On the power of deterministic reductions to $C_{=}P$

Frederic Green

Report LSI–91–17

On the Power of Deterministic Reductions to $C_=P$

(Preliminary Report)

Frederic Green *
Department of Mathematics and Computer Science
Clark University
Worcester, Massachusetts 01610
fgreen@clarku.bitnet

April 3, 1991

Abstract

The counting class $C_=P$, which captures the notion of "exact counting", while extremely powerful under various nondeterministic reductions, is quite weak under polynomial-time deterministic reductions. We discuss the analogies between NP and co-$C_=P$, which allow us to derive many interesting results for such deterministic reductions to co-$C_=P$. We exploit these results to obtain some interesting oracle separations. Most importantly, we show that there exists an oracle A such that $\oplus P^A \not\subseteq P^{C_=P^A}$ and $BPP^A \not\subseteq P^{C_=P^A}$. From this we can conclude that techniques that would prove that $C_=P$ and PP are polynomial-time Turing equivalent would not relativize.

1 Introduction

The class $C_=P$ (see section 2 for precise definitions) is an extremely powerful counting class which captures the notion of "exact counting". It can be characterized by nondeterministic machines which accept if and only if the number of accepting paths is exactly equal to a given number. The power of $C_=P$ can be seen from the following facts:

Facts:
(i) $PP^{PH} \subseteq NP^{C_=P}$.
(ii) $C_=P^{PH} \subseteq BP \cdot C_=P$.

Fact (i) is a consequence of Toda's theorem [12] that $PP^{PH} \subseteq P^{PP}$ combined with a theorem of Torán [14] which states $NP^{PP} = NP^{C_=P}$. It is significant since it states that $C_=P$ is hard for the polynomial hierarchy under nondeterministic reductions. Fact (ii) was proved

*Research supported by a grant from the Dirección General de Investigación Científica y Técnica (DGICYT), Spanish Ministry of Education, while the author was visiting the Facultat de Informàtica, Universitat Polytècnica de Catalunya, Barcelona.

by Toda and Ogiwara [13], and in a stronger form by Tarui [11]. It is significant since it says that $C_=P$ is hard for the polynomial hierarchy under randomized reductions.

The purpose of this paper is to observe that although $C_=P$ is quite powerful under nondeterministic or randomized reductions, nevertheless under deterministic reductions it appears to be quite weak. Specifically, we address primarily the following questions:

(1) How powerful are polynomial-time Turing reductions to $C_=P$?
(2) How powerful is a constant number of queries to $C_=P$?

In attempting to answer these questions, we find that in many respects the classes NP and *co*-$C_=P$ (or alternatively $C_{\neq}P$) are similar. In particular, NP has a complete problem, and is closed under union, intersection, and nondeterministic many-one reductions. The class $C_{\neq}P$ has all of these properties. It turns out that many proofs about restricted Turing reductions to NP, the Boolean hierarchy over NP, and related notions, depend only on these properties. Thus many interesting results about NP can be easily translated into analogous results about $C_{\neq}P$. For example, we find that the closure of $C_=P$ under polynomial time truth-table reductions is exactly as powerful as the closure under polynomial-time Turing reductions with logarithmically many queries (i.e., $P_{tt}^{C_=P} = P_{O(log(n))-T}^{C_=P}$; see theorem 6 and the remarks preceeding it). However, our main interest is in exploiting these results in order to answer the above questions.

Some insight into question 2 is gained immediately by examining the analogies between NP and $C_=P$. Using these analogies we find here that the query and Boolean hierarchies are just as closely intertwined for $C_=P$ as they are for NP [2]. We also find that the proof of Kadin [8] and Chang and Kadin [6] which shows that if the Boolean hierarchy over NP collapses then the polynomial hierarchy collapses, also works for the Boolean hierarchy over $C_=P$. Thus if the Boolean hierarchy over $C_=P$ collapses to some finite level, we find that the polynomial hierarchy *relative to PP* collapses to the Δ_2 level *relative to PP*. This links a collapse of the query hierarchy over $C_=P$ to a collapse of the polynomial hierarchy relative to PP. Note that this is in sharp contrast to the query hierarchy over PP, which collapses to PP [3]. These results are discussed in section 3.

Nothing is known about question 1, not even oracle separations, and up until now there has been no reason (other than intuition) to believe that $P^{C_=P}$ is any less powerful than P^{PP}. The intuition is quite strong, however. With queries to PP, there is a well-known binary search algorithm which enables us to count the number of accepting paths in nondeterministic computations. A PP oracle provides us pairs of the form $\langle x, y\rangle$ such that $f(x) \geq y$ and $f \in \#P$, from which it is possible to determine the value of $f(x)$ by binary search on y. On the other hand, a $C_=P$ oracle only provides information about the graph of a #P function, that is, pairs of the form $\langle x, y\rangle$ such that $f(x) = y$. It seems obvious that it would be impossible to determine the value of f from this information (in polynomial time) in the absence of nondeterminism. Thus for example, although it is well known that $\oplus P \subseteq P^{PP}$, it is not at all clear that $\oplus P \subseteq P^{C_=P}$. Indeed, it does not seem likely that any of the classes $\oplus P$, PP, or PH are in $P^{C_=P}$.

In this paper we obtain oracle separations such that all of the classes that are not obviously in $P^{C_=P}$ are in fact not contained in it. That is, we construct an oracle A relative to which $\oplus P$ and BPP are not contained in $P^{C_=P}$ (see section 5). A direct consequence of the separation

of BPP from $P^{C_=P}$ are separations of both PP and $\Sigma_2^p \cap \Pi_2^p$ from $P^{C_=P}$. The constructions are based on circuit lower bounds, building on a result of Gundermann, Nasser and Wechsung [7], as well as a new characterization of $P^{C_=P}$ (easily proved using the analogies between $C_{\neq}P$ and NP). The circuit lower bounds are for depth-2 circuits consisting of a single "equals" gate over AND-gates (called "EQ circuits"), and are actually lower bounds on the fanin of the AND-gates rather than lower bounds on the size of the circuits. In the context of circuits, the technique of [7] allows us to kill off EQ circuits by showing that they are always on if the bottom fanin is too small (see section 4).

We finally turn to oracle results relating to question 2. Gundermann, Nasser and Wechsung [7] obtained an oracle separation of the Boolean hierarchy over $C_=P$. Since the Boolean hierarchy is intertwined with the query hierarchy, in some relativized world, $k+1$ queries to $C_=P$ are more powerful than k. Here we consider the question of whether $k+1$ questions to *NP* cannot be answered by k questions to $C_=P$. In fact we find that in some relativized world, there are sets that are recognizable with $k+1$ queries to NP that cannot be recognized with k queries to $C_=P$. One may regard this as a generalization of the result of Torán [14] that relative to some oracle, NP is not contained in $C_=P$. In order to do this we again adapt the technique of [7] (section 4) to appropriate circuit problems. In section 6 an oracle is constructed which separates every level of the Boolean hierarchy over NP from a level of the Boolean hierarchy over $C_=P$. Clearly this simultaneously separates the Boolean hierarchies over NP and $C_=P$. The resulting construction for the Boolean hierarchy over NP is simpler than existing ones [5] although it apparently is not powerful enough to obtain random oracle results (see [4]).

