

 Open access • Journal Article • DOI:10.1007/BF01371729

A Relationship between Difference Hierarchies and Relativized Polynomial Hierarchies. — [Source link](#)

Richard Beigel, Richard Chang, Mitsunori Ogiwara

Institutions: Yale University, Cornell University, University of Electro-Communications

Published on: 01 Jan 1991 - Theory of Computing Systems V Mathematical Systems Theory (Cornell University)

Topics: Difference hierarchy and Polynomial (hyperelastic model)

Related papers:

- [ERRATUM: the polynomial time hierarchy collapses if the boolean hierarchy collapses](#)
- [The Boolean hierarchy I: structural properties](#)
- [The Boolean Hierarchy and the Polynomial Hierarchy: A Closer Connection](#)
- [The Boolean hierarchy II: applications](#)
- [The difference and truth-table hierarchies for NP](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/a-relationship-between-difference-hierarchies-and-57wm3msl3d>

**A Relationship between Difference
Hierarchies and Relativized
Polynomial Hierarchies**

Richard Beigel*
Richard Chang**
Mitsunori Ogiwara***

TR 91-1184
January 1991

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*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-880849 and CCR-8958528.

**Dept. of Computer Science, Upson Hall, Cornell University, Ithaca, NY 14853. Supported in part by NSF Research Grant CCR-22053.

***Dept. of Information Science, Tokyo Institute of Technology, Ookayama, Meguro-ku, Tokyo 152, Japan.

A Relationship between Difference Hierarchies and Relativized Polynomial Hierarchies

Richard Beigel*
Yale University

Richard Chang†
Cornell University

Mitsunori Ogiwara‡
Tokyo Institute of Technology

January, 1991

Abstract

Chang and Kadin have shown that if the difference hierarchy over NP collapses to level k , then the polynomial hierarchy (PH) is equal to the k th level of the difference hierarchy over Σ_2^P . We simplify their proof and obtain a slightly stronger conclusion: If the difference hierarchy over NP collapses to level k , then $\text{PH} = \left(\text{P}_{(k-1)\text{-tt}}^{\text{NP}}\right)^{\text{NP}}$. 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_{conj}^P - and \leq_m^{NP} -reductions, if the difference hierarchy over C collapses to level k , then $\text{PH}^C = \left(\text{P}_{(k-1)\text{-tt}}^{\text{NP}}\right)^C$. Then we show that the exact counting class C=P is closed under \leq_{disj}^P - and $\leq_m^{\text{co-NP}}$ -reductions. Consequently, if the difference hierarchy over C=P collapses to level k then PH^{PP} is equal to $\left(\text{P}_{(k-1)\text{-tt}}^{\text{NP}}\right)^{\text{PP}}$. 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 $\text{P}_{k\text{-tt}}^{\text{NP}}$: the restricted relativization $\text{P}_{k\text{-tt}}^{\text{NP}^C}$, and the full relativization $\left(\text{P}_{k\text{-tt}}^{\text{NP}}\right)^C$. If C is NP-hard, then we show that the two relativizations are different unless 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-23053.

‡Dept. of Information Science, Tokyo Institute of Technology, Ookayama, Meguro-ku, Tokyo 152, Japan.

1. Introduction

Numerous researchers [3,5,8,9,10,11,14,15,22,21,23] have studied the Boolean hierarchy over NP. This hierarchy intertwines the query hierarchies over NP, and is identical to the Hausdorff 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 [14] 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 Δ_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 [21], who showed that if the difference hierarchy over NP collapses to level k then the polynomial hierarchy is equal to $P_{O(k)\text{-tt}}^{\Sigma_2^P}$, 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 $\text{DIFF}_k(\Sigma_2^P)$, where $\text{DIFF}_k(C)$ 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 PH^{PP} , $\text{P}\#\text{P}$, or Wagner's [20] counting hierarchy.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 PH^{PP} collapse?
- $\text{P}\#\text{P} = \text{P}\#\text{P}^{[1]}$?
- Does the counting hierarchy collapse?
- $\text{PSPACE} = \text{P}\#\text{P}^{[1]}$?

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 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_{conj}^P - and \leq_m^{NP} -reductions, we show that

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

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_{disj}^P - and $\leq_m^{\text{co-NP}}$ -reductions. Our main results extend Chang and Kadin's result; our proof is also simpler.

