Unambiguous Computation: Boolean Hierarchies and Sparse Turing-Complete Sets

Lane A. Hemaspaandra∗
Department of Computer Science
University of Rochester
Rochester, NY 14627

Jörg Rothe†
Institut für Informatik
Friedrich-Schiller-Universität Jena
07743 Jena, Germany

Abstract

It is known that for any class $C$ closed under union and intersection, the Boolean closure of $C$, the Boolean hierarchy over $C$, and the symmetric difference hierarchy over $C$ all are equal. We prove that these equalities hold for any complexity class closed under intersection; in particular, they thus hold for unambiguous polynomial time (UP). In contrast to the NP case, we prove that the Hausdorff hierarchy and the nested difference hierarchy over UP both fail to capture the Boolean closure of UP in some relativized worlds.

Karp and Lipton proved that if nondeterministic polynomial time has sparse Turing-complete sets, then the polynomial hierarchy collapses. We establish the first consequences from the assumption that unambiguous polynomial time has sparse Turing-complete sets: (a) $\text{UP} \subseteq \text{Low}_2$, where $\text{Low}_2$ is the second level of the low hierarchy, and (b) each level of the unambiguous polynomial hierarchy is contained one level lower in the promise unambiguous polynomial hierarchy than is otherwise known to be the case.

1 Introduction

NP and NP-based hierarchies—such as the polynomial hierarchy [MS72, Sto77] and the Boolean hierarchy over NP [CGH+88, CGH+89, KSW87]—have played such a central role in complexity theory, and have been so thoroughly investigated, that it would be natural to take them as predictors of the behavior of other classes or hierarchies. However, over and over during the past decade it has

∗Supported in part by grants NSF-CCR-8957604, NSF-INT-9116781/JSPS-ENGR-207, and NSF-CCR-9322513, and by an NAS/NRC COBASE grant. Email: lane@cs.rochester.edu.

†Supported in part by a grant from the DAAD and by grant NSF-CCR-8957604. Work done in part while visiting the University of Rochester. Email: rothe@mipool.uni-jena.de.
been shown that NP is a singularly poor predictor of the behavior of other classes (and, to a lesser extent, that hierarchies built on NP are poor predictors of the behavior of other hierarchies).

As examples regarding hierarchies: though the polynomial hierarchy possesses downward separation (that is, if its low levels collapse, then all its levels collapse) \cite{MS72, Sto77}, downward separation does not hold “robustly” (i.e., in every relativized world) for the exponential time hierarchy \cite{HIS85, IT89} or for limited-nondeterminism hierarchies (\cite{HJ93}, see also \cite{BG94}). As examples regarding UP: NP has \( \leq_p^m \)-complete sets, but UP does not robustly possess \( \leq_p^m \)-complete sets \cite{HH88} or even \( \leq_p^T \)-complete sets \cite{HJV93}; NP positively relativizes, in the sense that it collapses to P if and only if it does so with respect to every tally oracle (\cite{LS86}, see also \cite{BBS86}), but UP does not robustly positively relativize \cite{HR92}; NP has “constructive programming systems,” but UP does not robustly have such systems \cite{Reg89}; NP (actually, nondeterministic computation) admits time hierarchy theorems \cite{HS65}, but it is an open question whether unambiguous computation has nontrivial time hierarchy theorems; NP displays upward separation (that is, NP \(- P \) contains sparse sets if and only if NE \( \neq \) E) \cite{HIS85}, but it is not known whether UP does (see \cite{HJ93}, which shows that R and BPP do not robustly display upward separation, and \cite{RRW94}, which shows that FewP does possess upward separation).

In light of the above list of the many ways in which NP parts company with UP, it is clear that we should not merely assume that results for NP hold for UP, but, rather, we must carefully check to see to what extent, if any, results for NP suggest results for UP. In this paper, we study, for UP, two topics that have been intensely studied for the NP case: the structure of Boolean hierarchies, and the effects of the existence of sparse Turing-complete/Turing-hard sets.

