On Closure Properties of Bounded 2-Sided Error Complexity Classes

Kenneth W. Regan*
SUNY/Buffalo
regan@cs.buffalo.edu

James S. Royer†
Syracuse University
royer@top.cis.syr.edu

June 14, 1993

Abstract

We show that if a complexity class $\mathcal{C}$ is closed downward under polynomial-time majority truth-table reductions ($\leq^{\mathrm{p}}_{\mathrm{mtt}}$), then practically every other "polynomial" closure property it enjoys is inherited by the corresponding bounded 2-sided error class BP[$\mathcal{C}$]. For instance, the Arthur-Merlin game class AM [Bab85] enjoys practically every closure property of NP. Our main lemma shows that for any relativizable class $\mathcal{D}$ which meets two fairly transparent technical conditions, we have $\mathcal{D}^{\mathrm{BP}[\mathcal{C}]} \subseteq \mathrm{BP}[\mathcal{D}^{\mathcal{C}}]$. Among our applications, we simplify the proof by S. Toda [Tod89, Tod91] that the polynomial hierarchy PH is contained in BP[$\oplus$P]. We also show that relative to a random oracle R, PH^R is properly contained in $\oplus\mathrm{P}^R$.

Keywords Computational complexity, theory of computation, polynomial-time hierarchy, randomness, oracles.

AMS/MOS classification(s) 68Q15.

*The first author was supported in part by NSF Grant CCR-9011248
†The second author was supported in part by NSF Grant CCR-89011154.

1. Overview

There has been much interest in relationships among BPP and related bounded two-sided error complexity classes [Ko82, Zac82, ZH86, Sch85, Sch88, Sch89, Kla90]. We show that many of the arguments in these papers are instances of the following "oracle interchange lemma": for complexity classes $\mathcal{C}$ and $\mathcal{D}$ which possess some commonly-found polynomial-time closure properties,

$$\mathcal{D}^{\mathrm{BP}[\mathcal{C}]} \subseteq \mathrm{BP}[\mathcal{D}^{\mathcal{C}}]. \tag{1}$$

Besides tightening several proofs in the literature, our formulation helps us study the exact conditions under which one can use these arguments. We believe that (1) is interesting in itself, especially for classes $\mathcal{D}$ such as NP and $\oplus$P whose computations may involve exponentially many branches and oracle queries, because it says that all the coinflips implicitly used to answer queries to BP[$\mathcal{C}$] can be replaced by polynomially many coinflips made at the outset by a $\mathcal{D}$-machine with an oracle from $\mathcal{C}$ itself.

We give several applications of our lemma, including a short and tighter proof of S. Toda's theorem [Tod89, Tod91] that the polynomial hierarchy PH is contained in BP[$\oplus$P]. Then we show that all "reasonable" closure properties of a class $\mathcal{C}$ meeting our conditions are inherited by its bounded two-sided analogue BP[$\mathcal{C}$]. In a final section we show that relative to a random oracle R, PH^R is properly contained in $\oplus\mathrm{P}^R$.

2. Preliminaries

We use the alphabets $\Sigma := \{0,1\}$ and $\Gamma := \{0,1,\#\}$, where '#' is an extra symbol used to form tuples. Given $x \in \Gamma^*$, there is a unique $k \geq 0$ and list of strings $x_1,\ldots,x_k \in \Sigma^*$ such that $x = x_1\#\cdots\#x_k$. We identify both languages and predicates with 0–1 valued functions (for predicates, $0 \equiv$ false and $1 \equiv$ true). For each $A \subseteq \Gamma^*$, we write $\overline{A}$ for the complement of A in Γ^*. For any language $A \subseteq \Gamma^*$ and string $u \in \Sigma^*$, we define the *projection of A along u*, written $\pi_u(A)$, to be $\{\, x \in \Gamma^* : x\#u \in A \,\}$. We also identify Σ^* with the natural numbers $\mathbf{N}$ in the standard manner. For any $m \geq 0$, Σ^m stands for the set of strings $x \in \Sigma^*$ whose length $|x|$ equals m. For each language A and $m \geq 0$, $A|_m := \{\, x \in A : |x| = m \,\}$ and $A|_{\leq m} := \{\, x \in A : |x| \leq m \,\}$. Double bars denote set cardinality; thus $\|\Sigma^m\| = 2^m$.

Languages over Γ are our main objects of study. These could be recoded over the alphabet Σ, but we prefer to distinguish Σ^* as a domain for quantifiers to range over. For brevity we indicate quantification restricted to strings of a given length m by superscripting m after the quantifier. We also employ the following extended quantifier notation:

Definition 1. For any predicate $Q(\cdot)$, $m \in \mathbf{N}$, and real $\delta \in (0,1)$:

(a) (*threshold counting*) $(T^m_\delta\, y)\, Q(y)$ holds iff $\|\{\, y \in \Sigma^m : Q(y) \,\}\| > \delta \cdot 2^m$.

(b) (*parity counting*) $(\oplus^m\, y)\, Q(y)$ holds iff $\|\{\, y \in \Sigma^m : Q(y) \,\}\|$ is odd.

Several well-known operators on complexity classes are then definable as follows. (We add square brackets to Toda's '$\cdot$' notation in order to clarify the order of application in compound expressions.)

Definition 2. Let $\mathcal{C}$ be any class of languages over Γ. Then a language $A \subseteq \Gamma^*$ belongs to co-$[\mathcal{C}]$ if $\overline{A}$ belongs to $\mathcal{C}$. A belongs to, respectively: (a) NP$[\mathcal{C}]$, (b) coNP$[\mathcal{C}]$, (c) $\oplus$P$[\mathcal{C}]$, (d) BP$[\mathcal{C}]$, (e) RP$[\mathcal{C}]$, iff there is a language $R \in \mathcal{C}$, a polynomial p, and a $\delta > 0$ such that for all $x \in \Gamma^*$, with $m := p(|x|)$:

(a) $A(x) = (\exists^m\, y)\, R(x\#y)$,

(b) $A(x) = (\forall^m\, y)\, R(x\#y)$,

(c) $A(x) = (\oplus^m\, y)\, R(x\#y)$,

(d) $(T^m_{.5+\delta}\, y)[\, A(x) = R(x\#y)\,]$,

(e) $(T^m_\delta\, y)[\, A(x) = R(x\#y)\,]$ and $\neg A(x) \Longrightarrow (\forall^m y)\, \neg R(x\#y)$.