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

While the class PP is closed under \leq_{tt}^p -reductions [12], it does not seem likely that PP is closed under \leq_m^{NP} reductions, for then we would have $\text{PARITYP} \subseteq \text{PP}$, which does not relativize [19]. 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 C=P ,² which is closely related to PP, is closed under \leq_{disj}^p - and $\leq_m^{\text{co-NP}}$ reductions, as we show (similar closure properties were obtained independently by Gundermann, Nasser, and Wechsung [13]). Applying our main result and a theorem of Toran, we find that the difference hierarchy over C=P does not collapse unless the polynomial hierarchy relative to PP collapses. This structural consequence complements a result of Gundermann, Nasser, and Wechsung [13], who constructed oracles that make the difference hierarchy over C=P proper.

2. Preliminaries

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

Definition 1.

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

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

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

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

Nondeterministic many-one reductions were defined by Ladner, Lynch, and Selman [16].

² 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 $A \leq_m^{\text{NP}} B$) 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 ΣC to denote the closure of C under \leq_m^{NP} reductions.

The “mind-change” technique was developed by Wagner and Wechsung [23] and Beigel [5] in order to prove absolute results about the nonadaptive query hierarchy over NP. Chang and Kadin applied a similar technique to the difference hierarchy over Σ_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.

Let A join B denote the join of A and B , that is $\{0x : x \in A\} \cup \{1x : x \in B\}$.

Lemma 5 (Mind-change). *Fix a natural number k . Let \prec be a polynomial-time computable partial order, with minimum element Λ . Let A and B be two-place predicates. Suppose that*

- (a) *there exists a polynomial p such that $B(x, h) \Rightarrow |h| \leq p(|x|)$, and*
- (b) *for all x , $B(x, \Lambda) = \text{true}$, and*
- (c) $\neg(\exists x)(\exists h_1 \prec \dots \prec h_{k+1})[B(x, h_1) \wedge \dots \wedge B(x, h_{k+1})]$.

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 $Q \in \left(\text{P}_{(k-1)\text{-tt}}^{\text{NP}}\right)^{B \text{ join } A}$.

Proof: Define

$$Q_m(x) \equiv (\exists h_1 \prec \dots \prec h_m)[(B(x, h_1) \wedge \dots \wedge B(x, h_m)) \wedge (A(x, \Lambda) \neq A(x, h_1) \neq \dots \neq A(x, h_m))].$$

Let M be the largest m such that $Q_m(x) = \text{true}$. If $h_M \prec h$ then $B(x, h) = \text{false}$ 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), $M \leq k - 1$. Thus,

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

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

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 [14], 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 $\text{co-}C$ predicates for all arguments with length $\leq n$. While authors working directly with Boolean hierarchies have had to consider separate cases for odd k and for even k , we consider only one case because we work with difference hierarchies. Like Wagner [21], we incorporate one or more hard sequences directly into a polynomial-length advice string, thus avoiding the need to construct sparse oracles as in [14] 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 PH^C . Recall that a single hard sequence allows us to reduce C predicates to $\text{co-}C$ predicates for all arguments with length $\leq n$. Then C ’s closure properties allow us (1) to reduce any NP^C predicate to a $\text{co-}C$ predicate for all arguments with length $\leq n$. Consequently, a single hard sequence allows us (2) to reduce any NP^{NP^C} predicate to an NP^C predicate for all arguments with length $\leq n$. We perform (2) and then (1) in order to reduce any NP^{NP^C} predicate to a $\text{co-}C$ 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 $\text{PH}^C \subseteq \left(\text{P}_{(2k-2)\text{-tt}}^{\text{NP}}\right)^C$. 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 P^{NP^C} predicate to a P^C predicate; thus we show that $\text{P}^{\text{NP}^C} \subseteq \left(\text{P}_{(k-1)\text{-tt}}^{\text{NP}}\right)^C$. Combined with (*) this implies that $\text{PH}^C \subseteq \left(\text{P}_{(k-1)\text{-tt}}^{\text{NP}}\right)^C$.

Theorem 6. *Assume*

- $\text{co-}C = \Sigma\text{co-}C$,
- C has \leq_m^p -complete sets,
- $\text{co-}C$ is closed under \leq_{conj}^p -reductions, and
- $\text{DIFF}_k(C) = \text{co-DIFF}_k(C)$.