2 Preliminaries

We assume the reader is familiar with complexity classes such as P, NP, Σ_k^p and PH (see, e.g., [1]). Let N be a polynomial time-bounded nondeterministic Turing machine. Then $\#acc_N(x)$ denotes the number of accepting paths of N on input x. #P is the class of functions f such that there exists a polynomial time-bounded nondeterministic machine N such that for all x, $f(x) = \#acc_N(x)$.

The class $C_=P$ [15] is defined to be the set of languages L such that there exist functions $f \in \#P$, $t \in FP$ and for all x, $x \in L$ if and only if $f(x) = t(x)$. The notation co-$C_=P$ denotes the class of sets whose complements are in $C_=P$. We will denote co-$C_=P$ alternatively by $C_{\neq}P$. PP is similarly defined, but with "$f(x) = t(x)$" replaced by "$f(x) \geq t(x)$".

It was recently remarked in [7] that in the definitions above one can replace the function $t \in FP$ by a function $g \in \#P$, and still obtain the same classes. We can furthermore assume without loss of generality that, in the resulting alternative definition of $C_=P$, $f(x) \geq g(x)$ for all x.

$\oplus P$ is defined as the set of languages L such that there exists a function $f \in \#P$ and for all x, $x \in L$ if and only if $f(x)$ is odd.

We denote by ESAT the standard complete language [15] for $C_=P$: ESAT $= \{\langle \mathcal{F}, n\rangle | \mathcal{F}$ is a Boolean formula with exactly n satisfying truth assignments $\}$. Obviously $\overline{\text{ESAT}}$ is complete for $C_{\neq}P$.

We define a family of circuits closely related to $C_=P$. An *EQ circuit of size m, order s*

and threshold t over the inputs $X = \{x_1, ...x_n\}$ consists of a set of AND gates, $c_i, 1 \leq i \leq m$, where each AND gate has fanin at most s and the circuit outputs 1 if and only if $\sigma(x) = t$, where by definition

$$\sigma(x) = \sum_{i=1}^{m} c_i(X).$$

An *NEQ circuit of size m, order s and threshold t* is defined similarly, except that it outputs 1 if and only if $\sigma(x) \neq t$. (Note that the AND-gates in this definition can have negated variables.)

Let $X = \{x_1, ...x_n\}$ be a set of inputs and let $c_i, 1 \leq i \leq m$ be a set of products of the form $\prod_{i \in S} x_i$ for some $S \subseteq \{1, ...n\}$. Suppose we always have $||S|| \leq s$. Associate an integer "weight" $w_i \in Z$ with each c_i. A circuit is called a *normalized EQ circuit of size m and order s* if it outputs 1 if and only if $\sigma(x) = 0$, where by definition

$$\sigma(x) = \sum_{i=1}^{m} w_i c_i$$

and we always have $\sigma(x) \geq 0$. A *normalized NEQ circuit of size m and order s* is defined similarly, except that it outputs 1 if and only if $\sigma(x) \neq 0$.

Using exactly the same techniques as in [7], it is easy to show that every EQ (resp. NEQ) circuit can be converted into an equivalent normalized EQ (resp. NEQ) circuit.

Proposition 1 *Let C be an EQ (NEQ) circuit of size m and order s. Then there exists an equivalent normalized EQ (NEQ) circuit of size $O((2^s m)^2)$ and order $O(2s)$.*

Proof: We have that $C = 1$ if and only if

$$\sum_{i=1}^{m} c_i(X) = t$$

where $X = \{x_1, ...x_n\}$. Then $C = 1$ if and only if

$$(\sum_{i=1}^{m} c_i(X) - t)^2 = 0.$$

Note that the left hand side is always ≥ 0. Rewrite each negated variable $\overline{x}_i$ as $1 - x_i$. Expressing each AND-gate c_i as a product, expand to obtain a sum of factors of the form $\prod_{i \in S} x_i$ (note that $||S|| \leq s$). This yields $O(2^s)$ terms for each of the m AND-gates. Then perform the square in the above equation. We obtain a sum of factors $c'_j(X)$ of the form $\prod_{i \in S'} x_i$, but now with coefficients that may not be 0 or 1 (and in fact can be negative). Now note that $||S'|| \leq 2s$. Thus the sum is of the form $\sum_{j=1}^{m'} w_j c'_j(X)$, where $m' = O((2^s m)^2)$, the sum is always ≥ 0, and it equals 0 if and only if $C = 1$. Furthermore, the order is $\leq 2s$.

A similar argument holds for NEQ. ∎

We now present other definitions that will be important in later sections. Let A and B be sets. We say $A \leq_m^{\mathrm{NP}} B$ if and only if there exist a polynomial p and a function $f \in \mathrm{FP}$ such that for any x, $x \in A$ if and only if $(\exists z, 1 \leq |z| \leq p(|x|))(f(\langle x, z \rangle) \in B)$. For any complexity class C, the notation $\exists C$ denotes the class of all sets L such that there exists a

polynomial p and a set $B \in \mathcal{C}$ such that for all x, $x \in L \Leftrightarrow (\exists z, |z| \leq p(|x|))(\langle x, z\rangle \in B)$. For any complexity class $\mathcal{C}$, $\mathrm{P}^{\mathcal{C}}_{k-\mathrm{T}}$ denotes the class of sets polynomial time Turing reducible to $\mathcal{C}$ with no more than k queries, $\mathrm{P}^{\mathcal{C}}_{tt}$ the class of sets polynomial-time truth-table reducible to $\mathcal{C}$, and $\mathrm{P}^{\mathcal{C}}_{O(f(n))-\mathrm{T}}$ the class of sets Turing reducible to $\mathcal{C}$ with $O(f(n))$ queries. The *query hierarchy* over $\mathcal{C}$ is defined as $\bigcup_{k=1}^{\infty} \mathrm{P}^{\mathcal{C}}_{k-\mathrm{T}}$.

Let $\mathcal{C}$ be a complexity class. Following [5], we define the *Boolean hierarchy over* $\mathcal{C}$ inductively as follows: Let $\mathrm{BH}_1(\mathcal{C}) = \mathcal{C}$, and, for all $k > 1$,

$$\mathrm{BH}_{2k}(\mathcal{C}) = \{L | L = L_1 \cap \overline{L}_2, L_1 \in \mathrm{BH}_{2k-1}(\mathcal{C}), L_2 \in \mathcal{C}\},$$

$$\mathrm{BH}_{2k+1}(\mathcal{C}) = \{L | L = L_1 \cup L_2, L_1 \in \mathrm{BH}_{2k}(\mathcal{C}), L_2 \in \mathcal{C}\}.$$

Finally, $\mathrm{BH}(\mathcal{C}) = \bigcup_{k=1}^{\infty}(\mathrm{BH}_k(\mathcal{C}))$.

The Boolean hierarchy can be defined in many different ways. One alternative definition that we will make use of is the following normal form [5]:

Proposition 2 *For any $k > 1$,*

$$\mathrm{BH}_{2k}(\mathcal{C}) = \bigcup_{i=1}^{k} L_i$$

where for each i, $L_i = L_{i1} \cap \overline{L}_{i2}$, $L_{i1}, L_{i2} \in \mathcal{C}$, and

$$\mathrm{BH}_{2k+1}(\mathcal{C}) = (\bigcup_{i=1}^{k} L_i) \cup L_{k+1}$$

where the L_i, $1 \leq i \leq k$, are as above, and $L_{k+1} \in \mathcal{C}$.

