223**, Springer-Verlag, 1986, 383-400.

[18] S. Zachos, M. Fürer. Probabilistic quantifiers vs. distrustful adversaries, Manuscript, 1985.