Journal of Universal Computer Science, vol. 4, no. 6 (1998), 574-588
submitted: 8/8/97, accepted: 6/5/98, appeared: 28/6/98 © Springer Pub. Co.

Query Order and the Polynomial Hierarchy

Edith Hemaspaandra
(Department of Mathematics, Le Moyne College, Syracuse, USA
edith@bamboo.lemoyne.edu)

Lane A. Hemaspaandra
(Department of Computer Science, University of Rochester, USA
lane@cs.rochester.edu)

Harald Hempel
(Institut für Informatik, Friedrich-Schiller-Universität Jena, Germany
hempel@informatik.uni-jena.de)

Abstract: Hemaspaandra, Hempel, and Wechsung [HHW] initiated the field of query order, which studies the ways in which computational power is affected by the order in which information sources are accessed. The present paper studies, for the first time, query order as it applies to the levels of the polynomial hierarchy. $\mathrm{P}^{\mathcal{C}:\mathcal{D}}$ denotes the class of languages computable by a polynomial-time machine that is allowed one query to $\mathcal{C}$ followed by one query to $\mathcal{D}$ [HHW]. We prove that the levels of the polynomial hierarchy are order-oblivious: $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_k^p:\Sigma_j^p}$. Yet, we also show that these ordered query classes form new levels in the polynomial hierarchy unless the polynomial hierarchy collapses. We prove that all leaf language classes—and thus essentially all standard complexity classes—inherit all order-obliviousness results that hold for P.

Key Words: query order, polynomial hierarchy, ordered computation, commutative queries, complexity classes, downward separation

Category: F.1.3

1 Introduction

Query order was introduced by Hemaspaandra, Hempel, and Wechsung [HHW] in order to study whether the order in which information sources are accessed has any effect on the class of problems that can be solved. In the everyday world, the order in which we access information is crucial, and the work of Hemaspaandra, Hempel, and Wechsung [HHW] shows that this real-world intuition holds true in complexity theory when the information one is accessing is from the boolean hierarchy. In particular, let $\mathrm{P}^{\mathcal{C}:\mathcal{D}}$ denote the class of languages L such that, for some $C \in \mathcal{C}$ and some $D \in \mathcal{D}$, L is accepted by some P transducer M that on any input may make at most one query to C followed by at most one query to D. Hemaspaandra, Hempel, and Wechsung show that, unless the polynomial hierarchy collapses, query order always matters when $\mathcal{C}$ and $\mathcal{D}$ are nontrivial levels of the boolean hierarchy [CGH$^+$88], except in two cases. In particular they prove that, for $1 \leq j \leq k$, $\mathrm{P}^{\mathrm{BH}_j:\mathrm{BH}_k} = \mathrm{P}^{\mathrm{BH}_k:\mathrm{BH}_j}$ if

$$j = k \text{ or } (j \text{ is even and } k = j + 1),$$

and they prove that unless the polynomial hierarchy collapses these are the only cases (satisfying $1 \leq j \leq k$) for which $\mathrm{P}^{\mathrm{BH}_j:\mathrm{BH}_k} = \mathrm{P}^{\mathrm{BH}_k:\mathrm{BH}_j}$.

The goal of the present paper is to study query order in the *polynomial* hierarchy. Section 3 shows that, in sharp contrast with the case of the boolean hierarchy, *query order never matters in the polynomial hierarchy*: For any j and k, $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_k^p:\Sigma_j^p}$. We prove this by providing for "$\mathrm{P}^{\mathcal{C}:\mathcal{D}} = \mathrm{P}^{\mathcal{D}:\mathcal{C}}$" a sufficient condition, which also has applications in other settings.

Of course, if for $j \leq k$, $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_k^p[1]}$, then our $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_k^p:\Sigma_j^p}$ theorem would be trivial. Here, as is standard, $\mathrm{P}^{\Sigma_k^p[1]}$ denotes the class of languages that are computable via polynomial-time 1-Turing reductions to Σ_k^p [LLS75]. In fact, the statement $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_k^p[1]}$, for $j < k$, might on casual consideration seem plausible, as certainly a Σ_k^p oracle can simulate the Σ_j^p query (when $j < k$) of $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p}$, can compute the answer to it, and then can based on the answer determine the Σ_k^p query of $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p}$ and can simulate it. (Footnote 1 explains why this argument fails to establish $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_k^p[1]}$.) Nonetheless, we show that, unless the polynomial hierarchy collapses, $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_\ell^p:\Sigma_m^p}$ only if $\{j,k\} = \{\ell,m\}$.

In Section 4, we show that *all* query order exchanges that hold for $\mathrm{P}^{\mathcal{C}:\mathcal{D}}$—not just all those we prove but rather all that are true—are automatically inherited by all leaf language classes, and thus by essentially all standard complexity classes. This shows that *these classes allow at least as many query order exchanges as* P *does*. We also note that some of these classes—in particular NP—allow (unless the polynomial hierarchy collapses) more order exchanges than P does.

2 Preliminaries

For standard notions not defined here, we refer the reader to any computational complexity textbook, e.g., [BC93, BDG95, Pap94].

We say a set is trivial if it is $\emptyset$ or Σ^*, and otherwise we say it is nontrivial. A complexity class is any collection of subsets of Σ^*. For each complexity class $\mathcal{C}$, let co$\mathcal{C}$ denote $\{L \mid \overline{L} \in \mathcal{C}\}$. The polynomial hierarchy is defined as follows: $\Sigma_0^p = \Pi_0^p = \Delta_0^p = \Delta_1^p = \mathrm{P}$ and, for each $i > 0$, $\Sigma_i^p = \mathrm{NP}^{\Sigma_{i-1}^p}$, $\Pi_i^p = \mathrm{co}\Sigma_i^p$, and $\Delta_i^p = \mathrm{P}^{\Sigma_{i-1}^p}$. Let $A \oplus B$ denote the disjoint union of the sets A and B, i.e., $A \oplus B = \{x0 \mid x \in A\} \cup \{x1 \mid x \in B\}$, and let $A \times B$ denote the Cartesian product of the sets A and B, i.e., $A \times B = \{\langle x, y\rangle \mid x \in A \text{ and } y \in B\}$.

In this paper we use oracles to represent databases that are queried. This does not mean that this is a "relativized worlds" oracle construction paper. It is not. Rather we use relativization in much the same way that it is used to build the polynomial hierarchy, namely, relativization by full, natural classes.

We now present the definitions that will allow us to discuss the ways—order of access, amount of access, etc.—that databases (modeled as oracles) are accessed. We use DPTM as a shorthand for "deterministic polynomial-time (oracle) Turing machine." Without loss of generality, we assume that such machines are clocked with clocks that are independent of the oracle. $M^A(x)$ denotes the computation of DPTM M with oracle A on input x. On occasion, when the oracle involved is clear from context and we are focusing on the action of M, we may write $M(x)$ and omit the oracle.

Definition 1. Let $\mathcal{C}$ and $\mathcal{D}$ be complexity classes.

