MIT/LCS/TM-252

ON BPP

Stathis Zachos

Hans Heller

December 1983

On BPP

Stathis Zachos, Hans Heller
MIT Laboratory for Computer Science
Cambridge, Ma

Abstract

It is shown that $L \in BPP$ iff $(x \in L \rightarrow \exists_m y \forall z P(x,y,z)) \wedge (x \notin L \rightarrow \forall y \exists_m z \neg P(x,y,z))$ for a polynomial time predicate P and for $|y|,|z| \leq poly(|x|)$, where $\exists_m y \Phi(y)$ means that $Pr(\{y | \Phi(y)\}) > 1/2 + \varepsilon$ for a fixed ε. Note that even the weaker conditions $\exists y \forall z P(x,y,z)$ and $\forall y \exists z \neg P(x,y,z)$ contradict each other and thus decide whether $x \in L$. Some of the consequences of the above are that various probabilistic polynomial time hierarchies collapse as well as that probabilistic oracles for algorithms as low as e.g. Σ_2^P do not add anything to the computing power of the corresponding classes; i.e. $NP^{NP^{BPP}} = NP^{NP}$.

Keywords: Probabilistic algorithms, polynomial time complexity classes, oracles, polynomial hierarchies.

1. Introduction

Many arguments in the theory of cryptography make use of probabilistic algorithms. The goal is to construct, if possible, (secure) schemes, which cannot be broken by probabilistic algorithms. The assumption is that problems solvable by probabilistic algorithms are easy or tractable; supposedly well below NP-complete problems. But in reality little is known about the power of such probabilistic e.g. BPP-algorithms. Thus a strong motivation for the following considerations is to understand BPP and to classify it as well as possible among other polynomial time complexity classes. Diagram 1 depicts pictorially some of the known inclusion relations among polynomial time complexity classes. For detailed descriptions of the classes we refer to [HU, GJ, G, Z].

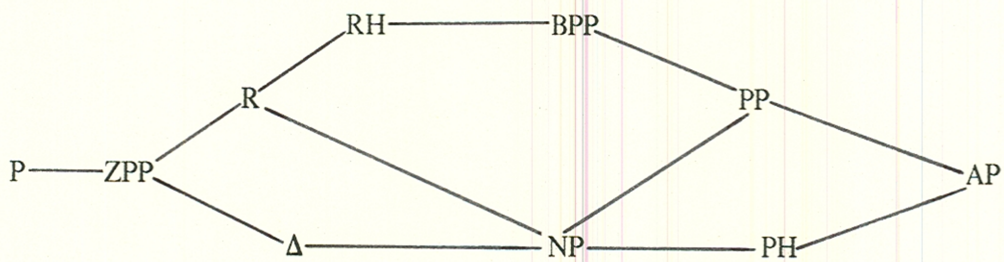

Diagram 1

In the following we are going to make use of some abbreviating notations:

1. In formulas, describing $x \in L$ or $x \notin L$, quantifiers are restricted to range over quantities with length at most a polynomial of the length of x. Thus for example

   $$x \in L \leftrightarrow \exists y \forall z P(x,y,z)$$

   for a polynomial time predicate P, is an alternate characterization of NP^{NP} (NP with oracle from NP) ; see [St, W].

2. $\exists_m y P(x,y)$ denotes that there is an $\varepsilon > 0$ so that for all (inputs) x: $Pr(\{y \mid P(x,y)\}) > 1/2 + \varepsilon$.

Using these notations let us review definitions of some of the above complexity classes:

<u>L∈P</u>: $x \in L \leftrightarrow P(x)$
for some polynomial time predicate P.

<u>L∈NP:</u> $x \in L \leftrightarrow \exists y P(x,y)$
for some polynomial time predicate P.

<u>L∈R:</u> $(x \in L \rightarrow \exists_m y P(x,y)) \wedge (x \notin L \rightarrow \forall y \neg P(x,y))$
for some polynomial time predicate P.

$\Delta = NP \cap \text{co-}NP$

$ZPP = R \cap \text{co-}R$

<u>L∈BPP:</u> $(x \in L \rightarrow \exists_m y P(x,y)) \wedge (x \notin L \rightarrow \exists_m y \neg P(x,y))$
for some polynomial time predicate P.

