SIAM J. COMPUT.
Vol. 20, No. 5, pp. 865-877, October 1991

# PP IS AS HARD AS THE POLYNOMIAL-TIME HIERARCHY*

SEINOSUKE TODA†

**Abstract.** In this paper, two interesting complexity classes, PP and $\oplus$P, are compared with PH, the polynomial-time hierarchy. It is shown that every set in PH is polynomial-time Turing reducible to a set in PP, and PH is included in BP $\cdot$ $\oplus$P. As a consequence of the results, it follows that PP $\subseteq$ PH (or $\oplus$P $\subseteq$ PH) implies a collapse of PH. A stronger result is also shown: every set in PP(PH) is polynomial-time Turing reducible to a set in PP.

**Key words.** polynomial-time hierarchy, probabilistic Turing machine, polynomial-time Turing reductions, parity, randomized reduction

**AMS(MOS) subject classifications.** 68Q15, 03D15

## 1. Introduction.

Since the notion of probabilistic Turing machines was introduced by Gill [5], much attention has been given to several questions about its computational power. One of those questions is whether PP is more powerful than PH (the polynomial-time hierarchy), where PP denotes the class of sets accepted by polynomial-time-bounded probabilistic Turing machines with two-sided unbounded error probability. In particular, it is important in the theory of computational complexity to ask whether PH is included in PP, or to ask whether all sets in PH are reducible to sets in PP under a suitable reducibility. This has been an open question discussed in many papers [1], [2], [10], [12], [15], [16], [19-21]. It was shown by Gill [5] that NP $\cup$ co-NP is included in PP. It is not known, however, whether $\Delta_2^P$ is included in PP. For this question, Beigel, Hemachandra, and Wechsung [3] have recently shown that $P^{NP[\log]}$ is included in PP. This is the strongest result known currently for the containment question of PH in PP. Some related results have been shown in [20].

In this paper, we give an affirmative answer to one of the above questions. We show that all sets in PH are $\leq_T^P$-reducible to a set in PP. Namely, our Main Theorem in this paper is stated as follows.

MAIN THEOREM. PH $\subseteq$ P(PP)

As an immediate consequence, we see that PP is not included in PH unless PH collapses to a finite level. This gives us evidence that PP is harder than PH. In the process of proving the Main Theorem, we show an interesting result about the hardness of the class $\oplus$P. This class was introduced by Papadimitriou and Zachos [13] and further investigated in several papers [13], [25], [15]. We show that all sets in PH are reducible to a set in this class under polynomial-time randomized reductions with two-sided bounded error probability. It was shown by Valiant and Vazirani [25] that all sets in NP are reducible to a set in $\oplus$P under polynomial-time randomized reductions with one-sided bounded error probability. These randomized reductions are stronger, but in other respects our result extends theirs. In fact, they asked how computationally difficult $\oplus$P is. Our result is an answer to their open question.

$\oplus P$
PH $\subseteq$ BPP

$\oplus P$
NP $\subseteq$ BPP

Our proof of the main theorem proceeds as follows. In § 3, we show that PH is included in BP $\cdot$ $\oplus$P, where BP $\cdot$ denotes the BP-operator introduced by Schöning [15]. Intuitively speaking, a set is in BP $\cdot$ $\oplus$P if and only if it is reducible to a set in $\oplus$P

* Received by the editors February 21, 1989; accepted for publication (in revised form) December 12, 1990. This research was supported in part by International Information Science Foundation grant 89 2 2 176.

† Department of Computer Science and Information Mathematics, University of Electro-communications, 1-5-1 Chofugaoka, Chofu-shi, Tokyo 182, Japan.

$BP \cdot \oplus P = BPP^{\oplus P}$

under a polynomial-time randomized reduction with two-sided bounded error probability. The proof of it is based on a result by Valiant and Vazirani [25] and a result by Schöning [15]. In § 4, we show that BP · ⊕P is included in P(PP). In fact, we will show a stronger result than this. The proof is based on a structural property of ⊕P discovered in this paper. At the end of § 4, we will mention a stronger result than the Main Theorem above: PP(PH) ⊆ P(PP). This result is obtained by combining the technique in this paper with a result by Köbler et al. [10].

$PP^{PH} \subseteq P^{PP}$

**2. Preliminaries.** We assume that the reader is familiar with the basic concepts of computational complexity theory. Let $\Sigma$ be a finite alphabet. For a string $w \in \Sigma^*$, $|w|$ denotes the length of $w$. For a set $L \subseteq \Sigma^*$, $\bar{L}$ denotes the complement of $L$. For a class **K** of sets, co-**K** denotes the class of sets whose complement is in **K**. Let $\Sigma^n$ (respectively, $\Sigma^{\leq n}$ and $\Sigma^{<n}$) denote the set of strings with length $n$ (respectively, length at most $n$ and less than $n$). For a finite set $X \subseteq \Sigma^*$, $\|X\|$ denotes the number of strings in $X$. Let **N** denote the set of natural numbers.

Our sets in this paper are over $\Sigma = \{0, 1, \#\}$ unless otherwise specified. The symbol # is usually used as a delimiter among strings of $\{0, 1\}^*$. A pairing function (respectively, a $k$-tuple function) over $\{0, 1\}^*$ is represented by delimiting two strings (respectively, $k$ strings) by this symbol.

Our models of computation are variations of *polynomial-time-bounded oracle Turing machines* (deterministic, nondeterministic, or probabilistic). Our oracle machines are usual ones. For an oracle machine M and an oracle set $A$, M($A$) denotes that M uses $A$ as an oracle. A polynomial-time-bounded deterministic (respectively, nondeterministic) oracle machine is abbreviated by an *oracle* P-*machine* (respectively, *oracle* NP-*machine*). A polynomial-time-bounded probabilistic oracle machine with two-sided unbounded error probability (respectively, with two-sided bounded error probability) is abbreviated by an *oracle* PP-*machine* (respectively, an *oracle* BPP-*machine*). In the unrelativized cases, we omit the term "oracle." For example, an oracle NP-machine with the empty set as an oracle is simply called an NP-*machine*.

For an oracle set $A$, P($A$) denotes the class of sets accepted by oracle P-machine with oracle $A$. NP($A$), PP($A$), and BPP($A$) are defined similarly. ⊕P($A$) denotes the class of sets $L$ for which there exists an oracle NP-machine M such that for each $x$, $x$ is in $L$ if and only if the number of accepting computation paths of M($A$) on $x$ is odd. This class was defined by Papadimitriou and Zachos [13]. For a class **K** of oracle sets, P(**K**) = ∪{P($A$): $A \in$ **K**}. Other classes are defined similarly. The unrelativized classes are defined by setting the oracle set to the empty set, and the specification of oracle set is omitted in this case.

We assume that all polynomial-time-bounded oracle machines M satisfy the following conditions.

(1) Its transition function has at most two possible transitions from each configuration.

(2) All computation paths of M are encoded into a string of $\{0, 1\}^*$ by the usual manner, where a computation path may contain possible answers from a given oracle, and the oracle answer "yes" (respectively, "no") is encoded by 0 (respectively, 1).

These assumptions are technical ones. Obviously, we lose no generality under these assumptions.

Let $X$ be a finite set of strings and $R$ be a predicate over strings. In this paper, we denote by Prob $(\{w \in X: R(w)\})$ the probability that $R(w)$ is true for randomly chosen $w$ from $X$ under uniform distribution. In [15], Schöning introduced the BP-operator, which produces a probabilistic class from a given class. He also defined

$\oplus$-operator in [16], as an abstraction of the class $\oplus$P. In [26], Wagner defined the counting operator, based on a characterization of PP in [14], [26]. We give those definitions here. The following definition of the counting operator is different from the original one; however, it is easy to see that both definitions define the same concept.

DEFINITION 2.1 [15], [16], [26]. Let $\mathbf{K}$ be a class of sets and let $L$ be a set. Then we define some new classes denoted by $\oplus \cdot \mathbf{K}$, $\mathsf{BP} \cdot \mathbf{K}$, and $\mathbf{C} \cdot \mathbf{K}$ as follows.