1. [HHW] Let $M^{A:B}$ denote DPTM M restricted to making at most one query to oracle A followed by at most one query to oracle B.
$$\mathrm{P}^{\mathcal{C}:\mathcal{D}} = \{L \subseteq \Sigma^* \mid (\exists C \in \mathcal{C})(\exists D \in \mathcal{D})(\exists \text{ DPTM } M)[L = L(M^{C:D})]\}.$$
2. Let $M^{(A,B)}_{1,1\text{-tt}}$ denote DPTM M restricted to making simultaneously at most one query to oracle A and at most one query to oracle B.
$$\mathrm{P}^{(\mathcal{C},\mathcal{D})}_{1,1\text{-tt}} = \{L \subseteq \Sigma^* \mid (\exists C \in \mathcal{C})(\exists D \in \mathcal{D})(\exists \text{ DPTM } M)[L = L(M^{(C,D)}_{1,1\text{-tt}})]\}.$$
3. Let $M^{A,B}$ denote DPTM M restricted to making at most one query to oracle A and at most one query to oracle B, in arbitrary order. Similarly, let $M^{A[1],B[poly]}$ denote DPTM M making at most one query to oracle A and polynomially many queries to B, in arbitrary order (it is even legal for the query to A to be sandwiched between queries to B).
$$\mathrm{P}^{\mathcal{C},\mathcal{D}} = \{L \subseteq \Sigma^* \mid (\exists C \in \mathcal{C})(\exists D \in \mathcal{D})(\exists \text{ DPTM } M)[L = L(M^{C,D})]\}.$$
$$\mathrm{P}^{\mathcal{C}[1],\mathcal{D}[poly]} =$$
$$\{L \subseteq \Sigma^* \mid (\exists C \in \mathcal{C})(\exists D \in \mathcal{D})(\exists \text{ DPTM } M)[L = L(M^{C[1],D[poly]})]\}.$$

As has been noted by the authors elsewhere [HHH97b], part 2 of Definition 1 is somewhat related to work of Selivanov [Sel94]. Independently of [HHH97b], Klaus Wagner [Wag] has made similar observations in a more general form (namely, applying to more than two sets and to more abstract classes) regarding the relationship between Selivanov's classes and parallel-access classes. For completeness, we repeat here, as the present paragraph, some text from [HHH97b] that presents the basic facts known about the relationship between the classes of Selivanov (for the case of "$\triangle$"s of two sets; see Wagner [Wag] for the case of more than two sets) and the classes discussed in this paper. Selivanov studied refinements of the polynomial hierarchy. Among the classes he considered, those closest to the classes we study in this paper are his classes

$$\Sigma^p_i \triangle \Sigma^p_j = \{L \mid (\exists A \in \Sigma^p_i)(\exists B \in \Sigma^p_j)[L = A \triangle B]\},$$

where $A \triangle B = (A - B) \cup (B - A)$. Note, however, that his classes seem to be different from our classes. This can be immediately seen from the fact that all our classes are closed under complementation, but the main theorem of Selivanov ([Sel94], see also the discussion and strengthening in [HHH97c]) states that no class of the form $\Sigma^p_i \triangle \Sigma^p_j$, with $i > 0$ and $j > 0$, is closed under complementation unless the polynomial hierarchy collapses. Nonetheless, the class $\Sigma^p_i \triangle \Sigma^p_j$ is not too much weaker than $\mathrm{P}^{(\Sigma^p_i,\Sigma^p_j)}_{1,1\text{-tt}}$, as it is not hard to see (by easy manipulations if $i \neq j$, and from the work of Wagner [Wag90] and Köbler, Schöning, and Wagner [KSW87] for the $i = j$ case) that, for all i and j, it holds that $\{L \mid (\exists L' \in \Sigma^p_i \triangle \Sigma^p_j)[L \leq^p_{1\text{-tt}} L']\} = \mathrm{P}^{(\Sigma^p_i,\Sigma^p_j)}_{1,1\text{-tt}}$. Here, as is standard, $\leq^p_{1\text{-tt}}$ denotes polynomial-time 1-truth-table reducibility [LLS75].

Let $\mathcal{C}$ be a complexity class. In the literature, $\leq^p_m$ denotes many-one polynomial-time reducibility. Similarly, we write $A \leq^{p,\mathcal{C}[1]}_m B$ if and only if there is a (total) function $f \in \mathrm{FP}^{\mathcal{C}[1]}$ such that, for all x, $x \in A \iff f(x) \in B$.

3 Query Order in the Polynomial Hierarchy

3.1 Order Exchange in the Polynomial Hierarchy

We first state and prove a sufficient condition for order exchange. This condition will apply to a large number of classes.

Theorem 2. *If $\mathcal{C}$ and $\mathcal{D}$ are classes such that $\mathcal{C}$ is closed under disjoint union and $\mathcal{C}$ is closed downwards under $\leq_m^{p,\mathcal{D}[1]}$, then*

$$\mathrm{P}^{\mathcal{C}:\mathcal{D}} = \mathrm{P}^{\mathcal{D}:\mathcal{C}} = \mathrm{P}^{(\mathcal{C},\mathcal{D})}_{1,1\text{-tt}}.$$

Proposition 3 notes that for complexity classes that have complete sets, closure under disjoint union follows from downward closure under many-one reductions. For most standard classes $\mathcal{C}$ this proposition can be used, when applying various theorems of this section, to remove the condition that $\mathcal{C}$ be closed under disjoint union.

Proposition 3. *If $\mathcal{C}$ has $\leq_m^p$-complete sets and $\mathcal{C}$ is closed downwards under $\leq_m^p$-reductions, then $\mathcal{C}$ is closed under disjoint union.*

Before proving Theorem 2 we first prove some results that will be helpful in the proof. Also, Theorem 5 may apply even in some cases where Theorem 2's hypothesis does not hold.

Definition 4. We say $\mathcal{C}$ "ands" $(\mathcal{C},\mathcal{D})$ if $(\forall C \in \mathcal{C})(\forall D \in \mathcal{D})[C \times D \in \mathcal{C}]$.

Theorem 5. *If $\mathcal{C}$ is closed under disjoint union, $\mathcal{C}$ "ands" $(\mathcal{C},\mathcal{D})$, and $\mathcal{C}$ "ands" $(\mathcal{C},\mathrm{co}\mathcal{D})$, then $\mathrm{P}^{\mathcal{C}:\mathcal{D}} \subseteq \mathrm{P}^{\mathcal{D}:\mathcal{C}}$.*

Proof. Suppose $L \in \mathrm{P}^{\mathcal{C}:\mathcal{D}}$ and let DPTM M, $C \in \mathcal{C}$, and $D \in \mathcal{D}$ be such that $L = L(M^{C:D})$. Without loss of generality, let M always query each of C and D exactly once, regardless of the answer of the first query (that is, even given an incorrect answer to the first query, M will always ask a second query). We describe a DPTM N and a set C' such that $C' \in \mathcal{C}$ and $L = L(N^{D:C'})$. Let $C' = \big((C \times \overline{D}) \oplus (C \times D)\big) \oplus C$, i.e., $C' =$

$$\{\langle y_1, y_2\rangle 00 \,|\, y_1 \in C \text{ and } y_2 \notin D\} \cup \{\langle y_1, y_2\rangle 10 \,|\, y_1 \in C \text{ and } y_2 \in D\} \cup \{y1 \,|\, y \in C\}.$$

On input x, DPTM $N^{D:C'}$ works as follows:

1. It determines the first and the two potential second queries of $M(x)$. Denote the first query by q_0 and the two potential second queries by q_y and q_n, where q_y is the query asked by $M(x)$ if the first query was answered "yes," and q_n the query asked if the first query was answered "no."
2. N queries q_n to D.
3. N determines the truth-table of $M(x)$ with respect to q_0 and q_y, with query q_n answered correctly. That is, let (X_1, X_2, X_3), $X_i \in \{\mathrm{A}, \mathrm{R}\}$, where A stands for accept and R for reject, be such that
   (a) X_1 is the outcome of $M(x)$ if both q_0 and q_y are answered "yes" (recall that if q_0 is answered "yes" then $M(x)$ asks q_y as its second query),

   (b) X_2 is the outcome of $M(x)$ if q_0 is answered "yes" and q_y is answered "no," and
   (c) X_3 is the outcome of $M(x)$ if q_0 is answered "no" and q_n is answered correctly.
4. There are eight different cases for (X_1, X_2, X_3). We have to show that each case can be handled in polynomial time with one query to C'. We will henceforward assume that there are more Rs than As in (X_1, X_2, X_3). (The remaining cases follow by complementation.) Depending on the determined truth-table (X_1, X_2, X_3), N does the following:
   (a) $(X_1, X_2, X_3) = (\mathrm{R}, \mathrm{R}, \mathrm{R})$. In this case, N will of course reject.
   (b) $(X_1, X_2, X_3) = (\mathrm{A}, \mathrm{R}, \mathrm{R})$. Then M accepts if and only if $q_0 \in C$ and $q_y \in D$. This is the case if and only if $\langle q_0, q_y\rangle 10 \in C'$. So N queries $\langle q_0, q_y\rangle 10$ to C' and accepts if and only if the answer is "yes."
   (c) $(X_1, X_2, X_3) = (\mathrm{R}, \mathrm{A}, \mathrm{R})$. Then M accepts if and only if $q_0 \in C$ and $q_y \notin D$. This is the case if and only if $\langle q_0, q_y\rangle 00 \in C'$. So N queries $\langle q_0, q_y\rangle 00$ to C' and accepts if and only if the answer is "yes."
   (d) $(X_1, X_2, X_3) = (\mathrm{R}, \mathrm{R}, \mathrm{A})$. Then M accepts if and only if $q_0 \notin C$. This is the case if and only if $q_0 1 \notin C'$. So N queries $q_0 1$ to C' and accepts if and only if the answer is "no."

It is clear from the construction that $L(M^{C:D}) = L(N^{D:C'})$ and thus $L \in \mathrm{P}^{\mathcal{D}:\mathcal{C}}$, since $C' \in \mathcal{C}$ by the closure properties in the theorem's hypothesis. □

Corollary 6. *If $\mathcal{C}$ and $\mathcal{D}$ are classes such that $\mathcal{C}$ is closed under disjoint union and $\mathcal{C}$ is closed downwards under $\leq_m^{p,\mathcal{D}[1]}$, then*

$$\mathrm{P}^{\mathcal{C}:\mathcal{D}} \subseteq \mathrm{P}^{\mathcal{D}:\mathcal{C}}.$$

Proof. If $\mathcal{C}$ contains only trivial sets, i.e., $\mathcal{C} \subseteq \{\emptyset, \Sigma^*\}$, then $\mathrm{P}^{\mathcal{C}:\mathcal{D}} = \mathrm{P}^{\mathcal{D}} = \mathrm{P}^{\mathcal{D}:\mathcal{C}}$ and we are done. So from now on we assume that $\mathcal{C}$ contains a nontrivial set. We will show that in this case we can apply Theorem 5, i.e., we will show that $\mathcal{C}$, which is closed under disjoint union, has also the properties that $\mathcal{C}$ "ands" $(\mathcal{C}, \mathcal{D})$ and $\mathcal{C}$ "ands" $(\mathcal{C}, \mathrm{co}\mathcal{D})$.

Let $C \in \mathcal{C}$ and $D \in \mathcal{D}$. We need to show that $C \times D \in \mathcal{C}$ and $C \times \overline{D} \in \mathcal{C}$. If $C \neq \Sigma^*$, then $C \times D \leq_m^{p,\mathcal{D}[1]} C$ by $f(\langle x, y\rangle) = x$ if $y \in D$ and some fixed element not in C if $y \notin D$. Since $\mathcal{C}$ is closed under $\leq_m^{p,\mathcal{D}[1]}$, it follows that $C \times D \in \mathcal{C}$. Similarly, if $C \neq \Sigma^*$, then $C \times \overline{D} \in \mathcal{C}$.

If $\Sigma^* \in \mathcal{C}$, it remains to show that $\Sigma^* \times D$ and $\Sigma^* \times \overline{D} \in \mathcal{C}$. Let $C \in \mathcal{C}$ be a nontrivial set (recall that we earlier eliminated the case in which $\mathcal{C}$ lacks nontrivial sets), and let $c \in C$ and $\widehat{c} \notin C$. Then $\Sigma^* \times D \leq_m^{p,\mathcal{D}[1]} C$ by $f(\langle x, y\rangle) = c$ if $y \in D$ and $\widehat{c}$ if $y \notin D$, and $\Sigma^* \times \overline{D} \leq_m^{p,\mathcal{D}[1]} C$ by $f(\langle x, y\rangle) = c$ if $y \notin D$ and $\widehat{c}$ if $y \in D$. □

Proof of Theorem 2. Let $\mathcal{C}$ and $\mathcal{D}$ be classes such that $\mathcal{C}$ is closed under disjoint union and $\mathcal{C}$ is closed downwards under $\leq_m^{p,\mathcal{D}[1]}$. We have to show that $\mathrm{P}^{\mathcal{C}:\mathcal{D}} = \mathrm{P}^{\mathcal{D}:\mathcal{C}} = \mathrm{P}^{(\mathcal{C},\mathcal{D})}_{1,1\text{-tt}}$. The containment $\mathrm{P}^{\mathcal{C}:\mathcal{D}} \subseteq \mathrm{P}^{\mathcal{D}:\mathcal{C}}$ follows from Corollary 6, and $\mathrm{P}^{(\mathcal{C},\mathcal{D})}_{1,1\text{-tt}} \subseteq \mathrm{P}^{\mathcal{C}:\mathcal{D}}$ is immediate from the definitions.

It remains to show that $\mathrm{P}^{\mathcal{D}:\mathcal{C}} \subseteq \mathrm{P}^{(\mathcal{C},\mathcal{D})}_{1,1\text{-tt}}$. Suppose $L \in \mathrm{P}^{\mathcal{D}:\mathcal{C}}$ and let DPTM M, $C \in \mathcal{C}$, and $D \in \mathcal{D}$ be such that $L = L(M^{D:C})$. Without loss of generality, let M always query both D and C. We now describe a DPTM N and a set C' such that $C' \in \mathcal{C}$ and $L = L(N_{1,1\text{-tt}}^{(C',D)})$. Define

$$C' = \{x \mid \text{the second query asked by } M^{D:C}(x) \text{ is in } C\}.$$

Since $\mathcal{C}$ is closed downwards under $\leq_m^{p,\mathcal{D}[1]}$, we clearly have $C' \in \mathcal{C}$.