For $\mathcal{C} :=$ P, the above are respectively equivalent to the usual definitions of NP, coNP, $\oplus$P, BPP, and RP in terms of machines. The following is immediate:

Proposition 3. *For any language class* $\mathcal{C}$, co-[NP$[\mathcal{C}]$] $=$ coNP[co-$\mathcal{C}$] *and* co-[BP$[\mathcal{C}]$] $=$ BP[co-$[\mathcal{C}]$]. *In particular,* co-[BP[NP]] $=$ BP[co-[NP]] $=$ BP[coNP]. $\square$

Applying (c) and then (d) to P gives Toda's class BP[$\oplus$P]. This class is also characterized by machines M which on any input x first flip polynomially-many coins to form a string $y \in \Sigma^*$, and then simulates a $\oplus$P machine Q on input $x\#y$. Corresponding to Q there is a P-machine P and a polynomial p such that for any x and y, $Q(x\#y)$ accepts iff the number of strings $z \in \Sigma^{p(|x|)}$ such that $P(x\#y\#z)$ accepts is odd. The natural definition of Toda's class relative to an oracle X, written $(\mathrm{BP}[\oplus\mathrm{P}])^X$, is in terms of oracle machines M of this kind. Without loss of generality we may suppose that M makes all of its coin-flips to form y and guesses to form z at the outset of every computation path, so that the queries are left to the machine P. This shows that for any oracle set X, $(\mathrm{BP}[\oplus\mathrm{P}])^X$ equals $\mathrm{BP}[\oplus\mathrm{P}[\mathrm{P}^X]]$, and also equals $\mathrm{BP}[\oplus\mathrm{P}^X]$.

In an *Arthur-Merlin protocol* [Bab85, BM88], the "Arthur" machine A, given an input x, first flips polynomially many coins and sends the resulting string y to the "Merlin" machine M. M reads $x\#y$ and nondeterministically computes a string z. A deterministic polynomial-time procedure then verifies a predicate $R(x\#y\#z)$. The protocol is said to recognize a language L if there exist $\delta > 0$ and a polynomial p such that for all x, with $m := p(|x|)$,

$$\begin{aligned} x \in L &\implies (T^m_{.5+\delta}\, y)(\exists^m z)\, R(x\#y\#z) \\ x \notin L &\implies (T^m_{.5+\delta}\, y)(\forall^m z)\, \neg R(x\#y\#z). \end{aligned}$$

This is equivalent to $L \in \mathrm{BP}[\mathrm{NP}]$. Relative to an oracle X, we may once again wlog. suppose that A makes all of its coin flips and M makes all of its nondeterministic moves before either machine makes any queries, so that all queries are left to the stage of verifying the P-predicate $R(x\#y\#z)$. This shows that for any oracle set X, $\mathrm{AM}^X = \mathrm{BP}[\mathrm{NP}^X] = \mathrm{BP}[\mathrm{NP}[\mathrm{P}^X]]$.

An attempt to extend this kind of argument to a general result of the form $(op[\mathcal{C}])^X = op[\mathcal{C}^X]$ may be found under the heading "associativity of relativization" in [HZ84]. The following proposition collects all the instances needed for applications in this paper.

Proposition 4. *For any oracle set X:*

$$\begin{array}{ll} \mathrm{NP}^X = \mathrm{NP}[\mathrm{P}^X] & \mathrm{AM}^X = \mathrm{BP}[\mathrm{NP}^X] \\ \oplus\mathrm{P}^X = \oplus\mathrm{P}[\mathrm{P}^X] & (\mathrm{BP}[\oplus\mathrm{P}])^X = \mathrm{BP}[\oplus\mathrm{P}^X] \\ & \\ \mathrm{BPP}^X = \mathrm{BP}[\mathrm{P}^X] & \Sigma_2^{p,X} = \mathrm{NP}[\mathrm{coNP}^X] = \mathrm{NP}[\text{co-}[\mathrm{NP}^X]]. \quad \square \end{array}$$

Last, we collect some known observations which quantify the extent to which probabilities can be amplified by conducting polynomially many repeated trials. A language A is *polynomial-time majority truth-table reducible to* a language B (written: $A \leq^{\mathrm{p}}_{\mathrm{mtt}} B$) iff there is a deterministic polynomial-time procedure which, for each x, constructs a set $S(x)$ of an odd number of strings such that $x \in A \iff$ more than half the strings in $S(x)$ are in B. Since S has polynomial size, this is far from the full power of threshold counting used in describing PP and related counting classes (see [Sch90]), and closure under $\leq^{\mathrm{p}}_{\mathrm{mtt}}$ can be considered a "moderately" strong property of complexity classes. Our slight improvement in the proof of *(d)* is apparently new.

Proposition 5 ([Ko82, Zac82, Sch89, Lau83]). *Let $\mathcal{C}$ be a collection of languages which is closed downward under $\leq^{\mathrm{p}}_{\mathrm{mtt}}$. Then:*

(a) For each $A \in \mathrm{BP}[\mathcal{C}]$ and polynomial r, there is a set $R \in \mathcal{C}$ and a polynomial q such that for all $m > 0$, $(T^{q(m)}_{1-\epsilon}\, y)(\forall x : |x| \leq m)[\, A(x) = R(x\#y)\,]$, where $\epsilon = 2^{-r(m)}$. That is, $(T^{q(m)}_{1-\epsilon}\, y)[\, A|_{\leq m} = (\pi_y R)|_{\leq m}\,]$.

(b) $\mathrm{RP}[\mathcal{C}] \subseteq \mathrm{BP}[\mathcal{C}]$.

(c) $\mathrm{BP}[\mathrm{BP}[\mathcal{C}]] = \mathrm{BP}[\mathcal{C}]$ *and* $\mathrm{RP}[\mathrm{RP}[\mathcal{C}]] = \mathrm{RP}[\mathcal{C}]$.

(d) $\mathrm{BP}[\mathcal{C}] \subseteq \mathrm{NP}[\mathrm{coNP}[\mathcal{C}]]$.

Proof Sketch. The counting arguments in (a) and the consequences (b) and (c) are well-known; the basic idea is that by polynomially many repeated trials, one can make the error probability for any given $x \in \Sigma^{\leq m}$ less than $2^{-(m+1)r(m)}$, so that the chance of failure for even a single such x is bounded by $2^{-r(m)}$. Previous proofs of (d) [Lau83, ZH86] involve the formula

$$x \in A \iff (\exists C \subseteq \Sigma^m)(\forall z \in \Sigma^m)(\text{for some } y \in C)\ R(x\#(y+z)),$$

where C is a collection of polynomially many strings and the addition is bitwise mod 2. They stipulate that the class $\mathcal{C}$ be closed downward under something like polynomial-time disjunctive truth-table reductions ($\leq^{\mathrm{p}}_{\mathrm{dtt}}$). However, a straightforward analysis replaces the 'for some $y \in C$' by 'for most $y \in C$', so that the proofs work for $\mathcal{C}$ closed under $\leq^{\mathrm{p}}_{\mathrm{mtt}}$. $\square$

3. The Interchange Lemma

We identify the notion of a *relativized class* $\mathcal{D}$ with a collection $\{ D_i : i \in \mathbf{N} \}$ of oracle Turing machines, such that for all oracle sets X, $\mathcal{D}^X = \{ L(D_i^X) \}$. Our results do not require that this collection be effectively enumerated. We also write $\mathcal{D}(X)$ for $\mathcal{D}^X$, and $\mathcal{D}^{\mathcal{C}}$ or $\mathcal{D}(\mathcal{C})$ for $\bigcup_{X \in \mathcal{C}} \mathcal{D}^X$. Our reason for the generality in this section is that many reducibility relations and operations on classes can be represented as relativized classes. For instance, the NP$[\,\cdot\,]$ operator is represented by the family $\mathcal{D}$ of polynomial-time bounded NTMs which on any input x, each computation path makes one oracle query, and accepts iff the answer to the query is 'yes.'

