# A Relationship between Difference Hierarchies and Relativized Polynomial Hierarchies

Richard Beigel* Yale University

Richard Chang† Cornell University

Mitsunori Ogiwara‡ University of Electro-Communications

January 5, 1995

**Abstract**

Chang and Kadin have shown that if the difference hierarchy over NP collapses to level $k$, then the polynomial hierarchy (PH) is equal the $k$th level of the difference hierarchy over $\Sigma_2^p$. We simplify their proof and obtain a slightly stronger conclusion: If the difference hierarchy over NP collapses to level $k$, then PH collapses to $\left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^{\mathrm{NP}}$, the class of sets recognized in polynomial time with $k-1$ nonadaptive queries to a set in $\mathrm{NP}^{\mathrm{NP}}$ and an unlimited number of queries to a set in NP. We also extend the result to classes other than NP: For any class $C$ that has $\leq_m^p$-complete sets and is closed under $\leq_{\mathrm{conj}}^p$- and $\leq_m^{\mathrm{NP}}$-reductions (alternatively, closed under $\leq_{\mathrm{disj}}^p$- and $\leq_m^{\text{co-NP}}$-reductions), if the difference hierarchy over $C$ collapses to level $k$, then $\mathrm{PH}^C = \left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^C$. Then we show that the exact counting class $\mathrm{C_=P}$ is closed under $\leq_{\mathrm{disj}}^p$- and $\leq_m^{\text{co-NP}}$-reductions. Consequently, if the difference hierarchy over $\mathrm{C_=P}$ collapses to level $k$ then $\mathrm{PH}^{\mathrm{PP}} (= \mathrm{PH}^{\mathrm{C_=P}})$ is equal to $\left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^{\mathrm{PP}}$. In contrast, the difference hierarchy over the closely related class PP is known to collapse.

Finally we consider two ways of relativizing the bounded query class $\mathrm{P}^{\mathrm{NP}}_{k\text{-tt}}$: the restricted relativization $\mathrm{P}^{\mathrm{NP}^C}_{k\text{-tt}}$, and the full relativization $\left(\mathrm{P}^{\mathrm{NP}}_{k\text{-tt}}\right)^C$. If $C$ is NP-hard, then we show that the two relativizations are different unless $\mathrm{PH}^C$ collapses.

---

*Dept. of Computer Science, P.O. Box 2158, Yale Station, New Haven, CT 06520-2158. beigel-richard@cs.yale.edu. Supported in part by NSF grants CCR-8808949 and CCR-8958528.

†Dept. of Computer Science, Upson Hall, Cornell University, Ithaca, NY 14853. Supported in part by NSF Research Grant CCR 88-23053.

‡Dept. of Computer Science, University of Electro-Communications, Chofu-si, Tokyo 182, Japan. This work was done while the author was at Dept. of Information Science, Tokyo Institute of Technology.

# 1. Introduction

Numerous researchers [3, 5, 8, 9, 10, 11, 16, 17, 24, 25, 26, 27] have studied the Boolean hierarchy over NP. This hierarchy intertwines the query hierarchies over NP, and is identical to the Haussdorf and the difference hierarchies over NP. (Similar relations hold among hierarchies over many classes other than NP [7].) A central question is whether these hierarchies collapse. Because they stand or fall together, it is sufficient to study a single one. We find that the difference hierarchy is the most amenable to analysis.

Kadin [16] was the first to discover non-trivial structural consequences of the collapse of the difference hierarchy over NP. He showed that if the difference hierarchy over NP collapses, then the polynomial hierarchy is equal to $\Delta_3^p$. Kadin's result can be understood as translating a collapse of one hierarchy upward to a collapse of a larger hierarchy. Kadin's result was improved by Wagner [25], who showed that if the difference hierarchy over NP collapses to level $k$ then the polynomial hierarchy is equal to $\mathrm{P}^{\Sigma_2^p}_{O(k)\text{-tt}}$, and independently by Chang and Kadin [11], who showed that if the difference hierarchy over NP collapses to level $k$, then the polynomial hierarchy is equal to $\mathrm{DIFF}_k(\Sigma_2^p)$, where $\mathrm{DIFF}_k(C)$ denotes the $k$th level of the difference hierarchy over $C$, defined in Section 2.

In contrast, Beigel, Reingold, and Spielman [6] have shown that the difference hierarchy over PP is equal to PP, yet it is not known whether this collapse translates upward to $\mathrm{PH}^{\mathrm{PP}}$, $\mathrm{P}^{\#\mathrm{P}}$, or Wagner's [23] counting hierarchy (CH).[1] None of the questions below has been answered; neither has anyone shown that the answer to any of them is negative even relative to an oracle.

- Does $\mathrm{PH}^{\mathrm{PP}}$ collapse?
- $\mathrm{P}^{\#\mathrm{P}} = \mathrm{P}^{\#\mathrm{P}[1]}$?
- Does the counting hierarchy collapse?
- $\mathrm{PSPACE} = \mathrm{P}^{\#\mathrm{P}[1]}$?

Separation of the levels of the counting hierarchy relative to an oracle is of special interest, because it is equivalent to separating the levels of the circuit class $\mathrm{TC}_0$.

The questions above motivate us to determine precisely which properties of NP cause a collapse of the difference hierarchy over NP to translate upward. For any class $C$ that has $\leq_m^p$-complete sets and is closed under $\leq_{\mathrm{conj}}^p$- and $\leq_m^{\mathrm{NP}}$-reductions, we show in Corollary 12 that

$$\mathrm{DIFF}_k(C) = \text{co-}\mathrm{DIFF}_k(C) \Rightarrow \mathrm{PH}^C = \left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^C.$$

By a symmetry argument, the result holds as well for any class $C$ that has $\leq_m^p$-complete sets and is closed under $\leq_{\mathrm{disj}}^p$- and $\leq_m^{\text{co-NP}}$-reductions. Our main results extend Chang and Kadin's result; our proof is also simpler.

[1] CH (resp., PH) is the smallest non-empty class $C$ such that $\mathrm{PP}^C \subseteq C$ (resp., $\mathrm{NP}^C \subseteq C$).

While the class PP is closed under $\leq_{tt}^{p}$-reductions [13], it does not seem likely that PP is closed under $\leq_{m}^{\mathrm{NP}}$ reductions, for then we would have PARITYP $\subseteq$ PP, which does not relativize [22]. Thus our main result does not seem to apply to the class PP. This explains, in part, why the collapse of the difference hierarchy over PP has not been shown to translate upward.

However, the class $\mathrm{C_{=}P}$,[2] which is closely related to PP, is closed under $\leq_{\mathrm{disj}}^{p}$- and $\leq_{m}^{\text{co-NP}}$ reductions, as we show in Theorem 15 (similar closure properties were obtained independently by Gundermann, Nasser, and Wechsung [15]). Applying our main result and a theorem of Toran, we find that the difference hierarchy over $\mathrm{C_{=}P}$ does not collapse unless the polynomial hierarchy relative to PP collapses. (Green [14] independently proved a similar theorem. However, our techniques give a stronger collapse of the polynomial hierarchy relative to PP.) This structural consequence complements a result of Gundermann, Nasser, and Wechsung [15], who constructed oracles that make the difference hierarchy over $\mathrm{C_{=}P}$ proper.

## 2. Preliminaries

We assume that the reader is familiar with oracle Turing machines. $\mathrm{PH}^C$ denotes $C \cup \mathrm{NP}^C \cup \mathrm{NP}^{\mathrm{NP}^C} \cup \cdots$. We define the difference hierarchy over a class $C$.

**Definition 1.**

- $\mathrm{DIFF}_1(C) = C$,
- $\mathrm{DIFF}_{k+1}(C) = \{L_1 - L_2 : L_1 \in C, L_2 \in \mathrm{DIFF}_k(C)\}$.

**Definition 2.** $\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}$ is the class of languages that are polynomial-time truth-table reducible to a language in NP, via a truth-table of norm $k$.

The sequence $\mathrm{P}_{1\text{-tt}}^{\mathrm{NP}}, \mathrm{P}_{2\text{-tt}}^{\mathrm{NP}}, \ldots$ is called the nonadaptive query hierarchy over NP. We define full relativizations of $\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}$ as follows:

**Definition 3.** $\left(\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}\right)^C$ is the class of languages that are computable in polynomial time with $k$ nonadaptive queries to a set in $\mathrm{NP}^C$ and an unlimited number of queries to $C$.