Then $\text{PH}^C = \left(\text{P}_{(k-1)\text{-tt}}^{\text{NP}}\right)^C$.

Proof: Let L be \leq_m^p -complete for C . Define

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

where ‘ $\langle \cdot, \cdot \rangle$ ’, ‘ $\langle \cdot, \cdot \rangle$ ’, and ‘ \cdot ’ are new characters. Then it is clear that L_k is \leq_m^p -complete for $\text{DIFF}_k(C)$. 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

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

Fix a positive integer n ; we will rely on the equation above only for $|x| \leq kn + 3(k-1)$. 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 $\Sigma\text{co-C} = \text{co-C}$. If there exists a *k-hard* string χ_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

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

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

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

Iterating that process, we define $\chi_{k-1}, \dots, \chi_{j+1}$ — stopping when $j = 1$ or when there exists no *j-hard* string of length n — and we define the corresponding functions h_{k-2}, \dots, h_j . (Since there may be several ways to choose the hard strings, we should write $h_{i\chi_k \dots \chi_{i+1}}$ 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 $\chi_k, \dots, \chi_{i+1}$).

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 $\chi_k, \dots, \chi_{j+1}$ be a maximal sequence of hard strings of length n . Then $x \in L$ iff x is *j-easy*. Thus, using the strings $\chi_k, \dots, \chi_{j+1}$ as advice, we can effectively reduce L to a *co-C* predicate for arguments of length $\leq n$.

We complete the proof by proving two things:

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

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

Proof of claim 1: Let Q be any P^{NP^C} predicate. Then $Q \in P^R$ for some $R \in \text{NP}^L$. Assume that R is reducible to L via a nondeterministic Turing machine M running in time $r(n)$. Using a maximal sequence $\chi_k, \dots, \chi_{j+1}$ of hard strings of length $r(n)$, we can reduce L to a co- C predicate. Since co- C is closed under \leq_{conj}^P -reductions, each computation of M can then be reduced to a single co- C predicate. Since $\Sigma\text{co-}C = \text{co-}C$, we reduce R to a co- C predicate, and finally reduce Q to a P^C predicate. Hence there is a P^C predicate A such that for all $x \in \Sigma^n$,

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

Then $Q(x)$ is true iff there exists a maximal sequence $\chi_k, \dots, \chi_{j+1}$ of hard strings of length $r(n)$ such that $A(x, \langle \chi_k, \dots, \chi_{j+1} \rangle)$ is true. Let $B(x, \langle \chi_k, \dots, \chi_{j+1} \rangle)$ be true iff χ_k is a k -hard string of length $r(n)$, \dots , and χ_{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 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 $\chi_k, \dots, \chi_{j+1}$ such that both $B(x, \langle \chi_k, \dots, \chi_{j+1} \rangle)$ and $A(x, \langle \chi_k, \dots, \chi_{j+1} \rangle)$ are true, and
- there does not exist a sequence χ'_k, \dots, χ'_j such that $B(x, \langle \chi_k, \dots, \chi_j \rangle)$ 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 5 Q is a $\left(P_{(k-1)\text{-tt}}^{\text{NP}}\right)^C$ predicate. ■ (claim 1)

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

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

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

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

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 5, Q is in $\left(P_{(2k-2)\text{-tt}}^{\text{NP}}\right)^C$. ■ (claim 2)

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

If we drop the restriction that $\text{co-}C$ be closed under \leq_{conj}^P -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 7. *Assume*

- $\text{co-}C = \Sigma\text{co-}C$,
- C has \leq_m^P -complete sets,
- $\text{DIFF}_k(C) = \text{co-DIFF}_k(C)$.

Then $\text{PH}^C = \left(\text{P}_{(2k-2)\text{-tt}}^{\text{NP}}\right)^C$.