For the Boolean hierarchy over NP, which has generated quite a bit of interest and the collapse of which is known to imply the collapse of the polynomial hierarchy \cite{Kad88, CK90a, BCO93}, a large number of definitions are known to be equivalent. For example, for NP, all the following coincide \cite{CGH+88}: the Boolean closure of NP, the Boolean (alternating sums) hierarchy, the nested difference hierarchy, and the Hausdorff hierarchy. The symmetric difference hierarchy also characterizes the Boolean closure of NP \cite{KSW87}. In fact, these equalities are known to hold for all classes that contain \( \Sigma^* \) and \( \emptyset \) and are closed under union and intersection \cite{Hau14, CGH+88, KSW87, BBJ+89, GNW90, CK90c, Cha91}. In Section 3, we prove that both the symmetric difference hierarchy (SDH) and the Boolean hierarchy (CH) remain equal to the Boolean closure (BC) even in the absence of the assumption of closure under union. That is, for any class \( \mathcal{K} \) containing \( \Sigma^* \) and \( \emptyset \) and closed under intersection (e.g., UP, US, and DP, first defined respectively in \cite{Val76}, \cite{BG82}, and \cite{PY84} and each of which is not currently known to be closed under union): SDH(\( \mathcal{K} \)) = CH(\( \mathcal{K} \)) = BC(\( \mathcal{K} \)). However, for the remaining two hierarchies, we show that not all classes containing \( \Sigma^* \) and \( \emptyset \) and closed under intersection robustly display equality. In
particular, the Hausdorff hierarchy over UP and the nested difference hierarchy over UP both fail

to robustly capture the Boolean closure of UP. In fact, the failure is relatively severe; we show

that even low levels of other Boolean hierarchies over UP—the third level of the symmetric differ-

ence hierarchy and the fourth level of the Boolean (alternating sums) hierarchy—fail to be robustly

captured by either the Hausdorff hierarchy or the nested difference hierarchy.

It is well-known, thanks to the work of Karp and Lipton ([KL80], see also the related refer-

cences given in Section 4), that if NP has sparse Turing-hard sets, then the polynomial hierarchy
collapses. Unfortunately, the promise-like definition of UP—its unambiguity, the very core of its

nature—seems to block any similarly strong claim for UP and the unambiguous polynomial hier-
archy (which was introduced recently by Niedermeier and Rossmanith [NR93]). Section 4 studies

this issue, and shows that if UP has sparse Turing-complete sets, then the levels of the unambiguous
polynomial hierarchy “slip down” slightly in terms of their location within the promise unambigu-
ous polynomial hierarchy (a version of the unambiguous polynomial hierarchy that requires only

that computations actually executed be unambiguous), i.e., the $k$th level of the unambiguous poly-
nomial hierarchy is contained in the $(k - 1)$st level of the promise unambiguous polynomial hier-
archy. Various related results are also established. For example, if UP has Turing-hard sparse sets,
then (a) $\text{UP} \subseteq \text{Low}_2$, where $\text{Low}_2$ is the second level of the low hierarchy [Sch83], and (b) the $k$th
level of the unambiguous polynomial hierarchy can be accepted via a deterministic polynomial-time
Turing transducer given access to both a $\Sigma^p_2$ set and the $(k - 1)$st level of the promise unambiguous
polynomial hierarchy.

2 Notations

In general, we adopt the standard notations of Hopcroft and Ullman [HU79]. Fix the alphabet $\Sigma = \{0, 1\}$. $\Sigma^*$ is the set of all strings over $\Sigma$. For each string $u \in \Sigma^*$, $|u|$ denotes the length of $u$. The empty string is denoted by $\epsilon$. For each set $L \subseteq \Sigma^*$, $|L|$ denotes the cardinality of $L$ and $\overline{L} = \Sigma^* - L$ denotes the complement of $L$. $L^{=n}$ ($L^{\leq n}$) is the set of all strings in $L$ having length $n$ (less than or equal to $n$). Let $\Sigma^n$ and $\Sigma^{\leq n}$ be shorthands for $(\Sigma^*)^=n$ and $(\Sigma^*)^{\leq n}$, respectively. A set $S$ is said to be sparse if there is a polynomial $q$ such that for every $m \geq 0$, $|S^{\leq m}| \leq q(m)$. To encode a pair of strings, we use a polynomial-time computable pairing function, $\langle \cdot, \cdot \rangle : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$, that has polynomial-time computable inverses; this notion is extended to encode every $k$-tuple of strings, in the standard way. Let $\leq_{\text{lex}}$ denote the standard quasi-lexicographical ordering on $\Sigma^*$, that is, for strings $x$ and $y$, $x \leq_{\text{lex}} y$ if either $x = y$, or $|x| < |y|$, or ($|x| = |y|$ and there exists some $z \in \Sigma^*$ such that $x = z0u$ and $y = z1v$). $x <_{\text{lex}} y$ indicates that $x \leq_{\text{lex}} y$ but $x \neq y$.