By relativizing a result of Beigel [5], it follows that every language in $\left(\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}\right)^C$ is the symmetric difference of a language in $\mathrm{P}^C$ and a language in $\mathrm{DIFF}_k(\mathrm{NP}^C)$. Thus $\left(\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}\right)^C$ is contained in $\mathrm{DIFF}_{k+1}(\mathrm{NP}^C) \cap \text{co-}\mathrm{DIFF}_{k+1}(\mathrm{NP}^C)$. In general, it is not known whether $\left(\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}\right)^C = \mathrm{P}_{k\text{-tt}}^{\mathrm{NP}^C}$. However, in Section 4 we will show that if $C$ is NP-hard under $\leq_m^p$-reductions, then $\left(\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}\right)^C = \mathrm{P}_{k\text{-tt}}^{\mathrm{NP}^C}$ implies that the $\mathrm{PH}^C$ collapses.

Nondeterministic many-one reductions were defined by Ladner, Lynch, and Selman [18].

[2] $\mathrm{C_{=}P}$ (resp., PP) is the class of languages accepted by polynomial-time bounded nondeterministic Turing machines that accept when exactly (resp., at least) half of the computations accept.

**Definition 4.** We say that $A$ is *NP many-one reducible* to $B$ (denoted $A \leq_m^{\mathrm{NP}} B$) if there exists a constant $i$ and a polynomial-time computable function $f$ of two variables such that

$$x \in A \iff (\exists y \,.\, |y| = |x|^i)[f(x,y) \in B].$$

We write $\Sigma C$ to denote the closure of $C$ under $\leq_m^{\mathrm{NP}}$ reductions.

We define co-NP many-one reductions as a counterpart of NP many-one reductions.

**Definition 5.** We say that $A$ is *co-NP many-one reducible* to $B$ (denoted $A \leq_m^{\text{co-NP}} B$) if there exists a constant $i$ and a polynomial-time computable function $f$ of two variables such that

$$x \in A \iff (\forall y \,.\, |y| = |x|^i)[f(x,y) \in B].$$

We write $\forall C$ to denote the closure of $C$ under $\leq_m^{\text{co-NP}}$ reductions.

**Definition 6.** We say that $A$ is *polynomial-time conjunctive reducible* to $B$ (denoted $A \leq_{\text{conj}}^p B$) if there exists a polynomial-time computable function $f$ which maps each input to a finite set of strings such that

$$x \in A \iff f(x) \subseteq B.$$

We say that $A$ is *polynomial-time disjunctive reducible* to $B$ (denoted $A \leq_{\text{disj}}^p B$) if there exists a polynomial-time computable function $f$ which maps each input to a finite set of strings such that

$$x \in A \iff f(x) \cap B \neq \emptyset.$$

Before we go on, we need to define the join and symmetric difference of two languages.

**Definition 7.** Let $A$ and $B$ be any two languages. We use $A$ join $B$ to denote the join of $A$ and $B$; and $A \triangle B$ to denote the symmetric difference of $A$ and $B$. That is,

$$\begin{aligned} A \text{ join } B &= \{0x : x \in A\} \cup \{1x : x \in B\} \\ X \triangle Y &= \{\langle x,y\rangle : (x \in X \text{ and } y \notin Y) \text{ or } (x \notin X \text{ and } y \in Y)\} \end{aligned}$$

Alternatively, $\langle x,y\rangle \in X \triangle Y$ iff $(x \in X \iff y \notin Y)$.

The "mind-change" technique was developed by Wagner and Wechsung [27] and Beigel [5] in order to prove absolute results about the nonadaptive query hierarchy over NP. In particular, the mind-change technique was used to show that any language in $\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}$ can be reduced to $X \triangle Y$ for some language $X \in \mathrm{P}$ and $Y \in \mathrm{DIFF}_k(\mathrm{NP})$ (which in turn shows that $\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}} \subseteq \mathrm{DIFF}_{k+1}(\mathrm{NP}) \cap \text{co-}\mathrm{DIFF}_{k+1}(\mathrm{NP})$). This technique was also used to show that $\mathrm{P}^{\mathrm{NP}}_{k\text{-T}} \subseteq \mathrm{P}^{\mathrm{NP}}_{(2^k-1)\text{-tt}}$. Chang and Kadin applied a similar technique to the difference hierarchy over $\Sigma_2^p$ in order to obtain a precise level of collapse in their results. Similarly, we require a relativized version of the mind-change technique. Because we are a bit more careful, we obtain a stronger collapse than Chang and Kadin.

The mind-change lemma, Lemma 9, that we want to prove is stated in a very general form. To assist the reader in understanding this lemma, we first sketch a mind-change proof on a concrete example. We give some details which help illustrate the statement of Lemma 9. The rest of the proof may be found in the literature [5, 12, 21, 24]. We will also use the following lemma in Section 4.

**Lemma 8.** *Let $L$ be any language in $\left(\mathrm{P}^{\mathrm{NP}}_{k\text{-tt}}\right)^C$. Then, $L \leq_m^p L_{\mathrm{P}^C} \triangle L_k$ where $L_{\mathrm{P}^C}$ and $L_{\mathrm{NP}^C}$ are respectively the $\leq_m^p$-complete languages for $\mathrm{P}^C$ and $\mathrm{DIFF}_k(\mathrm{NP}^C)$.*

**Proof sketch:** Let the language $L_{\mathrm{NP}^C}$ be $\leq_m^p$-complete for $\mathrm{NP}^C$. Fix a machine $D$ which accepts $L$ using polynomially many queries to $C$, and $k$ non-adaptive queries to $L_{\mathrm{NP}^C}$. On input $x$, $D$ computes a set of queries $\{q_1(x), \ldots, q_k(x)\}$ possibly with the help of the oracle $C$. Then, $D$ asks the $\mathrm{NP}^C$ oracle if each of the query strings $q_i(x)$ is in $L_{\mathrm{NP}^C}$. Let $h$ be an encoding of a finite set. Define the two place predicate $B(x, h)$ to be true if $h \subseteq \{q_1(x), \ldots, q_k(x)\} \cap L_{\mathrm{NP}^C}$. The predicate $B$ can be recognized in $\mathrm{NP}^C$, because $\mathrm{NP}^C$ is closed under $\leq^p_{\mathrm{conj}}$-reductions. As in all mind change proofs, we need an upper bound on the length of the longest chain $h_1 \subset h_2 \subset h_3 \ldots$ such that for all $h_i$, $B(x, h_i)$ is true. In our case, the longest chain starts with $\emptyset$ and ends with $\{q_1(x), \ldots, q_k(x)\}$, so the longest chain has length $k+1$. We say that $h$ is maximal if $B(x, h)$ is true, and for all $h'$, $h \subset h' \Rightarrow \neg B(x, h')$. In this proof, the maximal subset $h_{\max}$ is unique, because $h_{\max} = L_{\mathrm{NP}^C} \cap \{q_1(x), \ldots, q_k(x)\}$.

Now, we define another two place predicate $A(x, h)$ which is true if $h \subseteq \{q_1(x), \ldots, q_k(x)\}$, and $D$ accepts $x$ if we simulate the oracle $L_{\mathrm{NP}^C}$ by assuming that $q_i(x) \in L_{\mathrm{NP}^C} \iff q_i(x) \in h$. Since $A(x, h)$ can be computed by simulating $D$ without the $L_{\mathrm{NP}^C}$ oracle, the predicate $A$ can be recognized in $\mathrm{P}^C$. Note that $x \in L$ iff $A(x, h_{\max})$ is true.

The strategy in a mind-change proof is to find the longest alternating chain; i.e, find a chain $\emptyset = h_0 \subset h_1 \subset h_2 \subset \cdots \subset h_j$ such that for all $i$, $1 \leq i \leq j$, $B(x, h_i) = \mathit{true}$ and $A(x, h_{i-1}) \neq A(x, h_i)$. Let $Q_m(x)$ be the predicate that such an alternating chain of length $m$ exists. That is,

$$Q_m(x) \equiv (\exists h_1 \subset \cdots \subset h_m)[B(x, h_1) \wedge B(x, h_2) \wedge \cdots \wedge B(x, h_m) \wedge$$
$$A(x, \emptyset) \neq A(x, h_1) \neq A(x, h_2) \cdots \neq A(x, h_m)].$$

Since $\mathrm{NP}^C$ is closed under $\leq^p_{\mathrm{conj}}$- and $\leq^p_{\mathrm{disj}}$-reductions, $Q_m$ can be recognized in $\mathrm{NP}^C$. Now, let $M$ be the length of the longest alternating chain and $h_1 \subset \cdots \subset h_M$ be a witness that $Q_M$ is true. Since $h_1 \subset \cdots \subset h_M$ cannot be extended to a longer alternating chain, $A(x, h_M) = A(x, h_{\max})$. Thus, $x \in L \iff A(x, h_M) = \mathit{true}$.