Note that the above definition of R is decisive in the sense that $\exists y P(x,y)$ is enough to decide that $x \in L$, whereas for BPP this is not the case, because $\exists y P(x,y)$ and $\exists y \neg P(x,y)$ do not contradict each other.

<u>L∈PP:</u> $x \in L \leftrightarrow Pr(\{y | P(x,y)\}) > 1/2$
for some polynomial time predicate P.

For definitions of PH, AP=PSPACE, RH see [HU, CS, Z2].

It is helpful to have an algorithmic model for the above complexity classes in order to intuitively grasp properties of them. Nondeterministic Turing machines running in polynomial time represent the most widely spread computing model. For precise definitions see e.g. [HU, GJ]. For example in case of P all possible computation paths give the correct answer; in case of ZPP many paths give the correct answer, whereas the remaining paths give no answer at all (Las Vegas); in case of Δ there is at least one path that answers correctly, whereas the remaining paths give no answer at all; in case of BPP many paths give the correct answer, whereas the few remaining ones may give a wrong answer (Monte Carlo). Similarly computation trees for NP, R, PP have the known obvious structure.

A nondeterministic Turing machine can be augmented by a query tape and an oracle that can answer queries about some decision problem A without extra time costs. Thus for example NP^{SAT} represents the class of problems that can be solved by a nondeterministic Turing machine with NP behavior that can query an oracle for SAT. We can generalize this by allowing the oracle to be any one of some complexity class: ${C_1}^{C_2} = \{{C_1}^A | A \in C_2\}$. It turns out that some oracle classes collapse: $P^P = P$, $ZPP^{ZPP} = ZPP$, $BPP^{BPP} = BPP$, $\Delta^\Delta = \Delta$, $P^{ZPP} = ZPP$, $P^{BPP} = BPP$, $NP^\Delta = NP$, etc. For others it is known that one query to the oracle is enough to yield the whole power of the class: $NP^{NP} = NP^{NP[1]}$, $\Delta^{NP} = \Delta^{NP[1]}$, $R^R = R^{R[1]}$, $ZPP^R = ZPP^{R[1]}$,

$NP^{BPP} = NP^{BPP[1]}$.

$NP^{NP} = NP^{NP[1]}$ is essential for the alternate characterization of NP^{NP}, i.e. $x \in L \leftrightarrow \exists y \forall z P(x,y,z)$.

For all known inclusions the relativized inclusions are also valid: e.g. $\Delta^R \subseteq NP^R \subseteq NP^{NP}$.

Oracle querying is associative: e.g. $(NP^{NP})^{NP} = NP^{(NP \oplus NP^{NP})} = NP^{(NP^{NP})} = NP^{NP^{NP}}$. To persuade yourself of the above use your favorite model for NP^{NP} computations using oracles.

Another property, that we will be frequently using, is the robustness property of the $\exists_m$ quantifier (consequently of R, ZPP, BPP): The following requirements on a polynomial time predicate P are equivalent:

for a polynomial q and for all x: $Pr(\{y| P(x,y)\}) > 1/2 + 1/q(|x|)$
for a fixed ε and for all x: $Pr(\{y| P'(x,y)\}) > 1/2 + \varepsilon$
for a polynomial q and for all x: $Pr(\{y| P''(x,y)\}) > 1 - 1/2^{q(|x|}$.

Notice that $\exists_m y$ guarantees an overwhelming majority of witnesses.

2. BPP is Contained in the Polynomial Hierarchy

It seems very improbable that NP is contained in BPP. Evidence for this are the following facts:

1. BPP problems can be solved in practice with arbitrary small error probability, whereas this is not known to be the case for all NP problems.

2. Using random oracles, BPP collapses to P with probability one, whereas $NP \neq P$ with probability one [BG].

3. If we assume $NP \subseteq BPP$, we can deduce $R = NP$ and $PH \subseteq BPP$ and PH collapses at the second level, neither of which corresponds to our intuition [K,Z2].