(1) $L \in \oplus \cdot \mathbf{K}$ if there exist a set $A \in \mathbf{K}$ and a polynomial $p$ such that for all $x \in \Sigma^*$,

$$x \in L \leftrightarrow \|\{w \in \{0,1\}^{p(|x|)} \colon x \# w \in A\}\| \text{ is odd.}$$

(2) $L \in \mathsf{BP} \cdot \mathbf{K}$ if there exist a set $A \in \mathbf{K}$, a polynomial $p$, and a constant $\alpha > 0$ such that, for all $x \in \Sigma^*$,

$$\text{Prob}\,(\{w \in \{0,1\}^{p(|x|)} \colon x \# w \in A \leftrightarrow x \in L\}) \geqq \tfrac{1}{2}\alpha.$$

(3) $L \in \mathbf{C} \cdot \mathbf{K}$ if there exist a set $A \in \mathbf{K}$ and a polynomial $p$ such that for all $x \in \Sigma^*$,

$$\text{Prob}\,(\{w \in \{0,1\}^{p(|x|)} \colon x \# w \in A \leftrightarrow x \in L\}) > \tfrac{1}{2}.$$

It is easy to see that $\mathbf{C} \cdot \mathrm{P} = \mathrm{PP}$, $\mathsf{BP} \cdot \mathrm{P} = \mathrm{BPP}$, and $\oplus \cdot \mathrm{P} = \oplus\mathrm{P}$. The following propositions are basic properties of the above operators, which follow immediately from the definitions. These will be used implicitly in later arguments.

PROPOSITION 2.2. *Let* $\mathbf{K}_1$ *and* $\mathbf{K}_2$ *be arbitrary classes of sets. Then, the following statements hold.*

(1) *If* $\mathbf{K}_1 \subseteq \mathbf{K}_2$, *then* $\oplus \cdot \mathbf{K}_1 \subseteq \oplus \cdot \mathbf{K}_2$.

(2) *If* $\mathbf{K}_1$ *is closed under marked union with sets of the form* $\{x \# 0^{p(|x|)} \colon x \in \Sigma^*\}$ (*for arbitrary polynomial p such sets are in* P), *then* $\oplus \cdot \mathbf{K}_1$ *is closed under complementation.*

(3) *If* $\mathbf{K}_1$ *is such that, for all sets* $L \in \mathbf{K}_1$, *the set* $\{x \# x \colon x \in L\}$ *also belongs to* $\mathbf{K}_1$, *then* $\mathbf{K}_1 \subseteq \oplus \cdot \mathbf{K}_1$.

PROPOSITION 2.3. *Let* $\mathbf{K}_1$ *and* $\mathbf{K}_2$ *be any classes of sets. Then the following statements hold.*

(1) *If* $\mathbf{K}_1 \subseteq \mathbf{K}_2$, *then* $\mathsf{BP} \cdot \mathbf{K}_1 \subseteq \mathsf{BP} \cdot \mathbf{K}_2$.

(2) *co*-$\mathsf{BP} \cdot \mathbf{K}_1 \subseteq \mathsf{BP} \cdot$ *co*-$\mathbf{K}_1$. *Hence, if* $\mathbf{K}_1$ *is closed under complementation, then co*-$\mathsf{BP} \cdot \mathbf{K}_1 = \mathsf{BP} \cdot \mathbf{K}_1$.

(3) *If* $\mathbf{K}_1$ *is closed under padding* (*i.e.*, $L \in \mathbf{K}_1$ *implies* $\{x \# y \colon x \in L \text{ and } y \in \{0,1\}^*\} \in \mathbf{K}_1$ *for each set L*), *then* $\mathbf{K}_1 \subseteq \mathsf{BP} \cdot \mathbf{K}_1$.

PROPOSITION 2.4. *Let* $\mathbf{K}_1$ *and* $\mathbf{K}_2$ *be arbitrary classes of sets. Then the following statements hold.*

(1) *If* $\mathbf{K}_1 \subseteq \mathbf{K}_2$, *then* $\mathbf{C} \cdot \mathbf{K}_1 \subseteq \mathbf{C} \cdot \mathbf{K}_2$.

(2) *co*-$\mathbf{C} \cdot \mathbf{K}_1 \subseteq \mathbf{C} \cdot$ *co*-$\mathbf{K}_1$. *Hence, if* $\mathbf{K}_1$ *is closed under complementation, then co*-$\mathbf{C} \cdot \mathbf{K}_1 = \mathbf{C} \cdot \mathbf{K}_1$.

(3) *If* $\mathbf{K}_1$ *is closed under padding, then* $\mathbf{K}_1 \subseteq \mathbf{C} \cdot \mathbf{K}_1$.

(4) $\mathsf{BP} \cdot \mathbf{K}_1 \subseteq \mathbf{C} \cdot \mathbf{K}_1$.

We can easily see that all the classes to be built in this paper satisfy the closure properties mentioned in the above propositions (except possibly for complementation). For example, $\Pi_k^{\mathrm{P}} (k \geqq 0)$ is closed under taking marked union with the set of the form $\{x \# 0^{p(|x|)} \colon x \in \Sigma^*\}$; hence, $\oplus \cdot \Pi_k^{\mathrm{P}}$ is closed under complementation; we will use this fact in the next section.

In the later sections, we will be concerned with several reducibility notions defined below.

DEFINITION 2.5. Let $A$ and $B$ be arbitrary sets. $A$ is said to be $\leq_m^{\mathrm{P}}$*-reducible* to $B (A \leq_m^{\mathrm{P}} B)$ if there exists a function $f$ computable in polynomial time such that for each $x$, $x \in A$ if and only if $f(x) \in B$. $A$ is said to be $\leq_{\mathrm{maj}}^{\mathrm{P}}$*-reducible* to $B (A \leq_{\mathrm{maj}}^{\mathrm{P}} B)$ if there exists a function $f$ computable in polynomial time such that for each $x, f(x) = y_1 \# y_2 \# \cdots \# y_m$ $(m \geqq 1)$, and $x \in A$ if and only if the majority of $y_i$'s are in $B$. $A$ is said to be $\underline{\leq^{\mathrm{P}}_{\mathrm{T}}\text{-}reducible}$ to $B$ $(A \leq^{\mathrm{P}}_{\mathrm{T}} B)$ if there exists an oracle P-machine that accepts $A$ with oracle $B$. Let $\mathbf{K}$ be a class of sets and let $\leq^{\mathrm{P}}_{\mathrm{r}}$ denote an arbitrary reducibility. A set $L$ is $\leq^{\mathrm{P}}_{\mathrm{r}}$-*hard* for $\mathbf{K}$ if every set in $\mathbf{K}$ is $\leq^{\mathrm{P}}_{\mathrm{r}}$-reducible to $L$; if $L \in \mathbf{K}$ in addition, then $L$ is said to be $\leq^{\mathrm{P}}_{\mathrm{r}}$-*complete* for $\mathbf{K}$. Then we say that $\mathbf{K}$ is closed under $\leq^{p}_{r}$ if and only if for all sets $A$ and $B$, $A \leq^{p}_{r} B$ and $B \in \mathbf{K}$ implies $A \in \mathbf{K}$.

$A \in P^B$

Observe that for all sets $A$ and $B$, $\underline{A \leq^{\mathrm{P}}_{\mathrm{maj}} B \text{ implies } A \leq^{\mathrm{P}}_{\mathrm{T}} B}$; and hence, if a class $\mathbf{K}$ is closed under $\leq^{\mathrm{P}}_{\mathrm{T}}$, then the class is closed under $\leq^{\mathrm{P}}_{\mathrm{maj}}$.

## 3. $\oplus$P is hard for PH under randomized reducibility.

In this section, we show that $\oplus$P is hard for the polynomial-time hierarchy under polynomial-time randomized reducibility. More precisely, our main result in this section is stated as follows.

THEOREM 3.1. $\boxed{\mathrm{PH} \subseteq \mathsf{BP} \cdot \oplus\mathrm{P}.}$ Does it relativize? $\mathrm{PH}^A \subseteq \mathrm{BPP}^{\oplus P^A}$ ?