However, by the alternating condition of the chain, $A(x,\emptyset) = A(x,h_M)$ iff $M$ is even. So,

$$x \in L \iff A(x,\emptyset) \oplus (M \text{ is odd }).$$

Since $M$ is odd iff there is an odd number of true $Q_i$'s,

$$x \in L \iff A(x,\emptyset) \oplus (Q_1(x) \oplus \cdots \oplus Q_k(x)).$$

$A(x,\emptyset)$ can be computed in $\mathrm{P}^C$. Also, $Q_1(x) \oplus \cdots \oplus Q_k(x)$ is equal to $(Q_1(x) - (Q_2(x) - (\cdots - Q_k(x))))$, because $Q_{i+1}(x) \Rightarrow Q_i(x)$ for all $i$. Thus, we have reduced $L$ to $L_{\mathrm{P}^C} \triangle L_k$. ∎

We hope that the preceding proof gives the reader enough intuition to follow a mind-change proof in a more general and abstract setting. In the following lemma, the predicates $A$, $B$ and $Q_m$ are analogous to those in Lemma 8. Note that in the next lemma, the longest chain of $h$'s has length $k$ instead of $k+1$, and the longest alternating chain has length $k-1$. Also, there may not be a unique maximal $h$, hence the condition that $A(x,h)$ be the same for all maximal $h$.

**Lemma 9 (Mind-change).** *Fix a natural number $k$. Let $\prec$ be a polynomial-time computable partial order, with minimum element $\Lambda$. Let $A$ and $B$ be two-place predicates. Suppose that*

(a) *there exists a polynomial $p$ such that $B(x,h) \Rightarrow |h| \le p(|x|)$, and*

(b) *for all $x$, $B(x,\Lambda) = true$, and*

(c) $\neg(\exists x)(\exists h_1 \prec \cdots \prec h_{k+1})[B(x,h_1) \wedge \cdots \wedge B(x,h_{k+1})]$.

*We say that $h$ is* maximal *if*

$$B(x,h) \wedge \neg(\exists h')[(h \prec h') \wedge B(x,h')].$$

*Suppose that the value of $A(x,h)$ is the same for every maximal $h$, and define $Q(x)$ to be this value. Then, the predicate $Q$ is in $\left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^{B \text{ join } A}$.*

**Proof:** Define

$$\begin{aligned} Q_m(x) \equiv (\exists h_1 \prec \cdots \prec h_m)[\ (\ B(x,h_1) \wedge \cdots \wedge B(x,h_m)\ )\wedge \\ (\ A(x,\Lambda) \ne A(x,h_1) \ne \cdots \ne A(x,h_m)\ )\ ]. \end{aligned}$$

Let $M$ be the largest $m$ such that $Q_m(x) = true$. Let $h_1,\ldots,h_M$ be the witness that $Q_M(x) = true$. For all $h$, if $h_M \prec h$ then $B(x,h) = false$ or $A(x,h) = A(x,h_M)$. Therefore, $Q(x) = A(x,\Lambda)$ iff $M$ is even. $\Lambda \prec h_1$, so by (c), $M \le k-1$. Thus,

$$Q(x) = A(x,\Lambda) \oplus (Q_1(x) \oplus \cdots \oplus Q_{k-1}(x)).$$

By (a), $Q_m \in \mathrm{NP}^{B \text{ join } A}$; therefore, $Q \in \left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^{B \text{ join } A}$. ∎

## 3. Advice for collapsing hierarchies

Our main theorem could be obtained by a close inspection of Chang and Kadin's [11] proof for the case $C = \text{co-NP}$. Instead, we present our own proof, which is different and shorter. Our stronger collapse is due to the mind-change lemma in the preceding section. Like Kadin [16], we adaptively construct a maximal sequence of "hard" strings of each length, which we call a hard sequence. A single hard sequence allows us to reduce $C$ predicates to co-$C$ predicates for all arguments with length $\leq n$. In the construction of these hard sequences, it is important to exploit the structure of the complete language for the levels of the Boolean hierarchy. Each level of the Boolean hierarchy (or equivalently, of the Difference hierarchy) has several canonical $\leq_m^p$-complete languages. Previous authors have had to consider separate cases for odd $k$ and for even $k$, because they use the complete languages which have alternating intersections and unions. We consider only one case because we work with the complete languages which have a nested differences structure. Like Wagner [25], we incorporate one or more hard sequences directly into a polynomial-length advice string, thus avoiding the need to construct sparse oracles as in [16] or almost-tally oracles as in [11].

A major subtlety arises when one uses the hard sequences as polynomial-length advice in order to collapse $\text{PH}^C$. Recall that a single hard sequence allows us to reduce $C$ predicates to co-$C$ predicates for all arguments with length $\leq n$. Then $C$'s closure properties allow us (1) to reduce any $\text{NP}^C$ predicate to a co-$C$ predicate for all arguments with length $\leq n$. Consequently, a single hard sequence allows us (2) to reduce any $\text{NP}^{\text{NP}^C}$ predicate to an $\text{NP}^C$ predicate for all arguments with length $\leq n$. We perform (2) and then (1) in order to reduce any $\text{NP}^{\text{NP}^C}$ predicate to a co-$C$ predicate. However, (2) produces significantly longer arguments, so we need a different hard sequence when performing (1). Because of the need for two hard sequences, this shows only $(*)$ that $\text{PH}^C \subseteq \left(\text{P}^{\text{NP}}_{(2k-2)\text{-tt}}\right)^C$. Chang and Kadin devote considerable effort to overcoming this difficulty; they show how to construct both hard sequences, given a single hard sequence of sufficiently greater length. On the other hand, we note that only one hard sequence is needed in order to reduce a $\text{P}^{\text{NP}^C}$ predicate to a $\text{P}^C$ predicate; thus we show that $\text{P}^{\text{NP}^C} \subseteq \left(\text{P}^{\text{NP}}_{(k-1)\text{-tt}}\right)^C$. Combined with $(*)$ this implies that $\text{PH}^C \subseteq \left(\text{P}^{\text{NP}}_{(k-1)\text{-tt}}\right)^C$.

**Theorem 10.** *Let $C$ be a class having $\leq_m^p$-complete sets and closed under $\leq_m^{\text{co-NP}}$- and $\leq_{\text{disj}}^p$-reductions. If $\text{DIFF}_k(C) = \text{co-DIFF}_k(C)$, then $\text{PH}^C = \left(\text{P}^{\text{NP}}_{(k-1)\text{-tt}}\right)^C$.*

**Proof:** Let $C$ and $k$ be as in the hypothesis. Let $L$ be $\leq_m^p$-complete for $C$. Define

$$\begin{aligned} L_1 &= L, \\ L_{m+1} &= \{\langle x, y\rangle : x \in L \wedge y \notin L_m\}, \end{aligned}$$

where '$\langle$', '$\rangle$', and ',' are new characters. Then it is clear that $L_m$ is $\leq_m^p$-complete for $\text{DIFF}_m(C)$. Since $\text{DIFF}_k(C) = \text{co-DIFF}_k(C)$ by assumption, $L_k \leq_m^p \overline{L_k}$. Therefore there exists a polynomial-time computable function $h_k$ such that

$$w \in L_k \iff h_k(w) \notin L_k.$$

Fix a positive integer $n$; we will rely on the equation above only for $|w| \leq kn+3(k-1)$. Define

$$\langle f_{k-1}(x,y), g_{k-1}(x,y)\rangle = h_k(\langle x,y\rangle).$$

Then for $|x| = n$ and $|y| \leq (k-1)n + 3(k-2)$ we have $|\langle x,y\rangle| \leq kn + 3(k-1)$, so

$$x \in L \wedge y \notin L_{k-1} \iff f_{k-1}(x,y) \notin L \vee g_{k-1}(x,y) \in L_{k-1}.$$

We say that a string $x$ is $k$-*easy* if

$$(\exists y \,.\, |y| \leq (k-1)|x| + 3(k-2))[f_{k-1}(x,y) \notin L].$$

We say that $x$ is $k$-*hard* if

$$x \in L \wedge (\forall y \,.\, |y| \leq (k-1)|x| + 3(k-2))[f_{k-1}(x,y) \in L].$$