For sets $A$ and $B$, their join, $A \oplus B$, is $\{0x \mid x \in A\} \cup \{1x \mid x \in B\}$, and their symmet
The Boolean hierarchy is a natural extension of the classes NP [Coo71, Lev73] and DP $\equiv$ NP $\land$ coNP [PY84]. Both NP and DP contain natural problems, as do the levels of the Boolean hierarchy. For example, graph minimal uncolorability is known to be complete for DP [CM87]. Note that DP clearly is closed under intersection, but is not closed under union unless the polynomial hierarchy collapses (due to [Kad88], see also [CK90, Cha91]).
Definition 3.1 [CGH⁺88, KSW⁺87, Hau14]. Let $\mathcal{K}$ be any class of sets.

1. The Boolean ("alternating sums") hierarchy over $\mathcal{K}$:
   
   $$
   C_1(\mathcal{K}) \overset{\text{df}}{=} \mathcal{K}, \quad C_k(\mathcal{K}) \overset{\text{df}}{=} \begin{cases} 
   C_{k-1}(\mathcal{K}) \cup \mathcal{K} & \text{if } k \text{ odd} \\
   C_{k-1}(\mathcal{K}) \cap \text{co}\mathcal{K} & \text{if } k \text{ even}
   \end{cases}, \quad k \geq 2, \quad \text{CH}(\mathcal{K}) \overset{\text{df}}{=} \bigcup_{k \geq 1} C_k(\mathcal{K}).
   $$

2. The nested difference hierarchy over $\mathcal{K}$:
   
   $$
   D_1(\mathcal{K}) \overset{\text{df}}{=} \mathcal{K}, \quad D_k(\mathcal{K}) \overset{\text{df}}{=} \mathcal{K} - D_{k-1}(\mathcal{K}), \quad k \geq 2, \quad \text{DH}(\mathcal{K}) \overset{\text{df}}{=} \bigcup_{k \geq 1} D_k(\mathcal{K}).
   $$

3. The Hausdorff ("union of differences") hierarchy over $\mathcal{K}$:
   
   $$
   E_1(\mathcal{K}) \overset{\text{df}}{=} \mathcal{K}, \quad E_2(\mathcal{K}) \overset{\text{df}}{=} \mathcal{K} - \mathcal{K}, \quad E_k(\mathcal{K}) \overset{\text{df}}{=} E_2(\mathcal{K}) \cup E_{k-2}(\mathcal{K}), \quad k \geq 2, \quad \text{EH}(\mathcal{K}) \overset{\text{df}}{=} \bigcup_{k \geq 1} E_k(\mathcal{K}).
   $$

4. The symmetric difference hierarchy over $\mathcal{K}$:
   
   $$
   SD_1(\mathcal{K}) \overset{\text{df}}{=} \mathcal{K}, \quad SD_k(\mathcal{K}) \overset{\text{df}}{=} SD_{k-1}(\mathcal{K}) \Delta \mathcal{K}, \quad k \geq 2, \quad \text{SDH}(\mathcal{K}) \overset{\text{df}}{=} \bigcup_{k \geq 1} SD_k(\mathcal{K}).
   $$

It is easily seen that for any $X$ chosen from $\{C, D, E, SD\}$, if $\mathcal{K}$ contains $\emptyset$ and $\Sigma^*$, then for any $k \geq 1$,

$$
X_k(\mathcal{K}) \cup \text{co}X_k(\mathcal{K}) \subseteq X_{k+1}(\mathcal{K}) \cap \text{co}X_{k+1}(\mathcal{K}).
$$

The following fact is shown by an easy induction on $n$.

Fact 3.2 For every class $\mathcal{K}$ of sets and every $n \geq 1$, (a) $D_{2n-1}(\mathcal{K}) = \text{co}C_{2n-1}(\text{co}\mathcal{K})$, and (b) $D_{2n}(\mathcal{K}) = C_{2n}(\text{co}\mathcal{K})$.