We also make use of another definition of $\mathrm{B\bar{H}}$, which has not appeared previously, but which is easily proved to be equivalent to those mentioned above in the case that $\mathcal{C}$ is closed under union (which will be the case for the classes considered in this paper).

Proposition 3 *Suppose $\mathcal{C}$ is closed under union. Then for all $k > 1$,*

$$\mathrm{BH}_k(\mathcal{C}) = \{L | L = L_1 \cap L_2, \overline{L}_1 \in \mathrm{BH}_{k-1}(\mathcal{C}), L_2 \in \mathcal{C}\}.$$

Proof: The proof is by induction. The base case is obvious from the definition of $BH_2(\mathcal{C})$.

Assume by hypothesis that for some $k > 2$

$$\mathrm{BH}_{2k}(\mathcal{C}) = \{L | L = L_1 \cap L_2, \overline{L}_1 \in \mathrm{BH}_{2k-1}(\mathcal{C}), L_2 \in \mathcal{C}\}.$$

By definition, for any $L \in \mathrm{BH}_{2k+1}(\mathcal{C})$, we can write $L = L_1 \cup L_2$, $L_1 \in \mathrm{BH}_{2k}(\mathcal{C}), L_2 \in \mathcal{C}$. Using the inductive hypothesis, we can write $L_1 = L_3 \cap L_4$, where $\overline{L}_3 \in \mathrm{BH}_{2k-1}(\mathcal{C}), L_4 \in \mathcal{C}$. Thus $L = (L_3 \cap L_4) \cup L_2 = (L_3 \cup L_2) \cap (L_4 \cup L_2)$ where we have used the distributive law. Now $L_4 \cup L_2 = L'_4$, where $L'_4 \in \mathcal{C}$ since $\mathcal{C}$ is closed under union. Also $L_3 \cup L_2 = L'_3$ where $L'_3 \in \text{co-}\mathrm{BH}_{2k}(\mathcal{C})$ by definition. Thus for any $L \in \mathrm{BH}_{2k+1}(\mathcal{C})$ we can write $L = L'_3 \cap L'_4$, where $\overline{L}'_3 \in \mathrm{BH}_{2k}(\mathcal{C})$ and $L'_4 \in \mathcal{C}$, which proves the inductive step for $2k$.

The proof for $2k + 1$ is similar. ∎

3 Analogies Between NP and $C_{\neq}P$

It has been known for some time that $C_{\neq}P$ is closed under union ([15]). The closure under intersection was proved in [7]. That $C_{\neq}P$ is closed under $\leq_m^{NP}$-reductions is a simple observation, and is well-known [1], although it has not been explicitly stated in the literature. For completeness, we give a proof here.

Proposition 4 *$C_{\neq}P$ is closed under $\leq_m^{NP}$-reductions.*

Proof: Let $L \in C_{\neq}P$ and suppose $L' \leq_m^{NP} L$. We will show that $L' \in C_{\neq}P$. We know that there exist functions $f, g \in \#P$ that for any w, $f(w) \geq g(w)$ and $w \in L$ if and only if $f(w) \neq g(w)$. Since $L' \leq_m^{NP} L$, there exists a polynomial p and a function $h \in FP$ such that $x \in L' \Leftrightarrow (\exists z, |z| \leq p(|x|))(f(h(\langle x, z\rangle)) \neq g(h(\langle x, z\rangle))$. Since for any w, $f(w) \geq g(w)$, the predicate $(\exists z, |z| \leq p(|x|))(f(h(\langle x, z\rangle)) \neq g(h(\langle x, z\rangle))$ is true if and only if $\sum_z f(h(\langle x, z\rangle)) \neq \sum_z g(h(\langle x, z\rangle))$. But then using standard techniques we can implement these sums in terms of #P machines, i.e., there exist nondeterministic, polynomial time bounded machines N_1' and N_2' such that for all x, $x \in L' \Leftrightarrow \#acc_{N_1'}(x) \neq \#acc_{N_2'}(x)$. Hence $L' \in C_{\neq}P$. ∎

For the most part, we make use of this fact in the following form, which says that $C_{\neq}P$ is closed under existential quantification.

Corollary 5 $\exists C_{\neq}P = C_{\neq}P$.

Now many interesting results can be derived using analogous proofs for NP, with $C_{\neq}P$ playing the role of NP. In some cases we will omit the proofs, but to illustrate the correspondence we will include the more compact ones. For the first result, recall that truth-table reductions to NP are exactly as powerful as Turing reductions with logarithmically many queries, i.e., $P_{O(log(n))-T}^{NP} = P_{tt}^{NP}$ (see, e.g., [8]). Here we find the same is true of $C_{=}P$. This result has also been found by Toda. The result and the proof reported here were obtained independently [2]. The proof we give is typical of those that use the closure properties shared by NP and $C_{\neq}P$.

Theorem 6 $P_{O(log(n))-T}^{C_{=}P} = P_{tt}^{C_{=}P}$.

Proof: The inclusion from left to right is straightforward: the query tree of a $P_{O(log(n))-T}^{C_{=}P}$ machine can be written down in polynomial time, and a polynomial size truth table constructed from the query tree.

The inclusion from right to left follows from an argument similar to the one used to prove $P_{tt}^{NP} \subseteq P_{O(log(n))-T}^{NP}$, aided by corollary 5. Let M^A be a $P_{tt}^{C_{=}P}$ machine, where (without loss of generality) $A \in C_{\neq}P$, and M^A is bounded by the polynomial p. On input x, $|x| = n$,

[1] S. Toda, private communication. Ogiwara, Toda, and many others have made the same observation.

[2] Subsequently Ogiwara and Toda [10] have substantially generalized this result, exhibiting sufficient conditions for a complexity class C to obey $P_{tt}^{C} = P_{O(log(n))-T}^{C}$. They have also proved the very pretty result that $C_{=}P$ is closed under positive Turing reductions (and indeed exhibit sufficient conditions for any class to have this property).

suppose M produces the queries $S = \{q_1, q_2, ...q_{p(n)}\}$. We will show how to simulate the rest of the computation of M^A using logarithmically many queries to $C_{\neq}P$. First, we show that with logarithmically many queries to $C_{\neq}P$, we can determine the number l of q_i's such that $q_i \in \overline{ESAT}$. Consider a nondeterministic machine N that, on input $\langle S, k \rangle$, guesses k elements of S and then accepts iff all the guessed strings are elements of $\overline{ESAT}$. Using the closure of $C_{\neq}P$ under intersection, it is easy to see that N is an $\exists C_{\neq}P$ machine. By corollary 5 we can simulate N by some $C_{\neq}P$ machine N'. Note that by binary search on k, since $0 \leq k \leq p(n)$, we can determine l with log many queries. We now know both l, the number of positive answers to the queries in S, as well as $p(n) - l$, the number of negative answers. With one extra query to $C_{\neq}P$ we can simulate M^A. We construct a nondeterministic machine N'' which guesses l elements of S, and rejects if any one of them is in ESAT. If all the guessed elements are in $\overline{ESAT}$, then we know the set of queries with positive answers and with negative answers. Using this information, simulate M directly, accepting if and only if M accepts. It is easy to see that N'' is an $\exists C_{\neq}P$ machine, and therefore, again using corollary 5, a $C_{\neq}P$ machine. Hence with $O(log(n))$ queries to $C_{\neq}P$ we can simulate M^A. ∎

It is well known that NP = co-NP if and only if PH = NP. A similar phenomenon occurs if $C_{=}P$ is closed under complement.