Definition 6. A relativized class $\mathcal{D}$ has *polynomially bounded oracle use* iff for each $\mathcal{D}$-machine D there is a polynomial p such that for all n and oracle sets A, B,

$$A|_{\leq p(n)} = B|_{\leq p(n)} \implies L(D^A)|_{\leq n} = L(D^B)|_{\leq n}. \tag{2}$$

This is guaranteed if D on input x makes no queries of length $> p(|x|)$.

Definition 7. A relativized class $\mathcal{D}$ is *closed under oracle projections* if for each $\mathcal{D}$-machine D and oracle set A, there is a $\mathcal{D}$-machine D_* such that for all $x \in \Gamma^*$ and $u \in \Sigma^*$, $D_*^A(x\#u) = D^{\pi_u A}(x)$.

The following stronger condition gives a better intuitive picture of what Definition 7 involves. Given any oracle machine D, define D_π to be a machine which acts as follows, on any input $w \in \Gamma^*$ of the form $x\#u$ with $u \in \Sigma^*$: D_π simulates D on input x, but for each query string q submitted by D, D_π submits the query string $q\#u$ instead. If w does not have this form; i.e., if $w \in \Sigma^*$, then D_π just simulates D on input w.

Definition 8. A relativized class $\mathcal{D}$ is *closed syntactically under oracle projections* if for each $\mathcal{D}$-machine D, D_π is also a $\mathcal{D}$-machine.

Generally speaking, D_π is a machine of much the same "complexity type" as D. For instance, if D is an NP-machine, then so is D_π. If D is an oracle machine which on any input $w \in \Gamma^*$ only submits w itself as a query, then D_π is equivalent to D

itself. These remarks, and the obvious polynomial length bound on queries, suffice to verify that Definitions 6 and 8 hold for all the relativized classes $\mathcal{D}$ considered in this paper, in particular P, NP, coNP, and $\oplus$P. There is one exception: the "$\mathcal{D}(\mathcal{C}) = \mathcal{C} \cap \text{co-}\mathcal{C}$ operator" at the end of Section 5 gives an example of why the closure under oracle projections is needed for the results.

Lemma 9 (BP Interchange Lemma). *Let $\mathcal{D}$ be a relativized class which is closed under oracle projections and has polynomially bounded oracle use. Let $\mathcal{C}$ be a class which is closed downward under $\leq^{\mathrm{p}}_{\mathrm{mtt}}$. Then $\mathcal{D}^{\mathrm{BP}[\mathcal{C}]} \subseteq \mathrm{BP}[\mathcal{D}^{\mathcal{C}}]$.*

Proof. Let D be a $\mathcal{D}$-machine. The object is to show that for each oracle set $B \in \mathrm{BP}[\mathcal{C}]$, the language $L(D^B)$ is in $\mathrm{BP}[\mathcal{D}^{\mathcal{C}}]$. Take p to be a polynomial which bounds the oracle use of D as in (2). Then by Proposition 5a, there is a set $R \in \mathcal{C}$ and a polynomial q such that for all m, we have $(T^{q(m)}_{3/4}\, z)[\, B|_{\leq m} = (\pi_z R)|_{\leq m}\,]$, and, hence, setting $m = p(n)$,

$$(T^{q(p(n))}_{3/4}\, z)[\, B|_{\leq p(n)} = (\pi_z R)|_{\leq p(n)}\,].$$

Thus, since p bounds the oracle use of D, we have that

$$\text{for all } x \in \Gamma^*,\ (T^{q(p(|x|))}_{3/4}\, z)[\, D^B(x) = D^{\pi_z R}(x)\,].$$

By the closure of $\mathcal{D}$ under oracle projections, there is a $\mathcal{D}$-machine D_* such that for all $x \in \Gamma^*$ and $z \in \Sigma^*$, $D^R_*(x\#z) = D^{\pi_z R}(x)$. Hence,

$$\text{for all } x \in \Gamma^*,\ (T^{q(p(|x|))}_{3/4}\, z)[\, D^B(x) = D^R_*(x\#z)\,].$$

Since $R \in \mathcal{C}$, we have $L(D^B) \in \mathrm{BP}[\mathcal{D}^{\mathcal{C}}]$, as required. ☐

4. Applications

Our first applications are direct corollaries of Proposition 4 and the interchange lemma:

Corollary 10. *(a)* [Ko82] $\mathrm{BPP}^{\mathrm{BPP}} = \mathrm{BPP}$.

(b) [ZH86, Sch89] $\Sigma_2^{p,\mathrm{BPP}} = \Sigma_2^p$.

(c) [ZH86] $\mathrm{NP}^{\mathrm{BPP}} = \mathrm{NP}[\mathrm{BPP}]$.

(d) AM *equals the closure of* P *under the operators* $\mathrm{NP}[\,\cdot\,]$ *and* $\mathrm{BP}[\,\cdot\,]$.

Proof. (a) $\mathrm{BPP}^{\mathrm{BPP}} = \mathrm{BP}[\mathrm{P}^{\mathrm{BP}[\mathrm{P}]}] \subseteq \mathrm{BP}[\mathrm{BP}[\mathrm{P}^{\mathrm{P}}]]$ (by Lemma 9) $= \mathrm{BPP}$.

(b) $\Sigma_2^{p,\mathrm{BPP}} = (\mathrm{NP}[\mathrm{coNP}])^{\mathrm{BPP}} = \mathrm{NP}[\mathrm{coNP}^{\mathrm{BPP}}] \subseteq \mathrm{NP}[\mathrm{BP}[\mathrm{coNP}]]$ (by Lemma 9). By Proposition 5d we have $\mathrm{BP}[\mathrm{coNP}] \subseteq \mathrm{NP}[\mathrm{coNP}[\mathrm{coNP}]] = \mathrm{NP}[\mathrm{coNP}]$. Therefore, $\mathrm{NP}[\mathrm{BP}[\mathrm{coNP}]] \subseteq \mathrm{NP}[\mathrm{NP}[\mathrm{coNP}]] = \Sigma_2^p$.

(c) $\mathrm{NP}^{\mathrm{BPP}} = \mathrm{NP}[\mathrm{P}^{\mathrm{BPP}}] = \mathrm{NP}[\mathrm{BPP}]$.

(d) Clearly AM is contained in the closure, and $\mathrm{BP}[\mathrm{AM}] = \mathrm{AM}$. Let $\mathcal{D}$ represent the $\mathrm{NP}[\,\cdot\,]$ operator as above; then $\mathrm{NP}[\mathrm{AM}] = \mathcal{D}(\mathrm{BP}[\mathrm{NP}]) \subseteq \mathrm{BP}[\mathcal{D}(\mathrm{NP})]$ (by Lemma 9) $= \mathrm{BP}[\mathrm{NP}[\mathrm{NP}]] = \mathrm{AM}$. $\square$

Remark: Part (d) does not directly imply L. Babai's theorem that the constant-round Arthur-Merlin game hierarchy collapses to AM, because the definition given in [Bab85] is not the same as iterations of the $\mathrm{NP}[\,\cdot\,]$ and $\mathrm{BP}[\,\cdot\,]$ operators. To represent the inductive relation "Player A has a winning strategy at move k" in a k-round Arthur-Merlin game appears to require defining an $\mathrm{MA}[\,\cdot\,]$ operator, to which Lemma 9 doesn't directly apply. This is one case where the quantifier-interchange techniques of [Zac88] show to advantage, whereas our oracle interchange lemma seems cleaner and more general in other cases.