Proof. The base case holds by definition. Suppose (a) and (b) to be true for $n \geq 1$. Then,

$$
D_{2n+1}(\mathcal{K}) = \mathcal{K} \land (\text{co}\mathcal{K} \lor D_{2n-1}(\mathcal{K})) \overset{\text{hyp.}}{=} \mathcal{K} \land (\text{co}\mathcal{K} \lor \text{co}C_{2n-1}(\text{co}\mathcal{K}))
$$

$$
= \mathcal{K} \land \text{co}(\mathcal{K} \land C_{2n-1}(\text{co}\mathcal{K})) = \mathcal{K} \land \text{co}C_{2n}(\text{co}\mathcal{K})
$$

$$
= \text{co}(\text{co}\mathcal{K} \lor C_{2n}(\text{co}\mathcal{K})) = \text{co}C_{2n+1}(\text{co}\mathcal{K})
$$

shows (a) for $n + 1$, and

$$
D_{2n+2}(\mathcal{K}) = \mathcal{K} - (\mathcal{K} - D_{2n}(\mathcal{K})) \overset{\text{hyp.}}{=} \mathcal{K} \land (\text{co}\mathcal{K} \lor C_{2n}(\text{co}\mathcal{K})) = C_{2n+2}(\text{co}\mathcal{K})
$$

shows (b) for $n + 1$.  

\footnote{Hausdorff hierarchies \cite{Hau14}, see \cite{CGH⁺88, BBJ⁺89, NW90}, respectively, for applications to $\text{NP}$, $\text{R}$, and $\text{C-P}$, are interesting both in the case where, as in the definition here, the sets are arbitrary sets from $\mathcal{K}$, and, as is sometimes used in definitions, the sets from $\mathcal{K}$ are required to satisfy additional containment conditions. For classes closed under union and intersection, such as $\text{NP}$, the two definitions are identical, level by level \cite{Hau14}. In this paper, as, e.g., $\text{UP}$, is not known to be closed under union, the distinction is nontrivial.}
**Corollary 3.3** \(\text{CH}(\text{UP}) = \text{coCH}(\text{UP}) = \text{DH}(\text{coUP})\) and \(\text{CH}(\text{coUP}) = \text{coCH}(\text{coUP}) = \text{DH}(\text{UP})\).

We are interested in the Boolean hierarchies over classes closed under intersection (but perhaps not under union or complementation), such as UP, US, and DP. We state our theorems in terms of the class of primary interest to us in this paper, UP. However, many apply to any nontrivial class (i.e., any class containing \(\Sigma^*\) and \(\emptyset\)) closed under intersection (see Theorem 3.10). Although it has been proven in [CGH+88] and [KSW87] that all the standard normal forms of Definition 3.1 coincide for NP, the situation for UP seems to be different, as UP is probably not closed under union. (The closure of UP under intersection is straightforward.) Thus, all the relations among those normal forms have to be reconsidered for UP.

We first prove that the symmetric difference hierarchy over UP (or any class closed under intersection) equals the Boolean closure. Though Köbler, Schöning, and Wagner [KSW87] proved this for NP, their proof gateways through a class whose proof of equivalence to the Boolean closure uses closure under union, and thus the following result is not implicit in their paper.

**Theorem 3.4** \(\text{SDH}(\text{UP}) = \text{BC}(\text{UP})\).

**Proof.** The inclusion from left to right is clear. For the converse inclusion, it is sufficient to show that \(\text{SDH}(\text{UP})\) is closed under all Boolean operations, as \(\text{BC}(\text{UP})\), by definition, is the smallest class of sets that contains UP and is closed under all Boolean operations. Let \(L\) and \(L'\) be arbitrary sets in \(\text{SDH}(\text{UP})\). Then, for some \(k, \ell \geq 1\), there are sets \(A_1, \ldots, A_k, B_1, \ldots, B_\ell\) in UP representing \(L\) and \(L'\):

\[
L = A_1 \Delta \cdots \Delta A_k \quad \text{and} \quad L' = B_1 \Delta \cdots \Delta B_\ell.
\]

So

\[
L \cap L' = \left(\Delta_{i=1}^k A_i \right) \cap \left(\Delta_{j=1}^\ell B_j \right) = \Delta_{i \in \{1, \ldots, k\}, j \in \{1, \ldots, \ell\}} (A_i \cap B_j),
\]

and since UP is closed under intersection and \(\text{SDH}(\text{UP})\) is (trivially) closed under symmetric difference, we clearly have that \(L \cap L' \in \text{SDH}(\text{UP})\). Furthermore, since \(\overline{T} = \Sigma^* \Delta L\) implies that \(\overline{T} \in \text{SDH}(\text{UP})\), \(\text{SDH}(\text{UP})\) is closed under complementation. Since all Boolean operations can be represented in terms of complementation and intersection, our proof is complete. \(\square\)

Next, we show that for any class closed under intersection, instantiated below to the case of UP, the Boolean (alternating sums) hierarchy over the class equals the Boolean closure of the class. Our proof is inspired by the techniques used to prove equality in the case where closure under union may be assumed.