Let $N_{1,1\text{-tt}}^{(C',D)}$ on input x work as follows: $N_{1,1\text{-tt}}^{(C',D)}(x)$ simulates $M^{D:C}(x)$ until $M^{D:C}(x)$ asks its first query, call it q. Then $N_{1,1\text{-tt}}^{(C',D)}(x)$ queries "$x \in C'$?" and "$q \in D$?" N at this point has enough information to simulate the final action of M. We make this completely rigorous and formal as follows. Let $S_{C'}$ be Σ^* if $x \in C'$ and let $S_{C'}$ be $\emptyset$ if $x \notin C'$. Let S_D be Σ^* if $q \in D$ and let S_D be $\emptyset$ if $q \notin D$. $N_{1,1\text{-tt}}^{(C',D)}(x)$ accepts if and only if $M^{S_D:S_{C'}}(x)$ accepts (which $N(x)$ can easily determine given the answers to $N(x)$'s two queries). It is clear from the construction that $L(M^{D:C}) = L(N_{1,1\text{-tt}}^{(C',D)})$, and thus $L \in \mathrm{P}_{1,1\text{-tt}}^{(\mathcal{C},\mathcal{D})}$. □

In addition to leading to the "polynomial hierarchy is order-oblivious" results that this section will obtain, and leading to Section 4's applications to probabilistic and unambiguous classes, Theorem 2 has also played an important role in distinguishing robust Turing and many-one completeness [HHH97b].

The next theorem shows that if $\mathcal{C}$ and $\mathcal{D}$ are closed under disjoint union and are order-oblivious with respect to P transducers, then ordered access equals arbitrary-order access. Note that Theorem 7's hypothesis requires that *both* classes be closed under disjoint union, in contrast to the hypothesis of Theorem 5.

Theorem 7. *If $\mathcal{C}$ and $\mathcal{D}$ are complexity classes that are both closed under disjoint union, then*

$$\mathrm{P}^{\mathcal{C}:\mathcal{D}} = \mathrm{P}^{\mathcal{D}:\mathcal{C}} \Rightarrow \mathrm{P}^{\mathcal{C}:\mathcal{D}} = \mathrm{P}^{\mathcal{C},\mathcal{D}}.$$

Proof. Suppose that $\mathrm{P}^{\mathcal{C}:\mathcal{D}} = \mathrm{P}^{\mathcal{D}:\mathcal{C}}$. Since $\mathrm{P}^{\mathcal{C}:\mathcal{D}} \subseteq \mathrm{P}^{\mathcal{C},\mathcal{D}}$, we have only to show that $\mathrm{P}^{\mathcal{C},\mathcal{D}} \subseteq \mathrm{P}^{\mathcal{C}:\mathcal{D}}$. Let $L \in \mathrm{P}^{\mathcal{C},\mathcal{D}}$, and let DPTM M, $C \in \mathcal{C}$, and $D \in \mathcal{D}$ be such that $L = L(M^{C,D})$. Without loss of generality, we assume that $M^{C,D}$ always queries each oracle exactly once. Define

$$L_1 = \{x \in L \mid M^{C,D}(x) \text{ first queries } C\}.$$

$$L_2 = \{x \in L \mid M^{C,D}(x) \text{ first queries } D\}.$$

Let N be a DPTM such that $L_1 = L(N^{C:D})$. Since clearly $L_2 \in \mathrm{P}^{\mathcal{D}:\mathcal{C}}$, by our hypothesis there exists a DPTM T, and sets $C' \in \mathcal{C}$ and $D' \in \mathcal{D}$, such that $L_2 = L(T^{C':D'})$. Let $\widehat{C} = C \oplus C'$ and $\widehat{D} = D \oplus D'$. We describe a DPTM S such that $L = L(S^{\widehat{C}:\widehat{D}})$.

S on input x will work as follows: $S(x)$ simulates (appropriately tagging a 0 or a 1 onto the end of queries to address the appropriate part of the disjoint union) $M^{C,D}(x)$ until $M^{C,D}(x)$ makes its first query. Then $S(x)$ simulates $N^{C:D}(x)$ or $T^{C':D'}(x)$, depending on whether the first query of $M^{C,D}(x)$ was asked to C or D, respectively. Note that clearly $L = L(S^{\widehat{C}:\widehat{D}})$, and thus $L \in \mathrm{P}^{\mathcal{C}:\mathcal{D}}$. □

From Theorem 2 and Theorem 7 we have the following.

Corollary 8. *If $\mathcal{C}$ and $\mathcal{D}$ are classes such that $\mathcal{C}$ is closed downwards under $\leq_m^{p,\mathcal{D}[1]}$ and $\mathcal{C}$ and $\mathcal{D}$ are closed under disjoint union, then*

$$\mathrm{P}_{1,1\text{-tt}}^{(\mathcal{C},\mathcal{D})} = \mathrm{P}^{\mathcal{C}:\mathcal{D}} = \mathrm{P}^{\mathcal{D}:\mathcal{C}} = \mathrm{P}^{\mathcal{C},\mathcal{D}}.$$

Corollary 8 implies that query order does not matter in the polynomial hierarchy.

Corollary 9. *1. For all $j,k \geq 0$, $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_k^p:\Sigma_j^p}$.*
2. For all $j,k \geq 0$ such that $j \neq k$, $\mathrm{P}_{1,1\text{-tt}}^{(\Sigma_j^p,\Sigma_k^p)} = \mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_j^p,\Sigma_k^p}$.

Proof. Note that for $j = k$ the claim of part 1 is trivial. Assume $j < k$ (the $j > k$ case is similar). It is immediately clear that Σ_k^p is closed downwards under $\leq_m^{p,\Sigma_j^p[1]}$ and it is well-known that Σ_j^p and Σ_k^p are both closed under disjoint union. So we can apply Corollary 8. Thus, both parts of the theorem are established. $\square$

Note that in part 2 of Corollary 9 we need $j \neq k$, since otherwise we would have included the claim that two truth-table queries to Σ_k^p have as much computational power as two Turing queries. However, that would imply that the boolean hierarchy over Σ_k^p collapses to the 2-truth-table closure of Σ_k^p, which in turn would imply that the polynomial hierarchy collapses. The last implication refers to the well-known fact that if the boolean hierarchy collapses then the polynomial hierarchy collapses; this fact was first proven by Kadin [Kad88], and the strongest known collapse of the polynomial hierarchy from a given collapse of the boolean hierarchy is the one recently obtained by Hemaspaandra et al. [HHH98] and, independently, by Reith and Wagner [RW98].

We also have the following.

Corollary 10. *1. For all $k \geq 0$ and $j > 0$,*

$$\mathrm{P}_{1,1\text{-tt}}^{(\Delta_j^p,\Sigma_k^p)} = \mathrm{P}^{\Delta_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_k^p:\Delta_j^p} = \mathrm{P}^{\Sigma_k^p,\Delta_j^p} = \begin{cases} \Delta_j^p & \text{if } j > k \\ \mathrm{P}^{\Sigma_k^p[1],\Sigma_{j-1}^p[poly]} & \text{if } j \leq k. \end{cases}$$

2. For all $j,k \geq 0$, $\mathrm{P}_{1,1\text{-tt}}^{(\Delta_j^p,\Sigma_k^p\cap\Pi_k^p)} =$

$$\mathrm{P}^{\Delta_j^p:\Sigma_k^p\cap\Pi_k^p} = \mathrm{P}^{\Sigma_k^p\cap\Pi_k^p:\Delta_j^p} = \mathrm{P}^{\Sigma_k^p\cap\Pi_k^p,\Delta_j^p} = \begin{cases} \Delta_j^p & \text{if } j > k \\ \Sigma_k^p \cap \Pi_k^p & \text{if } j \leq k. \end{cases}$$