Thus trying to prove $BPP \subseteq NP$ or $BPP \subseteq \Sigma_k^P$ for some $k > 1$, seems to be a more reasonable project. As a matter of fact Sipser showed $BPP \subseteq \Sigma_4^P$ and Gacs and Lautemann improved this to $BPP \subseteq \Sigma_2^P$[S,L]. We give here a simplified proof of this fact, which is the start of several improvements that we are going to prove in the next section; in addition our proof shows that a poly-size circuit argument [A] is basically enough to show $BPP \subseteq NP^{NP}$. In order to demonstrate this we formulate the concept of a comb with polynomially many teeth and then we prove a lemma about combs which we are going to use throughout the paper.

Def.: C_n a comb of size n is a collection of binary numbers (teeth of the comb), such that for all $z \in C_n$ $|z| \leq n$ and $card(C_n) \leq n$.

Remark: C_n can be encoded into one number of polynomial length and decoded from it in time polynomial in n.

Lemma 1: If $\forall x_{|x|<n} \exists_m y_{|y|<n} \Phi(x,y)$ then $\exists C_n \forall x_{|x|<n}$[for some tooth $y \in C_n$: $\Phi(x,y)$]

Proof:

Consider the matrix $M[x,y]=\Phi(x,y)$, $0 \leq x,y \leq 2^n-1$. Thus $\forall x Pr(\{y| M[x,y]=true\}) > 1/2+\varepsilon$ (all rows contain many "true") and therefore $Pr(\{(x,y)| M[x,y]=true\}) > 1/2+\varepsilon$ (M contains many "true") and $\exists y_1$: $Pr(\{x|M[x,y_1]=true\}) > 1/2 + \varepsilon$ (some column contains many "true"). Remove from matrix M all rows x where $M[x,y_1]=true$, remove column y_1 and call the new matrix M'. M' has at most half as many rows as M. M' similarly contains many "true" and thus there is a column y_2 in M' that contains many "true". Proceed analogously to obtain $\{y_1,...,y_n\}$ halving the number of rows each time and thus covering all rows of the original M with $\{y_1,...,y_n\}=C_n$.

q.e.d.

Roughly speaking, Lemma 1 says that we can interchange the quantifiers $\forall x$ and $\exists y$ provided that for all x there are many y.

Remark: If P(x,y,z)is a polynomial time predicate, then $P'(x,C_{p(|x|)},y) = \bigvee_{z \in C_{p(|x|)}} P(x,y,z)$ is also.

Theorem 1: (Sipser, Gacs, Lautemann) $BPP \subseteq NP^{NP}$

Proof:

Let $L \in BPP$ i.e.

$(x \in L \rightarrow \exists_m y\ P(x,y)) \wedge (x \notin L \rightarrow \exists_m y \neg P(x,y))$

Assume w.l.o.g. that

$x \in L \rightarrow Pr(\{y| P(x,y)\}) > 1-1/2^{|x|}$

$x \notin L \rightarrow Pr(\{y| P(x,y)\}) < 1-1/2^{|x|}$

Let p(|x|) be a polynomial bound for the BPP computation on x. Take a comb $C_{p(|x|)}$ and slide it across the leaves of the BPP computation tree. Shift-rotating the comb by s corresponds to replacing every tooth y_i by $u_i=(s+y_i) \bmod 2^{p(|x|)}$.

Claim 1: $x \in L \rightarrow \exists C_{p(|x|)} \forall$shifts s [for a tooth u of the s-shifted comb:P(x,u)]

Proof:

$x \in L \rightarrow$ [because $L \in BPP$] $\exists_m y P(x,y) \rightarrow$ [consider shifts] $\forall s \exists_m y P(x,(s+y) \bmod 2^{p(|x|)}) \rightarrow$ [because of lemma 1] $\exists C_{p(|x|)} \forall s$[for a tooth u: P(x,u)]

q.e.d.1

Claim 2: $\exists C_{p(|x|)} \forall$ shifts s [for a tooth u of the s-shifted comb: $P(x,u)] \rightarrow x \in L$

Proof: by a pigeon hole argument.

There exists a comb $C_{p(|x|)}$ and a $z_i \in C_{p(|x|)}$ such that $\mathrm{card}\{s \mid P(x,(s+z_i) \bmod 2^{p(|x|)})\} > 2^{p(|x|)}/p(|x|)$.