Note that if $x$ is $k$-easy then $x \in L$, and furthermore that the set of all $k$-easy strings is in $\Sigma$co-$C$ = co-$C$. If there exists a $k$-hard string $\chi_k$ of length $n$ then we have

$$(\forall y \,.\, |y| \leq (k-1)n + 3(k-2))[y \notin L_{k-1} \iff g_{k-1}(\chi_k, y) \in L_{k-1}],$$

so for $|y| \leq (k-1)n + 3(k-2)$

$$y \in L_{k-1} \iff g_{k-1}(\chi_k, y) \notin L_{k-1}.$$

Define

$$h_{k-1}(w) = g_{k-1}(\chi_k, w).$$

Then for $|w| \leq (k-1)n + 3(k-2)$ we have

$$w \in L_{k-1} \iff h_{k-1}(w) \notin L_{k-1}.$$

Iterating that process, we define $\chi_{k-1}, \ldots, \chi_{j+1}$ — stopping when $j = 1$ or when there exists no $j$-hard string of length $n$ — and we define the corresponding functions $h_{k-2}, \ldots, h_j$. (Since there may be several ways to choose the hard strings, we should write $h_{i\chi_k\cdots\chi_{i+1}}$ instead of simply $h_i$, but we don't.) The $i$-*easy* and the $i$-*hard* strings are defined by this iterative process as well (again depending implicitly on the choice of $\chi_k, \ldots, \chi_{i+1}$).

For $i < j$ we say that there are no $i$-easy strings. If $j = 1$ we encounter a special case. For $|x| \leq n$ we have

$$x \in L \iff h_1(x) \notin L.$$

We say that $x$ is 1-easy if $h_1(x) \notin L$. There are no 1-hard strings.

Let $x$ be a particular string of length $\leq n$, and let $\chi_k, \ldots, \chi_{j+1}$ be a maximal sequence of hard strings of length $n$. Then $x \in L$ iff $x$ is $j$-easy. Thus, using the strings

$\chi_k, \ldots, \chi_{j+1}$ as advice, we can effectively reduce $L$ to a co-$C$ predicate for arguments of length $\leq n$. More precisely, define $H$ to be the set of strings $\langle x, \langle \chi_k, \cdots, \chi_{j+1}\rangle\rangle$ such that $x$ is $j$-easy with respect to $\chi_k, \cdots, \chi_{j+1}$; that is, the function $f_j$ derived from $\chi_k, \cdots, \chi_{j+1}$ witnesses that

$$(\exists y \,.\, |y| \leq j|x| + 3(j-1))[f_j(x, y) \notin L].$$

It is not hard to see that $H$ is $\leq_m^{\mathrm{NP}}$ reducible to a set in co-$C$, so $H \in$ co-$C$, and that $x \in L$ iff $\langle x, \langle \chi_k, \cdots, \chi_{j+1}\rangle\rangle \in L$ for a maximal sequence of hard strings $\chi_k, \cdots, \chi_{j+1}$.

We complete the proof by proving two things:

**Claim 1:** $\mathrm{P}^{\mathrm{NP}^C} \subseteq \left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^C$.

**Claim 2:** $\mathrm{NP}^{\mathrm{NP}^C} \subseteq \left(\mathrm{P}^{\mathrm{NP}}_{(2k-2)\text{-tt}}\right)^C$.

**Proof of Claim 1:** Let $Q$ be any $\mathrm{P}^{\mathrm{NP}^C}$ predicate. Then $Q \in \mathrm{P}^R$ for some $R \in \mathrm{NP}^L$. Assume that $R$ is reducible to $L$ via a nondeterministic Turing machine $M$ running in time $r(n)$. Each computation path of $M$ on $x$ can be encoded into a string of length $\leq r(n)$, which consists of all nondeterministic guesses and answers obtained from an oracle. Also, $x \in R$ iff there is an accepting computation path in which every query answered affirmatively is in $L$ and every query answered negatively is not in $L$. Define $D$ to be the set of strings $\langle x, \langle s\rangle\rangle$ such that there is an accepting computation path of $M$ on $x$ in which every query answered negatively is not in $L$ and every query answered affirmatively is $j$-easy w.r.t. $\langle s\rangle$, where $\langle s\rangle$ has $k-j$ elements. Since $j$-easiness w.r.t. $\langle s\rangle$ can be tested by a co-$C$ predicate and co-$C$ is closed under $\leq_{\mathrm{conj}}^p$- and $\leq_m^{\mathrm{NP}}$-reductions, $D \in$ co-$C$, and if $\langle s\rangle$ is a maximal sequence, then $x \in R$ iff $\langle x, \langle s\rangle\rangle \in D$.

Hence there is a $\mathrm{P}^C$ predicate $A$ such that for all $x \in \Sigma^n$,

$$Q(x) = A(x, \langle \chi_k, \ldots, \chi_{j+1}\rangle).$$

Then $Q(x)$ is true iff there exists a maximal sequence $\chi_k, \ldots, \chi_{j+1}$ of hard strings of length $r(n)$ such that $A(x, \langle \chi_k, \ldots, \chi_{j+1}\rangle)$ is true. Let $B(x, \langle \chi_k, \ldots, \chi_{j+1}\rangle)$ be true iff $\chi_k$ is a $k$-hard string of length $r(n)$, ..., and $\chi_{j+1}$ is $(j+1)$-hard of length $r(n)$. Testing whether an individual string is $i$-hard is in $C$; therefore $B$ is a $\mathrm{P}^C$ predicate.

$Q(x)$ is true if and only if there exists $j$, $1 \leq j \leq k$, such that

- there exists a sequence $\chi_k, \ldots, \chi_{j+1}$ such that both $B(x, \langle \chi_k, \ldots, \chi_{j+1}\rangle)$ and $A(x, \langle \chi_k, \cdots, \chi_{j+1}\rangle)$ are true, and
- there does not exist a sequence $\chi'_k, \ldots, \chi'_j$ such that $B(x, \langle \chi'_k, \ldots, \chi'_j\rangle)$ is true.

We define $\langle s\rangle \prec \langle s'\rangle$ iff the sequence $s'$ is a proper extension of $s$. Then any chain of elements each satisfying $B(x, h)$ has length $\leq k$. By Lemma 9 $Q$ is a $\left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^C$ predicate. ▮ (Claim 1)

**Proof of Claim 2:** Let $Q$ be any $\mathrm{NP}^{\mathrm{NP}^C}$ predicate. Proceeding as above, we find an $\mathrm{NP}^C$ predicate $A$ and a polynomial $r(n)$ such that for every $n$, every maximal sequence $\chi_k, \ldots, \chi_{j+1}$ of hard strings of length $r(n)$, and every $x \in \Sigma^n$,

$$Q(x) = A(x, \langle \chi_k, \ldots, \chi_{j+1} \rangle).$$

Applying a similar argument to $A$, we find a co-$C$ predicate $A'$ and a polynomial $r'(n)$ such that for every $n$, every maximal sequence $\chi_k, \ldots, \chi_{j+1}$ of hard strings of length $r(n)$, and every maximal sequence $\chi'_k, \ldots, \chi'_{j'+1}$ of hard strings of length $r'(n)$, and every $x \in \Sigma^n$,

$$Q(x) = A'(x, (\langle \chi_k, \ldots, \chi_{j+1} \rangle, \langle \chi'_k, \ldots, \chi'_{j'+1} \rangle)).$$

We define $(\langle s_1 \rangle, \langle s_2 \rangle) \prec (\langle s'_1 \rangle, \langle s'_2 \rangle)$ iff $s_1 \preceq s'_1$, $s_2 \preceq s'_2$ and at least one of the extensions is a proper extension. Then any chain of elements each satisfying $B(x,h)$ has length $\leq 2k-1$. By Lemma 9, $Q$ is in $\left(\mathrm{P}^{\mathrm{NP}}_{(2k-2)\text{-tt}}\right)^C$. ∎ (Claim 2)

By Claim 2, $\mathrm{PH}^C \subseteq \mathrm{P}^{\mathrm{NP}^C}$, which is equal to $\left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^C$, by Claim 1. ∎

If we drop the restriction that co-$C$ be closed under $\leq^p_{\mathrm{conj}}$-reductions, then we obtain a weaker collapse. It is frustrating that we do not know how to obtain as strong a collapse as above; the need for hard strings for different lengths is the culprit.