Corollary 7 $C_{=}P = C_{\neq}P$ *if and only if* $PH^{PP} = C_{=}P$.

Proof: The "if" part is clear. Then suppose $C_{=}P = C_{\neq}P$. Torán [14] has proved that $NP^{PP} \subseteq NP^{C_{=}P} \subseteq \exists C_{=}P$. Hence if $C_{=}P = C_{\neq}P$, $NP^{PP} \subseteq \exists C_{=}P = \exists C_{\neq}P \subseteq C_{\neq}P = C_{=}P$. ∎

We conclude this section with some remarks on the Boolean and query hierarchies. Not surprisingly, many of the properties of BH(NP) are shared by $BH(C_{=}P)$. For example, the levels of $BH(C_{=}P)$ have complete problems analogous to those for the levels of BH(NP). Another important example for this paper is that, just as in the case of NP, there is a tight intertwining relationship between the Boolean and query hierarchies over $C_{=}P$. The proof of this fact follows Beigel's proof for NP [2]. In fact it was pointed out in [2] that the proof for this theorem only depends on the existence of a complete problem, and closure under intersection, union and $\leq_m^{NP}$-reducibility.

Theorem 8 *For all* $k \geq 1$, $BH_{2k-1}(C_{=}P) \cup \text{co-}BH_{2k-1}(C_{=}P) \subseteq P_{k-T}^{C_{=}P} \subseteq BH_{2k}(C_{=}P) \cap \text{co-}BH_{2k}(C_{=}P)$.

It is natural to ask if the Boolean hierarchy over $C_{=}P$ collapses, or, equivalently, if the query hierarchy over $C_{=}P$ collapses to some finite level. (Corollary 7 represents a first step in this direction.) As mentioned in the introduction, up to now the only known separation is a relativized one (see [7] and section 5). In contrast for NP, structural relationships between the Boolean hierarchy over NP and the polynomial hierarchy are known. It was proved by Kadin that if the Boolean hierarchy collapses to a finite level then PH collapses to $P^{NP^{NP}}$ [8],[6]. Stated more precisely, if for any k, $BH_k(NP) = \text{co-}BH_k(NP)$ then $PH \subseteq \Delta_2^{NP}$ (in fact, Chang and Kadin show the much sharper consequence $PH \subseteq BH_k(NP^{NP})$). We observe here that Chang and Kadin's proof of this fact carries over directly to $C_{\neq}P$. We simply exploit the analogy between $C_{\neq}P$ and NP in their proof. Corollary 5 allows us to do "oracle replacement" in nondeterministic computations when it is possible to find "small $C_{\neq}P$ machines" for $C_{=}P$. The "hard string/easy string" argument similarly holds with $\overline{ESAT}$ playing the role of SAT.

Theorem 9 *If for any k, $\mathrm{BH}_k(\mathrm{C}_=\mathrm{P}) \subseteq \text{co-}\mathrm{BH}_k(\mathrm{C}_=\mathrm{P})$ then $\mathrm{PH}^{\mathrm{PP}} \subseteq \mathrm{BH}_k(\mathrm{NP}^{\mathrm{PP}}) \subseteq \Delta_2^{\mathrm{PP}}$.*

Thus making use of theorem 8 we have,

Corollary 10 *If for any k, $\mathrm{P}^{\mathrm{C}_=\mathrm{P}}_{(k+1)-\mathrm{T}} \subseteq \mathrm{P}^{\mathrm{C}_=\mathrm{P}}_{k-\mathrm{T}}$, then $\mathrm{PH}^{\mathrm{PP}} \subseteq \mathrm{P}^{\mathrm{NP}^{\mathrm{PP}}}$.*

4 A Technical Lemma

We frequently use the following lemma to establish the relativized separations. It is the same as lemma 30 in [7], translated into the context of circuits. In addition, we generalize the lemma so that it can be applied to Boolean functions which are symmetric with respect to certain (disjoint) subsets of the input variables, a generalization of the usual definition of symmetric function. More precisely, in the following, for any subset $S \subseteq \{x_1, ...x_n\}$ we define, for any truth assignment to the x's, $y(S) = ||\{j|(x_j \in S) \wedge (x_j = 1)\}||$. For some Boolean function f over the variables $\{x_1, ...x_n\}$, suppose there exist disjoint subsets $S_1, S_2, ..., S_k \subseteq \{x_1, ...x_n\}$ such that if we put $y_j = y(S_j)$ for each $j, 1 \leq j \leq k$, then f can be expressed as a function g of the numbers y_j, that is, there is a function g such that $(\forall z_i, 1 \leq i \leq k)(\forall x$, s.t. $(\forall j, 1 \leq j \leq k)(z_j = y_j))(f(x_1, ...x_n) = g(y_1, ...y_k))$. We then say that f is *symmetric* in the subsets $S_1, ...S_n$ via the function $g(y_1, ...y_k)$.

Lemma 11 *Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be a Boolean function over the variables $\{x_1, ...x_n\}$, and suppose f is symmetric in the subsets $S_1, ...S_k \subseteq \{x_1, ...x_n\}$ via the function $g(y_1, ...y_k)$. Let s be a function $s : \mathbb{N} \rightarrow \mathbb{N}$. Let C be an EQ circuit of order $o(s(n))$ such that $f(x_1...x_n) = 1 \Rightarrow C = 1$. Furthermore, suppose that $g(y_1, ...y_k) = 1$ for $\Omega(s(n))$ values of $y_1, y_2, ...y_k$, respectively. Then for all input settings, $C = 1$.*

Proof: By proposition 1, we can assume that C is a normalized EQ circuit and still maintain an order of $o(s(n))$. Let C consist of terms c_i, $i = 1, ...m$, each of degree bounded by $o(s(n))$. Thus $C = 1$ if and only if $\sum_{l=1}^{m} w_l c_i(x) = 0$. Since C is normalized, the quantity $\sigma(x) = \sum_{l=1}^{m} w_l c_i(x) \geq 0$ for any setting of the inputs. We know that for any choice of y_j's such that $g(y_1, ...y_k) = 1$, that $\sigma(x) = 0$. Let us compute the sum of $\sigma(x)$ over input settings x for any fixed $y_j, 1 \leq j \leq k$. Let the notation "$x : y$" denote that the assignments x can vary given fixed values for the y_j's. One can reverse the order of the sums: $\sum_{x:y} \sigma(x) = \sum_{l=1}^{m} w_l \sum_{x:y} c_l(x)$. Let $S_{lj} = \{i|x_i \in S_j$ and x_i is an input to $c_l\}$ and $s_{lj} = ||S_{lj}||$. Thus for each l, $c_l(x) = \prod_{j=1}^{k}(\prod_{i \in S_{lj}} x_i)$. Letting $n_j = ||S_j||$ where $1 \leq j \leq k$, it is not hard to show that

$$\sum_{x:y} c_l(x) = \prod_{j=1}^{k} \binom{n_j - s_{jl}}{y_j - s_{jl}}.$$

Observe that, by hypothesis, $\sum_{j=1}^{k} s_{jl} = o(s(n))$, so that the quantity

$$\prod_{j=1}^{k} y_j!(n_j - y_j)! \prod_{j=1}^{k} \binom{n_j - s_{jl}}{y_j - s_{jl}} = \prod_{j=1}^{k} [(n_j - s_{jl})! \prod_{i=0}^{s_{jl}-1} (y_j - i)]$$

is a polynomial in each individual y_j of degree $o(s(n))$. Therefore the quantity

$$\prod_{j=1}^{k} y_j!(n_j - y_j)! \sum_{x:y} \sigma(x)$$

is a polynomial $p(y_1, ...y_k)$ of degree $o(s(n))$ in each y_j. Since for any x such that $g(y_1, ...y_k) = 1$ we also have $\sigma(x) = 0$, it follows from the definition of $p(y_1, ...y_k)$ that $p(y_1, ...y_k) = 0$ for any such x. Conversely, note that since $\sigma(x) \geq 0$ for any x, if for any choice of $y_1, ...y_k$ we have $p(y_1, ...y_k) = 0$, then for all x with this choice of $y_1, ...y_k$, it follows that $\sigma(x) = 0$ and therefore $C = 1$.