Therefore $\Pr(\{s \mid P(x,(s+z_i) \bmod 2^{p(|x|)})\}) > 1/p(|x|) > 1/2^{|x|}$, for sufficiently large x.

Thus it is not true that $\exists_m y \neg P(x,y)$ and therefore $x \in L$.

q.e.d.2

Thus

$x \in L \leftrightarrow \exists C \forall s \Phi(x,C,s)$

for a polynomial time predicate Φ, where $\Phi(x,C,s) \leftrightarrow$ for some z in C: $P(x,(s+z) \bmod 2^{p(|x|)})$;

i.e. $L \in NP^{NP}$.

q.e.d.

Corollary: $BPP \subseteq \Delta^{NP[1]}$

Proof:

For NP^{NP} one oracle query is enough and BPP is closed under complements.

q.e.d.

Theorem 2: (Gacs) $BPP \subseteq R^{NP}$

Proof: See [Si].

Corollary: $BPP \subseteq ZPP^{NP}$

3. Classes with Decisive Characterization which Contain BPP

In section 2 we have seen that $BPP \subseteq ZPP^{NP}$. For relativizations, however, we know that $BPP^X = P^X$ with probability 1 and $P^X \neq NP^X$ with probability 1 [see BG]. Therefore $BPP^X \subset (ZPP^{NP})^X$ with probability 1. This shows that it will be difficult to prove $BPP = ZPP^{NP}$. For that reason we try to find a tighter characterization of BPP (below ZPP^{NP}). Define a class A of languages by

$L \in A$: $(x \in L \rightarrow \exists_m y \forall z P(x,y,z)) \wedge (x \notin L \rightarrow \forall y \exists z \neg P(x,y,z))$ for some polynomial time predicate P

Proposition: $A \subseteq R^{NP[1]}$

Notice that we do not know, whether $R^{NP[1]} \subseteq A$. For problems in $R^{NP[1]}$ distinct computation paths can query the oracle (once) and receive positive or negative answers, with the use of which then these paths might lead to an accepting answer; whereas in case of A the "paths" only make use of negative answers .

To prove the next theorem we need a stronger version of Lemma 1:

Lemma 2: If $\forall x_{|x|<n} \exists_m y_{|y|<n} \Phi(x,y)$ then $\exists_m C_k \forall x_{|x|<n}$[for some tooth $y \in C_k$: $\Phi(x,y)$] ,where $k = n+2$

Proof:

$Pr(\{C_k | \exists x_{|x|<n}[\text{for all teeth } y\in C_k: \neg\Phi(x,y)]\}) = Pr(\cup_{|x|<n}\{C_k | \text{ for all teeth } y\in C_k: \neg\Phi(x,y)\})$

$\leq \Sigma_{|x|<n} Pr(\{C_k | \text{ for all teeth } y\in C_k: \neg\Phi(x,y)\}) \leq \Sigma_{|x|<n}(1/2)^k = 2^n(1/2)^k < 1/4 < 1/2 - \varepsilon.$

Therefore for most of the C_k $\forall x_{|x|<n}[\text{for some tooth } y \in C_k: \Phi(x,y)]$ holds.

q.e.d.

Theorem 3: BPP $\subseteq$ A

Proof:
Let $L\in$BPP. We will show $L\in$A.

Claim 1: $x\in L \rightarrow \exists_m C_{p(|x|)} \forall \text{shifts } s[\text{for a tooth } u: P(x,u)]$
Proof: As in claim 1 of theorem BPP $\subseteq NP^{NP}$ replacing $\exists C_{p(|x|)}$ by $\exists_m C_{p(|x|)}$ using Lemma 2 instead of Lemma 1.

Claim 2: as in BPP $\subseteq NP^{NP}$

q.e.d.

Corollary: BPP $\subseteq$ A$\cap$co-A $\subseteq ZPP^{NP[1]}$

Define now a class B of languages by

L$\in$B: $(x\in L \rightarrow \forall y\exists_m zP(x,y,z)) \wedge (x\notin L \rightarrow \exists y\forall z\neg P(x,y,z))$ for some polynomial time predicate P.