---

\(^2\)Due essentially to its closure under union and intersection, and this reflects a more general behavior of classes closed under union and intersection, as studied by Bertoni et al. ([BBJ+89], see also [Hau14, CGH+88, KSW87, CK90b, Cha91]).
Theorem 3.5 \[ \text{CH(UP)} = \text{BC(UP)}. \]

Proof. We will prove that SDH(UP) \( \subseteq \text{CH(UP)}. \) By Theorem 3.4, this will suffice.

Let \( L \) be any set in SDH(UP). Then there is a \( k > 1 \) (the case \( k = 1 \) is trivial) such that \( L \in \text{SD}_k(\text{UP}). \) Let \( U_1, \ldots, U_k \) be the witnessing UP sets; that is, \( L = U_1 \Delta U_2 \Delta \cdots \Delta U_k. \) By the inclusion-exclusion rule, \( L \) satisfies the equalities below. For odd \( k, \)

\[
L = \left( \ldots \left( \left( U_1 \cup U_2 \cdots \cup U_k \right) \cap \left( \bigcup_{j_1 < j_2} (U_{j_1} \cap U_{j_2} \cap U_{j_3}) \right) \right) \cup \left( \bigcup_{j_1 < j_2 < j_3} (U_{j_1} \cap U_{j_2} \cap U_{j_3}) \right) \right) \cap \cdots \cap \left( \bigcup_{j_1 < \cdots < j_k} (U_{j_1} \cap \cdots \cap U_{j_k}) \right),
\]

where each subscripted \( j \) term must belong to \( \{1, \ldots, k\}. \) For even \( k, \) we similarly have:

\[
L = \left( \ldots \left( \left( U_1 \cup U_2 \cdots \cup U_k \right) \cap \left( \bigcup_{j_1 < j_2} (U_{j_1} \cap U_{j_2} \cap U_{j_3}) \right) \right) \cup \left( \bigcup_{j_1 < j_2 < j_3} (U_{j_1} \cap U_{j_2} \cap U_{j_3}) \right) \right) \cap \cdots \cap \left( \bigcup_{j_1 < \cdots < j_k} (U_{j_1} \cap \cdots \cap U_{j_k}) \right).
\]

For notational convenience, let us use \( A_1, \ldots, A_k \) to represent the respective terms in the above expressions (ignoring the complementations). By the closure of UP under intersection, each \( A_i, \) \( 1 \leq i \leq k, \) is the union of \( \binom{k}{i} \) UP sets \( B_{i,1}, \ldots, B_{i,\binom{k}{i}}. \) Using the fact that \( \emptyset \) is clearly in UP, we can easily turn the union of \( n \) arbitrary UP sets (or the intersection of \( n \) arbitrary coUP sets) into an alternating sum of \( 2n - 1 \) UP sets. So for instance, \( A_1 = U_1 \cup U_2 \cup \cdots \cup U_k \) can be written

\[
\left( \cdots \left( \left( (U_1 \cap \emptyset) \cup U_2 \right) \cap \emptyset \right) \cup \cdots \cup U_k \right),
\]
call this \( C_1. \) Clearly, \( C_1 \in \text{C}_{2k-1}(\text{UP}). \) To transform the above representation of \( L \) into an alternating sum of UP sets, we need two (trivial) transformations holding for any \( m \geq 1 \) and for arbitrary sets \( S \) and \( T_1, \ldots, T_m:\)

\[
S \cap (T_1 \cup T_2 \cdots \cup T_m) = \left( \cdots \left( S \cap T_1 \right) \cap T_2 \right) \cap \cdots \cap T_m \tag{1}
\]

\[
S \cup (T_1 \cup T_2 \cdots \cup T_m) = \left( \cdots ((S \cup T_1) \cup T_2) \cup \cdots \right) \cup T_m. \tag{2}
\]

Using (1) with \( S = C_1 \) and \( T_1 = B_{2,1}, \ldots, T_m = B_{2,\binom{k}{1}} \) and the fact that \( \emptyset \) is in UP, \( A_1 \cap \overline{A_2} \) can be transformed into an alternating sum of UP sets, call this \( C_2. \) Now apply (2) with \( S = C_2 \) and \( T_1 = B_{3,1}, \ldots, T_m = B_{3,\binom{k}{2}} \) to obtain, again using that \( \emptyset \) is in UP, an alternating sum \( C_3 = \left( A_1 \cap \overline{A_2} \right) \cup A_3 \) of UP sets, and so on. Eventually, this procedure of alternately applying (1) and (2) will yield an alternating sum \( C_k \) of sets in UP that equals \( L. \) Thus, \( L \in \text{CH(UP)}. \) \( \square \)
Corollary 3.6  SDH(UP) and CH(UP) are both closed under all Boolean operations.