Before proving this, we give an intuitive explanation of the proof to the reader. We first show that $\Sigma^{\mathrm{P}}_k$ is included in $\mathsf{BP} \cdot \oplus \cdot \Pi^{\mathrm{P}}_{k-1}$ for each $k \geqq 1$ (see Lemma 3.3). This generalizes a result due to Valiant and Vazirani [25] in which they showed that all NP-complete sets are reducible to a set in $\oplus$P under randomized polynomial-time reducibility. Our proof technique is essentially the same as theirs. We next observe that it is possible to swap a $\oplus$-operator and a $\mathsf{BP}$-operator. In particular, we show that $\oplus \cdot \mathsf{BP} \cdot \oplus\mathrm{P} \subseteq \mathsf{BP} \cdot \oplus \cdot \oplus\mathrm{P}$ (see Lemma 3.6). Furthermore, we observe that it is possible to reduce two consecutive $\mathsf{BP}$-operators (respectively, two consecutive $\oplus$-operators) to one operator: It was shown by Papadimitriou and Zachos [13] that $\oplus\mathrm{P}(\oplus\mathrm{P}) = \oplus\mathrm{P}$. This implies that $\oplus \cdot \oplus\mathrm{P} = \oplus\mathrm{P}$, and we also show that $\mathsf{BP} \cdot \mathsf{BP} \cdot \oplus\mathrm{P} = \mathsf{BP} \cdot \oplus\mathrm{P}$ (see Lemma 3.7). At the end of this section, we put all this together to prove Theorem 3.1, using an induction on the levels of the polynomial-time hierarchy.

Now we begin to show the lemmas mentioned above. Following Valiant and Vazirani [25], we shall view strings of $\{0, 1\}^n$ as $n$-dimensional vectors from the vector space GF$[2]^n$. We denote by $u \cdot v$ the inner product of two vectors $u$ and $v$ over GF[2]. In [25], they showed the following result.

THEOREM 3.2 [25]. *Let $n \geqq 1$ and let $S \subseteq \{0, 1\}^n$ be a nonempty set. Suppose $w_1, w_2, \cdots, w_n$ are randomly chosen from $\{0, 1\}^n$. Let $S_0 = S$ and let*

$$S_i = \{v \in S \colon v \cdot w_1 = v \cdot w_2 = \cdots = v \cdot w_i = 0\}$$

*for each $1 \leqq i \leqq n$. Let $P_n(S)$ be the probability that $\|S_i\| = 1$ for some $0 \leqq i \leqq n$. Then, $P_n(S) \geqq \frac{1}{4}$.*

LEMMA 3.3. *For each $k \geqq 1$, $\Sigma^{\mathrm{P}}_k \cup \Pi^{\mathrm{P}}_k \subseteq \mathsf{BP} \cdot \oplus \cdot \Pi^{\mathrm{P}}_{k-1}$.*

*Proof.* By Propositions 2.2(2) and 2.3(2), $\mathsf{BP} \cdot \oplus \cdot \Pi^{\mathrm{P}}_{k-1}$ is closed under complementation. Hence, it suffices to show that $\Sigma^{\mathrm{P}}_k \subseteq \mathsf{BP} \cdot \oplus \cdot \Pi^{\mathrm{P}}_{k-1}$. Let $L \in \Sigma^{\mathrm{P}}_k$. Then it was shown by Stockmeyer [18] and Wrathall [28] that there exist a set $A \in \Pi^{\mathrm{P}}_{k-1}$ and a polynomial $p$ such that for every $x$, $x \in L$ if and only if $x \# y \in A$ for some $y \in \{0, 1\}^{p(|x|)}$. We define a set $C$ as follows:

$$C = \{x \# w_1 w_2 \cdots w_{p(|x|)} \colon \text{for each } i,\ 1 \leqq i \leqq p(|x|),\ w_i \in \{0, 1\}^{p(|x|)}, \text{ and for some } j,$$

$$0 \leq j \leq p(|x|),\ \|\{y \in \{0, 1\}^{p(|x|)} \colon x \# y \in A \wedge (\forall i \leqq j)[w_j \cdot y = 0]\}\| \text{ is odd}\}.$$

We first show that $C$ is in $\oplus \cdot \Pi^{\mathrm{P}}_{k-1}$. Since $\oplus \cdot \Pi^{\mathrm{P}}_{k-1}$ is closed under complementation, it suffices to show that $C$'s complement, $\bar{C}$, is in $\oplus \cdot \Pi^{\mathrm{P}}_{k-1}$. This can be done as follows: Given arbitrary strings $x$ and $z = w_1 w_2 \cdots w_{p(|x|)}$ such that, for each $i$, $1 \leqq i \leqq p(|x|)$, $w_i \in \{0, 1\}^{p(|x|)}$,

$$x \# z \in \bar{C}$$
$$\leftrightarrow \text{for each } j,\ 0 \leqq j \leqq p(|x|),\ \|\{y \in \{0,1\}^{p(|x|)}: x \# y \in A \wedge (\forall i \leq j)[w_j \cdot y = 0]\}\| \text{ is even}$$
$$\leftrightarrow \prod_{j=0}^{p(|x|)} (\|\{y: x \# y \in A \wedge (\forall i \leq j)[w_j \cdot y = 0]\}\| + 1) \text{ is odd.}$$

Hence we may define a set $B \in \Pi^{\mathrm{P}}_{k-1}$ and a polynomial $q$ such that for every $x \# z$, $(|z| = p(|x|)^2)$,

$$\|\{u \in \{0,1\}^{q(|x|)}: x \# z \# u \in B\}\| = \prod_{j=0}^{p(|x|)} (\|\{y: x \# y \in A \wedge (\forall i \leq j)[w_j \cdot y = 0]\}\| + 1).$$

The set $B$ is defined by

$$B = \{x \# w_1 \cdots w_{p(|x|)} \# a_1 y_1 a_2 y_2 \cdots a_{p(|x|)} w_{p(|x|)}: \text{for every } j,\ 1 \leqq j \leqq p(|x|),$$
$$w_j \in \{0,1\}^{p(|x|)},\ a_j \in \{0,1\},\ y_j \in \{0,1\}^{p(|x|)}, \text{ and}$$
$$(\forall j, 1 \leqq j \leqq p(|x|))[a_j y_j = 0^{p(|x|)+1} \vee (a_j = 1 \wedge x \# y_j \in A \wedge (\forall i \leq j)[w_i \cdot y_j = 0])]\}.$$

It is easy to see that $B \in \Pi^{\mathrm{P}}_{k-1}$ and that $B$ and the polynomial $q(n) = p(n)(1 + p(n))$ satisfy the required condition; that is, the set $B$ and the polynomial $q$ witness $\bar{C} \in \oplus \cdot \Pi^{\mathrm{P}}_{k-1}$.

Next, we show $L \in \mathsf{BP} \cdot \oplus \cdot \Pi^{\mathrm{P}}_{k-1}$ by using the set $C$. Let $x$ be a string and let $w_1, w_2, \cdots, w_{p(|x|)}$ be randomly chosen from $\{0,1\}^{p(|x|)}$. We define

$$S_0 = \{y \in \{0,1\}^{p(|x|)}: x \# y \in A\}$$

and

$$S_i = \{y \in S_0: w_1 \cdot y = w_2 \cdot y = \cdots = w_i \cdot y\}$$

for each $1 \leqq i \leqq p(|x|)$. Let $P_{p(|x|)}(S_0)$ be the probability that $\|S_i\| = 1$ for some $0 \leqq i \leqq p(|x|)$. Then it is easy to see that

$$\text{Prob}\ (\{w_1 \cdots w_{p(|x|)} \in \{0,1\}^{p(|x|)^2}: x \# w_1 w_2 \cdots w_{p(|x|)} \in C\}) \geqq P_{p(|x|)}(S_0).$$

Hence, from Theorem 3.2,

(1) $x \in L \rightarrow \text{Prob}\ (\{u \in \{0,1\}^{p(|x|)^2}: x \# u \in C\}) \geqq \frac{1}{4}$ and