Proposition: B $\subseteq$ A

Proof:
Let $L\in$B. We will show $L\in$A.
$x\in L \rightarrow \forall y\exists_m zP(x,y,z) \rightarrow$ (by Lemma 2) $\exists_m C_{p(|x|)}\forall y[\text{for a tooth } u\in C_{p(|x|)}: P(x,y,u)]$

On the other hand:
$\exists_m C_{p(|x|)}\forall y[\text{for a tooth } u\in C_{p(|x|)}: P(x,y,u)] \rightarrow \exists C_{p(|x|)}\forall y[\text{for a tooth } u\in C_{p(|x|)}: P(x,y,u)]$

$\rightarrow \forall y\exists C_{p(|x|)}[\text{for a tooth } u\in C_{p(|x|)}: P(x,y,u)] \rightarrow \forall y\exists zP(x,y,z) \rightarrow$ (because $L\in$B) $x\in L$.

Thus L satisfies the definition of A for the polynomial time predicate $\vee_{u \text{ in } C}P(x,y,u)$.

q.e.d.

Theorem 4: BPP $\subseteq$ B

Proof: Let $L \in BPP$.

Claim 1: $x \in L \rightarrow \forall C_{p(|x|)} \exists_m s$[for all teeth $u: P(x,u)$]

Proof:

$x \in L \rightarrow$ (since $L \in BPP$) $\exists_m y P(x,y) \rightarrow \forall u \quad Pr(\{s| \neg P(x,(s+u) \bmod 2^{p(|x|)})\}) < 1/2^{|x|}$

$$\rightarrow \forall C_{p(|x|)} \quad Pr(\{s| \text{ for some tooth } u \text{ in } C_{p(|x|)} \ \neg P(x,(s+u) \bmod 2^{p(|x|)})\}) <$$

$$< \Sigma_{u \in C_{p(|x|)}} 1/2^{|x|} = p(|x|)/2^{|x|} < 1/2 - \varepsilon$$

$$\rightarrow \forall C_{p(|x|)} \exists_m s[\text{for all teeth } u\text{: } P(x,u)]$$

Claim 2: $\forall C_{p(|x|)} \exists s$[for all teeth: $P(x,u)$] $\rightarrow x \in L$

Proof of the contraposition $x \notin L \rightarrow$ ...as in claim 1 of $BPP \subseteq NP^{NP}$

q.e.d.

Corollary: $BPP \subseteq B \cap$ co-B

4. Decisive Characterizations of BPP

We proceed now to define some decisive classes K_1 to K_4 which will first turn out to be all equal. Then we prove that they coincide with BPP.

$L \in K_1$: $(x \in L \rightarrow \exists_m y \forall z P(x,y,z)) \wedge (x \notin L \rightarrow \exists_m z \forall y \neg P(x,y,z))$
for some polynomial time predicate P.

$L \in K_2$: $(x \in L \rightarrow \forall y \exists_m z P(x,y,z)) \wedge (x \notin L \rightarrow \forall z \exists_m y \neg P(x,y,z))$
for some polynomial time predicate P.

$L \in K_3$: $(x \in L \rightarrow \exists_m y \forall z P(x,y,z)) \wedge (x \notin L \rightarrow \forall y \exists_m z \neg P(x,y,z))$
for some polynomial time predicate P.

$L \in K_4$: $(x \in L \rightarrow \forall y \exists_m z P(x,y,z)) \wedge (x \notin L \rightarrow \exists_m y \forall z \neg P(x,y,z))$
for some polynomial time predicate P.

Remark: $K_3 = \text{co-}K_4$, $K_1 = \text{co-}K_1$, $K_2 = \text{co-}K_2$

Lemma: $K_1 \subseteq K_3 \subseteq K_2$, $K_1 \subseteq K_4 \subseteq K_2$

Proof:

Trivial. Note that $\exists_m y \forall z \Phi(y,z)$ implies $\forall z \exists_m y \Phi(y,z)$

q.e.d.