Now we obtain Toda's theorem that $\mathrm{PH} \subseteq \mathrm{BP}[\oplus\mathrm{P}]$ as a quick consequence of Lemma 9 and the relativized forms of two earlier-known theorems in the literature, namely that $\oplus\mathrm{P}^{\oplus\mathrm{P}} = \oplus\mathrm{P}$ [PZ83] and that $\mathrm{NP} \subseteq \mathrm{RP}[\oplus\mathrm{P}]$. The latter was actually stated in the form "parity-SAT is NP-hard under randomized reductions" in [VV86]; to show that this relativizes in our given form, we include a proof sketch.

Proposition 11 ([PZ83]). *For any oracle set* X, $\oplus\mathrm{P}^{\oplus\mathrm{P}^X} = \oplus\mathrm{P}^X$.

Corollary 12. *For any oracle set* X, $\oplus\mathrm{P}^X$ *is closed downward under polynomial time Turing reducibility. In particular, for any set* $T \in \oplus\mathrm{P}^X$ *and polynomial* p, *the language* $R := \{\, x : (\exists i \leq p(|x|))[\, x\#i \in T] \,\}$ *also belongs to* $\oplus\mathrm{P}^X$.

Proposition 13. (cf. [VV86, Tod89]) *For any oracle set* X, $\mathrm{NP}^X \subseteq \mathrm{RP}[\oplus\mathrm{P}^X]$.

Proof Sketch. Fix an $A \in \mathrm{NP}^X$ and let $B \in \mathrm{P}^X$ and polynomial p be such that $A = \{\, x : \text{ for some } y \in \Sigma^{p(|x|)},\ x\#y \in B \,\}$. For each $x \in \Gamma^*$, define $S^x := \{\, y \in \Sigma^{p(|x|)} : x\#y \in B \,\}$. So, $x \in A$ iff S^x is nonempty.

By convention, for each n, we view a string $w \in \Sigma^{p(n)^2}$ as the concatenation of $p(n)$-many strings $w_1, \ldots, w_{p(n)}$, where each w_j has length $p(n)$. For each $x \in \Gamma^*$, $n := |x|$, $w \in \Sigma^{p(n)^2}$, and $j < p(n)$, define

$$S^x_{w,j} := \{\, y \in S^x : (\forall i \leq j)[\, w_i \cdot y = 0 \,] \,\},$$

where $\cdot$ is the GF_2 inner product on $\Sigma^{p(n)}$. Then, for each x, w, and k as above, $S^x = S^x_{w,0} \supseteq S^x_{w,1} \supseteq \cdots \supseteq S^x_{w,p(n)}$. By the key counting lemma in [VV86], whenever S^x is nonempty, at least $1/4$ of the strings $w \in \Sigma^{p(n)^2}$ are such that for some j, $\|S^x_{w,j}\| = 1$. Define (again with $n := |x|$),

$$R \;:=\; \left\{ x\#w : w \in \Sigma^{p(n)^2} \text{ and } (\exists j \leq p(n)) \left[\, (\oplus^{p(n)}\, y)\, y \in S^x_{w,j} \,\right] \right\}.$$

By applying Corollary 12 to the '$\exists$' quantifier, we obtain that $R \in \oplus\mathrm{P}^X$. By the [VV86] counting lemma we also have that (i) $(T^{p(n)}_{1/4}\, w)[\, R(x\#w) = A(x) \,]$, and (ii) no witness string w makes $R(x\#w)$ hold when $x \notin A$. Therefore, it follows that A belongs to $\mathrm{RP}[\oplus\mathrm{P}^X]$. □

Our proof of Toda's theorem now goes through quickly.

Theorem 14 ([Tod89, Tod91]). $\mathrm{PH} \subseteq \mathrm{BP}[\oplus\mathrm{P}]$.

Proof. By Proposition 13, we have that for all oracle sets X, $\mathrm{NP}^X \subseteq \mathrm{BP}[\oplus\mathrm{P}^X]$. Hence

$$\begin{array}{rcll}
\mathrm{NP}^{\mathrm{NP}} & \subseteq & \mathrm{BP}[\oplus\mathrm{P}^{\mathrm{NP}}] & \\
 & \subseteq & \mathrm{BP}[\oplus\mathrm{P}^{\mathrm{BP}[\oplus\mathrm{P}]}] & \text{(since } \mathrm{NP} \subseteq \mathrm{BP}[\oplus\mathrm{P}]\text{)} \\
 & \subseteq & \mathrm{BP}[\mathrm{BP}[\oplus\mathrm{P}^{\oplus\mathrm{P}}]] & \text{(by Lemma 9)} \\
 & \subseteq & \mathrm{BP}[\mathrm{BP}[\oplus\mathrm{P}]] & \text{(by Proposition 11)} \\
 & \subseteq & \mathrm{BP}[\oplus\mathrm{P}] & \text{(by Proposition 5c).}
\end{array}$$

By iterating this argument, the theorem follows. □

Since the relativized polynomial hierarchy PH^X is the union of the levels P^X, NP^X, $\mathrm{NP}^{(\mathrm{NP}^X)}$, and so on, the above proof relativizes straightforwardly to yield:

Corollary 15. *For any oracle set X,* $\mathrm{PH}^X \subseteq \mathrm{BP}[\oplus\mathrm{P}^X]$. □

The next result is a little stronger than the lemma $\mathrm{BP}[\oplus\mathrm{P}[\mathrm{BP}[\oplus\mathrm{P}]]] = \mathrm{BP}[\oplus\mathrm{P}]$ shown in Toda's paper.

Proposition 16. $\oplus\mathrm{P}^{\mathrm{BPP}^{\oplus\mathrm{P}}} = \mathrm{BPP}^{\oplus\mathrm{P}} = \mathrm{BP}[\oplus\mathrm{P}]$.

Proof. $\mathrm{BPP}^{\oplus\mathrm{P}} = \mathrm{BP}[\mathrm{P}^{\oplus\mathrm{P}}]$ and $\mathrm{P}^{\oplus\mathrm{P}} = \oplus\mathrm{P}$. □

This gives us equality in Lemma 9 in the case $\mathcal{C} = \mathcal{D} = \oplus\mathrm{P}$. There are also cases in which equality fails. M. Santha [San87] constructs an oracle X such that $\mathrm{AM}^X \setminus \Sigma_2^{p,X}$ is nonempty. Since $\mathrm{NP}[\mathrm{BPP}^X] \subseteq \Sigma_2^{p,X}$ (by Corollary 10b) and $\mathrm{AM}^X = \mathrm{BP}[\mathrm{NP}[\mathrm{P}^X]]$, taking $\mathcal{C} := \mathrm{P}^X$ and $\mathcal{D}$ to represent the $\mathrm{NP}[\,\cdot\,]$ operator gives us $\mathcal{D}^{\mathrm{BP}[\mathcal{C}]} \subset \mathrm{BP}[\mathcal{D}^{\mathcal{C}}]$. We do not have good general conditions under which equality holds.[1] It is also interesting to ask whether the 1-sided error in Proposition 13 can be exploited here—for instance, whether PH is contained in $\mathrm{RP}[\oplus\mathrm{P}]$. J. Tarui [Tar91, Tar93] shows that $\mathrm{PH} \subseteq \mathrm{RP}^{\mathrm{PP}}$ and $\mathrm{PH} \subseteq \mathrm{RP}^{\mathrm{C_=P}}$, but we do not see how to apply his techniques for $\mathrm{RP}^{\oplus\mathrm{P}}$, which equals $\mathrm{RP}[\oplus\mathrm{P}]$.