(2) $x \notin L \rightarrow \text{Prob}\ (\{u \in \{0,1\}^{p(|x|)^2}: x \# u \in C\}) = 0.$

The probability of (2) follows from the fact that for all $x$, if $x \notin L$, then $x \# y \notin A$ for every $y \in \{0,1\}^{p(|x|)}$. To amplify the probability in (1), we further define a set $D$ as follows:

$$D = \{x \# u_1 u_2 u_3: |u_i| = p(|x|)^2 \text{ for each } i = 1,2,3 \text{ and } x \# u_i \in C \text{ for some } i = 1,2,3\}.$$

Then we obtain that for each $x$,

(3) $x \in L \rightarrow \text{Prob}\ (\{u_1 u_2 u_3 \in \{0,1\}^{3p(|x|)^2}: x \# u_1 u_2 u_3 \in D\})$

$= 1 - (\text{Prob}\ (\{u \in \{0,1\}^{p(|x|)^2}: x \# u \notin C\}))^3$

$\geqq 1 - \frac{27}{64} = \frac{1}{2} + \frac{5}{64}$ and

(4) $x \notin L \rightarrow \text{Prob}\ (\{u_1 u_2 u_3 \in \{0,1\}^{3p(|x|)^2}: x \# u_1 u_2 u_3 \in D\}) = 0.$

By using the same argument as when showing $C \in \oplus \cdot \Pi^{\mathrm{P}}_{k-1}$, we can show $D \in \oplus \cdot \Pi^{\mathrm{P}}_{k-1}$. This implies $L \in \mathsf{BP} \cdot \oplus \cdot \Pi^{\mathrm{P}}_{k-1}$. □

It was shown by Papadimitriou and Zachos [13] that $\oplus\mathrm{P}(\oplus\mathrm{P}) = \oplus\mathrm{P}$. This implies the following theorem. For the sake of making this paper more self-contained, we provide a sketch of their proof.

THEOREM 3.4 [13]. $\oplus\mathrm{P}(\oplus\mathrm{P}) = \oplus\mathrm{P}$. *Hence we have that* $\oplus \cdot \oplus\mathrm{P} = \oplus\mathrm{P}$ *and that* $\oplus\mathrm{P}$ *is closed under* $\le^{\mathrm{P}}_{\mathrm{maj}}$.

*Proof Sketch.* Let $L$ be a set in $\oplus\mathrm{P}(\oplus\mathrm{P})$. Then there exist a set $A \in \oplus\mathrm{P}$ and an oracle NP-machine M such that for every $x$, $x \in L$ if and only if the number of accepting paths of $\mathrm{M}(A)$ on input $x$ is odd. Let $\mathrm{M}_1$ be an NP-machine that witnesses $A \in \oplus\mathrm{P}$. Then we define an NP-machine $\mathrm{M}_2$ working on a given input $x$ as follows:

(1) $\mathrm{M}_2$ first guesses a computation path $w$ of M on input $x$, which includes possible oracle answers to the query strings appearing in $w$.

(2) If $w$ is a rejecting path of M on $x$, then $\mathrm{M}_2$ enters a rejecting state; otherwise, it goes to the next step.

(3) Let $y_1, y_2, \cdots, y_m$ $(z_1, z_2, \cdots, z_l)$ be all the query strings which appear in $w$ and whose corresponding oracle answers in $w$ are "yes" (respectively, "no"). Then $\mathrm{M}_2$ simulates $\mathrm{M}_1$ successively for each $y_i$ and each $z_j$ in the following manner:

(a) For each $y_i$, it simply simulates $\mathrm{M}_1$. If $\mathrm{M}_1$ enters a rejecting state, then so does $\mathrm{M}_2$; otherwise, it proceeds to the next simulation.

(b) For each $z_i$, it nondeterministically selects one of the following processes:

(i) $\mathrm{M}_2$ goes to the next simulation. (Intuitively speaking, this process can be regarded as a dummy-accepting path of $\mathrm{M}_1$ on input $z_i$.)

(ii) $\mathrm{M}_2$ simulates $\mathrm{M}_1$ on $z_i$. If $\mathrm{M}_1$ enters a rejecting state, then so does $\mathrm{M}_2$; otherwise, it goes to the next simulation.

(4) $\mathrm{M}_2$ enters an accepting state.

We can classify all possible accepting computation paths of M on input $x$ into two groups, one of which consists of accepting paths of $\mathrm{M}(A)$ on $x$ (group 1), and the other consists of the remaining ones (group 2). Obviously, every accepting path in group 1 contains correct oracle answers of the oracle set $A$, and every accepting path in group 2 contains a wrong oracle answer. From the definition of $\mathrm{M}_2$, we can easily see that every accepting path in group 1 is followed by an odd number of accepting paths in steps 3 and 4, and every accepting path in group 2 is followed by an even number of accepting paths in those steps. From this observation, it is not difficult to see that for every $x$, $x \in L$ if and only if the number of accepting paths of $\mathrm{M}_2$ on input $x$ is odd. We leave the verification to the interested reader. Thus $L$ is in $\oplus\mathrm{P}$. The other statements are immediate from the first one. □

The following theorem was shown by Schöning [15].

THEOREM 3.5 [15]. *Let* $\mathbf{K}$ *be a class of sets which is closed under* $\le^{\mathrm{P}}_{\mathrm{maj}}$. *Then for all sets* $A \in \mathsf{BP} \cdot \mathbf{K}$ *and all polynomials* $q$, *there exist a set* $B$ *and a polynomial* $p$ *such that, for all* $n$,

$$\mathrm{Prob}\,(\{y \in \{0,1\}^{p(n)} \colon (\forall x, |x| = n)[x \# y \in B \leftrightarrow x \in A]\}) \geqq 1 - 2^{-q(n)}.$$

LEMMA 3.6. $\oplus \cdot \mathsf{BP} \cdot \oplus\mathrm{P} \subseteq \mathsf{BP} \cdot \oplus\mathrm{P}$.

*Proof.* Let $L \in \oplus \cdot \mathsf{BP} \cdot \oplus\mathrm{P}$. Then there exist a set $A \in \mathsf{BP} \cdot \oplus\mathrm{P}$ and a polynomial $p$ such that for each $x$, $x \in L$ if and only if

$$\|\{w \colon |w| = p(|x|) \text{ and } x \# w \in A\}\|$$

is odd. Furthermore, there exist a set $B \in \oplus\mathrm{P}$, a polynomial $q$, and a constant $\alpha > 0$ such that for each $y$,

$$\mathrm{Prob}\,(\{u \in \{0,1\}^{q(|y|)} \colon y \# u \in B \leftrightarrow y \in A\}) \geqq \tfrac{1}{2} + \alpha.$$

Since $\oplus\mathrm{P}$ is closed under $\le^{\mathrm{P}}_{\mathrm{maj}}$, we may assume, from Theorem 3.5, that for all $m$,

$$\mathrm{Prob}\,(\{u \in \{0,1\}^{q(m)} \colon (\forall y, |y| = m)[y \# u \in B \leftrightarrow y \in A]\}) \geqq \tfrac{3}{4}.$$

Hence we have that for all $x$ of length $n$,

$$\text{Prob}\,(\{u \in \{0,1\}^{q(n+1+p(n))}\colon (\forall w \in \{0,1\}^{p(n)})[x \# w \# u \in B \leftrightarrow x \# w \in A]\}) \geqq \tfrac{3}{4},$$

and hence,

$$(1) \quad \text{Prob}\,(\{u \in \{0,1\}^{q(n+1+p(n))}\colon \{w \in \{0,1\}^{p(n)}\colon x \# w \# u \in B\} = \{w \in \{0,1\}^{p(n)}\colon x \# w \in A\}\}) \geqq \tfrac{3}{4}.$$

For a string $x$ of length $n$, assume $x \in L$. Then $\|\{w \in \{0,1\}^{p(n)}\colon x \# w \in A\}\|$ is odd. Hence from (1),

$$(2) \quad \text{Prob}\,(\{u \in \{0,1\}^{q(n+1+p(n))}\colon \|\{w \in \{0,1\}^{p(n)}\colon x \# w \# u \in B\}\| \text{ is odd}\}) \geqq \tfrac{3}{4}.$$