**Theorem 11.** *Let $C$ be a class having $\leq^p_m$-complete sets and closed under $\leq^{\text{co-NP}}_m$-reductions, If $\mathrm{DIFF}_k(C) = \text{co-}\mathrm{DIFF}_k(C)$, then $\mathrm{PH}^C = \left(\mathrm{P}^{\mathrm{NP}}_{(2k-2)\text{-tt}}\right)^C$.*

**Proof sketch:** This differs from the preceding proof only in the two claims.

**Claim 1:** $\mathrm{P}^{\mathrm{NP}^C} \subseteq \left(\mathrm{P}^{\mathrm{NP}}_{(2k-2)\text{-tt}}\right)^C$.

**Claim 2:** $\mathrm{NP}^{\mathrm{NP}^C} \subseteq \left(\mathrm{P}^{\mathrm{NP}}_{(4k-4)\text{-tt}}\right)^C$.

**Proof of Claim 1:** Using one maximal sequence of hard strings we can reduce a co-$C$ predicate to a $C$ predicate. Since $C$ is closed under $\leq^{\text{co-NP}}_m$-reductions, and *a fortiori* closed under $\leq^p_{\mathrm{conj}}$ reduction, verifying a path of an $\mathrm{NP}^C$ computation can be reduced to a single $C$ predicate. Using another maximal sequence of hard strings, we can reduce that $C$ predicate to a co-$C$ predicate. Since $\Sigma\text{co-}C = \text{co-}C$, we thus reduce an $\mathrm{NP}^C$ predicate to a co-$C$ predicate. Applying Lemma 9, we have $\mathrm{P}^{\mathrm{NP}^C} \subseteq \left(\mathrm{P}^{\mathrm{NP}}_{(2k-2)\text{-tt}}\right)^C$.

**Proof of Claim 2:** Using a total of four maximal sequences of hard strings, we reduce an $\mathrm{NP}^{\mathrm{NP}^C}$ predicate to a co-$C$ predicate. Applying Lemma 9, we have $\mathrm{NP}^{\mathrm{NP}^C} \subseteq \left(\mathrm{P}^{\mathrm{NP}}_{(4k-4)\text{-tt}}\right)^C$. ∎

**Corollary 12.** *Let $C$ be a class having $\leq^p_m$-complete sets and closed under $\leq^{\mathrm{NP}}_m$- and $\leq^p_{\mathrm{conj}}$-reductions. If $\mathrm{DIFF}_k(C) = \text{co-}\mathrm{DIFF}_k(C)$, then $\mathrm{PH}^C = \left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^C$.*

**Proof:** We note that for any $C$

$$\mathrm{DIFF}_k(\text{co-}C) = \begin{cases} \text{co-DIFF}_k(C) & \text{if } k \text{ is odd,} \\ \mathrm{DIFF}_k(C) & \text{if } k \text{ is even.} \end{cases}$$

So, $\mathrm{DIFF}_k(C) = \text{co-DIFF}_k(C) \iff \mathrm{DIFF}_k(\text{co-}C) = \text{co-DIFF}_k(\text{co-}C)$. Therefore, the corollary is equivalent to Theorem 10. ∎

Thus we extend the results of Kadin [16], Wagner [25], and Chang and Kadin [11]:

**Corollary 13.** *If* $\mathrm{DIFF}_k(\mathrm{NP}) = \text{co-DIFF}_k(\mathrm{NP})$ *then* $\mathrm{PH} = \left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^{\mathrm{NP}}$.

**Proof:** Assume that $\mathrm{DIFF}_k(\mathrm{NP}) = \text{co-DIFF}_k(\mathrm{NP})$. Since $\mathrm{NP} = \Sigma\mathrm{NP}$ and NP has $\le^p_m$-complete sets, Corollary 12 implies $\mathrm{PH}^{\mathrm{NP}} = \left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^{\mathrm{NP}}$, so $\mathrm{PH} = \left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^{\mathrm{NP}}$. ∎

Our results can be placed in the context of lowness [1, 4], though they lose quite a bit of strength in the translation.

**Corollary 14.** *Let $C$ be a class having $\le^p_m$-complete sets and closed under $\le^{\mathrm{NP}}_m$- and $\le^p_{\mathrm{conj}}$-reductions. If* $\mathrm{DIFF}_k(C) = \text{co-DIFF}_k(C)$, *then all $C$-complete sets are in* $\widehat{\mathrm{EL}}_4$.

**Proof:** By Corollary 12, all $C$-complete sets $L$, satisfy $(\Delta^p_4)^L = (\Delta^p_3)^L$. Therefore $(\Delta^p_4)^L = (\Delta^p_3)^{\mathrm{NP},L}$, which is the definition of $\widehat{\mathrm{EL}}_4$ [1]. ∎

The class $\mathrm{C_{=}P}$ was defined by Wagner [23]. For a polynomial-time bounded nondeterministic Turing machine $M$ let $\#acc_M$ and $\#rej_M$ be functions mapping $x$ to the number of accepting and rejecting paths of $M$ on $x$, respectively. A language $L$ belongs to $\mathrm{C_{=}P}$ if and only if there exists a polynomial-time bounded nondeterministic Turing machine $M$ such that $x \in L$ iff $\#acc_M(x) - \#rej_M(x) = 0$.

We show that $\mathrm{C_{=}P}$ is closed under $\le^p_{\mathrm{disj}}$-reductions and $\le^{\text{co-NP}}_m$ reductions. These results have appeared in [19]. Similar closure properties were obtained independently by Gundermann, Nasser, and Wechsung [15].

**Theorem 15.**

(a) $\mathrm{C_{=}P}$ *is closed under* $\le^p_{\mathrm{disj}}$*-reductions.*

(b) $\mathrm{C_{=}P}$ *is closed under* $\le^{\text{co-NP}}_m$*-reductions.*

**Proof:** Assume for concreteness that the reductions run in time bounded by $n^k$.

(a) By standard low-degree polynomial techniques for closure properties [6, Lemma 5], it suffices to construct a uniform sequence of multivariate polynomials $\{p_n(x_1, \ldots, x_{n^k})\}$ having degree $n^{O(1)}$ and coefficients bounded in absolute value by $2^{n^{O(1)}}$ satisfying

$$p_n(x_1, \ldots, x_{n^k}) = 0 \iff x_1 = 0 \vee \cdots \vee x_{n^k} = 0.$$

Let $p_n(x_1, \ldots, x_{n^k}) = x_1 \cdots x_{n^k}$.

(b) For this part, it suffices to construct a similar sequence of polynomials $\{p_n(x_1, \ldots, x_{2^{n^k}}\}$ satisfying

$$p_n(x_1, \ldots, x_{2^{n^k}}) = 0 \iff x_1 = 0 \wedge \cdots \wedge x_{2^{n^k}} = 0.$$

Let $p_n(x_1, \ldots, x_{2^{n^k}}) = x_1^2 + \cdots + x_{2^{n^k}}^2$.

∎

**Corollary 16.** *If* $\mathrm{DIFF}_k(\mathrm{C_{=}P}) = \text{co-}\mathrm{DIFF}_k(\mathrm{C_{=}P})$ *then* $\mathrm{PH}^{\mathrm{PP}} = \left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^{\mathrm{PP}}$.

**Proof:** Assume that $\mathrm{DIFF}_k(\mathrm{C_{=}P}) = \text{co-}\mathrm{DIFF}_k(\mathrm{C_{=}P})$. Since $\text{co-}\mathrm{C_{=}P} = \Sigma\text{co-}\mathrm{C_{=}P}$ and $\text{co-}\mathrm{C_{=}P}$ is closed under $\leq^p_{\mathrm{conj}}$-reductions, Theorem 10 implies that $\mathrm{PH}^{\mathrm{C_{=}P}} = \left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^{\mathrm{C_{=}P}}$. By a result of Toran [22], $\mathrm{NP}^{\mathrm{PP}} = \mathrm{NP}^{\mathrm{C_{=}P}}$ (because one can guess the exact threshold); therefore

$$\mathrm{PH}^{\mathrm{PP}} = \mathrm{PH}^{\mathrm{C_{=}P}} = \left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^{\mathrm{C_{=}P}} \subseteq \left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^{\mathrm{PP}}.$$

∎

# 4. Relativizing Bounded Query Classes

It is natural to ask whether our Corollary 13 is really stronger than Chang and Kadin's theorem [11], i.e., does

$$\mathrm{PH} \subseteq \mathrm{DIFF}_k(\mathrm{NP}^{\mathrm{NP}}) \not\Rightarrow \mathrm{PH} \subseteq \left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^{\mathrm{NP}}?$$