5. Closure Properties

It is generally known that an effective reducibility relation $\leq_{\mathrm{r}}$ can be represented by a relativized class $\mathcal{D}$. For instance, polynomial-time Turing reducibility $\leq_{\mathrm{T}}^{\mathrm{p}}$ is represented by the collection of oracle P-machines, and polynomial-time many reducibility $\leq_{\mathrm{m}}^{\mathrm{p}}$, by the subcollection of P-machines which on any input make exactly one query, and accept iff the answer to that query is 'yes'. Then a language class $\mathcal{C}$ is closed downward under $\leq_{\mathrm{r}}$ iff $\mathcal{D}(\mathcal{C}) \subseteq \mathcal{C}$.

What may not be so well known is that many other commonly-studied closure properties of language classes $\mathcal{C}$ can be represented by relativized classes, so long as one tolerates an extension to query machines with more than one oracle and

[1] In [AW90] the claim $\mathrm{P}^{\mathrm{BP}[\mathcal{C}]} = \mathrm{BP}[\mathrm{P}^{\mathcal{C}}]$ (for classes $\mathcal{C}$ closed downward under $\leq_{\mathrm{mtt}}^{\mathrm{p}}$) is stated without proof, but this claim is unfounded [E. Allender, personal communication]. We do not have any contrary evidence, however. With $\mathcal{C} := \mathrm{NP}$ this becomes the open question of whether $\mathrm{BP}[\mathrm{P}^{\mathrm{NP}}] = \mathrm{P}^{\mathrm{AM}}$, which we have studied without result.

query tape. For instance, the property of being closed under finite unions can be represented by a single oracle machine D which has two distinguished query tapes and takes a pair $\widehat{A} := (A_1, A_2)$ of sets as oracle. On any input x, D writes x on each query tape in turn and accepts iff at least one of the two answers is 'yes'; thus we write $L(D^{\widehat{A}}) = A_1 \cup A_2$. Then $\mathcal{C}$ is closed under finite union iff $\mathcal{D}(\mathcal{C}) = \mathcal{C}$, where $\mathcal{D} := \{ D \}$. Other Boolean closure properties can be treated similarly.

We use the following notation for tuples $\widehat{A} := (A_1, \ldots, A_k)$ and $\widehat{B} := (B_1, \ldots, B_k)$ of languages over Γ, used as oracles by "k-ary" query machines. Given $m \in \mathbf{N}$ and $u \in \Sigma^*$, write $\widehat{A}|_{\leq m} = \widehat{B}|_{\leq m}$ if for each j, $1 \leq j \leq k$, $A_j|_{\leq m} = B_j|_{\leq m}$, and also write $\pi_u \widehat{A}$ for $(\pi_u A_1, \ldots, \pi_u A_k)$. Then, for collections $\mathcal{D} = \{ D_i : i \in \mathbf{N}^+ \}$ of oracle machines of various arities k, the definitions of *polynomially bounded oracle use* and *closure under (syntactic) oracle projections* are the same as in Definitions 6–8, on substituting $\widehat{A}$ and $\widehat{B}$ for the oracles A and B. Observe that the representations $\mathcal{D}$ of Boolean closure properties have polynomial-bounded oracle use and are syntactically closed under oracle projections, since the machines D only submit their input w as a query. In general we feel the following is justified:

Definition 17. A closure property Π is a *reasonable polynomial closure property* if it is representable by a collection $\mathcal{D} = \{ D_i : i \in \mathbf{N} \}$ of oracle machines, such that $\mathcal{D}$ has polynomially bounded oracle use and is closed under oracle projections.

We show that if $\mathcal{C}$ is closed downward under $\leq^{\mathrm{p}}_{\mathrm{mtt}}$, then practically every other closure property of $\mathcal{C}$ is inherited by the associated bounded 2-sided error class, $\mathrm{BP}[\mathcal{C}]$.

Theorem 18. *Let $\mathcal{C}$ be a class which is closed downward under $\leq^{\mathrm{p}}_{\mathrm{mtt}}$. Then every reasonable polynomial closure property of $\mathcal{C}$ is also a closure property of* $\mathrm{BP}[\mathcal{C}]$.

Proof. The proof of Lemma 9 remains valid for machines D which have multiple query tapes, on substituting $\widehat{B}$ and $\widehat{R}$ for the oracles B and R. Hence, given $\mathcal{D}_\Pi$ representing Π (a given reasonable polynomial closure property), we have

$$\begin{aligned} \mathcal{D}_\Pi(\mathrm{BP}[\mathcal{C}]) &\subseteq \mathrm{BP}[\mathcal{D}_\Pi(\mathcal{C})] && \text{(by Lemma 9)} \\ &= \mathrm{BP}[\mathcal{C}] && \text{(since } \mathcal{D}_\Pi \text{ is a closure property of } \mathcal{C}\text{).} \end{aligned}$$

□

To illustrate, we can state that BPP enjoys every reasonable closure property of P, and that AM enjoys every reasonable closure property of NP.

An interesting and strong property of NP is that it is closed under *nondeterministic positive Turing reducibility* (written $\leq_{\rm pos}^{\rm np}$; see [Sel82, Sch89]). An OTM D is *positive* if for all oracle sets A, B, $A \subseteq B \Longrightarrow L(D^A) \subseteq L(D^B)$. Then for any two languages A and B, $A \leq_{\rm pos}^{\rm np} B$ if there is a positive oracle NP-machine N such that $A = L(N^B)$. NP is also closed under the *gamma-reductions* introduced by L. Adleman and K. Manders [AM77] and studied further by T. Long [Lon82], who rechristened them *strong nondeterministic polynomial time reductions*: $A \leq_{\rm m}^{\rm snp} B$ iff there is a polynomial-time NTM N such that for all inputs x, at least one computation path of $N(x)$ outputs a string y, and $x \in A \Longrightarrow$ every output string y belongs to B, while $x \notin A \Longrightarrow$ every output string y belongs to $\overline{B}$. These reductions are important because there are natural problems in NP which are complete under $\leq_{\rm m}^{\rm snp}$, but which are not known to be complete under $\leq_{\rm T}^{\rm p}$ [AM77].

Proposition 19. AM *is closed downward under* $\leq_{\rm pos}^{\rm np}$ *and under* $\leq_{\rm m}^{\rm snp}$.