Conversely, assume $x \notin L$. Then $\|\{w \in \{0,1\}^{p(n)}\colon x \# w \in A\}\|$ is even. Hence from (1),

$$(3) \quad \text{Prob}\,(\{u \in \{0,1\}^{q(n+1+p(n))}\colon \|\{w \in \{0,1\}^{p(n)}\colon x \# w \# u \in B\}\| \text{ is odd}\}) \leqq \tfrac{1}{4}.$$

We now define $B'$ and $A'$ by

$$B' = \{x \# u \# w\colon |u| = q(|x|+1+p(|x|)),\ |w| = p(|x|), \text{ and } x \# w \# u \in B\}$$

and

$$A' = \{x \# u\colon |u| = q(|x|+1+p(|x|)) \text{ and } x \# u \# w \in B' \text{ for an odd number of } w \in \{0,1\}^{p(|x|)}\}.$$

Then we obtain that $\text{Prob}\,(\{u \in \{0,1\}^{q(|x|+1+p(|x|))}\colon x \# u \in A' \leftrightarrow x \in L\}) \geq \frac{3}{4}$ from (2) and (3) above. It is easy to see that $B' \in \oplus\text{P}$ and $A' \in \oplus \cdot \oplus\text{P}$. Hence, $A'$ is in $\oplus\text{P}$ from Theorem 3.4. This implies $L \in \text{BP} \cdot \oplus\text{P}$. □

LEMMA 3.7. $\text{BP} \cdot \text{BP} \cdot \oplus\text{P} = \text{BP} \cdot \oplus\text{P}$.

*Proof.* It suffices to show the inclusion $\text{BP} \cdot \text{BP} \cdot \oplus\text{P} \subseteq \text{BP} \cdot \oplus\text{P}$. Let $L \in \text{BP} \cdot \text{BP} \cdot \oplus\text{P}$. Then there exist a set $A \in \text{BP} \cdot \oplus\text{P}$ and a polynomial $p$ such that for each $x$,

$$\text{Prob}\,(\{w \in \{0,1\}^{p(|x|)}\colon x \# w \in A \leftrightarrow x \in L\}) \geqq \tfrac{3}{4}.$$

Furthermore, there exist a set $B \in \oplus\text{P}$ and a polynomial $q$ such that for each $y$,

$$\text{Prob}\,(\{u \in \{0,1\}^{q(|y|)}\colon y \# u \in B \leftrightarrow y \in A\}) \geqq \tfrac{7}{8}.$$

Note that we are using Theorem 3.5 in this setting. We now define a set $C$ by

$$C = \{x \# wu\colon |w| = p(|x|),\ |u| = q(|x \# w|) = q(|x|+1+p(|x|)), \text{ and } x \# w \# u \in B\}.$$

It is easy to see that $C$ is in $\oplus\text{P}$. For a string $x$, if $x \in L$, then

$$\begin{aligned}
(1) \quad & \text{Prob}\,(\{wu \in \{0,1\}^{p(|x|)+q(|x|+1+p(|x|))}\colon x \# wu \in C\}) \\
&= \text{Prob}\,(\{w \in \{0,1\}^{p(|x|)}\colon x \# w \in A\}) \\
&\qquad \times \text{Prob}\,(\{u \in \{0,1\}^{q(|x|+1+p(|x|))}\colon x \# w \# u \in B\} \mid x \# w \in A) \\
&\qquad + \text{Prob}\,(\{w \in \{0,1\}^{p(|x|)}\colon x \# w \notin A\}) \\
&\qquad \times \text{Prob}\,(\{u \in \{0,1\}^{q(|x|+1+p(|x|))}\colon x \# w \# u \in B\} \mid x \# w \notin A) \\
&\geqq \tfrac{3}{4} \cdot \tfrac{7}{8} = \tfrac{21}{32} = \tfrac{1}{2} + \tfrac{5}{32}
\end{aligned}$$

where Prob $(X/Y)$ denotes the conditional probability of the event $X$ under the condition $Y$.

Conversely, if $x \notin L$, then

$$\begin{aligned}(2)\qquad &\text{Prob}\,(\{wu \in \{0,1\}^{p(|x|)+q(|x|+1+p(|x|))}: x \# wu \in C\})\\ &= \text{Prob}\,(\{w \in \{0,1\}^{p(|x|)}: x \# w \in A\})\\ &\qquad \times \text{Prob}\,(\{u \in \{0,1\}^{q(|x|+1+p(|x|))}: x \# w \# u \in B\} \,|\, x \# w \in A)\\ &\qquad + \text{Prob}\,(\{w \in \{0,1\}^{p(|x|)}: x \# w \notin A\})\\ &\qquad \times \text{Prob}\,(\{u \in \{0,1\}^{q(|x|+1+p(|x|))}: x \# w \# u \in B\} \,|\, x \# w \notin A)\\ &\leqq \tfrac{1}{4}\cdot 1 + 1 \cdot \tfrac{1}{8} = \tfrac{3}{8} = \tfrac{1}{2} - \tfrac{1}{8}.\end{aligned}$$

Thus we have that for every $x$,

$$\text{Prob}\,(\{v \in \{0,1\}^{p(|x|)+q(|x|+1+p(|x|))}: x \# v \in C \leftrightarrow x \in L\}) \geqq \tfrac{1}{2} + \tfrac{1}{8}.$$

This implies that $L \in \mathsf{BP}\cdot\oplus\mathrm{P}$. □

Now we can prove Theorem 3.1.

*Proof of Theorem* 3.1. We prove this theorem by induction on the levels of the polynomial-time hierarchy. The inclusion $\Sigma_0^{\mathrm{P}} = P \subseteq \mathsf{BP}\cdot\oplus\mathrm{P}$ is obvious. We now assume $\Sigma_{k-1}^{\mathrm{P}} \subseteq \mathsf{BP}\cdot\oplus\mathrm{P}$ for some $k > 0$. It is easy to see that $\mathsf{BP}\cdot\oplus\mathrm{P}$ is closed under complementation. Hence we have the inclusion $\Pi_{k-1}^{\mathrm{P}} \subseteq \mathsf{BP}\cdot\oplus\mathrm{P}$. Then,

$$\begin{aligned}\Sigma_k^{\mathrm{P}} &\subseteq \mathsf{BP}\cdot\oplus\cdot\Pi_{k-1}^{\mathrm{P}} \quad \text{(from Lemma 3.3)}\\ &\subseteq \mathsf{BP}\cdot\oplus\cdot\mathsf{BP}\cdot\oplus\mathrm{P} \quad \text{(from inductive assumption)}\\ &= \mathsf{BP}\cdot\oplus\mathrm{P} \quad \text{(from Lemma 3.6 and Lemma 3.7).}\end{aligned}$$

Thus we conclude that $\mathrm{PH} = \bigcup_{k\geqq 0}\Sigma_k^{\mathrm{P}} \subseteq \mathsf{BP}\cdot\oplus\mathrm{P}$.

It was shown by Schöning [15] that $\Pi_k^{\mathrm{P}} \subseteq \mathsf{BP}\cdot\Sigma_k^{\mathrm{P}}$ implies $\Sigma_{k+1}^{\mathrm{P}} = \Pi_{k+1}^{\mathrm{P}}$ for every $k \geqq 1$, which is regarded as a refinement of the result by Karp and Lipton [7]. Combining this result with Theorem 3.1, we observe that $\oplus\mathrm{P}$ is harder than PH unless PH collapses to a finite level. More precisely, we obtain the following corollary.

COROLLARY 3.8. *For every* $k \geqq 1$, *if* $\oplus\mathrm{P} \subseteq \Sigma_k^{\mathrm{P}}$, *then* $\mathrm{PH} = \Sigma_{k+1}^{\mathrm{P}}$. *Hence* $\oplus\mathrm{P} \subseteq \mathrm{PH}$ *implies that* PH *collapses at a finite level.*

The second statement in this corollary follows from the fact that $\oplus\mathrm{P}$ has a complete set under $\leq_{\mathrm{m}}^{\mathrm{P}}$-reducibility.