Lemma 3: If $\forall x_{|x|<n} \exists_m y_{|y|<n} \Phi(x,y)$ then $\exists_m C_k \forall C_k'$[for most teeth $y \in C_k$ and $x \in C_k'$: $\Phi(x,y)$], where $k=2n+4$

Proof:

We will first show

$\exists_m C_k \forall x_{|x|<n}$[for most teeth $y \in C_k$: $\Phi(x,y)$]

Then the claim of the lemma follows immediately.

$\Pr(\{C_k | \exists x_{|x|<n}[\text{for most teeth } y \in C_k: \neg\Phi(x,y)]\}) = \Pr(\cup_{|x|<n}\{C_k | \text{for most teeth } y \in C_k: \neg\Phi(x,y)\})$

$\leq \Sigma_{|x|<n}\{C_k | \text{for most teeth } y \in C_k: \neg\Phi(x,y)\}) \leq \Sigma_{|x|<n}(1/2)^{k/2} = 2^n(1/2)^{n+2} = 1/4 < 1/2 - \varepsilon.$

q.e.d.

Theorem 5: $K_2 \subseteq K_1$

Proof:
Let $L \in K_2$

$x \in L \rightarrow \forall y \exists_m z P(x,y,z) \rightarrow$ (Lemma 3) $\exists_m C \forall C'[\text{for most } z \in C \text{ and } y \in C': P(x,y,z)]$

$x \notin L \rightarrow \forall z \exists_m y \neg P(x,y,z) \rightarrow$ (lemma 3) $\exists_m C' \forall C[\text{for most } z \in C \text{ and } y \in C': \neg P(x,y,z)]$

$\rightarrow \exists_m C' \forall C \neg[\text{for most } z \in C \text{ and } y \in C': P(x,y,z)]$

Therefore $L \in K_1$

q.e.d.

Corollary: $K_1 = K_2 = K_3 = K_4 =: K$

Proposition: $K \subseteq B$

Proof: $K_4 \subseteq B$ is obvious

q.e.d.

Theorem 6: $BPP \subseteq K$

Proof:
Let $L \in BPP$.
Similarly as in claim 1 of $BPP \subseteq B$ and claim 1 of $BPP \subseteq A$ we can show that

$x \in L \rightarrow \forall C_{p(|x|)} \exists_m s[\text{for all teeth } u = (z+s) \bmod 2^{p|x|}, z \in C_{p(|x|)}: P(x,u)]$

and $x \notin L \rightarrow \exists_m C_{p(|x|)} \forall s[\text{ for some tooth } u: \neg P(x,u)]$

q.e.d.

Proposition: $K \subseteq BPP$

Proof:
Let $L \in K_2$.
$x \in L \rightarrow \exists_m \langle y,z \rangle: P(x,\langle y,z \rangle)$
$x \notin L \rightarrow \exists_m \langle y,z \rangle: \neg P(x,\langle y,z \rangle)$
Therefore $L \in BPP$.

q.e.d.

From the above then follows our:

MAIN THEOREM: $K = BPP$

5. Various Consequences

Note that any of the K_1, K_2, K_3, K_4 characterizations of BPP are decisive, that is, even if we replace $\exists_m$ quantifiers by $\exists$, the simplified clauses for $x \in L$ and $x \notin L$ contradict each other and thus they allow us to decide whether $x \in L$.

Another interesting fact is that possible probabilistic hierarchies built by $\exists_m \forall$ (resp. $\forall \exists_m$) quantifier repetitions collapse.

e.g.

$L \in BPP$ iff there is some polynomial time predicate P such that

$$(x \in L \rightarrow \exists_m x_1 \forall x_2 \exists_m x_3 \forall x_4 P(x_1,x_2,x_3,x_4))$$
$$\wedge \quad (x \notin L \rightarrow \forall x_1 \exists_m x_2 \forall x_3 \exists_m x_4 \neg P(x_1,x_2,x_3,x_4)), \quad \text{etc.}$$

Our hope was to show $BPP = ZPP^R$, but as we discuss below this does not seem to be an easy problem.

$L \in R_1$:

$$(x \in L \rightarrow \exists_m y \forall z P(x,y,z))$$
$$\wedge \quad (x \notin L \rightarrow \forall y \exists_m z P(x,y,z))$$
$$\wedge \quad \forall x,y(\exists z \neg P(x,y,z) \rightarrow \exists_m z \neg P(x,y,z))$$

for some polynomial time predicate P.