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

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

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

Proof of claim 1: Using one maximal sequence of hard strings we can reduce a $\text{co-}C$ predicate to a C predicate. Since $\text{co-}C$ is closed under \leq_m^{NP} -reductions, C is closed under $\leq_m^{\text{co-NP}}$ -reductions, and *a fortiori* closed under \leq_{conj}^P reductions. Thus verifying a path of an NP computation can be reduced to a single C predicate. Using a second maximal sequence of hard strings, we can reduce that C predicate to a $\text{co-}C$ predicate. Since $\Sigma\text{co-}C = \text{co-}C$, we thus reduce an NP^C predicate to a $\text{co-}C$ predicate. Applying Lemma 5, we have $\text{P}^{\text{NP}^C} \subseteq \left(\text{P}_{(2k-2)\text{-tt}}^{\text{NP}}\right)^C$.

Proof of claim 2: Using a total of four maximal sequences of hard strings, we reduce an NP^{NP^C} predicate to a $\text{co-}C$ predicate. Applying Lemma 5, we have $\text{P}^{\text{NP}^C} \subseteq \left(\text{P}_{(4k-4)\text{-tt}}^{\text{NP}}\right)^C$. ■

Corollary 8. *Assume*

- $C = \Sigma C$,
- C has \leq_m^P -complete sets,
- C is closed under intersection, and
- $\text{DIFF}_k(C) = \text{co-DIFF}_k(C)$.

Then

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

Proof: We note that for any C

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

Thus

$$\text{DIFF}_k(C) = \text{co-DIFF}_k(C) \iff \text{DIFF}_k(\text{co-}C) = \text{co-DIFF}_k(\text{co-}C).$$

Thus the corollary is equivalent to Theorem 6. ■

Thus we extend the results of Kadin [14], Wagner [21], and Chang and Kadin [11]:

Corollary 9. *If $\text{DIFF}_k(\text{NP}) = \text{co-DIFF}_k(\text{NP})$ then $\text{PH} = \left(\text{P}_{(k-1)\text{-tt}}^{\text{NP}}\right)^{\text{NP}}$.*

Proof: Assume that $\text{DIFF}_k(\text{NP}) = \text{co-DIFF}_k(\text{NP})$. Since $\text{NP} = \Sigma\text{NP}$ and NP has \leq_m^p -complete sets, Corollary 8 implies $\text{PH}^{\text{NP}} = \left(\text{P}_{(k-1)\text{-tt}}^{\text{NP}}\right)^{\text{NP}}$, so $\text{PH} = \left(\text{P}_{(k-1)\text{-tt}}^{\text{NP}}\right)^{\text{NP}}$. ■

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

Corollary 10. *Assume*

- $C = \Sigma C$,
- C has \leq_m^p -complete sets,
- C is closed under intersection, and
- $\text{DIFF}_k(C) = \text{co-DIFF}_k(C)$.

Then all C -complete sets are in $\widehat{\text{EL}}_4$.

Proof: By Corollary 8, all C -complete sets L , satisfy

$$(\Delta_4^p)^L = (\Delta_3^p)^L.$$

Therefore

$$(\Delta_4^p)^L = (\Delta_3^p)^{\text{NP},L},$$

which is the definition of $\widehat{\text{EL}}_4$ [1]. ■

4. Closure Properties of $C=P$

The class $C=P$ was defined by Wagner [20]. A language L belongs to $C=P$ if and only if there exists a polynomial-time bounded nondeterministic Turing machine N such that $x \in L$ iff the number of accepting paths of machine N on input x is equal to the number of rejecting paths.

A \leq_{disj}^P -reduction is a polynomial-time truth-table reduction in which the truth-table predicate is logical-or. A \leq_{conj}^P -reduction is a polynomial-time truth-table reduction in which the truth-table predicate is logical-and. A co-NP machine M many-one reduces A to B if, for all x ,

$$x \in A \iff (\forall \rho)[f_\rho(x) \in B],$$

where $f_\rho(x)$ denotes the output of M 's computation ρ on input x . We show that $C=P$ is closed under \leq_{disj}^P -reductions and $\leq_m^{\text{co-NP}}$ reductions. These results have appeared in [17]. Similar closure properties were obtained independently by Gundermann, Nasser, and Wechsung [13].

Theorem 11.

- (a) $C=P$ is closed under \leq_{disj}^P -reductions.
- (b) $C=P$ is closed under $\leq_m^{\text{co-NP}}$ -reductions.