## 4. PP is $\leqq_{\mathrm{T}}^{\mathrm{P}}$-hard for PH.

In this section, we prove the following theorem.

THEOREM 4.1. $\mathbf{C}\cdot\oplus\mathrm{P} \subseteq \mathrm{P(PP)}$.

It is easy to see that $\mathsf{BP}\cdot\oplus\mathrm{P} \subseteq \mathbf{C}\cdot\oplus\mathrm{P}$. Hence the Main Theorem in § 1 follows immediately from this theorem and from Theorem 3.1.

The following lemma plays an important role in the proof of Theorem 4.1, and depends on an interesting numerical property. For an NP-machine N and an input $y$, let $\#\mathrm{acc}_{\mathrm{N}}(y)$ denote the number of accepting computation paths of N on input $y$.

LEMMA 4.2. *Let $X$ be a set in* $\oplus\mathrm{P}$ *and let $q$ be a polynomial. Then, there exists an* NP-*machine* $\mathrm{N}_1$ *such that for each input $y$ of length $n$,*

(1) *if* $y \notin X$, *then* $\#\mathrm{acc}_{\mathrm{N}_1}(y) \equiv 0 \pmod{2^{q(n)}}$, *and*

(2) *if* $y \in X$, *then* $\#\mathrm{acc}_{\mathrm{N}_1}(y) \equiv -1 \pmod{2^{q(n)}}$.

Before proving this lemma, we give an intuitive explanation about our proof of Theorem 4.1 and about the role of Lemma 4.2. Let $L$ be a set in $\mathbf{C}\cdot\oplus\mathrm{P}$. Then there exist a set $X \in \oplus\mathrm{P}$ and a polynomial $p$ such that for every $x$,

$$\text{Prob}\,(\{w \in \{0,1\}^{p(|x|)}: x \# w \in X \text{ iff } x \in L\}) > \tfrac{1}{2}.$$

Let $N_1$ be an NP-machine satisfying the conditions in Lemma 4.2 for the set $L$ and the polynomial $q(n) = n$. Consider an NP-machine N which operates as follows. Given an input $x$ of length $n$, N first guesses a string $w$ of length $p(n)$, and it begins to simulate $N_1$ on input $x \# w$. If $N_1$ enters an accepting state, then N enters an accepting state; otherwise, it enters a rejecting state. Let $\# X[x]$ denote the number of strings $w$ such that $|w| = p(n)$ and $x \# w \in X$. From Lemma 4.2, it is not difficult to see that $\# \mathrm{acc}_N(x) = 2^{n+1+p(n)} \cdot k_x - \# X[x]$ for some natural number $k_x \geqq 0$. By a standard binary search technique, $\# \mathrm{acc}_N(x)$ can be computed in polynomial time with an oracle set from PP. After this, we can also get the value of $\# X[x]$ within polynomial time by computing $\# \mathrm{acc}_N(x) \bmod 2^{n+1+p(n)}$, because $\# X[x] \leqq 2^{p(n)} < 2^{n+1+p(n)}$. Finally, we decide to accept the input $x$ if and only if $\# X[x] > 2^{p(n)-1}$. Hence, we can conclude that $\mathbf{C} \cdot \oplus \mathrm{P} \subseteq \mathrm{P(PP)}$.

The recurrence relation in the next lemma provides the key numerical property. Intuitively speaking, it gives us a whole exponential factor of freedom in our counting.

simple polyrec definition.

LEMMA 4.3. *For an integer $m$, define the sequence $s_0, s_1, s_2, \cdots$, inductively by $s_0 = m$ and for $i \geqq 1$, $\underline{s_i = 3 \cdot s_{i-1}^4 + 4 \cdot s_{i-1}^3}$. Then,*

(1) *if $m$ is even, then for all $i$, $s_i$ is a multiple of $2^{2^i}$, and*

(2) *if $m$ is odd, then for all $i$, $s_i + 1$ is a multiple of $2^{2^i}$.*

*Proof.* The statement (1) is obvious from the definition of the sequence. We prove (2). The case $i = 0$ is obvious. We assume that for some $i > 0$, $s_{i-1} = 2^{2^{i-1}} \cdot k_{i-1} - 1$ for some positive integer $k_{i-1}$. Then, from the definition of $s_i$,

$$\begin{aligned} s_i &= 3 \cdot s_{i-1}^4 + 4 \cdot s_{i-1}^3 \\ &= 3 \cdot (2^{2^{i-1}} \cdot k_{i-1} - 1)^4 + 4 \cdot (2^{2^{i-1}} \cdot k_{i-1} - 1)^3 \\ &= 3 \cdot 2^{2^{i+1}} \cdot k_{i-1}^4 - 8 \cdot 2^{3 \cdot 2^{i-1}} \cdot k_{i-1}^3 + 6 \cdot 2^{2^i} \cdot k_{i-1}^2 - 1 \\ &= 2^{2^i} \cdot (3 \cdot 2^{2^i} \cdot k_{i-1}^4 - 8 \cdot 2^{2^{i-1}} \cdot k_{i-1}^3 + 6 \cdot k_{i-1}^2) - 1. \end{aligned}$$

Hence $s_i$ is a multiple of $2^{2^i}$. □

Accordingly, we make the following recursive definition of a function $f_N : \Sigma^* \times \mathbf{N} \to \mathbf{N}$, where $\Sigma$ is the input alphabet of a given NTM N.

DEFINITION 4.4. For an NP-machine N and an input $y$, define

$$f_N(y, 0) = \# \mathrm{acc}_N(y), \quad \text{and for } i \geqq 1,$$

$$f_N(y, i) = 3 \cdot (f_N(y, i-1))^4 + 4 \cdot (f_N(y, i-1))^3.$$

LEMMA 4.5. *Let N be an NP-machine, and let $q(n)$ be a polynomial. Then for all input $y$ of length $n$,*

(1) *if $\# \mathrm{acc}_N(y)$ is even, then $f_N(y, \lceil \log_2 q(n) \rceil) \equiv 0 \pmod{2^{q(n)}}$, and*

(2) *if $\# \mathrm{acc}_N(y)$ is odd, then $f_N(y, \lceil \log_2 q(n) \rceil) \equiv -1 \pmod{2^{q(n)}}$.*

*Proof.* Since $2^{2^{\lceil \log_2 q(n) \rceil}} = 2^{q(n)+k} = 2^{q(n)} \cdot 2^k$ for some natural number $k$, this follows immediately from Lemma 4.3. □

Given a set $X \in \oplus \mathrm{P}$, it follows from the definition of $\oplus \mathrm{P}$ that there is an NP-machine N such that for all $y$, $y \in X$ if and only if $\# \mathrm{acc}_N(y)$ is odd. Hence, to obtain Lemma 4.2, it remains to show how to construct an NP-machine Q such that for all $y$,

$$\# \mathrm{acc}_Q(y) = f_N(y, \lceil \log_2 q(n) \rceil).$$

This is accomplished in the following lemma.