We will now show that for all possible values of $y_1, ...y_k$, $p(y_1, ...y_k) = 0$. This will prove that for all input settings, $C = 1$.

Let $Y_1, Y_2, ...Y_k$ respectively denote the sets of values of $y_1, ...y_k$ such that $g(y_1, ...y_k) = 1$. By hypothesis, for all $i, 1 \leq i \leq k$, $||Y_i|| = \Omega(s(n))$. We know that

$$(\forall y_1 \in Y_1)(\forall y_2 \in Y_2)...(\forall y_k \in Y_k)(p(y_1, ...y_k) = 0).$$

Fix any $y_1, y_2, ...y_{k-1}$, where $y_1 \in Y_1, y_2 \in Y_2, ...$ and $y_{k-1} \in Y_{k-1}$. Then $p(y_1, ...y_{k-1}, y_k)$ is a polynomial in y_k of degree $o(s(n))$. However, $g(y_1, ...y_{k-1}, y_k) = 1$ for $\Omega(s(n))$ values of y_k, and hence $p(y_1, ...y_{k-1}, y_k) = 0$ for this many values of y_k. Therefore $p(y_1, ...y_{k-1}, y_k) = 0$ for all values of y_k, $0 \leq y_k \leq ||S_k||$. Proceeding inductively in this fashion, we find that

$$(\forall y_1, 0 \leq y_1 \leq ||S_1||)...(\forall y_k, 0 \leq y_k \leq ||S_k||)(p(y_1, ...y_k) = 0).$$

This proves the lemma. ∎

Note that we can replace "EQ" with "NEQ" and "$C = 1$" with "$C = 0$" in the above lemma and that it remains true.

Lemma 11 will be used to eliminate EQ (respectively, NEQ) circuits by showing that they have constant values.

5 Separations of $P^{C_=P}$ from P^{PP}

As explained in the introduction, intuitively PP appears to be more powerful than $C_=P$ in deterministic reductions, even though it is no more powerful than $C_=P$ in nondeterministic reductions. We show here that any proofs that Turing reductions to $C_=P$ are as powerful as Turing reductions to PP would, at least, not relativize. These separations leave open the possibility that, while P^{PP} machines can count (via the binary search technique mentioned in the introduction), $P^{C_=P}$ perhaps cannot.

First we need an alternative characterization of $P^{C_=P}$. Following Wagner [16], we define, for any complexity class $\mathcal{C}$ and for any bounding function b, the class $\mathcal{C}(b)$ as follows: $A \in \mathcal{C}(b)$ if and only if there exists a $B \in \mathcal{C}$ such that for any x, for all z where $1 \leq z \leq b(|x|)$, $\chi_B(\langle x, z\rangle) \leq \chi_B(\langle x, z-1\rangle)$, and $x \in A$ if and only if $max\{z | 0 \leq z \leq b(|x|) - 1$ and $\langle x, z\rangle \in B\}$ is odd. $\mathcal{C}(2^{poly})$ is defined as $\bigcup_{c \in N} \mathcal{C}(2^{n^c})$. It was shown in [16] that $NP(2^{poly}) = P^{NP}$. We can prove the analogous result here.

Theorem 12 $C_{\neq}P(2^{poly}) = P^{C_=P}$.

Proof: ($\subseteq$): Let $A \in \mathrm{C}_{\neq}\mathrm{P}(2^{poly})$ and let B be as in the definition above (with $C = \mathrm{C}_{\neq}\mathrm{P}$), and b the bounding function (of the form $2^{p(n)}$). Define the set $S = \{\langle x, y\rangle | 0 \leq y \leq b(|x|), (\exists z)(z > y, 0 \leq z \leq b(|x|) \wedge \langle x, z\rangle \in B\}$. By binary search, we can find the maximum y such that $\langle x, y\rangle \in S$ in time $p(|x|)$, using S as an oracle. We can easily tell if the maximum y is odd. Finally, note that $S \in \exists \mathrm{C}_{\neq}\mathrm{P} = \mathrm{C}_{\neq}\mathrm{P}$. Then $A \in \mathrm{P}^{\mathrm{C}_{=}\mathrm{P}}$.

($\supseteq$): Let $L \in \mathrm{P}^{\mathrm{C}_{=}\mathrm{P}}$ via machine M which makes queries to a set $A \in \mathrm{C}_{\neq}\mathrm{P}$. Define the set B' as follows. $B' = \{\langle x, z\rangle | z = \langle r_1, ...r_{p(|x|)}, a\rangle, a \in \{0,1\}, M$, using query answers $r_1, ...r_{p(|x|)}$, produces queries $q_1, ...q_{p(|x|)}$, and (M accepts $x \Leftrightarrow a = 1) \wedge (\forall i)(r_i = 1 \Rightarrow q_i \in A)\}$. It is clear that $B' \in \mathrm{C}_{\neq}\mathrm{P}$. Furthermore, M accepts x if and only if the maximum z such that $\langle x, z\rangle \in B'$ is odd. Thus

$$\begin{aligned} x \in L &\Leftrightarrow max\{z | \langle x, z\rangle \in B'\} \neq 0 \bmod 2 \\ &\Leftrightarrow max\{z | (\exists w \geq z)\langle x, w\rangle \in B'\} \neq 0 \bmod 2 \\ &\Leftrightarrow max\{z | \langle x, z\rangle \in B\} \neq 0 \bmod 2 \end{aligned}$$

where $B = \{\langle x, z\rangle | (\exists w \geq z)(\langle x, w\rangle \in B')\}$. Now since $B' \in \mathrm{C}_{\neq}\mathrm{P}$, $B \in \exists \mathrm{C}_{\neq}\mathrm{P} = \mathrm{C}_{\neq}\mathrm{P}$. Furthermore, $\chi_B(\langle x, z\rangle) \leq \chi_B(\langle x, z-1\rangle)$. This proves the theorem. ∎

Observe that the statement $max\{z | \langle x, z\rangle \in B\} \neq 0 \bmod 2$ is equivalent to the statement $\sum_{z=0}^{b(|x|)} \chi_B(\langle x, z\rangle) \neq 0 \bmod 2$, because of the monotonicity of the χ_B's in z. Thus the separation results can be understood in terms of a simple and highly constrained circuit model.

We say a circuit has *polylog order* if there exists a polynomial q such that for n inputs, the order of the circuit is $q(log(n))$.