Proof: This follows from the low-degree polynomial techniques of Beigel, Reingold, and Spielman [6]. Note that

- (a) $x_1 = 0 \vee \dots \vee x_k = 0 \iff x_1 \cdots x_k = 0$.
- (b) $x_1 = 0 \wedge \dots \wedge x_k = 0 \iff x_1^2 + \dots + x_k^2 = 0$.

■

Corollary 12. *If $\text{DIFF}_k(C=P) = \text{co-DIFF}_k(C=P)$ then $\text{PH}^{\text{PP}} = \left(\text{P}_{(k-1)\text{-tt}}^{\text{NP}}\right)^{\text{PP}}$.*

Proof: Assume that $\text{DIFF}_k(C=P) = \text{co-DIFF}_k(C=P)$. Since $\text{co-}C=P = \Sigma \text{co-}C=P$ and $\text{co-}C=P$ is closed under \leq_{conj}^P -reductions, Theorem 6 implies that $\text{PH}^{C=P} = \left(\text{P}_{(k-1)\text{-tt}}^{\text{NP}}\right)^{C=P}$. By a result of Toran [19], $\text{NP}^{\text{PP}} = \text{NP}^{C=P}$ (because one can guess the exact threshold); therefore

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

■

5. Relativizing Bounded Query Classes

It is natural to ask whether our Corollary 9 is really stronger than Chang and Kadin's theorem [11], i.e., does

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

It is clear that

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

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

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

so we would really like to know the answer to:

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

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

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

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 $\text{NP}^{(\text{NP}^C)}$, and it does not matter that the base NP machine does not have direct access to the oracle C , because it can ask the 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 $\text{P}_{k\text{-tt}}^{(\text{NP}^C)}$, which is the class of languages recognized by polynomial time Turing machines which are allowed k parallel queries to the NP^C oracle. In the second approach, the polynomial time base machine can ask k parallel queries to the NP^C oracle and polynomially many serial queries to the C oracle. This is a full relativization. We denote this second class as $\left(\text{P}_{k\text{-tt}}^{\text{NP}}\right)^C$. In what follows, we show that if C is sufficiently hard, then the two relativizations are different unless 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 [14,21,11]. The first proof modifies a technique from [2], and is relatively simple, but it only collapses PH^C to $(\Sigma_3^p)^C$. The second proof is more difficult, combining two hard/easy-formulas arguments; it collapses PH^C to $\left(\text{P}_{k\text{-tt}}^{\text{NP}}\right)^{\text{NP}^C}$.

Definition 13. Let X and Y be any two languages, we define the set

$$X \Delta Y = \{ \langle x, y \rangle : (x \in X \text{ and } y \notin Y) \text{ or } (x \notin X \text{ and } y \in Y) \}$$

Alternatively, $\langle x, y \rangle \in X \Delta Y$ iff $x \in X \iff y \notin Y$.

Proposition 14. Let C be any class such that $\text{NP} \subseteq \text{P}^C$. If $\text{P}_{k\text{-tt}}^{\text{NP}^C} = \left(\text{P}_{k\text{-tt}}^{\text{NP}}\right)^C$ then $\text{PH}^C = \left(\Sigma_3^P\right)^C$.

Proof: Let L_P , L_{PC} and L_k be \leq_m^p -complete for P , P^C , and $\text{DIFF}_k(\text{NP}^C)$, respectively. By careful analysis of the mind-change proof in [5], one can show that $L_{\text{PC}} \Delta L_k$ is \leq_m^p -complete for $\left(\text{P}_{k\text{-tt}}^{\text{NP}}\right)^C$ and $L_P \Delta L_k$ is \leq_m^p -complete for $\text{P}_{k\text{-tt}}^{\text{NP}^C}$. Thus $\text{P}_{k\text{-tt}}^{\text{NP}^C} = \left(\text{P}_{k\text{-tt}}^{\text{NP}}\right)^C$ if and only if

$$L_{\text{PC}} \Delta L_k \leq_m^p L_P \Delta L_k.$$

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 $\text{DIFF}_k(\text{NP}^C) \subseteq \text{co-DIFF}_k(\text{NP}^C)/\text{poly}$, so $\text{NP}^C \subseteq \text{co-NP}^C/\text{poly}$, so $\text{PH}^C \subseteq \left(\Sigma_3^P\right)^C$.