Proposition: $R_1 \subseteq K_3 \subseteq BPP$

Proposition: $R^R = R^{R[1]} = R_1$

Proof:

First note that as in the case of NP^{NP} the base R-machine can itself generate answers of the oracle R, proceed accordingly and at the end verify only those queries for which it assumed a negative answer. Thus for $L \in R^R$ the first two clauses of the definition of R_1 are true. The third clause is necessary to ensure that the oracle machine has R behavior even if its answers do not contribute to the base machines result. On the other hand $R_1 \subseteq R^R$ is clear.

q.e.d.

Thus the reason we could not show BPP $\subseteq R^R$ is that we could not characterize an L$\in$BPP in such a way that the third condition for R_1 would be satisfied. Note that similarly $NP^R \subseteq$ co-B.

The insight we obtained about BPP looking at these proofs, as well as the final characterization BPP = K led us to the following result.

Theorem 7: $NP^{NP^{BPP}} = NP^{NP}$

Sketch of a proof: Let $L \in NP^{NP^{BPP}} = NP^{NP[1]^{BPP[1]}}$. Note that only one query per path and only with the same answer for all paths is enough for the BPP (as well as the NP) oracle.

$x \in L \leftrightarrow \exists x_1 \forall x_2 \exists_m x_3 \forall x_4 P(x,x_1,x_2,x_3,x_4)$

Using lemma 1: $x \in L \leftrightarrow \exists \langle x_1,C \rangle \forall x_2 \bigvee_{x_3 \in C} \forall x_4 P(x,x_1,x_2,x_3,x_4)$

Therefore $L \in NP^{NP}$

q.e.d.

This last theorem shows that using a BPP oracle does not add any computing power to classes as low as $\Sigma_2^P \cap \Pi_2^P$.

6. References

[A] ADLEMAN, L. (1978), Two Theorems on Random Polynomial Time, FOCS 19, 75-83.

[BG] BENNET, C.H., and GILL, J. (1981), Relative to a Random Oracle A, $P^A \neq NP^A \neq co\text{-}NP^A$ with Probability 1, SIAM J. Comput. 10, 96-112.

[CS] CHANDRA, A.K., and STOCKMEYER, L.J,(1976), Alternation, FOCS 17, 98-108.

[GJ] GAREY, M.R., and JOHNSON, D.S., (1979), Computers and Intractabilty: a Guide to the Theory of NP-completeness, Freeman, San Francisco, Ca.

[G] GILL, J., (1977), Computational Complexity of Probabilistic Turing Machines, SIAM J. Comput. 6, 675-695.

[HU] HOPCROFT, J.E., and ULLMAN, J.D., (1979), Introduction to Automata Theory, Languages ,and Computation, Addison-Wesley, Reading, Mass.

[K] Ker-I-Ko, (1982), Some Observations on Probabilistic Algorithms and NP-Hard Problems, Inf. Proc. Let. 14,39-43.

[L] LAUTEMANN, C., (1983), BPP and the Polynomial Hierarchy, Techn. Uni. Berlin, Informatik, Report 83-06.

[Si] SIPSER, M., (1983), A Complexity Theoretic Approach to Randomness, STOC 15, 330-335.

[St] STOCKMEYER, L.J., (1976), The Polynomial-Time Hierarchy, Theoret. Comp. Science, 3, 1-22.

[W] WRATHALL, C. (1976), Complete Sets and the Polynomial-Time Hierarchy, Theoret. Comp. Science, 3, 23-33.

[Z] ZACHOS, S., (1982), Robustness of Probabilistic Computational Complexity Classes under Definitional Perturbations, Information and Control, 54, 143-154.

[Z2] ZACHOS, S., (1983), Collapsing Probabilistic Polynomial Hierarchies, Conf. on Comp. Compl. Th., S. Barbara, 75-81.

Table of Contents

Abstract 0

1. Introduction 1

2. BPP is Contained in the Polynomial Hierarchy 3

3. Classes with Decisive Characterization which Contain BPP 5

4. Decisive Characterizations of BPP 7

5. Various Consequences 9

6. References 11