LEMMA 4.6. *Let* N *be an* NP-*machine, and let t be a polynomial which bounds the runtime of* N. *Then we can find an* NTM Q *which takes inputs of the form* $y \# 1^i$, *and a constant* $c > 0$ *such that for all* $y, i$:

(1) $\#\mathrm{acc}_Q(y \# 1^i) = f_N(y, i)$, *and*

(2) *all computations of* Q *on input* $y \# 1^i$ *halt within* $c \cdot 4^i \cdot (t(|y|)+1)$ *steps.*

*Proof.* Intuitively speaking, the required machine Q is designed so that for each input $y \# 1^i$, it executes itself recursively on input $y \# 1^{i-1}$ according to the definition of $f_N$. This can be done by using stack operations. Furthermore, since the depth of recursive executions for input $y \# 1^i$ is at most $i$, and at most four sequential calls are made at each level, the required time bound is obtained. We now describe Q as a recursive procedure, using a stack for the recursive executions.

```
PROCEDURE Q(y, i), where the input is written in the form $y \# 1^i$.
Step 1: if $i = 0$ then simulate M on input y;
                    if M enters an accepting state
                      then return "ACCEPT" else return "REJECT";
Step 2: guess one of the following subprocesses nondeterministically;
    (subprocess 1)
    branch away nondeterministically into three branches;
    execute the following in each branch;
    for $j := 1$ to 4 do
        execute $Q(y, i-1)$ recursively;
        {this can be done by pushing $i, j$ and return position into a stack before
         execution and by popping those off the stack after execution}
        if this call to $Q(y, i-1)$ returns "REJECT" then return "REJECT"
    od;
    return "ACCEPT";
    (subprocess 2)
    branch away nondeterministically into four branches;
    execute the following in each branch;
    for $j := 1$ to 3 do
        execute $Q(y, i-1)$ recursively;
        {this can be done by pushing $i, j$ and return position into a stack before
         execution and by popping those off the stack after execution}
        if this call to $Q(y, i-1)$ returns "REJECT" then return "REJECT"
    od;
    return "ACCEPT."
```

By induction on $i$, it is not difficult to show that for each input $y \# 1^i$, the number of accepting computation paths of Q is equal to $f_N(y, i)$. The essence of this proof is to estimate the runtime of the above machine. Let $y \# 1^i$ be an input for Q and let $T(y, i)$ denote the runtime of Q on input $y \# 1^i$. It is not difficult to see that stack operations and the other bookkeeping operations in Step 2 can be done within time at most $O(i)$, say $c \cdot i + c$ for some $c > 0$, if we denote natural numbers by unary notation. Furthermore, the operations in Step 1 can be done within a constant time, say $c > 0$. Then, we obtain the following inequalities from the definition of $M_1$:

$$T(y, 0) \leqq t(|y|) + c, \tag{1}$$

and

$$T(y, i) \leqq 4 \cdot (T(y, i-1) + c \cdot i + c) \quad \text{for each } i > 0. \tag{2}$$

From this, we have:

(3) $T(y, i) \leqq 4^i \cdot t(|y|) + \Sigma_{k=1}^{i} 4^k \cdot (c \cdot i + c) = 4^i \cdot t(|y|) + \frac{4}{3} \cdot (4^i - 1) \cdot (c \cdot i + c)$.

Thus we finally have $T(y, i) \leqq O(4^i \cdot (t(|y|) + i))$. This completes the proof.

*Proof of Lemma* 4.2. Let $\mathrm{N}_1$ on input $y$ simulate Q of Lemma 4.6 on input $y \# 1^{\lceil \log_2 q(|y|) \rceil}$. Then $\mathrm{N}_1$ runs in polynomial time in $|y|$, and satisfies the properties required in Lemma 4.2. □

Before deducing Theorem 4.1, we state and prove a technically stronger result. A function $h: \Sigma^* \rightarrow \mathbf{N}$ is said to belong to the class #P [23], [24] if there is an NP-machine N such that for all $x \in \Sigma^*$, $h(x) = \#\mathrm{acc}_{\mathrm{N}}(x)$. Then $\mathrm{P}^{\#\mathrm{P}[1]}$ stands for the class of sets which can be solved in polynomial time with one free evaluation of a #P function. Papadimitriou and Zachos [13] showed that $\mathrm{P}^{\mathrm{NP}[\log]} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$ (calling the latter class "#P"). We show that the whole polynomial-time hierarchy is contained in $\mathrm{P}^{\#\mathrm{P}[1]}$, as a consequence of Theorem 4.7.

THEOREM 4.7. $\mathbf{C} \cdot \oplus \mathrm{P} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$.

*Proof.* Let $L \in \mathbf{C} \cdot \oplus \mathrm{P}$. Then there is a set $X \in \oplus \mathrm{P}$ and a polynomial $p$ such that for all $x$, putting $W_x = \{w \in \{0,1\}^{p(|x|)} : x \# w \in X\}$, we have $x \in L$ if and only if $\|W_x\| > 2^{p(|x|)-1}$. Then from Lemma 4.2, we can find an NP-machine N such that for all inputs $y$, with $m = |y|$,

(1) if $y \in X$, then for some integer $\alpha_y > 0$, $\#\mathrm{acc}_{\mathrm{N}}(y) = 2^m \cdot \alpha_y - 1$, and

(2) if $y \notin X$, then for some integer $\beta_y \geqq 0$, $\#\mathrm{acc}_{\mathrm{N}}(y) = 2^m \cdot \beta_y$.

Now let Z be an NTM which on every input $x$ of length $n$ does this:

(1) Guess $w \in \{0,1\}^{p(n)}$.

(2) Simulate N on input $x \# w$, accepting if and only if N accepts.

Then Z clearly runs in polynomial time. Now writing $\tilde{W}_x$ for $\{0,1\}^{p(n)} - W_x$, we have:

$$\begin{aligned}
\#\mathrm{acc}_{\mathrm{Z}}(x) &= \sum_{w \in W_x} \#\mathrm{acc}_{\mathrm{N}}(x \# w) + \sum_{w \in \tilde{W}_x} \#\mathrm{acc}_{\mathrm{N}}(x \# w) \\
&= \sum_{w \in W_x} (2^{|x \# w|} \cdot \alpha_{x \# w} - 1) + \sum_{w \in \tilde{W}_x} (2^{|x \# w|} \cdot \beta_{x \# w}) \\
&= 2^{|x|+1+p(|x|)} \cdot \left( \sum_{w \in W_x} \alpha_{x \# w} + \sum_{w \in \tilde{W}_x} \beta_{x \# w} \right) - \|W_x\|.
\end{aligned}$$

Since every $\alpha_{x \# w}$ and $\beta_{x \# w}$ is integral, it follows that $\#\mathrm{acc}_{\mathrm{Z}}(x) + \|W_x\|$ is a multiple of $2^{|x|+1+p(|x|)}$. Since $\|W_x\| \leqq 2^{p(|x|)} < 2^{|x|+1+p(|x|)}$, it follows that $\|W_x\|$ can be computed simply by complementing the last $p(|x|)$ bits of $\#\mathrm{acc}_{\mathrm{Z}}(x)$ in binary notation. That is to say, $x \in L$ if and only if the $p(|x|)$th bit of $\#\mathrm{acc}_{\mathrm{Z}}(x)$ from the right is a "0." Since Z is a polynomial-time-bounded NTM, $\#\mathrm{acc}_{\mathrm{Z}}(\cdot)$ is a #P function, and the theorem follows. □

*Proof of Theorem* 4.1. It is well known that for every #P function $h$, its *graph* $\{x \# k : h(x) \leqq k\}$ belongs to PP [14]. By the standard binary search technique, $h(x)$ can be computed with $O(|x|)$-many queries to its graph. So $\mathbf{C} \cdot \oplus \mathrm{P}$, and hence PH, is included in P(PP). □

The following corollary is straightforward from the Main Theorem.

COROLLARY 4.8. *For each* $k \geqq 0$, *if* $\mathrm{PP} \subseteq \Sigma_k^{\mathrm{P}}$, *then* PH *collapses to* $\Sigma_k^{\mathrm{P}}$. *Furthermore, if* $\mathrm{PP} \subseteq \mathrm{PH}$, *then* PH *collapses to a finite level.*

*Proof.* Assume $\mathrm{PP} \subseteq \Sigma_k^{\mathrm{P}}$. It is well known that PP is closed under complementation. Hence we have $\mathrm{PP} \subseteq \Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}}$ from the assumption. It is also well known that $\mathrm{P}(\Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}}) \subseteq \Sigma_k^{\mathrm{P}}$ for each $k \geqq 0$. From the main theorem, we have

$$\mathrm{PH} \subseteq \mathrm{P(PP)} \subseteq \mathrm{P}(\Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}}) \subseteq \Sigma_k^{\mathrm{P}}.$$

$NP \subseteq P^{PP}$

$\not\Downarrow$

$NP^{PP} \subseteq (P^{PP})^{PP}$

$PP(PH(A)) \subseteq PP(BPP(\oplus P(A))) \subseteq PP(\oplus P(A)) \subseteq P(PP(A))$

Th3.1: $PH(A) \subseteq BPP(\oplus P(A))$

The second statement is easily obtained from the fact that PP has a complete set under $\leq_m^P$-reducibility. □

At the end of this section, we observe a result stronger than the Main Theorem. It was shown by Köbler et al. [10] that PP(BPP) = PP. Furthermore, the equality can be relativized to all oracle sets. More precisely, we have the following result.