It is clear that

$$\left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^{\mathrm{NP}} \subset \mathrm{DIFF}_k(\mathrm{NP}^{\mathrm{NP}})$$

unless PH collapses, because equality would imply that $\mathrm{DIFF}_k(\mathrm{NP}^{\mathrm{NP}})$ is closed under complementation. However, Chang and Kadin's theorem implies that

$$\mathrm{PH} \subseteq \mathrm{DIFF}_k(\mathrm{NP}^{\mathrm{NP}}) \cap \text{co-}\mathrm{DIFF}_k(\mathrm{NP}^{\mathrm{NP}}),$$

so we would really like to know the answer to:

$$\left(\mathrm{P}^{\mathrm{NP}}_{(k-1)\text{-tt}}\right)^{\mathrm{NP}} \subset \mathrm{DIFF}_k(\mathrm{NP}^{\mathrm{NP}}) \cap \text{co-}\mathrm{DIFF}_k(\mathrm{NP}^{\mathrm{NP}})?$$

Currently we are unable to establish proper containment under plausible complexity assumptions. In considering that question, we came to the related question:

$$\mathrm{P}^{\mathrm{NP}^{\mathrm{NP}}}_{k\text{-tt}} \subset \left(\mathrm{P}^{\mathrm{NP}}_{k\text{-tt}}\right)^{\mathrm{NP}}?$$

The question above is interesting because it involves restricted relativizations. Relativizing the polynomial hierarchy is straightforward. For example, $\Sigma_2^{p,C}$ can be defined as $\mathrm{NP}^{(\mathrm{NP}^C)}$, and it does not matter that the base NP machine does not have direct access to the oracle $C$, because it can ask the $\mathrm{NP}^C$ oracle, instead.

However, there are two ways to relativize a bounded query hierarchy. In the first approach, the oracle $C$ is attached to the NP oracle only. This is a restricted relativization. We denote this class as $\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}^C}$, which is the class of languages recognized by polynomial time Turing machines which are allowed $k$ parallel queries to the $\mathrm{NP}^C$ oracle. In the second approach, the polynomial time base machine can ask $k$ parallel queries to the $\mathrm{NP}^C$ oracle and polynomially many serial queries to the $C$ oracle. This is a full relativization. We denote this second class as $\left(\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}\right)^C$. In what follows, we show that if $C$ is sufficiently hard, then the two relativizations are different unless $\mathrm{PH}^C$ collapses. This is an example of natural interest, where we have good circumstantial evidence that restricted relativizations are strictly less powerful than full relativizations.

We have two proofs of this. Both proofs use ideas that are substantially different from those in [11, 16, 25]. The first proof modifies a technique from [2], and is relatively simple, but it only collapses $\mathrm{PH}^C$ to $(\Sigma_3^p)^C$. The second proof is more difficult, combining two hard/easy-formulas arguments; it collapses $\mathrm{PH}^C$ to $\left(\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}\right)^{\mathrm{NP}^C}$.

**Proposition 17.** *Let $C$ be any class such that $\mathrm{NP} \subseteq \mathrm{P}^C$. If $\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}^C} = \left(\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}\right)^C$ then $\mathrm{PH}^C = (\Sigma_3^p)^C$.*

**Proof:** Let $L_{\mathrm{P}}$, $L_{\mathrm{P}^C}$ and $L_k$ be $\leq_m^p$-complete for P, $\mathrm{P}^C$, and $\mathrm{DIFF}_k(\mathrm{NP}^C)$, respectively. Using relativized versions of the mind-change proof in [5], one can show that $L_{\mathrm{P}^C} \triangle L_k$ is $\leq_m^p$-complete for $\left(\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}\right)^C$ and $L_{\mathrm{P}} \triangle L_k$ is $\leq_m^p$-complete for $\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}^C}$ (for details consult Lemma 8 and the literature [12, 21, 24] ). Thus $\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}^C} = \left(\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}\right)^C$ if and only if

$$L_{\mathrm{P}^C} \triangle L_k \leq_m^p L_{\mathrm{P}} \triangle L_k.$$

Fix a polynomial-time computable function $h$ that performs that reduction. For each $m$, we will construct polynomial-size advice allowing us to reduce $L_k$ to $\overline{L_k}$ on strings of length $\leq m$. Thus $\mathrm{DIFF}_k(\mathrm{NP}^C) \subseteq \text{co-}\mathrm{DIFF}_k(\mathrm{NP}^C)/\mathrm{poly}$, so $\mathrm{NP}^C \subseteq \text{co-}\mathrm{NP}^C/\mathrm{poly}$, so $\mathrm{PH}^C \subseteq (\Sigma_3^p)^C$.

Let $|S|$ denote the cardinality of the set $S$. Let $(\exists^{\geq\alpha} y \in S)$ denote "for at least $\alpha$ elements $y$ of $S$." Fix a length $m$. Throughout the construction of the advice let $\langle x', y'\rangle$ denote $h(x, y)$.
Begin construction:
Let $S = \{0,1\}^{\leq m}$.

Begin loop:

**Case 1** $(\exists x \in \{0,1\}^{\leq m})(\exists^{\geq \frac{1}{4}|S|} y \in S)[x \in L_{\mathrm{P}^C} \iff x' \notin L_{\mathrm{P}}]$: Choose such an $x$, and incorporate $x$ into the advice for length $m$. Let $S = S - \{y : x \in L_{\mathrm{P}^C} \iff x' \notin L_{\mathrm{P}}\}$. If $S = \emptyset$ then exit the loop.

**Case 2** $(\forall x \in \{0,1\}^{\leq m})(\exists^{\geq \frac{3}{4}|S|} y \in S)[x \in L_{\mathrm{P}^C} \iff x' \in L_{\mathrm{P}}]$: Discard all advice constructed so far for length $m$. For length $\leq m$, there is a nonuniform random polynomial-time algorithm to m-reduce $L_{\mathrm{P}^C}$ to $L_{\mathrm{P}}$: Input $x$; choose a random $y \in S$; compute $x'$; then $x \in L_{\mathrm{P}^C}$ iff $x' \in L_{\mathrm{P}}$. The nonuniform randomness can be simulated by incorporating a polynomial number of elements of $S$ into the advice, as in Schöning's proof that BPP $\in$ P/poly [20]. Exit the loop.

End loop.
End construction.

If case 2 is ever reached then the construction produces advice sufficient to reduce $L_{\mathrm{P}^C}$ to $L_{\mathrm{P}}$ for length $\leq m$. Since $\mathrm{NP} \subseteq \mathrm{P}^C$, this advice certainly allows us to reduce $L_k$ to $\overline{L_k}$ for length $\leq m$.

If case 2 is not reached then, after a linear number of iterations, $S$ becomes empty, so we have polynomial-size advice sufficient for a $\mathrm{P}^C$-algorithm to m-reduce $L_k$ to $\overline{L_k}$ for length $\leq m$, via the following algorithm: Input $y$; exhaustively search the advice for a string $x$ such that $x \in L_{\mathrm{P}^C} \iff x' \notin L_{\mathrm{P}}$; then $y \in L_k$ iff $y' \notin L_k$.

Thus $\mathrm{DIFF}_k(\mathrm{NP}^C) \subseteq \text{co-DIFF}_k(\mathrm{NP}^C)/\mathrm{poly}$, as promised. ∎

Now we prove the stronger result.

**Theorem 18.** *Let $C$ be any class such that* $\mathrm{NP} \subseteq \text{co-NP}^C$. *If* $\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}^C} = \left(\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}\right)^C$ *then* $\mathrm{PH}^C = \left(\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}\right)^{\mathrm{NP}^C}$.

**Proof:** Let $L_{\mathrm{P}}$, $L_{\mathrm{P}^C}$ and $L_{\mathrm{NP}^C}$ be $\leq_m^p$-complete for P, $\mathrm{P}^C$, and $\mathrm{NP}^C$, respectively. Let $L_k$ be defined by

$$L_1 = L_{\mathrm{NP}^C}, \quad L_{k+1} = \{ \langle x, y \rangle : x \in L_{\mathrm{NP}^C} \text{ or } y \notin L_k \}.$$

(Technically, $L_k$ is not complete for $\mathrm{DIFF}_k(\mathrm{NP}^C)$, but rather $\overline{L_k}$ is complete for $\mathrm{DIFF}_k(\text{co-NP}^C)$.) As we have mentioned before, $L_{\mathrm{P}^C} \triangle L_k$ is $\leq_m^p$-complete for $\left(\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}\right)^C$ and $L_{\mathrm{P}} \triangle L_k$ is $\leq_m^p$-complete for $\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}^C}$. Thus $\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}^C} = \left(\mathrm{P}_{k\text{-tt}}^{\mathrm{NP}}\right)^C$ if and only if

$$L_{\mathrm{P}^C} \triangle L_k \leq_m^p L_{\mathrm{P}} \triangle L_k.$$

Fix a polynomial-time computable function $h$ that performs that reduction. Fix $m$. We will construct advice that either lets us reduce $L_{\mathrm{P}^C}$ to $L_{\mathrm{P}}$ for all strings of length $\leq m$ or else lets us reduce $L_{\mathrm{NP}^C}$ to $\overline{L_{\mathrm{NP}^C}}$ for all strings of length $\leq m$.