3. For all $j,k \geq 0$, $\mathrm{P}_{1,1\text{-tt}}^{(\Sigma_j^p\cap\Pi_j^p,\Sigma_k^p\cap\Pi_k^p)} =$

$$\mathrm{P}^{\Sigma_j^p\cap\Pi_j^p:\Sigma_k^p\cap\Pi_k^p} = \mathrm{P}^{\Sigma_k^p\cap\Pi_k^p:\Sigma_j^p\cap\Pi_j^p} = \mathrm{P}^{\Sigma_k^p\cap\Pi_k^p,\Sigma_j^p\cap\Pi_j^p} = \Sigma_{\max(j,k)}^p \cap \Pi_{\max(j,k)}^p.$$

Proof. We first prove part 1. If $j > k$, then a Δ_j^p machine can simulate $\mathrm{P}^{\Delta_j^p,\Sigma_k^p}$, and it is unconditionally immediate that $\mathrm{P}_{1,1\text{-tt}}^{(\Delta_j^p,\Sigma_k^p)}$ contains Δ_j^p. If $0 < j \leq k$, then Σ_k^p is closed under $\leq_m^{p,\Delta_j^p[1]}$ and thus, by Corollary 8, $\mathrm{P}_{1,1\text{-tt}}^{(\Delta_j^p,\Sigma_k^p)} = \mathrm{P}^{\Delta_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_k^p:\Delta_j^p} = \mathrm{P}^{\Delta_j^p,\Sigma_k^p}$. Since $\Delta_j^p = \mathrm{P}^{\Sigma_{j-1}^p}$, $\mathrm{P}^{\Sigma_k^p:\Delta_j^p} \subseteq \mathrm{P}^{\Sigma_k^p[1],\Sigma_{j-1}^p[poly]}$.

It remains to show that $\mathrm{P}^{\Sigma_k^p[1],\Sigma_{j-1}^p[poly]} \subseteq \mathrm{P}^{\Sigma_k^p:\Delta_j^p}$. Suppose $L \in \mathrm{P}^{\Sigma_k^p[1],\Sigma_{j-1}^p[poly]}$ and let DPTM M, $A \in \Sigma_k^p$, and $B \in \Sigma_{j-1}^p$ be such that $L = L(M^{A[1],B[poly]})$. Without loss of generality, let M ask all its queries to B before asking anything to A. (If M does not have the desired property, replace it with a machine that, before asking anything to A, asks to B the queries M would ask to B if the A query were answered "yes" and also asks to B the queries M would ask to B if the A query were answered "no" and then queries A and uses the appropriate set of already obtained answers to complete the simulation of the original M.) We will denote this with the notation $L = L(M^{B[poly]:A[1]})$. Also, without loss of generality assume that $M^{B[poly]:A[1]}$ on input x asks exactly one query a_x to A.

Now let us describe a DPTM N and sets A' and C such that $A' \in \Sigma_k^p$, $C \in \Delta_j^p$, and $L(N^{A':C}) = L$.

$$A' = \{x \in \Sigma^* \mid a_x \in A\},$$
$$C = \{x \mid M^{B[poly]:\emptyset[1]}(x) \text{ accepts}\} \oplus \{x \mid M^{B[poly]:\Sigma^*[1]}(x) \text{ accepts }\}.$$

Note that the use of $\emptyset$ and Σ^* in the definition of C is just a way to study the effect, respectively, of "no" and "yes" oracle answers. Clearly we have $A' \in \Sigma_k^p$ and $C \in \Delta_j^p$. On input x, $N^{A':C}$ will work as follows: $N^{A':C}(x)$ first queries "$x \in A'$." If the answer to "$x \in A'$" is "yes," then N accepts if and only if $x1 \in C$ and if the answer to "$x \in A'$" is "no," then N accepts if and only if $x0 \in C$. It is immediate that $L(N^{A':C}) = L(M^{B[poly]:A[1]})$ and thus $L \in \mathrm{P}^{\Sigma_k^p:\Delta_j^p}$. This completes the proof of part 1 of the corollary.

We now turn to proving part 2. First note that both Δ_j^p and $\Sigma_k^p \cap \Pi_k^p$ are trivially contained in $\mathrm{P}_{1,1\text{-tt}}^{(\Delta_j^p,\Sigma_k^p\cap\Pi_k^p)}$. The $j > k$ case now follows from part 1, since $\mathrm{P}^{\Sigma_k^p\cap\Pi_k^p,\Delta_j^p} \subseteq \mathrm{P}^{\Sigma_k^p,\Delta_j^p} = \Delta_j^p$. If $j \leq k$, then a Σ_k^p machine can simulate $\mathrm{P}^{\Delta_j^p,\Sigma_k^p\cap\Pi_k^p}$. (This simulation is an easy variation of the standard simulation showing that $\mathrm{P}^{\Sigma_k^p\cap\Pi_k^p} = \Sigma_k^p \cap \Pi_k^p$, which itself is a straightforward generalization of the early work [Sel74, Sel79] noting $\mathrm{P}^{\mathrm{NP}\cap\mathrm{coNP}} = \mathrm{NP} \cap \mathrm{coNP}$.) Since $\mathrm{P}^{\Delta_j^p,\Sigma_k^p\cap\Pi_k^p}$ is closed under complement, it follows that $\mathrm{P}^{\Delta_j^p,\Sigma_k^p\cap\Pi_k^p} \subseteq \Sigma_k^p \cap \Pi_k^p$.

We now prove part 3. As in part 2, $\Sigma_j^p \cap \Pi_j^p$ and $\Sigma_k^p \cap \Pi_k^p$ are trivially contained in $\mathrm{P}_{1,1\text{-tt}}^{(\Sigma_j^p\cap\Pi_j^p,\Sigma_k^p\cap\Pi_k^p)}$. Also, a $\Sigma_{\max(j,k)}^p$ machine can (even if $j = k$) simulate $\mathrm{P}^{\Sigma_j^p\cap\Pi_j^p,\Sigma_k^p\cap\Pi_k^p}$ so, by complementation, we have $\mathrm{P}^{\Sigma_j^p\cap\Pi_j^p,\Sigma_k^p\cap\Pi_k^p} \subseteq \Sigma_{\max(j,k)}^p \cap \Pi_{\max(j,k)}^p$. □

3.2 Query Order Classes Differ from Standard Polynomial Hierarchy Levels and from Each Other

In Section 1 we mentioned that though $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_k^p[1]}$, $0 < j < k$, might seem a tempting claim,[1] the claim is false unless the polynomial hierarchy collapses. In fact, we will prove something much stronger, namely that, unless the polynomial

[1] The reason the tempting proof implicitly sketched in the introduction is not valid is that, though Σ_k^p indeed can simulate Σ_j^p, $j < k$, Σ_k^p can neither pass an extra bit of information back to the "base" P machine nor—in the crucial case in which the