Definition 13 Let c_i, $1 \leq i \leq m$, be a set of NEQ circuits, over the inputs $\{x_1, ...x_n\}$, each of polylog order in n, with the property that $c_i = 1 \Rightarrow c_{i-1} = 1$. A $\mathrm{P}^{\mathrm{C}_{=}\mathrm{P}}$*-circuit of size* m is a circuit which outputs 1 if and only if $||\{i | 1 \leq i \leq m, c_i = 1\}||$ is odd.

Proposition 14 *Let M be a $\mathrm{P}^{\mathrm{C}_{=}\mathrm{P}^A}$-machine. Then for any input x, there exists a $\mathrm{P}^{\mathrm{C}_{=}\mathrm{P}}$-circuit C of size $O(2^{n^{O(1)}})$, $n = |x|$, whose inputs are the characteristic functions of A, such that $C = 1$ if and only if M accepts x.*

Proof: By theorem 12 and the remarks following it, we can assume there is a set $B \in \mathrm{C}_{\neq}\mathrm{P}^A$ and a polynomial p such that for any x, M accepts x if and only if

$$\sum_{z=0}^{2^{p(|x|)}} \chi_B(\langle x, z\rangle) \neq 0 \bmod 2.$$

Let N be a $\mathrm{C}_{\neq}\mathrm{P}^A$ machine recognizing B. Because N correctly recognizes B, it is clear that if N accepts on input $\langle x, z\rangle$ then N also accepts on input $\langle x, z-1\rangle$. In a standard fashion, in any computation of N we can guess answers to queries to A and verify that they are correct at the ends of the computation paths. From this we can obtain an NEQ circuit of order polynomial in $|x|$ whose inputs are the characteristic functions of A and which, for any input to N of the form $\langle x, z\rangle$, outputs 1 if and only if N accepts. Note that the order of this circuit

is polylog since the number of inputs ($\chi_A(w)$) is exponential in $|x|$. Assuming the input x to M to be fixed, call this circuit c_z. Evidently M accepts x if and only if

$$\sum_{z=0}^{2^{p(|x|)}} c_z \neq 0 \bmod 2$$

and furthermore $c_z = 1 \Rightarrow c_{z-1} = 1$. Since $1 \leq z \leq 2^{p(|x|)}$, the circuit has size $O(2^{n^{O(1)}})$ where $n = |x|$. ∎

In the proof of the following we make use of lemma 11, restricted to the case of $k = 1$, that is, for ordinary symmetric functions.

Lemma 15 *Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be a symmetric Boolean function such that if we put $y = ||\{i|x_i = 1\}||$, $f(x_1, ...x_n) = 1$ for $\Omega(n)$ values of y and $f(x_1, ...x_n) = 0$ for $\Omega(n)$ values of y. Then no $\mathrm{P}^{\mathrm{C}=\mathrm{P}}$-circuit can compute f.*

Proof: Let $C(x_1, ...x_n)$ be a $\mathrm{P}^{\mathrm{C}=\mathrm{P}}$-circuit with NEQ subcircuits $c_i, 1 \leq i \leq m$, which computes $f(x_1, ...x_n)$. We will show that for all input settings, $C = 0$. Suppose the number of NEQ subcircuits m is odd (the proof when m is even is the same, as will be obvious). Then whenever $f(x) = 0$, we must have $c_m = 0$, since if $c_m = 1$ all the c_i's with $i < m$ will also have to be 1, and hence we would have $C = 1$. By hypothesis, for $\Omega(n)$ values of y, $f(x) = 0$. Since c_m is of order $o(n)$ but is zero for $\Omega(n)$ values of y, we can conclude from lemma 11 that $c_m = 0$ for all input settings. This eliminates c_m from consideration. Consider the remaining $m - 1$ gates. Since $m - 1$ is even, if $f(x) = 1$ we must have $c_{m-1} = 0$, otherwise an even number of NEQ circuits would yield 1. c_{m-1} can be eliminated just as c_m, since $f(x) = 1$ for $\Omega(n)$ values of y. Proceeding inductively, we find that all the c_i's equal 0 for all input settings, and therefore $C = 0$ for all input settings. ∎

The parity function $\oplus(x_1, ...x_n)$ is defined to be 1 if and only if $||\{i|x_i = 1\}||$ is odd.

Lemma 16 *No $\mathrm{P}^{\mathrm{C}=\mathrm{P}}$-circuit can compute the parity function.*

Proof: $\oplus(x_1, ...x_n)$ is a symmetric function which is 1 for $\Omega(n)$ values of $y = ||\{i|x_i = 1\}||$ and is 0 for $\Omega(n)$ values of y. Applying lemma 15, the result is immediate. ∎

Theorem 17 *There exists an oracle A such that $\oplus\mathrm{P}^A \not\subseteq \mathrm{P}^{\mathrm{C}=\mathrm{P}^A}$.*

Using the same technique it is also possible to prove that in some relativized world the polynomial hierarchy is not contained in $\mathrm{P}^{\mathrm{C}=\mathrm{P}}$. In fact we can construct an oracle such that BPP is not contained in $\mathrm{P}^{\mathrm{C}=\mathrm{P}}$, which provides a simultaneous proof that there is an oracle such that neither $\Sigma_2^p \cap \Pi_2^p$ nor PP are contained in $\mathrm{P}^{\mathrm{C}=\mathrm{P}}$. For this purpose we define a function related to BPP. Define the "strict majority" function $T_{3/4}$ as follows:

$$T_{3/4}(x_1, ...x_n) = \begin{cases} 1 & \text{if } \sum_{i=1}^n x_i \geq \frac{3}{4}n, \\ 0 & \text{if } \sum_{i=1}^n x_i \leq \frac{1}{4}n, \\ ? & \text{otherwise.} \end{cases}$$

We say a circuit C *computes* $T_{3/4}$ if $T_{3/4}(x_1, ...x_n) = 1 \Rightarrow C(x_1, ...x_n) = 1$ and $T_{3/4}(x_1, ...x_n) = 0 \Rightarrow C(x_1, ...x_n) = 0$.

Lemma 18 *No* $\mathrm{P}^{\mathrm{C}_=\mathrm{P}}$*-circuit computes the function* $T_{3/4}$.

Proof: $T_{3/4}(x_1, ...x_n)$ is a symmetric function which is 1 for $\Omega(n)$ values of $y = ||\{i|x_i = 1\}||$ and is 0 for $\Omega(n)$ values of y. The result follows from lemma 15. ∎

In the proof of the following, as well as in the following section, s_i^n denotes the i^{th} string of length n.

Theorem 19 *There exists an oracle* A *such that* $\mathrm{BPP}^A \not\subseteq \mathrm{P}^{\mathrm{C}_=\mathrm{P}^A}$.

Proof: Define the test language $L(A) = \{1^n|$ at least 3/4 of the strings of length n are in $A\}$. Clearly, for any A such that for each length n, either 3/4 of the strings of that length are in A or no more than 1/4 are in A, then $L(A) \in \mathrm{BPP}^A$. For such A's, $L(A) = \{1^n|T_{3/4}(\chi_A(s_1^n), \chi_A(s_2^n), ...\chi_A(s_{2^n}^n))\}$. Let $M_i, i \in \mathbb{N}$ be an enumeration of $\mathrm{P}^{\mathrm{C}_=\mathrm{P}}$-machines. Using lemma 18 one can easily show that A can be constructed such that for all i, $1^{n_i} \in L(A) \Leftrightarrow M_i$ rejects 1^{n_i}. ∎

Corollary 20 *There exists an oracle* A *such that* $\Sigma_2^{p,A} \cap \Pi_2^{p,A} \not\subseteq \mathrm{P}^{\mathrm{C}_=\mathrm{P}^A}$ *and* $\mathrm{PP}^A \not\subseteq \mathrm{P}^{\mathrm{C}_=\mathrm{P}^A}$.