Note that the proofs of Theorems 3.5 and 3.4 implicitly give a recurrence yielding an upper bound on the level-wise containments. We find the issue of equality to BC(UP), or lack thereof, to be the central issue, and thus we focus on that. Nonetheless, we point out in the corollary below that losing the assumption of closure under union seems to have exacted a price: though the hierarchies SDH(UP) and CH(UP) are indeed equal, the above proof embeds SD_k(UP) in an exponentially higher level of the C hierarchy. Similarly, the proof of Theorem 3.4 embeds C_k(UP) in an exponentially higher level of SDH(UP).

Corollary 3.7 (to the proofs of Theorems 3.5 and 3.4)

1. For each \( k \geq 1 \), \( \text{SD}_k(\text{UP}) \subseteq \text{C}_{2^k-1+k-2}(\text{UP}) \).

2. For each \( k \geq 1 \), \( \text{C}_k(\text{UP}) \subseteq \text{SD}_{T(k)}(\text{UP}) \), where \( T(k) = \begin{cases} 2^k - 1 & \text{if } k \text{ is odd} \\ 2^k - 2 & \text{if } k \text{ is even.} \end{cases} \)

Proof. For an \( \text{SD}_k(\text{UP}) \) set \( L \) to be placed into the \( R(k) \)th level of CH(UP), \( L \) is represented (in the proof of Theorem 3.5) as an alternating sum of \( k \) terms \( A_1, \ldots, A_k \), each \( A_i \) consisting of \( \binom{k}{i} \) UP sets \( B_{i,j} \). In the subsequent transformation of \( L \) according to the equations (1) and (2), each \( A_i \) requires as many as \( \binom{k}{i} - 1 \) additional terms \( \emptyset \) or \( \overline{\emptyset} \), respectively, to be inserted, and each such insertion brings us one level higher in the C hierarchy. Thus,

\[
R(k) = \sum_{i=1}^{k} \binom{k}{i} + \left( \binom{k}{i} - 1 \right) = -k + 2 \sum_{i=1}^{k} \binom{k}{i} = 2^{k+1} - k - 2.
\]

A close inspection of the proof of \( \text{C}_k(\text{UP}) \subseteq \text{SD}_{T(k)}(\text{UP}) \) according to Theorem 3.4 leads to the recurrence:

\[
T(1) = 1 \quad \text{and} \quad T(k) = \begin{cases} 2T(k-1) + 3 & \text{if } k > 1 \text{ is odd} \\ 2T(k-1) & \text{if } k > 1 \text{ is even,} \end{cases}
\]

since any set \( L \in \text{C}_k(\text{UP}) \) can be represented by sets \( A \in \text{C}_{k-1}(\text{UP}) \) and \( B \in \text{UP} \) as follows:

\[
L = A \cup B = \overline{A} \cap B = \Sigma^* \Delta (\Sigma^* \Delta A) \cap (\Sigma^* \Delta B) & \quad \text{if } k \text{ is odd,} \\
L = A \cap \overline{B} = A \cap (\Sigma^* \Delta B) & \quad \text{if } k \text{ is even.}
\]

The above recurrence is in (almost) closed form:

\[
T(k) = \begin{cases} 2^k - 1 & \text{if } k \geq 1 \text{ is odd} \\ 2^k - 2 & \text{if } k \geq 1 \text{ is even,} \end{cases}
\]
as can be proven by induction on \( k \) (we omit the trivial induction base): For odd \( k \) (i.e., \( k = 2n - 1 \) for \( n \geq 1 \), assume \( T(2n - 1) = 2^{2n-1} - 1 \) to be true. Then,

\[
T(2n + 1) = 2T(2n) + 3 = 4T(2n - 1) + 3 \overset{\text{hyp.}}{=} 4 \left( 2^{2n-1} - 1 \right) + 3 = 2^{2n+1} - 1.
\]

For even \( k \) (i.e., \( k = 2n \) for \( n \geq 1 \)), assume \( T(2n) = 2^{2n} - 2 \) to be true. Then,

\[
T(2n + 2) = 2T(2n + 1) = 2(2T(2n) + 3) \overset{\text{hyp.}}{=} 4 \left( 2^{2n} - 2 \right) + 6 = 2^{2n+2} - 2.
\] 