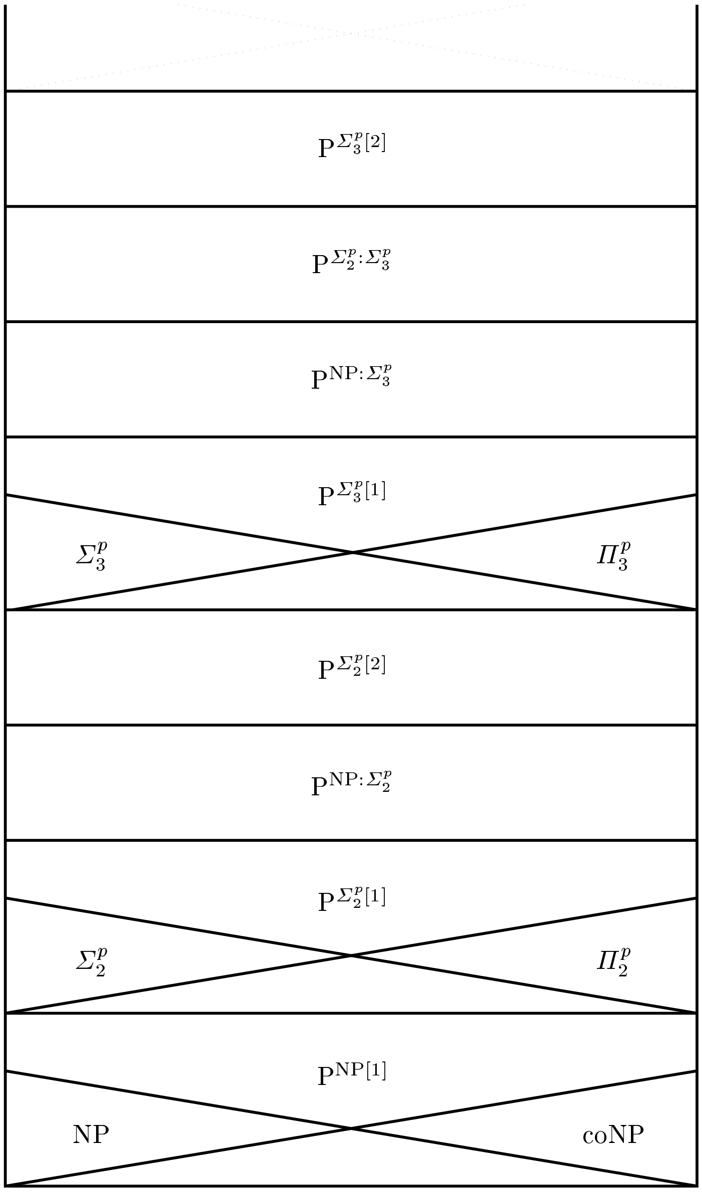

Figure 1: All the classes shown are distinct, unless the polynomial hierarchy collapses (see Theorem 11).

hierarchy collapses, $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_\ell^p:\Sigma_m^p}$ if and only if $\{j,k\} = \{\ell,m\}$. The "if" direction is trivial. Theorem 11 establishes the "only if" direction.

Theorem 11. *Let $j,k,\ell,m \geq 0$. If $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_\ell^p:\Sigma_m^p}$ then $\{j,k\} = \{\ell,m\}$ or the polynomial hierarchy collapses.*

This theorem will follow from a result of this paper combined with the results and techniques of [HHH97c]. The following proposition is a strong and counter-intuitive downward translation result that has recently been established.

Proposition 12. *(Special case of [HHH97c, Theorem 2.3]) Let $0 < j$ and $j < k$. If $\Delta_j^p \triangle \Sigma_k^p = \Sigma_j^p \triangle \Sigma_k^p$, then $\Sigma_k^p = \Pi_k^p = \mathrm{PH}$.*

For all j and k, it holds that $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} \subseteq \Delta_{j+1}^p \triangle \Sigma_k^p$. Why? For $j \geq k$ it is immediate as in that case $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} \subseteq \mathrm{P}^{\Sigma_j^p[2]}$. For $j < k$ it follows essentially by the technique of the *proof* of [HHH, Lemma 2.3]. So, for all j and k we have $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} \subseteq \Delta_{j+1}^p \triangle \Sigma_k^p \subseteq \Sigma_{j+1}^p \triangle \Sigma_k^p \subseteq \mathrm{P}^{\Sigma_{j+1}^p:\Sigma_k^p}$, and thus we have the following corollary.

Corollary 13. *Let $0 \leq j$ and $j < k-1$. If $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_{j+1}^p:\Sigma_k^p}$, then $\Sigma_k^p = \Pi_k^p = \mathrm{PH}$.*

We note the strength of this collapse. The conclusion obtains a collapse of the hierarchy to a level that is generally thought to be lower (a priori) than the level of either of the classes whose equality was assumed in the hypothesis. That is, this is an actual *downward* translation of equality, in contrast with the far more common behavior of upward translation of equality (see, e.g., [Wag87, Wag89, RRW94], for examples and discussion).

We now can prove Theorem 11.

Proof of Theorem 11. Suppose that $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_\ell^p:\Sigma_m^p}$ and that (without loss of generality in light of Corollary 9) $j \leq k$ and $\ell \leq m$. Suppose that either $k < m$ or ($k = m$ and $j < \ell$). First note that if $k < m$, then $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} \subseteq \Sigma_m^p \subseteq \mathrm{P}^{\Sigma_\ell^p:\Sigma_m^p}$. Since $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_\ell^p:\Sigma_m^p}$, it follows immediately that $\Sigma_m^p = \Pi_m^p = \mathrm{PH}$. So suppose that $k = m$ and $j < \ell$. Then $\mathrm{P}^{\Sigma_j^p:\Sigma_k^p} = \mathrm{P}^{\Sigma_{j+1}^p:\Sigma_k^p}$. If $j < k-1$, then $\Sigma_k^p = \Pi_k^p = \mathrm{PH}$ by Corollary 13. Finally, suppose that $j = k-1$. Then the class of sets that 2-truth-table reduce to Σ_k^p sets equals the class of sets that 2-Turing reduce to Σ_k^p sets, which itself is well-known ([Kad88], see also [HHH98, RW98]) to imply that PH collapses. □

So it is clear from Theorem 11 that query order classes do not equal standard "bounded query" classes but rather form new intermediate levels in the polynomial hierarchy, unless the polynomial hierarchy collapses (see Figure 1).

We conclude this section by mentioning some very recent related work that was inspired by the present paper. In this paper, our basic model is of ordered

base P machine uses the answer to the Σ_j^p query to decide whether to treat the Σ_k^p answer via the strictly positive truth-table or the strictly negative truth-table—can it complement itself (as that seemingly requires $\Sigma_k^p = \Pi_k^p$). That is, the tempting claim fails due to a 1-bit information-passing bottleneck.

access to two sets. Wagner [Wag] and Beigel and Chang [BC97] build on the work of the current paper by studying machines that have ordered access to truth-table groups of queries and they show that in that case too order does not matter. We consider this work to be important and interesting, and to broaden the range of models to which the questions of this paper can be applied. We also mention that the work is not strictly stronger than our work as Beigel and Chang discuss only sets from the polynomial hierarchy and Wagner has somewhat different hypotheses than we do on the classes involved, especially regarding our intermediate results that separate out exactly what hypotheses imply what conclusions. Also, in contrast to the key hierarchy collapse result of the present paper, which guarantees and proves a downward translation of equality, the analogous hierarchy collapses of those papers obtain from weaker hypotheses weaker collapses (namely, the collapse results of those papers related to query-order-based language classes merely assert that the hierarchy collapses, and they rely either on the upward-equality-translation work of Selivanov or make no specific collapse-level claim at all). Finally, as we will discuss later in more detail, the work of Section 4 applies fully to the cases discussed in these papers. A survey paper by Hemaspaandra et al. [HHH97a] provides a detailed overview of query order.