Let $|S|$ denote the cardinality of the set S . Let $(\exists^{\geq \alpha} y \in S)$ denote “for at least α 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 $(\exists x \in \{0, 1\}^{\leq m})(\exists^{\geq \frac{1}{4}|S|} y \in S)[x \in L_{\text{PC}} \iff x' \notin L_P]$: Choose such an x , and incorporate x into the advice for length m . Let $S = S - \{y : x \in L_{\text{PC}} \iff x' \notin L_P\}$. If $S = \emptyset$ then exit the loop.

Case 2 $(\forall x \in \{0, 1\}^{\leq m})(\exists^{\geq \frac{3}{4}|S|} y \in S)[x \in L_{\text{PC}} \iff x' \in L_P]$: 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_{PC} to L_P : Input x ; choose a random $y \in S$; compute x' ; then $x \in L_{\text{PC}}$ iff $x' \in L_P$. The nonuniform randomness can be simulated by incorporating a polynomial number of elements of S into the advice, as in Schöning’s proof that $\text{BPP} \in \text{P}/\text{poly}$ [18]. Exit the loop.

End loop.

End construction.

If case 2 is ever reached then the construction produces advice sufficient to reduce L_{PC} to L_P for length $\leq m$. Since $\text{NP} \subseteq \text{P}^C$, 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 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 $x \in L_{PC} \iff x' \notin L_P$; then $y \in L_k$ iff $y' \notin L_k$.

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

Now we prove the stronger result.

Theorem 15. *Let C be any class such that $\text{NP} \subseteq \text{co-NP}^C$. If $\text{P}_{k\text{-tt}}^{\text{NP}^C} = (\text{P}_{k\text{-tt}}^{\text{NP}})^C$ then $\text{PH}^C = (\text{P}_{k\text{-tt}}^{\text{NP}})^{\text{NP}^C}$.*

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

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

(Technically, L_k is not complete for $\text{DIFF}_k(\text{NP}^C)$, but rather $\overline{L_k}$ is complete for $\text{DIFF}_k(\text{co-NP}^C)$.) By careful analysis of the mind-change proof in [5], one can show that $L_{PC} \triangle L_k$ is \leq_m^p -complete for $(\text{P}_{k\text{-tt}}^{\text{NP}})^C$ and $L_P \triangle L_k$ is \leq_m^p -complete for $\text{P}_{k\text{-tt}}^{\text{NP}^C}$. Thus $\text{P}_{k\text{-tt}}^{\text{NP}^C} = (\text{P}_{k\text{-tt}}^{\text{NP}})^C$ if and only if

$$L_{PC} \triangle L_k \leq_m^p L_P \triangle L_k.$$

Fix a polynomial-time computable function h that performs that reduction. Fix m . We will construct advice that either lets us reduce L_{PC} to L_P for all strings of length $\leq m$ or else lets us reduce L_{NP^C} to $\overline{L_{\text{NP}^C}}$ 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 $\vec{\chi} = \langle \chi_1, \dots, \chi_j \rangle$ 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. $|\chi_i| \leq m$.
2. $\forall \vec{p} = \langle p_1, \dots, p_{k-i} \rangle \in \{0, 1\}^{\leq m \times (k-i)}, \forall u \in \{0, 1\}^{\leq m},$

$$(u \in L_{PC} \iff v \notin L_P) \Rightarrow y_i \notin L_{\text{NP}^C},$$

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

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_{PC} to L_P .

Claim 4: If $\vec{\chi}$ is a maximal-length hard sequence and $|\vec{\chi}| < k$, then $\vec{\chi}$ induces a nondeterministic reduction from $\overline{L_{\text{NP}^C}}$ to L_{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_{NP^c} to $\overline{L_{\text{NP}^c}}$ (recall that $\text{NP} \subseteq \text{co-NP}^c$). We order hard sequences by length, so a chain of hard sequences contains at most $k + 1$ elements. Applying Lemma 5, we collapse PH^c to $(\text{P}_{k\text{-tt}}^{\text{NP}})^{\text{NP}^c}$.