**Remark 3.8** The upper bound in the second part of the above proof can be slightly improved using the fact that \( \Sigma^* \Sigma^* \Delta \Sigma^* \Delta A = \emptyset \Delta A = A \) for any set \( A \). This gives the recurrence:

\[
T(1) = 1 \quad \text{and} \quad T(k) = \begin{cases} 
2T(k - 1) + 1 & \text{if } k > 1 \text{ is odd} \\
2T(k - 1) & \text{if } k > 1 \text{ is even},
\end{cases}
\]

or, equivalently, \( T(1) = 1, T(2) = 2, \) and \( T(k) = 2^{k-1} + T(k - 2) \) for \( k \geq 3 \). Though this shows that the upper bound given in the above proof is not optimal, the new bound is not a strong improvement, as it still embeds \( C_k(\text{UP}) \) in an exponentially higher level of SDH(UP). We propose as an interesting task the establishment of tight level-wise containments, at least up to the limits of relativizing techniques, between the hierarchies SDH(UP) and CH(UP), both of which capture the Boolean closure of UP.

We conjecture that there is some relativized world in which an exponential increase (though less dramatic than the particular exponential increase of Corollary 3.7) indeed is necessary.

Theorem 3.9 below shows that each level of the nested difference hierarchy is contained in the same level of both the C and the E hierarchy. Surprisingly, it turns out (see Theorem 3.13 below) that, relative to a recursive oracle, even the fourth level of CH(UP) and the third level of SDH(UP) are not subsumed by any level of the EH(UP) hierarchy. Consequently, neither the D nor the E normal forms of Definition 3.1 capture the Boolean closure of UP.

**Theorem 3.9** For every \( k \geq 1 \), \( D_k(\text{UP}) \subseteq C_k(\text{UP}) \cap E_k(\text{UP}) \).

**Proof.** For the first inclusion, by [CH85, Proposition 2.1.2], each set \( L \in D_k(\text{UP}) \) can be represented as

\[
L = A_1 - (A_2 - (\cdots (A_{k-1} - A_k) \cdots)),
\]

where \( A_i = \bigcap_{1 \leq j \leq i} L_j \), \( 1 \leq i \leq k \), and the \( L_j \)'s are the original UP sets representing \( L \). Note that since the proof of [CH85, Proposition 2.1.2] only uses intersection, the sets \( A_i \) are in UP. A special
case of [CH85, Proposition 2.1.3] says that sets in $D_k(UP)$ via decreasing chains such as the $A_i$ are in $C_k(UP)$, and so $L \in C_k(UP)$.

The proof of the second inclusion is done by induction on the odd and even levels separately. The induction base follows by definition in either case. For odd levels, assume $D_{2n-1}(UP) \subseteq E_{2n-1}(UP)$ to be valid, and let $L$ be any set in $D_{2n+1}(UP) = UP - (UP - D_{2n-1}(UP))$. By our inductive hypothesis, $L$ can be represented as

$$L = A - \left( B - \left( \bigcup_{i=1}^{n-1} \left( C_i \cap \overline{D_i} \right) \cup E \right) \right),$$

where $A, B, C_i, D_i$, and $E$ are sets in $UP$. Thus,

$$L = A \cap \left( B \cap \left( \bigcup_{i=1}^{n-1} \left( C_i \cap \overline{D_i} \right) \cup E \right) \right)$$

$$= A \cap \left( \overline{B} \cup \left( \bigcup_{i=1}^{n-1} \left( C_i \cap \overline{D_i} \right) \cup E \right) \right)$$

$$= (A \cap \overline{B}) \cup \left( \bigcup_{i=1}^{n-1} A \cap C_i \cap \overline{D_i} \right) \cup (A \cap E)$$

$$= \left( \bigcup_{i=1}^{n} F_i \cap \overline{D_i} \right) \cup G,$$

where $F_i = A \cap C_i$, for $1 \leq i \leq n - 1$, $F_n = A$, $D_n = B$, and $G = A \cap E$. Since $UP$ is closed under intersection, each of these sets is in $UP$. Thus, $L \in E_{2n+1}(UP)$. The proof for the even levels is analogous except that the set $E$ is dropped. \qed

Note that most of the above proofs used only the facts that the class is closed under intersection and contains $\Sigma^*$ and $\emptyset$:

**Theorem 3.10** Theorems 3.4, 3.5, and 3.9 and Corollaries 3.6 and 3.7 apply to all classes that contain $\Sigma^*$ and $\emptyset$ and are closed under intersection.