THEOREM 4.9 [10]. *For all oracle sets A,* PP(BPP(*A*)) = PP(*A*).

From this theorem, we have the following theorem.

THEOREM 4.10. PP(PH) *is included in* P(PP).

it even relativizes: $PP(PH(A)) \subseteq P(PP(A))$

$BP\cdot\oplus P = BPP^{\oplus P} \subseteq PP^{\oplus P}$

*Proof.* It is easy to see that PP(PH) ⊆ PP(BP · ⊕P) ⊆ PP(BPP(⊕P)). These inclusions follow from Theorem 3.1 and from the definition of BP-operator. From Theorem 4.9, we have PP(PH) ⊆ PP(⊕P). It is not hard to show that PP(⊕P) = C · P(⊕P) = C · ⊕P. Some techniques for showing this have appeared in [21], [26]. Hence we obtain this theorem from Theorem 4.1.

$PP^{\oplus P} = PP^{P^{\oplus P}} = PP^{\oplus P}$

**5. Concluding remarks.** In this paper, we showed that every set in PH (and in PP(PH)) is polynomial-time Turing reducible to a set in PP. We also show a similar result about ⊕P. There are some further questions that are related to this work. A simple question is whether we can show, by using a different kind of reducibility such as polynomial-time truth-table reducibility, that PH is reducible to PP. In fact, we showed that PH is included in $P^{\#P[1]}$; this is a somewhat stronger statement than PH ⊆ P(PP). On the other hand, it is well known that every set which is $\leq_{tt}^P$-reducible to a set in PP is in $P^{\#P[1]}$; but the converse is unknown. Hence the answer to the above question will give us a somewhat stronger result than the present result. The other interesting question is whether $C_=P$ [26], [21] is as hard as PH. A more important question is whether NP(PP) is included in P(PP), or whether PP(PP) is included in P(PP). It is also interesting to find oracle sets that separate those classes from each other.

!

This means $NP \subseteq P^{PP}$ does not relativize ??

**Acknowledgment.** I am very thankful to Osamu Watanabe for helpful discussions and some nice advice, to Lane Hemachandra for his comments on this work, and to Kenneth Regan for his many suggestions on the earlier version of this paper. I am very thankful to the Program Committee of the 4th IEEE Conference on Structure in Complexity Theory in 1989 for giving me an opportunity to talk about this result at the conference; concerning this, I have to thank Richard Beigel, Lane Hemachandra, and Gerd Wechsung for giving much of their lecture time to me. I want to thank the referees of this paper. In particular, one of the referees gave me many suggestions which made the quality of this paper much better. I would also like to thank Ron Book and Janos Simon for their suggestions.

## REFERENCES

[1] D. ANGLUIN, *On counting problems and the polynomial-time hierarchy*, Theoret. Comput. Sci., 12 (1980), pp. 161–173.

[2] J. L. BALCÁZAR, R. V. BOOK, AND U. SCHÖNING, *The polynomial-time hierarchy and sparse oracles*, J. Assoc. Comput. Mach., 33 (1986), pp. 603–617.

[3] R. BEIGEL, L. A. HEMACHANDRA, AND G. WECHSUNG, *On the power of probabilistic polynomial-time*: $P^{NP[\log]} \subseteq PP$, in Proc. 4th IEEE Conference on Structure in Complexity Theory, 1989, pp. 225–227.

[4] JIN-YI CAI AND L. A. HEMACHANDRA, *On the power of parity polynomial time*, in Proc. Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science 349, 1989, Springer-Verlag, Berlin, pp. 229–240.

[5] J. GILL, *Computational complexity of probabilistic Turing machines*, SIAM J. Comput., 6 (1977), pp. 675–695.

[6] L. A. HEMACHANDRA, *On ranking*, in Proc. 2nd IEEE Conference on Structure in Complexity Theory, 1987, pp. 103–117.

[7] R. KARP AND R. LIPTON, *Some connections between nonuniform and uniform complexity classes*, in Proc. 12th ACM Symposium on Theory of Computing, 1980, pp. 302–309.

[8] K. KO, *Some observations on the probabilistic algorithms and* NP*-hard problems*, Inform. Process. Lett., 14 (1982), pp. 39–43.

[9] J. KÖBLER, U. SCHÖNING, AND J. TORAN, *On counting and approximation*, Acta Inform., 26 (1989), pp. 363–379.

[10] J. KÖBLER, U. SCHÖNING, S. TODA, AND J. TORAN, *Turing machines with few accepting computations and low sets for* PP, in Proc. 4th IEEE Conference on Structure in Complexity Theory, 1989, pp. 208–215.

[11] C. LAUTEMAN, BPP *and the polynomial-time hierarchy*, Inform. Process. Lett., 14 (1983), pp. 215–217.

[12] T. J. LONG AND A. L. SELMAN, *Relativizing complexity classes with sparse oracles*, J. Assoc. Comput. Mach., 33 (1986), pp. 618–627.

[13] C. H. PAPADIMITRIOU AND S. ZACHOS, *Two remarks on the power of counting*, in Proc. 6th Gesellschaft für Informatik Conference on Theoretical Computer Science, Lecture Notes in Computer Science 145, 1983, Springer-Verlag, Berlin, pp. 269–276.

[14] J. SIMON, *On the difference between one and many*, in Proc. 4th Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science 52, 1977, Springer-Verlag, Berlin, pp. 480–491.

[15] U. SCHÖNING, *Probabilistic complexity classes and lowness*, in Proc. 2nd IEEE Conference on Structure in Complexity Theory, 1987, pp. 2–8; also in J. Comput. System Sci., 39 (1989), pp. 84–100.

[16] ———, *The power of counting*, in Proc. 3rd IEEE Conference on Structure in Complexity Theory, 1988, pp. 2–9.

[17] M. SIPSER, *A complexity theoretic approach to randomness*, in Proc. 15th ACM Symposium on Theory of Computing, 1983, pp. 330–335.

[18] L. J. STOCKMEYER, *The polynomial-time hierarchy*, Theoret. Comput. Sci., 3 (1977), pp. 1–22.

[19] ———, *On approximation algorithms for* #P, SIAM J. Comput., 14 (1985), pp. 849–861.

[20] S. TODA, *Restricted relativizations of probabilistic polynomial-time*, Theoret. Comput. Sci., 1990, accepted.

[21] J. TORAN, *An oracle characterization of the counting hierarchy*, in Proc. 3rd IEEE Conference on Structure in Complexity Theory, 1988, pp. 213–223.

[22] L. G. VALIANT, *Relative complexity of checking and evaluating*, Inform. Process. Lett., 5 (1976), pp. 20–23.

[23] ———, *The complexity of computing the permanent*, Theoret. Comput. Sci., 8 (1979), pp. 189–201.

[24] ———, *The complexity of reliability and enumeration problems*, SIAM J. Comput., 8 (1979), pp. 410–421.

[25] L. G. VALIANT AND V. V. VAZIRANI, NP *is as easy as detecting unique solutions*, Theoret. Comput. Sci., 47 (1986), pp. 85–93.

[26] K. WAGNER, *The complexity of combinatorial problems with succinct input representation*, Acta Inform., 23 (1986), pp. 325–356.

[27] ———, *Some observations on the connection between counting and recursion*, Theoret. Comput. Sci., 47 (1986), pp. 131–147.

[28] C. WRATHALL, *Complete sets and the polynomial-time hierarchy*, Theoret. Comput. Sci., 3 (1977), pp. 23–33.

[29] S. ZACHOS, *Robustness of probabilistic computational complexity classes under definitional perturbations*, Inform. and Control, 54 (1982), pp. 143–154.

[30] S. ZACHOS AND H. HELLER, *A decisive characterization of* BPP, Inform. and Control, 69 (1986), pp. 125–135.