4 Base Classes Other Than P

We show that a wide variety of classes inherit all order exchanges that hold for P. For example, if $\mathrm{P}^{\mathcal{C}:\mathcal{D}} \subseteq \mathrm{P}^{\mathcal{D}:\mathcal{C}}$ then $\mathrm{PP}^{\mathcal{C}:\mathcal{D}} \subseteq \mathrm{PP}^{\mathcal{D}:\mathcal{C}}$. Thus order exchanges proven for P—such as those of Section 3.1 of this paper and those of Hemaspaandra, Hempel, and Wechsung [HHW]—can immediately be applied to many other classes.

We prove our result in a very general form, and then state some corollaries and applications to make the meaning of the theorem more concrete. For classes $\mathcal{D}_1$ and $\mathcal{D}_2$ for which relativization has been defined, we say that $\mathcal{D}_1$ is robustly contained in $\mathcal{D}_2$ if, for each A, $\mathcal{D}_1^A \subseteq \mathcal{D}_2^A$. $\mathcal{D}^{\mathcal{C}[1]}$ will mean that each path of the base machine makes at most one call to $\mathcal{C}$. $\mathcal{D}^{\mathcal{C}_1:\mathcal{C}_2}$ will mean that each path of the base machine makes at most one call to $\mathcal{C}_1$ followed by at most one call to $\mathcal{C}_2$.

Definition 14. Let $\mathcal{D}$ be a complexity class for which relativization is defined. We say that $\mathcal{D}$ is *sane* if

$$(\forall \mathcal{C}_1, \mathcal{C}_2)[\mathcal{D}^{\mathcal{C}_1:\mathcal{C}_2} = \mathcal{D}^{(\mathrm{P}^{\mathcal{C}_1:\mathcal{C}_2})[1]}].$$

The important point to note is that essentially all standard complexity classes within the realm of potentially feasible computation (classes from P to PSPACE) are sane. In particular, bringing work of Bovet, Crescenzi, and Silvestri into notational analogy with more recent terminology, let us say that a relativizable complexity class $\mathcal{D}$ is leaf-definable if $\mathcal{D}$ "admits a C-Class representation" in the formal sense (which we will not repeat here) defined by Bovet, Crescenzi, and Silvestri ([BCS92], see also [BCS95]) and the representation itself holds also in all relativized worlds (under the natural extension of their work to ordered oracle access, following exactly their discussion of relativization). Bovet, Crescenzi, and Silvestri [BCS92] prove that essentially all standard classes in the realm of potentially feasible computation "admit C-Class representations," they observe that these representations all relativize, and we comment that their observation clearly holds also for ordered access. The reason this is relevant is that it is easy to see that all leaf-definable classes are sane. Thus, the following result says that essentially all standard complexity classes inherit every order exchange possessed by P.

Theorem 15. *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be sane complexity classes, and let $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, and $\mathcal{C}_4$ be complexity classes. If $\mathcal{D}_1$ is robustly contained in $\mathcal{D}_2$ and $\mathrm{P}^{\mathcal{C}_1:\mathcal{C}_2} \subseteq \mathrm{P}^{\mathcal{C}_3:\mathcal{C}_4}$, then*

$$\mathcal{D}_1^{\mathcal{C}_1:\mathcal{C}_2} \subseteq \mathcal{D}_2^{\mathcal{C}_3:\mathcal{C}_4}.$$

Proof. The theorem holds via the following inclusion chain:

$$\mathcal{D}_1^{\mathcal{C}_1:\mathcal{C}_2} \overset{a}{\subseteq} \mathcal{D}_1^{(\mathrm{P}^{\mathcal{C}_1:\mathcal{C}_2})[1]} \overset{b}{\subseteq} \mathcal{D}_1^{(\mathrm{P}^{\mathcal{C}_3:\mathcal{C}_4})[1]} \overset{c}{\subseteq} \mathcal{D}_2^{(\mathrm{P}^{\mathcal{C}_3:\mathcal{C}_4})[1]} \overset{d}{\subseteq} \mathcal{D}_2^{\mathcal{C}_3:\mathcal{C}_4}.$$

Inclusion (b) follows from the assumption that $\mathrm{P}^{\mathcal{C}_1:\mathcal{C}_2} \subseteq \mathrm{P}^{\mathcal{C}_3:\mathcal{C}_4}$ and inclusion (c) follows from the assumption that $\mathcal{D}_1$ is robustly contained in $\mathcal{D}_2$. Inclusions (a) and (d) hold via the fact that the classes are sane. □

Corollary 16. *Let $\mathcal{D}$ be any sane complexity class. If $\mathrm{P}^{\mathcal{C}_1:\mathcal{C}_2} \subseteq \mathrm{P}^{\mathcal{C}_2:\mathcal{C}_1}$ then*

$$\mathcal{D}^{\mathcal{C}_1:\mathcal{C}_2} \subseteq \mathcal{D}^{\mathcal{C}_2:\mathcal{C}_1}.$$

We give some examples of how this can be applied. BH_j will denote the jth level of the boolean hierarchy [CGH+88], and as is standard DP [PY84] denotes the second level of the boolean hierarchy. Note that Bovet, Crescenzi, and Silvestri [BCS92] have proven that BPP, UP, and PP are leaf-definable classes. Thus, these classes are sane.

Example 1. 1. $\mathrm{PP}^{\mathrm{NP}:\Sigma_2^p} = \mathrm{PP}^{\Sigma_2^p:\mathrm{NP}}$.
2. $\mathrm{BPP}^{\mathrm{BH}_{50}:\mathrm{BH}_{25}} \subseteq \mathrm{BPP}^{\mathrm{BH}_{80}:\mathrm{BH}_{10}} = \mathrm{BPP}^{\mathrm{BH}_{10}:\mathrm{BH}_{80}} \subseteq \mathrm{BPP}^{\mathrm{BH}_{25}:\mathrm{BH}_{38}}$.
3. $\mathrm{UP}^{\mathrm{DP}:\mathrm{BH}_3} = \mathrm{UP}^{\mathrm{BH}_3:\mathrm{DP}} = \mathrm{UP}^{\mathrm{NP}:\mathrm{BH}_3}$.

Parts 2 and 3 of the example hold due to Corollary 16 in light of [HHW], which proves that the class of languages computable via a polynomial-time machine given one query to the jth level of the boolean hierarchy followed by one query to the kth level of the boolean hierarchy equals $\mathrm{R}^p_{j+2k-1\text{-tt}}(\mathrm{NP})$ if j is even and k is odd, and equals $\mathrm{R}^p_{j+2k\text{-tt}}(\mathrm{NP})$ otherwise, where $\mathrm{R}^p_{\ell\text{-tt}}(\mathrm{NP})$ equals the class of languages that ℓ-truth-table reduce to NP sets. Part 1 follows, as an application of Corollary 16, from Corollary 9.

Though Theorem 15 says that *all* order exchanges of P apply to essentially all standard complexity classes, it of course remains possible that certain path-based classes may possess additional order exchanges. For example, though Section 3.2 showed that P ordered query classes create new intermediate polynomial hierarchy levels unless the polynomial hierarchy collapses, this clearly is not the case for NP or Σ_k^p.