Proof. Let D be a positive oracle NP-machine, and let D_π be the machine given before Definition 8, which on any input of the form $x\#u$ simulates D on input x, but replaces each query q made by D by the query $q\#u$. Then D_π is also a positive oracle NP-machine. Hence $\leq_{\rm pos}^{\rm np}$ defines a reasonable polynomial-time closure property of NP. Thus the conclusion for $\leq_{\rm pos}^{\rm np}$ follows from Theorem 18.

Now suppose $B \in$ AM and $A \leq_{\rm m}^{\rm snp} B$ via an NTM N. Let N' be an oracle NTM which on any input x simulates $N(x)$. Along any computation path, if $N(x)$ outputs a string y, then N' queries y, and the computation path accepts iff the oracle answers 'yes.' Then N' is a positive NOTM, and $L(N'^B) = A$. Hence by the closure of AM under $\leq_{\rm pos}^{\rm np}$, $A \in$ AM. □

Now we examine a certain closure property of NP which is represented by machines with two query tapes. We define it by a "class operator" $\mathcal{D}_{\mathcal{E}}(\cdot)$ which takes a relativized class $\mathcal{E}$ as a parameter; we will show that $\mathcal{D}_{\rm NP}({\rm NP}) = {\rm NP}$, and use this to draw conclusions about AM.

Definition 20. Let $\mathcal{E} := \{ E_i : i \in \mathbf{N} \}$ be a relativized class. Then define the property of "reflection by $\mathcal{E}$" to be the relativized class $\mathcal{D}_{\mathcal{E}} := \{ D_i : i \in \mathbf{N} \}$ given by machines D_i which operate as follows: Each D_i has two query tapes, and on any input $x \in \Gamma^*$, begins to simulate $E_i(x)$. Whenever E_i submits a query string q, D_i queries q on both of its tapes. If the answers are

(yes,no),	D_i simulates a 'yes' answer to E_i's query;
(no,yes),	D_i simulates a 'no' answer to E_i's query;
(yes,yes),	the current computation path of D_i accepts;
(no,no),	the current computation path of D_i rejects.

If an accepting (resp. rejecting) ID of E_i is reached in the simulation, then the current computation path of D_i accepts (resp. rejects).

Lemma 21.

(a) For any class $\mathcal{C}$, $\mathcal{D}_{\mathcal{E}}(\mathcal{C})$ contains $\mathcal{E}^{\mathcal{C} \cap \text{co-}\mathcal{C}}$.

(b) If $\mathcal{E}$ has polynomially bounded oracle use, then so does $\mathcal{D}_{\mathcal{E}}$.

(c) If $\mathcal{E}$ is syntactically closed under oracle projections, then so is $\mathcal{D}_{\mathcal{E}}$.

(d) $\mathcal{D}_{\mathrm{NP}}(\mathrm{NP}) = \mathrm{NP}$.

Proof. Statements (a) and (b) are immediate, while (c) follows because the only queries made by D_i are copies of queries made by E_i. We prove (d). Let E_i be an NP-machine, and let D_i correspond to E_i as in Definition 20. Let $A, B \in \mathrm{NP}$, let R_A and R_B be P-predicates for A and B as in Definition 2(a), and let $L := L(D_i^{(A,B)})$. Suppose $x \in L$. Then there exists a computation path $\vec{c}$ of D_i and a number $k \geq 0$ such that

(i) $\vec{c}$ contains k-many queries $q_1, \ldots, q_k$ made by E_i,

(ii) For each j, $1 \leq j \leq k-1$, the responses by the two oracles of D_i to the query q_j are either (yes,no) or (no,yes).

(iii) Either query q_k receives a (yes,yes) response, or: q_k receives a (yes,no) or (no,yes) response, and $\vec{c}$ leads to an accepting ID of E_i without any further queries.

Conversely, if such a computation path $\vec{c}$ of $D_i^{(A,B)}$ on input x exists, then $x \in L$. A certificate for '$x \in L$' consists of $\vec{c}$ and strings $w_1, w_2, \ldots, w_k, w_{k'}$ such that for each j, $1 \leq j \leq k$: if the response to query q_j represented in $\vec{c}$ is (yes,no) then $R_A(q_j, w_j)$; if it is (no,yes), then $R_B(q_j, w_j)$; and if q_k receives (yes,yes) then $R_A(q_k, w_k)$ & $R_B(q_k, w_{k'})$, else $w_{k'}$ is the accepting ID of E_i. Since the quantities k, $|\vec{c}|$, and $|w_1|, \ldots, |w_k|, |w_{k'}|$ are all bounded by a polynomial in $|x|$, $L \in \mathrm{NP}$. □

Our first application gives a more-detailed treatment of a result in [Kla90].

Corollary 22 ([Kla90]). $\mathrm{AM}^{\mathrm{AM}\cap\mathrm{coAM}} = \mathrm{AM}$.

Proof. Since NP has the closure properties required of $\mathcal{E}$ in Lemma 21 and is also closed under $\leq^{\mathrm{p}}_{\mathrm{mtt}}$, we have, using $\mathcal{D}_{\mathrm{NP}}(\mathrm{NP}) = \mathrm{NP}$:

$$\mathrm{AM}^{\mathrm{AM}\cap\mathrm{coAM}} = \mathrm{BP}[\mathrm{NP}^{\mathrm{AM}\cap\mathrm{coAM}}] \subseteq \mathrm{BP}[\mathcal{D}_{\mathrm{NP}}(\mathrm{AM})] =$$

$$\mathrm{BP}[\mathcal{D}_{\mathrm{NP}}(\mathrm{BP}[\mathrm{NP}])] \subseteq \mathrm{BP}[\mathrm{BP}[\mathcal{D}_{\mathrm{NP}}(\mathrm{NP})]] = \mathrm{BP}[\mathrm{BP}[\mathrm{NP}]] = \mathrm{AM}.$$

□

Corollary 23 ([Sch89]). $\Sigma_2^p(\mathrm{AM} \cap \mathrm{coAM}) = \Sigma_2^p$.

Proof. $\Sigma_2^p(\mathrm{AM}\cap\mathrm{coAM}) = \mathrm{NP}[\text{co-}[\mathrm{NP}^{\mathrm{AM}\cap\mathrm{coAM}}]] = \mathrm{NP}[\text{co-}[\mathrm{AM}]] = \mathrm{NP}[\mathrm{BP}[\mathrm{coNP}]] \subseteq \mathrm{NP}[\mathrm{NP}[\mathrm{coNP}[\mathrm{coNP}]]] = \Sigma_2^p$. □

Unfortunately, the closure of $\mathcal{C}$ under $\leq^{\mathrm{p}}_{\mathrm{mtt}}$ does not seem enough by itself to ensure that $\mathcal{D}_{\mathcal{C}}(\mathcal{C})$ equals the "reflection" $\mathcal{C}^{\mathcal{C}\cap\text{co-}\mathcal{C}}$ of $\mathcal{C}$—the proof above that this works for NP uses something like the closure of NP under $\leq^{\mathrm{np}}_{\mathrm{pos}}$, as in [Sch88]. Hence even with our tools, we do not have a general result of the kind, "If $\mathcal{C}$ equals its own reflection, then so does $\mathrm{BP}[\mathcal{C}]$"; this remains an open technical problem.