Proof: It is well known that $\mathrm{BPP} \subseteq \mathrm{PP} \cap \Sigma_2^p \cap \Pi_2^p$ via a proof that relativizes (e.g., [9]). ∎

6 An Oracle Interlocking the Query Hierarchies Over NP and C$_=$P

In this section we construct an oracle which gives an optimal separation of the Boolean and query hierarchies over NP and C$_=$P. This represents a technical improvement of the oracle of Gundermann, Nasser and Wechsung [7].

It is possible to reduce this separation to circuit lower bounds. We now give this reduction.

Define the function $f_{2k,n}$ in $2kn$ variables $X = \{x_{ij}|i = 1..n, j = 1..2k\}$ as follows:

$$f_{2k,n}(X) = \bigvee_{j=1}^{k} [(\bigvee_{i=1}^{n} x_{i,(2j-1)}) \wedge (\bigwedge_{i=1}^{n} \overline{x}_{i,2j})]$$

Similarly, we define $f_{2k+1,n}(X \cup \{x_{i,2k+1}\})$ as

$$f_{2k+1,n}(X \cup \{x_{i,2k+1}\}) = f_{2k,n}(X) \vee (\bigvee_{i=1}^{n} x_{i,2k+1})$$

Observe that $f_{k,n}$ is symmetric in the subsets $S_j = \{x_{ij}|1 \leq i \leq n\}$, where $1 \leq j \leq k$, via the following function g: Setting $y_j = ||\{i|x_{ij} = 1\}||$, when k is even, $g(y_1, ...y_k) = 1$ if and only if for some odd $j \leq k-1$ we have $y_j > 0$ and $y_{j+1} = 0$. When k is odd, $g(y_1, ...y_k) = 1$ if and only if $y_k > 0$ or for some odd $j \leq k-2$ we have $y_j > 0$ and $y_{j+1} = 0$.

A $\mathrm{BH}_k^{\mathrm{C}_=\mathrm{P}}$ circuit is defined inductively as follows. A $\mathrm{BH}_1^{\mathrm{C}_=\mathrm{P}}$ circuit is a normalized EQ circuit of polylog order. A $\mathrm{BH}_{2k}^{\mathrm{C}_=\mathrm{P}}$ circuit is a circuit of the form $C_1 \vee C_2$ where C_1 is a $\mathrm{BH}_{2k-1}^{\mathrm{C}_=\mathrm{P}}$

circuit and C_2 is a normalized NEQ circuit of polylog order. A $\mathrm{BH}^{\mathrm{C_=P}}_{2k+1}$ circuit is a circuit of the form $C_1 \wedge C_2$ where C_1 is a $\mathrm{BH}^{\mathrm{C_=P}}_{2k}$ circuit and C_2 is a normalized EQ circuit of polylog order.

The main theorem which allows a complete separation of BH(NP) from BH($\mathrm{C_=P}$) is the following. It is here that we use the full power of lemma 11, with arbitrary k.

Theorem 21 *For sufficiently large n and all $k \geq 1$, no $\mathrm{BH}^{\mathrm{C_=P}}_{k}$ circuit can compute $f_{k,n}$.*

Proof: The proof is by induction. For the base case, suppose $f_{1,n}$ is computable by a $\mathrm{BH}^{\mathrm{C_=P}}_{1}$, i.e., an EQ circuit C^0. Define $y_1 = ||\{i|x_{i,1} = 1\}||$. We know that if $y_1 > 0$ then $C^0 = 1$. Since C^0 has polylog order, it has order $o(n)$. But $C^0 = 1$ for $n - 1 = \Omega(n)$ values of y_1, and hence by lemma 11, for all input settings $C^0 = 1$. However then $C^0 = 1$ when $f_{1,n} = 0$, a contradiction.

For the inductive step, we first consider even levels. Suppose by hypothesis that no $\mathrm{BH}^{\mathrm{C_=P}}_{2k-1}$-circuit can compute $f_{2k-1,n}$. Suppose however that there exists a $\mathrm{BH}^{\mathrm{C_=P}}_{2k}$-circuit C to compute $f_{2k,n}$ correctly. C is of the form $C_1 \vee C_2$, where C_1 is a $\mathrm{BH}^{\mathrm{C_=P}}_{2k-1}$-circuit and C_2 is an NEQ circuit of polylog order, and hence of order $o(n)$. Let $y_j = ||\{i|x_{ij} = 1\}||$, as in the discussion preceeding this theorem. Also note from that discussion that whenever $y_j > 0$ for all EVEN j, $f_{2k,n} = 0$. Since for all j, $0 \leq y_j \leq n$, the function $g(y_1, ...y_{2k})$ is zero for $\Omega(n)$ values of $y_1, ...y_{2k}$, respectively. Now whenever $g(y_1, ...y_{2k}) = 0$, $C_2 = 0$. Then by lemma 11, for all input settings $C_2 = 0$. We can thus ignore C_2. By setting y_{2k} to some value greater than zero, we find that C_1 computes $f_{2k-1,n}$ correctly, which contradicts the induction hypothesis.

The argument for odd values $2k + 1$ is similar to the base case. We assume by hypothesis that no $\mathrm{BH}^{\mathrm{C_=P}}_{2k}$-circuit can compute $f_{2k,n}$, but that there exists a $\mathrm{BH}^{\mathrm{C_=P}}_{2k+1}$-circuit of the form $C_1 \wedge C_2$ which computes f_{2k+1} correctly. We define as before $y_j = ||\{i|x_{i,j} = 1\}||$, $1 \leq j \leq 2k + 1$, now noting that whenever $y_{2k+1} > 0$ we have $f_{2k+1,n} = 1$. As before we find that for all input settings $C_2 = 1$, and by setting $y_{2k+1} = 0$ we conclude that C_1 computes $f_{2k,n}$ correctly, which is a contradiction. ∎

The next proposition establishes the relationship between $\mathrm{BH}_k(\mathrm{C_=P}^A)$ and $\mathrm{BH}^{\mathrm{C_=P}}_{k}$-circuits. In the following, $x_{i,j}$ denotes $\chi_A(\langle s^n_i, j\rangle)$.

Proposition 22 *Let $k \geq 1$. For any $\mathrm{BH}_{2k-1}(\mathrm{C_=P}^A)$ machine M, for any input x there exists a $\mathrm{BH}^{\mathrm{C_=P}}_{2k-1}$-circuit C with inputs $x_{i,j}$ as defined above such that $C = 1$ if and only if M accepts x. Similarly, for any co-$\mathrm{BH}_{2k}(\mathrm{C_=P}^A)$ machine M, for any input x there exists a $\mathrm{BH}^{\mathrm{C_=P}}_{2k}$-circuit C with inputs $x_{i,j}$ as defined above such that $C = 1$ if and only if M accepts x.*

Proof: The proof is by induction. For $k = 1$, it is easy to show that for any $\mathrm{C_=P}^A$ machine M and input x, there exists an EQ circuit C such that M accepts x if and only if $C = 1$. The order of the circuit is polynomial in $|x|$. However, since there are exponentially many inputs of the form $x_{i,j}$, the order is polylog in the number of inputs to C.