Let $\{0,1\}^{\leq m \times k}$ denote the set of $k$-tuples of strings of length $\leq m$.

A sequence $\vec{\chi} = \langle \chi_1, \ldots, \chi_j \rangle$ is a *hard sequence* for length $m$ if $0 \leq j \leq k$ and all of the following conditions hold, for $1 \leq i \leq j$:

1. $|\chi_i| \le m$ and $\chi_i \notin L_{\mathrm{NP}^C}$.

2. $\forall \vec{p} = \langle p_1, \ldots, p_{k-i}\rangle \in \{0,1\}^{\le m\times(k-i)}$, $\forall u \in \{0,1\}^{\le m}$,

$$(u \in L_{\mathrm{P}^C} \iff v \notin L_{\mathrm{P}}) \Rightarrow y_i \notin L_{\mathrm{NP}^C},$$

where $\langle v, y_1, \ldots, y_i, \vec{q}\rangle = h(u, \vec{\chi}, \vec{p})$.

The structure of the proof is as follows.

**Claim 1:** There exists a hard sequence.

**Claim 2:** If $\vec{\chi}$ is a hard sequence and $|\vec{\chi}| = k$, then $\vec{\chi}$ induces a deterministic reduction from $L_{\mathrm{P}^C}$ to $L_{\mathrm{P}}$.

**Claim 3:** Suppose that $\vec{\chi}$ is a hard sequence and $|\vec{\chi}| = j < k$. Then, $\forall \vec{p} = \langle p_1, \ldots, p_{k-j}\rangle \in \{0,1\}^{\le m\times(k-j)}$, $\forall u \in \{0,1\}^{\le m}$,

$$(u \in L_{\mathrm{P}^C} \iff v \notin L_{\mathrm{P}}) \Rightarrow (\vec{p} \in L_{k-j} \iff \vec{q} \notin L_{k-j}),$$

where $\langle v, \vec{y}, \vec{q}\rangle = h(u, \vec{\chi}, \vec{p})$.

**Claim 4:** If $\vec{\chi}$ is a maximal-length hard sequence and $|\chi| < k$, then $\vec{\chi}$ induces a nondeterministic reduction from $\overline{L_{\mathrm{NP}^C}}$ to $L_{\mathrm{NP}^C}$.

Since the length of a hard sequence is bounded, Claim 1 implies that a maximal-length hard sequence $\vec{\chi}$ exists. By Claims 2 and 4, $\vec{\chi}$ induces a reduction from $L_{\mathrm{NP}^C}$ to $\overline{L_{\mathrm{NP}^C}}$ (recall that $\mathrm{NP} \subseteq \text{co-NP}^C$). We order hard sequences by length, so a chain of hard sequences contains at most $k+1$ elements. Applying Lemma 9, we collapse $\mathrm{PH}^C$ to $\left(\mathrm{P}^{\mathrm{NP}}_{k\text{-tt}}\right)^{\mathrm{NP}^C}$.

**Proof of Claim 1:** The empty sequence is a hard sequence. ▮ (Claim 1)

**Proof of Claim 2:** Suppose that $\vec{\chi} = \chi_1, \ldots, \chi_k$ is a hard sequence. Let $u \in \{0,1\}^{\le m}$, let $\langle v, \vec{y}\rangle = h(u, \vec{\chi})$ and let $y_1, \ldots, y_k = \vec{y}$. We will prove, by contradiction, that $u \in L_{\mathrm{P}^C} \iff v \in L_{\mathrm{P}}$. Suppose not. Then $u \in L_{\mathrm{P}^C} \iff v \notin L_{\mathrm{P}}$. Then by condition 2, $y_i \notin L_{\mathrm{NP}^C}$ for $i = 1, \ldots, k$. By condition 1, $\chi_i \notin L_{\mathrm{NP}^C}$ as well for $i = 1, \ldots, k$. Therefore, by the definition of $L_k$, $\vec{\chi} \in \mathrm{L}_k \iff \vec{y} \in \mathrm{L}_k$ (iff $k$ is even). However, this contradicts the fact that $h$ is a reduction from $L_{\mathrm{P}^C} \triangle L_k$ to $L_{\mathrm{P}} \triangle L_k$. Therefore for all $u \in \{0,1\}^{\le m}$, we have

$$u \in L_{\mathrm{P}^C} \iff v \in L_{\mathrm{P}}.$$

Thus the following algorithm reduces $L_{\mathrm{P}^C}$ to $L_{\mathrm{P}}$ for strings of length $\le m$: Input $u$; let $\langle v, \vec{y}\rangle = h(u, \vec{\chi})$; then $u \in L_{\mathrm{P}^C} \iff v \in L_{\mathrm{P}}$. ▮ (Claim 2)

**Proof of Claim 3:** Let $\vec{p} \in \{0,1\}^{\leq m \times k-j}$, $u \in \{0,1\}^{\leq m}$, and $\langle v, \vec{\chi}, \vec{q}\rangle = h(u, \vec{y}, \vec{p})$. By the definition of $h$,

$$\langle u, \vec{\chi}, \vec{p}\rangle \in L_{\mathrm{P}^C} \bigtriangleup L_k \iff \langle v, \vec{y}, \vec{q}\rangle \in L_{\mathrm{P}} \bigtriangleup L_k.$$

Suppose that $u \in L_{\mathrm{P}^C} \iff v \notin L_{\mathrm{P}}$. Then

$$\langle \vec{\chi}, \vec{p}\rangle \in L_k \iff \langle \vec{y}, \vec{q}\rangle \notin L_k.$$

By conditions 1 and 2, $\chi_i \notin L_{\mathrm{NP}^C}$ and $y_i \notin L_{\mathrm{NP}^C}$, for $1 \leq i \leq j$. Therefore, by the definition of $L_k$, $\langle \vec{\chi}, \vec{p}\rangle \in L_k$ iff $\vec{p} \in L_{k-j}$, and $\langle \vec{y}, \vec{q}\rangle \in L_k$ iff $\vec{q} \in L_{k-j}$. Therefore $\vec{p} \in L_{k-j} \iff \vec{q} \notin L_{k-j}$. ▮ (Claim 3)

**Proof of Claim 4:** Suppose $s \notin L_{\mathrm{NP}^C}$. Since $\vec{\chi}$ is maximal, $\langle \chi_1, \ldots, \chi_i, s\rangle$ does not satisfy condition 2 in the definition of hard sequences — which is exactly what we need.

Conversely, suppose $s \in L_{\mathrm{NP}^C}$. By Claim 3, $\forall \vec{p} \in \{0,1\}^{\leq m \times (k-i-1)}$, $\forall u \in \{0,1\}^{\leq m}$, $(u \in L_{\mathrm{P}^C} \iff v \notin L_{\mathrm{P}})$ implies

$$\langle s, \vec{p}\rangle \in L_{k-i} \iff \langle t, \vec{q}\rangle \notin L_{k-i}.$$

where $\langle v, y_1, \ldots, y_i, t, \vec{q}\rangle = h(u, \vec{\chi}, s, \vec{p})$. By expanding the definition of $L_{k-i}$, we have

$$(s \in L_{\mathrm{NP}^C} \text{ or } \vec{p} \notin L_{k-i-1}) \iff (t \notin L_{\mathrm{NP}^C} \text{ and } \vec{q} \in L_{k-i-1})$$

Since $s \in L_{\mathrm{NP}^C}$, $t \notin L_{\mathrm{NP}^C}$. So, when $s \in L_{\mathrm{NP}^C}$,

$$\forall \vec{p} \in \{0,1\}^{\leq m \times (k-i-1)},\ \forall u \in \{0,1\}^{\leq m},\ (u \in L_{\mathrm{P}^C} \iff v \notin L_{\mathrm{P}}) \Rightarrow t \notin L_{\mathrm{NP}^C}.$$

Thus, using $\chi$ as advice, an $\mathrm{NP}^C$ algorithm can m-reduce $\overline{L_{\mathrm{NP}^C}}$ to $L_{\mathrm{NP}^C}$, for strings of length $\leq m$, as follows: Input $s$; guess $\vec{p} \in \{0,1\}^{\leq m \times (k-i-1)}$ and $u \in \{0,1\}^{\leq m}$; let $\langle v, y_1, \ldots, y_i, t, \vec{q}\rangle = h(u, \vec{\chi}, \vec{p})$; if $u \in L_{\mathrm{P}^C} \iff v \notin L_{\mathrm{P}}$ and $t \in L_{\mathrm{NP}^C}$ then accept, else reject. ▮ (Claim 4)

It follows from Claim 2 that a hard sequence of length $k$ induces a deterministic reduction from $C$ to P for strings of length $\leq m$. Therefore a hard sequence of length $k$ induces a reduction from $\mathrm{NP}^C$ to NP for strings of length $\leq m$. By assumption, $\mathrm{NP} \subseteq \text{co-NP}^C$, so a hard sequence of length $k$ induces a reduction from $\mathrm{NP}^C$ to co-$\mathrm{NP}^C$ for strings of length $\leq m$.