Proof of claim 1: The empty sequence is a hard sequence. ■ (claim 1)

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

$$u \in L_{\text{PC}} \iff v \in L_{\text{P}}.$$

Thus the following algorithm reduces L_{PC} to L_{P} for strings of length $\leq m$: Input u ; let $(v, \vec{y}) = h(u, \vec{\chi})$; then $u \in L_{\text{PC}} \iff v \in L_{\text{P}}$. ■ (claim 2)

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

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

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

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

$$\langle u, \vec{\chi}, \vec{p} \rangle \in L_{\text{PC}} \Delta L_k \iff \langle v, \vec{y}, \vec{q} \rangle \in L_{\text{P}} \Delta L_k.$$

Suppose that $u \in L_{\text{PC}} \iff v \notin L_{\text{P}}$. 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, $\chi_i \notin L_{\text{NP}^c}$ and $y_i \notin L_{\text{NP}^c}$, 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 $s \notin L_{\text{NP}^c}$. Since $\vec{\chi}$ is maximal, $\langle \chi_1, \dots, \chi_i, s \rangle$ does not satisfy condition 2 in the definition of hard sequences — which is exactly what we need.

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

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

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

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

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

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

Thus, using χ as advice, an NP^C algorithm can m-reduce $\overline{L_{\text{NP}^C}}$ to L_{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 $\langle v, y_1, \dots, y_i, t, \vec{q} \rangle = h(u, \vec{\chi}, \vec{p})$; if $u \in L_{\text{PC}} \iff v \notin L_{\text{P}}$ and $t \in L_{\text{NP}^C}$ 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 NP^C to NP for strings of length $\leq m$. By assumption, $\text{NP} \subseteq \text{co-NP}^C$, so a hard sequence of length k induces a reduction from NP^C to co-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-NP^C to NP^C for strings of length $\leq m$. Thus, a maximal hard sequence of any length induces a deterministic reduction from co-NP^C to NP^C for strings of length $\leq m$.

Therefore, every $\text{P}^{\Sigma_2^p, C}$ languages is recognized by a P^{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 5

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

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

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

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

Corollary 16. *If $\left(\text{P}_{k-tt}^{\text{NP}} \right)^{\text{NP}} = \text{P}_{k-tt}^{\text{NP}^{\text{NP}}}$, then $\text{PH} \left(\text{P}_{k-tt}^{\text{NP}} \right)^{\text{NP}^{\text{NP}}}$.*

Corollary 17. *If $\left(\text{P}_{k-tt}^{\text{NP}} \right)^{\text{PP}} = \text{P}_{k-tt}^{\text{NP}^{\text{PP}}}$, then $\text{PH}^{\text{PP}} = \left(\text{P}_{k-tt}^{\text{NP}} \right)^{\text{NP}^{\text{PP}}}$.*

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*, 1989.
- [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 Comput. Sci.* To appear.
- [6] R. Beigel, N. Reingold, and D. Spielman. PP is closed under intersection. YALEU/DCS/TR 803, Yale University, Dept. of Computer Science, 1990.
- [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*. Springer-Verlag, 1989. LNCS.
- [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. Lecture Notes in Computer Science 223.
- [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] L. Fortnow and N. Reingold. PP is closed under truth-table reductions. Technical Report 30, Dept. of Computer Science, U. Chicago, 1990.
- [13] 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.

- [14] J. Kadin. The polynomial time hierarchy collapses if the Boolean hierarchy collapses. *SIAM J. Comput.*, 17(6):1263–1282, Dec. 1988.
- [15] 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.
- [16] R. E. Ladner, N. A. Lynch, and A. L. Selman. A comparison of polynomial time reducibilities. *Theoretical Comput. Sci.*, 1:103–123, 1975.
- [17] M. Ogiwara. On the computational power of exact counting. Unpublished manuscript, Apr. 1990.
- [18] U. Schöning. *Complexity and Structure*. Springer-Verlag, Berlin, 1985.
- [19] J. Torán. *Structural Properties of The Counting Hierarchies*. PhD thesis, Facultat d'Informàtica de Barcelona, 1988.
- [20] K. W. Wagner. The complexity of combinatorial problems with succinct input representation. *Acta Inf.*, 23:325–356, 1986.
- [21] K. W. Wagner. Number-of-query hierarchies. Technical Report 4, Universitat Augsburg, Institut für Mathematik, 1989.
- [22] K. W. Wagner. Bounded query classes. *SIAM J. Comput.*, 19(5):833–846, Oct. 1990.
- [23] 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*.