We are also interested in exploring other "class operators" of this kind. For instance, the variant of $\mathcal{D}_{\mathcal{E}}$ whereby the response '(no,no)' causes a universal branch by D_i (rather than rejection as above) may be interesting in the study of *promise problems*. There are, however, natural class operators $\mathcal{D}(\cdot)$ which, at least intuitively speaking, fail to be closed under oracle projections because they incorporate a "looking-back" mechanism of the kind studied by P. Chew and M. Machtey [CM81]. The following is a representative example:

Definition 24. Let $\mathcal{D}_{\cap\mathrm{co}}$ be the relativized class consisting of a single oracle machine D with two query tapes, which operates as follows on any input x:

> Submit each of the first $|x|$-many strings $w_i \in \Gamma^*$ as a query on each tape. If some w_i receives two 'yes' answers or two 'no' answers, then *reject*.
> (Call this routine *Check*(x).)
>
> If *Check*(x) passes, then submit x as a query on both tapes, and *accept* iff the answers are respectively 'yes' and 'no'.

Then for any language class $\mathcal{C}$, $\mathcal{D}_{\cap\mathrm{co}}(\mathcal{C})$ contains $\mathcal{C} \cap$co-$\mathcal{C}$, since if A and $\overline{A}$ both belong to $\mathcal{C}$, then $L(D^{(A,\overline{A})})$ equals A, and hence belongs to $\mathcal{D}_{\cap\mathrm{co}}(\mathcal{C})$. If $\mathcal{C}$ contains all finite and all co-finite sets, then $\mathcal{D}_{\cap\mathrm{co}}(\mathcal{C}) = \mathcal{C} \cap \text{co-}\mathcal{C}$. Note that $\mathcal{D}_{\cap\mathrm{co}}$ has polynomially-bounded oracle use.

If the proof of Lemma 9 worked for $\mathcal{D}_{\cap\mathrm{co}}$, then we would have in particular that $\mathrm{AM} \cap \mathrm{coAM} = \mathcal{D}_{\cap\mathrm{co}}(\mathrm{BP[NP]}) \subseteq \mathrm{BP}[\mathcal{D}(\mathrm{NP})] = \mathrm{BP[NP} \cap \mathrm{coNP]}$. Put another way, we'd have $\mathrm{BP[NP]} \cap \mathrm{BP[coNP]} = \mathrm{BP[NP} \cap \mathrm{coNP]}$. To our knowledge, this problem is open; nor do the proofs we know (see e.g. [Sch88]) that the graph-isomorphism problem is in $\mathrm{NP} \cap \mathrm{coAM}$ show that it is in $\mathrm{BP[NP} \cap \mathrm{coNP]}$. The lack is that this singleton class $\mathcal{D}_{\cap\mathrm{co}}$ is not closed under oracle projections, owing to the "looking-back" *Check* routine. When one attempts the proof of Lemma 9 for $\mathcal{D}_{\cap\mathrm{co}}$, *Check* trips across the few strings u which make $\widehat{B} \neq \pi_u \widehat{R}$, and this shuts down the whole computation. If one closes $\mathcal{D}_{\cap\mathrm{co}}$ out syntactically under oracle projections to obtain a class $\mathcal{D}_\pi$, then the property $\mathcal{D}_\pi(\mathrm{NP}) = \mathrm{NP} \cap \mathrm{coNP}$ appears to be lost.

6. Random Oracles

The following is essentially an abstraction of the proof of Theorem 5 in [BG81]. Let μ be the standard product measure on $2^{\mathbf{N}}$. All the oracle properties we consider are first-order definable, so that the subsets of $2^{\mathbf{N}}$ they define are Borel, and hence measurable. Say a property $\Psi(\cdot)$ *holds for a random oracle set* R iff $\mu(\{\, R \subseteq \mathbf{N} : \Psi(R)\,\}) = 1$. We note that for all the classes $\mathcal{D}$ in this paper which are closed downward under $\leq_{\mathrm{m}}^{\mathrm{p}}$, and any oracle A, $\mathcal{D}^A$ is closed downward under $\leq_{\mathrm{m}}^{\mathrm{p},A}$; i.e. if f

is computable in polynomial time with oracle A, $D \in \mathcal{D}^A$, and C reduces to D via f, then $C \in \mathcal{D}^X$.

Theorem 25. *Let $\mathcal{D}$ be a relativized class which has polynomial bounded oracle use, such that for all A, $\mathcal{D}^A$ is closed downward under $\leq_{\mathrm{m}}^{\mathrm{p},A}$. Then for a random oracle set R, $\mathrm{BP}[\mathcal{D}^R] \subseteq \mathcal{D}^R$.*

Proof. Let $\{ D_i \}$ be a representation of $\mathcal{D}$ by OTMs with corresponding polynomial bounds $\{ r_i \}$ on their oracle use. For every i and polynomial p_j, let M_{ij} be an OTM which behaves as follows, for any input x and oracle A:

> Make queries to the first $p_j(|x|)$ strings of length $r_i(|x| + p_j(|x|)) + 1$ and call the 0–1 string of results y.
> Simulate D_i^A on input $x\#y$.

Then for all oracle sets A, $L(M_{ij}^A) \in \mathcal{D}^A$, because $L(M_{ij}^A)$ reduces to $L(D_i^A)$ by the function which maps x to $x\#y$, which is P^A-computable.

Now let $k > 0$. For all i, j, and x define E_{ijx} to be the set of oracles A such that *(i)* the proportion of strings $y \in \Sigma^{p_j(|x|)}$ such that D_i^A accepts $x\#y$ is either less than $2^{-q(|x|)}$ or greater than $1 - 2^{-q(|x|)}$, where $q(n) := 2n + i + j - k$ for all n, and *(ii)* M_{ij}^A on input x disagrees with the answer of this overwhelming majority for $D_i^A(x\#y)$. Because D_i never queries those strings which are used by M_{ij} to obtain the bits for y, $\mu(E_{ijx}) \leq 2^{-q(|x|)}$. The rest of the analysis, showing that with this choice of q, $\mu(\cup_{ijx} E_{ijx}) < 2^{-k}$ (which can be made arbitrarily small), and that $\mathrm{BP}[\mathcal{D}^A] = \mathcal{D}^A$ for any $A \notin \cup_{ijx} E_{ijx}$, is the same as that in [BG81]. $\square$

Corollary 26. *Relative to a random oracle R, PH^R is properly contained in $\oplus\mathrm{P}^R$.*

Proof. By Corollary 15, for any oracle set R, $\mathrm{PH}^R \subseteq \mathrm{BP}[\oplus\mathrm{P}^R]$. Since the relativized class $\oplus\mathrm{P}$ has the closure properties required in the statement of Theorem 25, it follows that for a random oracle set R, $\mathrm{PH}^R \subseteq \oplus\mathrm{P}^R$. J. Cai [Cai89] and L. Babai [Bab87] proved that for a random oracle set R, $\oplus\mathrm{P}^R$ is not contained in PH^R. $\square$

Definition 27 ([NW88]). For any relativized class $\mathcal{C}$, $\mathit{Almost}[\mathcal{C}]$ denotes the class of languages L such that $\mu(\{ A \subseteq \mathbf{N} : L \in \mathcal{C}^A \}) = 1$.