It follows from Claim 4, that a maximal hard sequence of length $< k$ induces a deterministic reduction from co-$\mathrm{NP}^C$ to $\mathrm{NP}^C$ for strings of length $\leq m$. Thus, a maximal hard sequence of any length induces a deterministic reduction from co-$\mathrm{NP}^C$ to $\mathrm{NP}^C$ for strings of length $\leq m$.

Therefore, every $\mathrm{P}^{\Sigma_2^{p,C}}$ languages is recognized by a $\mathrm{P}^{\mathrm{NP}^C}$ machine using a single maximal hard sequence as advice for each length. Note that the set of hard sequences belongs to co-NP$^C$. If we order hard sequences by length, then any chain has length $\leq k+1$. So, by Lemma 9

$$\mathrm{P}^{\Sigma_2^{p,C}} \subseteq \left(\mathrm{P}^{\mathrm{NP}}_{k\text{-tt}}\right)^{\mathrm{NP}^C}.$$

A similar argument, using two maximal hard sequences per length, shows that

$$\Sigma_3^{p,C} \subseteq \mathrm{P}^{\Sigma_2^{p,C}}.$$

Thus, $\mathrm{PH}^C \subseteq \left(\mathrm{P}^{\mathrm{NP}}_{k\text{-tt}}\right)^{\mathrm{NP}^C}$. ∎

**Corollary 19.** *If* $\left(\mathrm{P}^{\mathrm{NP}}_{k\text{-tt}}\right)^{\mathrm{NP}} = \mathrm{P}^{\mathrm{NP}^{\mathrm{NP}}}_{k\text{-tt}}$, *then* $\mathrm{PH} = \left(\mathrm{P}^{\mathrm{NP}}_{k\text{-tt}}\right)^{\mathrm{NP}^{\mathrm{NP}}}$.

**Corollary 20.** *If* $\left(\mathrm{P}^{\mathrm{NP}}_{k\text{-tt}}\right)^{\mathrm{PP}} = \mathrm{P}^{\mathrm{NP}^{\mathrm{PP}}}_{k\text{-tt}}$, *then* $\mathrm{PH}^{\mathrm{PP}} = \left(\mathrm{P}^{\mathrm{NP}}_{k\text{-tt}}\right)^{\mathrm{NP}^{\mathrm{PP}}}$.

Acknowledgments. We are grateful to Bill Gasarch for advice on the exposition and also to Pankaj Rohatgi for helpful discussions.

# References

[1] E. Allender and L. A. Hemachandra. Lower bounds for the low hierarchy. In *Proceedings of the 16th ICALP*, pages 31–45. Springer-Verlag, 1989. Volume 372 of *Lecture Notes in Computer Science*.

[2] A. Amir, R. Beigel, and W. I. Gasarch. Some connections between bounded query classes and non-uniform complexity. In *Proceedings of the 5th Annual Conference on Structure in Complexity Theory*, pages 232–243. IEEE Computer Society Press, July 1990.

[3] A. Amir and W. I. Gasarch. Polynomial terse sets. *Information and Computation*, 77:37–56, Apr. 1988.

[4] J. L. Balcázar, R. V. Book, and U. Schöning. Sparse sets, lowness and highness. *SIAM J. Comput.*, 15(3):739–747, Aug. 1986.

[5] R. Beigel. Bounded queries to SAT and the Boolean hierarchy. *Theoretical Computer Science*, 84(2):199–223, July 1991.

[6] R. Beigel, N. Reingold, and D. Spielman. PP is closed under intersection. To appear in *Journal of Computer and System Sciences*. An earlier version appeared in *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 1–9. ACM Press, May 1991.

[7] A. Bertoni, D. Bruschi, D. Joseph, M. Sitharam, and P. Young. Generalized Boolean hierarchies and Boolean hierarchies over RP. In *Proceedings of the 7th Conference on Fundamentals of Computation Theory*, pages 35–46. Springer-Verlag, 1989. Volume 380 of *Lecture Notes in Computer Science*.

[8] J. Cai. *On Some Most Probable Separations of Complexity Classes*. PhD thesis, Cornell University, Ithaca, NY, 1986.

[9] J. Cai, T. Gundermann, J. Hartmanis, L. A. Hemachandra, V. Sewelson, K. W. Wagner, and G. Wechsung. The Boolean hierarchy I: structural properties. *SIAM J. Comput.*, 17(6):1232–1252, Dec. 1988.

[10] J. Cai and L. A. Hemachandra. The Boolean hierarchy: Hardware over NP. In *Structure in Complexity Theory*, pages 105–124. Springer-Verlag, June 1986. Volume 223 of *Lecture Notes in Computer Science*.

[11] R. Chang and J. Kadin. The Boolean hierarchy and the polynomial hierarchy: a closer connection. In *Proceedings of the 5th Annual Conference on Structure in Complexity Theory*, pages 169–178. IEEE Computer Society Press, July 1990.

[12] R. Chang and J. Kadin. On computing Boolean connectives of characteristic functions. Technical Report TR 90-1118, Cornell Department of Computer Science, May 1990. To appear in *Mathematical Systems Theory*.

[13] L. Fortnow and N. Reingold. PP is closed under truth-table reductions. In *Proceedings of the 6th Annual Conference on Structure in Complexity Theory*, pages 13–15. IEEE Computer Society Press, June 1991.

[14] F. Green. On the power of deterministic reductions to $C_{=}P$. To appear in *Mathematical Systems Theory*.

[15] T. Gundermann, N. Nasser, and G. Wechsung. A survey of counting classes. In *Proceedings of the 5th Annual Conference on Structure in Complexity Theory*, pages 140–153. IEEE Computer Society Press, July 1990.

[16] J. Kadin. The polynomial time hierarchy collapses if the Boolean hierarchy collapses. *SIAM J. Comput.*, 17(6):1263–1282, Dec. 1988.

[17] J. Köbler, U. Schöning, and K. W. Wagner. The difference and truth-table hierarchies for NP. *RAIRO Theoretical Informatics and Applications*, 21:419–435, 1987.

[18] R. E. Ladner, N. A. Lynch, and A. L. Selman. A comparison of polynomial time reducibilities. *Theoretical Comput. Sci.*, 1:103–123, 1975.

[19] M. Ogiwara. On the computational power of exact counting. Unpublished manuscript, Apr. 1990.

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

[21] M. Sitharam. *Generalized Bounded Query Hierarchies*. PhD thesis, University of Wisconsin—Madison, 1990.

[22] J. Torán. Complexity classes defined by counting quantifiers. *Journal of the ACM*, 38(3):753–774, July 1991.

[23] K. W. Wagner. The complexity of combinatorial problems with succinct input representation. *Acta Inf.*, 23:325–356, 1986.

[24] K. Wagner. Bounded query computations. In *Proceedings of the 3rd Structure in Complexity Theory Conference*, pages 260–277, June 1988.

[25] K. W. Wagner. Number-of-query hierarchies. TR 4, Universitat Augsburg, Institut fur Mathematik, 1989.

[26] K. W. Wagner. Bounded query classes. *SIAM J. Comput.*, 19(5):833–846, Oct. 1990.

[27] G. Wechsung and K. Wagner. On the Boolean closure of NP. In *Proceedings of the 1985 International Conference on Fundamentals of Computation Theory*, pages 485–493. Springer-Verlag, 1985. Volume 199 of *Lecture Notes in Computer Science*.