Theorem 17. *If $i \geq 1$ and $j, k \geq 0$, then*

$$(\Sigma_i^p)^{\Sigma_j^p:\Sigma_k^p} = (\Sigma_i^p)^{\Sigma_k^p:\Sigma_j^p} = \Sigma_{i+\max(j,k)}^p.$$

Proof. Without loss of generality, suppose $j \leq k$. Then $\Sigma_{i+\max(j,k)}^p = \Sigma_{i+k}^p = (\Sigma_i^p)^{\Sigma_k^p} = (\Sigma_i^p)^{\Sigma_k^p[1]}$ is clear in light of the quantifier characterization of the levels of the polynomial hierarchy [Wra77, Sto77]. Furthermore, $(\Sigma_i^p)^{\Sigma_k^p[1]} \subseteq (\Sigma_i^p)^{\Sigma_j^p:\Sigma_k^p} \subseteq (\Sigma_i^p)^{\Sigma_k^p} = \Sigma_{i+k}^p$ and similarly $(\Sigma_i^p)^{\Sigma_k^p[1]} \subseteq (\Sigma_i^p)^{\Sigma_k^p:\Sigma_j^p} \subseteq (\Sigma_i^p)^{\Sigma_k^p} = \Sigma_{i+k}^p$. □

Relatedly, classes may also trivially exhibit certain equalities based on class-specific features. For example, it follows trivially from $\mathrm{NP} \subseteq \mathrm{PP}$ and the (non-trivial) result of Fortnow and Reingold [FR96] regarding the $\leq_{tt}^p$ closure of PP that $\mathrm{PP} = \mathrm{PP}^{\mathrm{NP:PP}} = \mathrm{PP}^{\mathrm{PP:NP}}$.

Finally, as we mentioned earlier, other papers have suggested varying the model of this paper to include multiple queries to many oracles in various patterns of truth-table and ordered access. We note that the approach of this section applies completely to such cases (modifying the definitions of sanity and leaf-definability to reflect whatever access model one is using).

References

[BC93] D. Bovet and P. Crescenzi. *Introduction to the Theory of Complexity*. Prentice Hall, 1993.

[BC97] R. Beigel and R. Chang. Commutative queries. In *Proceedings of the 5th Israeli Symposium on Theory of Computing and Systems*, pages 159–165. IEEE Computer Society Press, June 1997.

[BCS92] D. Bovet, P. Crescenzi, and R. Silvestri. A uniform approach to define complexity classes. *Theoretical Computer Science*, 104(2):263–283, 1992.

[BCS95] D. Bovet, P. Crescenzi, and R. Silvestri. Complexity classes and sparse oracles. *Journal of Computer and System Sciences*, 50(3):382–390, 1995.

[BDG95] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, 2nd edition, 1995.

[CGH⁺88] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy I: Structural properties. *SIAM Journal on Computing*, 17(6):1232–1252, 1988.

[FR96] L. Fortnow and N. Reingold. PP is closed under truth-table reductions. *Information and Computation*, 124(1):1–6, 1996.

[HHH] E. Hemaspaandra, L. Hemaspaandra, and H. Hempel. A downward collapse within the polynomial hierarchy. *SIAM Journal on Computing*. To appear.

[HHH97a] E. Hemaspaandra, L. Hemaspaandra, and H. Hempel. An introduction to query order. *Bulletin of the EATCS*, (63):93–107, 1997.

[HHH97b] E. Hemaspaandra, L. Hemaspaandra, and H. Hempel. $\mathrm{R}_{1\text{-}tt}^{SN}(\mathrm{NP})$ distinguishes robust many-one and Turing completeness. In *Proceedings of the 3rd Italian Conference on Algorithms and Complexity*, pages 49–60. Springer-Verlag *Lecture Notes in Computer Science #1203*, March 1997. Final version appears in *Theory of Computing Systems*, v. 31, pp. 307–325, 1998.

[HHH97c] E. Hemaspaandra, L. Hemaspaandra, and H. Hempel. Translating equality downwards. Technical Report TR-657, Department of Computer Science, University of Rochester, Rochester, NY, April 1997. Available on-line at http://www.cs.rochester.edu/trs/theory-trs.html.

[HHH98] E. Hemaspaandra, L. Hemaspaandra, and H. Hempel. What's up with downward collapse: Using the easy-hard technique to link boolean and polynomial hierarchy collapses. Technical Report TR-682, Department of Computer Science, University of Rochester, Rochester, NY, February 1998. Available on-line at http://www.cs.rochester.edu/trs/theory-trs.html.

[HHW] L. Hemaspaandra, H. Hempel, and G. Wechsung. Query order. *SIAM Journal on Computing*. To appear.

[Kad88] J. Kadin. The polynomial time hierarchy collapses if the boolean hierarchy collapses. *SIAM Journal on Computing*, 17(6):1263–1282, 1988. Erratum appears in the same journal, 20(2):404.

[KSW87] J. Köbler, U. Schöning, and K. Wagner. The difference and truth-table hierarchies for NP. *RAIRO Theoretical Informatics and Applications*, 21:419–435, 1987.

[LLS75] R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1(2):103–124, 1975.

[Pap94] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[PY84] C. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). *Journal of Computer and System Sciences*, 28(2):244–259, 1984.

[RRW94] R. Rao, J. Rothe, and O. Watanabe. Upward separation for FewP and related classes. *Information Processing Letters*, 52(4):175–180, 1994.

[RW98] S. Reith and K. Wagner. On boolean lowness and boolean highness. Technical Report TR 195, Institut für Informatik, Bayerische Julius-Maximilians-Universität Würzburg, Würzburg, Germany, March 1998.

[Sel74] A. Selman. On the structure of NP. *Notices of the AMS*, 2:A–498, 1974. Erratum in Volume 2, page 310.

[Sel79] A. Selman. P-selective sets, tally languages, and the behavior of polynomial time reducibilities on NP. *Mathematical Systems Theory*, 13:55–65, 1979.

[Sel94] V. Selivanov. Two refinements of the polynomial hierarchy. In *Proceedings of the 11th Annual Symposium on Theoretical Aspects of Computer Science*, pages 439–448. Springer-Verlag *Lecture Notes in Computer Science #775*, February 1994.

[Sto77] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.

[Wag] K. Wagner. A note on parallel queries and the symmetric-difference hierarchy. *Information Processing Letters*. To appear.

[Wag87] K. Wagner. Number-of-query hierarchies. Technical Report 158, Institut für Mathematik, Universität Augsburg, Augsburg, Germany, October 1987.

[Wag89] K. Wagner. Number-of-query hierarchies. Technical Report 4, Institut für Informatik, Bayerische Julius-Maximilians-Universität Würzburg, Würzburg, Germany, February 1989.

[Wag90] K. Wagner. Bounded query classes. *SIAM Journal on Computing*, 19(5):833–846, 1990.

[Wra77] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3:23–33, 1977.

Acknowledgements

We thank Gerd Wechsung for his warm encouragement, and we thank Lance Fortnow, Maren Hinrichs, Leen Torenvliet, and Gerd Wechsung for valuable conversations and suggestions. An extended abstract of this work appears in the proceedings of the 11th Conference on Fundamentals of Computation Theory. We thank anonymous referees for many helpful suggestions. This work was supported in part by grants NSF-INT-9513368/DAAD-315-PRO-fo-ab and

NSF-CCR-9322513, and was done in part while the first two authors were visiting Friedrich-Schiller-Universität Jena and while the third author was visiting Le Moyne College, Syracuse.