Corollary 28. $\mathrm{BP}[\oplus\mathrm{P}] \subseteq \mathit{Almost}[\oplus\mathrm{P}]$.

Whether equality holds here runs into the problem that a single relativized $\oplus$P computation may access exponentially many bits of the oracle, whereas the BP[$\oplus$P] computations have only polynomially many coin flips. The relativized $\oplus$P computations can be modeled by depth-2 circuits whose bottom level is a single parity gate. It is interesting to ask whether these can be "fooled" by strong pseudorandom generators which generate exponentially many bits from a polynomial-length random seed, along the lines of [NW88] for relativized PH computations.

Problem. *Is* $\mathrm{BP}[\oplus\mathrm{P}] = \mathit{Almost}[\oplus\mathrm{P}]$*?*

A positive answer would yield yet another proof of $\mathrm{PH} \subseteq \mathrm{BP}[\oplus\mathrm{P}]$ using random oracle sets R, via $\mathrm{NP}^{\mathrm{NP}} \subseteq \mathrm{BP}[\oplus\mathrm{P}^{\mathrm{BP}[\oplus\mathrm{P}]}] = \mathrm{BP}[\oplus\mathrm{P}^{\oplus\mathrm{P}^R}] = \mathrm{BP}[\oplus\mathrm{P}^R] = \oplus\mathrm{P}^R = \mathrm{BP}[\oplus\mathrm{P}]$.

Nisan and Wigderson also proved that $\mathit{Almost}[\mathrm{NP}] = \mathrm{BP}[\mathrm{NP}]$, and similarly for the other Σ_k^p and Π_k^p levels of the polynomial hierarchy. It follows that $\mathit{Almost}[\mathrm{NP} \cap \mathrm{coNP}]$ equals $\mathrm{AM} \cap \mathrm{coAM}$. However, this leaves the following open question:

Problem. *Is* $\mathrm{BP}[\mathrm{NP} \cap \mathrm{coNP}] = \mathit{Almost}[\mathrm{NP} \cap \mathrm{coNP}]$*?*

Last, we remark that under the "Random Oracle Hypothesis" of [BG81], PH would be (strictly) contained in $\oplus$P. Our initial reaction is to disbelieve this even somewhat more than the hypothesis BPP = P. It is interesting to ask how these two assertions are related. In conclusion, we hope that our results add understanding to the effect of sources of randomness in polynomial-time computations.

Acknowledgments. We would like to thank Richard Beigel, Ronald Book, Seinosuke Toda, and several anonymous referees for helpful comments and suggestions on this work.

References

[AM77] L. Adleman and K. Manders. Reducibility, randomness, and intractability. In *The Proceedings of the 9th Annual ACM Symposium on the Theory of Computing*, pages 151–163, 1977.

[AW90] E. Allender and K. Wagner. Counting hierarchies: Polynomial time and constant depth circuits. *EATCS Bulletin*, 40:182–194, February 1990.

[Bab85] L. Babai. Trading group theory for randomness. In *The Proceedings of the 17th Annual ACM Symposium on the Theory of Computing*, pages 421–429, 1985.

[Bab87] L. Babai. Random oracles separate PSPACE from the polynomial-time hierarchy. *Information Processing Letters*, 26:51–53, 1987.

[BG81] C. Bennett and J. Gill. Relative to a random oracle A, $\mathrm{P}^A \neq \mathrm{NP}^A \neq \mathrm{coNP}^A$ with probability 1. *SIAM Journal on Computing*, 10:96–113, 1981.

[BM88] L. Babai and S. Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36:254–276, 1988.

[Cai89] J. Cai. With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy. *Journal of Computer and System Sciences*, 38:68–85, 1989.

[CM81] P. Chew and M. Machtey. A note on structure and looking-back applied to the relative complexity of computable functions. *Journal of Computer and System Sciences*, 22:53–59, 1981.

[HZ84] P. Hinman and S. Zachos. Probabilistic machines, oracles, and quantifiers. In *Proceedings, Oberwohlfach Recursion-Theory Week, Lecture Notes in Mathematics 1141*, pages 159–192. Springer-Verlag, 1984.

[Kla90] A. Klapper. Generalized lowness and highness and probabilistic complexity classes. *Mathematical Systems Theory*, 22:37–45, 1990.

[Ko82] K. Ko. Some observations on the probabilistic algorithms and NP-hard problems. *Information Processing Letters*, 14:39–43, 1982.

[Lau83] C. Lautemann. BPP and the polynomial hierarchy. *Information Processing Letters*, 17:215–217, 1983.

[Lon82] T. Long. Strong nondeterministic polynomial reducibilities. *Theoretical Computer Science*, 21:1–25, 1982.

[NW88] N. Nisan and A. Wigderson. Hardness vs. randomness. In *The Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science*, pages 2–11, 1988.

[PZ83] C. H. Papadimitriou and S. Zachos. Two remarks on the power of counting. In *The 6th GI Conference on Theoretical Computer Science*, Lecture Notes in Computer Science No. 145, pages 269–276. Springer-Verlag, 1983.

[San87] M. Santha. Relativized Arthur-Merlin vs. Merlin-Arthur games. In *Foundations of Software Theory and Theoretical Computer Science*, Lecture Notes in Computer Science No. 287, pages 437–442. Springer-Verlag, 1987.

[Sch85] U. Schöning. *Complexity and Structure*, volume 211 of *Lecture Notes in Computer Science*. Springer-Verlag, 1985.

[Sch88] U. Schöning. Graph isomorphism is in the low hierarchy. *Journal of Computer and System Sciences*, 37:312–323, 1988.

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

[Sch90] U. Schöning. The power of counting. In A. Selman, editor, *Complexity Theory Retrospective*, pages 204–223. Springer-Verlag, 1990.

[Sel82] A. Selman. Reductions on NP and p-selective sets. *Theoretical Computer Science*, 19:287–304, 1982.

[Tar91] J. Tarui. Randomized polynomials, threshold circuits, and the polynomial hierarchy. In *The Proceedings of the 8th Annual Symposium on Theoretical Aspects of Computer Science*, volume 480 of *Lecture Notes in Computer Science*, pages 238–250. Springer-Verlag, 1991.

[Tar93] J. Tarui. Probabilistic polynomials, AC^0 functions, and the polynomial-time hierarchy. *Theoretical Computer Science*, 113:167–183, 1993.

[Tod89] S. Toda. On the computational power of PP and $\oplus$P. In *The Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 514–519, 1989.

[Tod91] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20:865–877, 1991.

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

[Zac82] S. Zachos. Robustness of probabilistic computational complexity classes under definitional perturbations. *Information and Computation*, 54:143–154, 1982.

[Zac88] S. Zachos. Probabilistic quantifiers and games. *Journal of Computer and System Sciences*, 36:433–451, 1988.

[ZH86] S. Zachos and H. Heller. A decisive characterization of BPP. *Information and Computation*, 69:125–135, 1986.