Suppose the proposition is true for some $k > 1$. Consider any co-$\mathrm{BH}_{2k}(\mathrm{C_=P}^A)$-machine M. By proposition 3, there exist a $\mathrm{BH}_{2k-1}(\mathrm{C_=P}^A)$-machine M_1 and a $\mathrm{C_{\neq}P}$-machine M_2 such that for any input x, M accepts x if and only if either M_1 or M_2 accept x. By the induction

hypothesis, we can replace M_1 with a $\mathrm{BH}_{2k-1}^{\mathrm{C_=P}}$-circuit C_1. By an argument similar to the base case, we can replace M_2 with an NEQ circuit C_2. This yields a circuit of the form $C_1 \vee C_2$ which outputs 1 if and only if M accepts x. $C_1 \vee C_2$ is a $\mathrm{BH}_{2k}^{\mathrm{C_=P}}$ circuit. The argument that we can find an appropriate $\mathrm{BH}_{2k+1}^{\mathrm{C_=P}}$-circuit for any $\mathrm{BH}_{2k+1}(\mathrm{C_=P}^A)$-machine is similar, again using proposition 3. ∎

We are now in a position to explain how the oracle separations follow from theorem 21. Define, for each k, the test languages $L_k(A)$ related to the functions $f_{k,n}$ as follows.

$$L_{2k}(A) = \{1^n | \bigvee_{i=1}^{k} [(\exists z, |z| = n)(\langle z, 2i-1\rangle \in A) \wedge (\forall z, |z| = n)(\langle z, 2i\rangle \notin A)]\}$$

$$L_{2k+1}(A) = \{1^n | 1^n \in L_{2k}(A) \vee [(\exists z, |z| = n)(\langle z, 2k+1\rangle \in A)]\}$$

It is clear from proposition 2 that for all k and A, $L_k(A) \in \mathrm{BH}_k(\mathrm{NP})$. Furthermore, for any x, $x \in L_k(A) \Leftrightarrow f_{k,2^n}(X(A)) = 1$ where the arguments $X(A)$ for $f_{k,2^n}$ are defined by $x_{i,j} = \chi_A(\langle s_i^n, j\rangle)$, as above.

Combining these observations with theorem 21 and proposition 22, we conclude that for some A, for all even k, $L_k(A) \notin \text{co-}\mathrm{BH}_k(\mathrm{C_=P}^A)$, and for all odd k, $L_k(A) \notin \mathrm{BH}_k(\mathrm{C_=P}^A)$. This proves

Theorem 23 *There exists an oracle A such that for odd k, $\mathrm{BH}_k(\mathrm{NP}^A) \not\subseteq \mathrm{BH}_k(\mathrm{C_=P}^A)$ and for even k, $\mathrm{BH}_k(\mathrm{NP}^A) \not\subseteq \text{co-}\mathrm{BH}_k(\mathrm{C_=P}^A)$.*

Then using theorem 8, we have,

Corollary 24 *There exists an oracle A such that $\mathrm{P}_{(k+1)-\mathrm{T}}^{\mathrm{NP}^A} \not\subseteq \mathrm{P}_{k-\mathrm{T}}^{\mathrm{C_=P}^A}$.*

7 Open Problems

Oracle separations are naturally not satisfying in these times of techniques that do not relativize, and thus the results of this paper raise more questions than they answer. It would be interesting to see sharper results along the lines of Theorem 9, in particular using techniques which exploit *differences* rather than similarities between NP and $\mathrm{C_{\neq}P}$ (indeed, it is possible that Theorem 9 is vacuous since it is possible with our current state of knowledge that P^{PP} =PSPACE). Similarly, some plausible separation of $\mathrm{P}^{\mathrm{C_=P}}$ from P^{PP} without oracles would be very interesting: Does the hypothesis $\mathrm{P}^{\mathrm{C_=P}} = \mathrm{P}^{\mathrm{PP}}$ have any dramatic consequences?

Acknowledgements

I wish to thank the members of the Departament de Llenguatges i Sistemes Informàtics, UPC Barcelona, where this work was done, for their hospitality, with special thanks to José Balcázar for helping to make the visit possible and for valuable comments on the paper. I thank Jacobo Torán for many discussions on this subject, Mitsunori Ogiwara and Seinosuke Toda for sharing a preliminary version of their results with me, and all of the above for their comments. Conversations with Antoni Lozano and Gerd Wechsung are also gratefully acknowledged.

References

[1] J. L. Balcázar, J. Díaz, and J. Gabarró, *Structural Complexity Theory I*, Volume II of *EATCS Monographs on Theoretical Computer Science*, Springer-Verlag, New York, 1988.

[2] R. Beigel, "Bounded Queries to SAT and the Boolean Hierarchy", Johns Hopkins Tech Report 87-8 (1988), to appear in Theoretical Computer Science.

[3] R. Beigel, N. Reingold and D. Spielman, "PP is Closed Under Intersection", STOC 1991.

[4] J.-y. Cai "Probability One Separation of the Boolean Hierarchy", *4th Annual Symposium on Theoretical Aspects of Computer Science*, Springer-Verlag Lecture Notes in Computer Science 247 (1987) 148-158.

[5] J.-y. Cai, T. Gunderman, J. Hartmanis, L. A. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung, "The Boolean Hierarchy I: Structural Properties", *SIAM Journal of Computing*, **17** (1988) 1232-1252.

[6] R. Chang and J. Kadin, "The Boolean Hierarchy and the Polynomial Hierarchy: a Closer Connection", *5th Annual Conference on Structure in Complexity Theory* (1990) 169-178.

[7] T. Gundermann, N. A. Nasser and G. Wechsung, "A Survey on Counting Classes", *5th Annual Conference on Structure in Complexity Theory* (1990) 140-153.

[8] J. Kadin, "Restricted Turing Reducibilities and the Structure of the Polynomial Time Hierarchy", Ph.D. thesis, Cornell University, February 1988.

[9] C. Lautenmann, "BPP and the Polynomial Hierarchy", *Information Processing Letters* **17** (1983) 215-217.

[10] M. Ogiwara and S. Toda, "On Polynomial-Time Reducibility Notions to Certain Complexity Classes", manuscript, April 1991.

[11] J. Tarui, "Randomized Polynomials, Threshold Circuits, and the Polynomial Hierarchy", *Proceedings of the 8th Annual Symposium on Theoretical Aspects of Computer Science* (1991) 238-250.

[12] S. Toda, "On the computational power of PP and $\oplus$P", *Proceedings 30th IEEE Symposium on Foundations of Computer Science* (1989) 514-519.

[13] S. Toda and M. Ogiwara, "Counting Classes Are as Hard as the Polynomial- Time Hierarchy", to appear in *6th Annual Conference on Structure in Complexity Theory* (1991).

[14] J. Torán, "Structural Properties of the Counting Hierarchy", Ph.D. thesis, Facultat d'Informatica de Barcelona, 1988.

[15] K. Wagner, "Compact Descriptions and the Counting Polynomial Time Hierarchy", *Acta Informatica* **23** (1986) 325-356.

[16] K. Wagner, "Bounded Query Classes", *SIAM Journal of Computing* **19** (1990) 833-846.