**Remark 3.11** Although $DP$ is closed under intersection but seems to lack closure under union (unless the polynomial hierarchy collapses to $DP$ [Kad88, CK90b, Cha91]) and thus Theorem 3.10 in particular applies to $DP$, we note that the known results about Boolean hierarchies over $NP$ [CGH+88, KSW87] in fact even for the $DP$ case imply stronger results than those given by our Theorem 3.10 due to the very special structure of $DP$. Indeed, since, e.g., $E_k(DP) = E_{2k}(NP)$ for any $k \geq 1$ (and the same holds for the other hierarchies), it follows immediately that all the
level-wise equivalences among the Boolean hierarchies (and also their ability to capture the Boolean closure) that are known to hold for NP also hold for DP even in the absence of the assumption of closure under union. This appears to contrast with the UP case (see Remark 3.8).

The following combinatorial lemma will be useful in proving Theorem 3.13.

**Lemma 3.12 [CHV93]** Let $G = (S, T, E)$ be any directed bipartite graph with out-degree bounded by $d$ for all vertices. Let $S' \subseteq S$ and $T' \subseteq T$ be subsets such that $S' \supseteq \{s \in S \mid (\exists t \in T) [\langle s, t \rangle \in E]\}$, and $T' \supseteq \{t \in T \mid (\exists s \in S) [\langle t, s \rangle \in E]\}$. Then either:

1. $\|S'\| \leq 2d$, or
2. $\|T'\| \leq 2d$, or
3. $(\exists s \in S') (\exists t \in T') [\langle s, t \rangle \notin E \land \langle t, s \rangle \notin E]$.

For papers concerned with oracles separating internal levels of Boolean hierarchies over classes other than those of this paper, we refer the reader to ([CGH+88, Cai87, GNW90, BJY90, Cro94], see also [GW87]). Theorem 3.13 is optimal, as clearly $C_3(UP) \subseteq EH(UP)$ and $SD_2(UP) \subseteq EH(UP)$, and both these containments relativize.

**Theorem 3.13** There are recursive oracles $A$ and $D$ (though we may take $A = D$) such that

1. $C_4(UP^A) \not\subseteq EH(UP^A)$, and
2. $SD_3(UP^D) \not\subseteq EH(UP^D)$.

**Corollary 3.14** There is a recursive oracle $A$ such that

1. $EH(UP^A) \neq BC(UP^A)$ and $DH(UP^A) \neq BC(UP^A)$, and
2. $EH(UP^A)$ and $DH(UP^A)$ are not closed under all Boolean operations.

**Proof of Theorem 3.13.** Although the theorem claims there is an oracle keeping $C_4(UP)$ from being contained in any level of $EH(UP)$, we will only prove that for any fixed $k$ we can ensure that $C_4(UP)$ is not contained in $E_k(UP)$, relative to some oracle $A^{(k)}$. In the standard way, by interleaving diagonalizations, the sequence of oracles, $A^{(k)}$, can be combined into a single oracle, $A$, that fulfills the claim of the theorem. An analogous comment holds for the second claim of the

\(^3\)As Fact 3.2 shows that $DH(UP) = CH(coUP)$, this oracle $A$ also separates the Boolean (alternating sums) hierarchy over $coUP$ from the fourth level of the same hierarchy over UP and, thus, from $BC(UP)$.
theorem, with a sequence of oracles $D^{(k)}$ yielding a single oracle $D$. Similarly, both statements of the theorem can be satisfied simultaneously via just one oracle, via interleaving with each other the constructions of $A$ and $D$. Though below we construct just $A^{(k)}$ and $D^{(k)}$, as a notational shorthand we’ll use $A$ and $D$ below to represent $A^{(k)}$ and $D^{(k)}$.

Before the actual construction of the oracles, we state some preliminaries that apply to the proofs of both statements in the theorem.

For any $n \geq 0$ and any string $v \in \Sigma^{\leq n}$, define $S^n_v \overset{\text{df}}{=} \{vw | vw \in \Sigma^n\}$. The sets $S^n_v$ are used to distinguish between different segments of $\Sigma^n$ in the definition of the test languages, $L_A$ and $L_D$.

Fix any standard enumeration of all NPOMs. Fix any $k > 0$. We need only consider even levels of EH(UP), as each odd level is contained in some even level. Call any collection of $2k$ NPOMs, $H = \langle N_{1,1}, \ldots, N_{k,1}, N_{1,2}, \ldots, N_{k,2} \rangle$, a potential (relativized) $E_{2k}(\text{UP})$ machine, and for any oracle $X$, define its language to be:

$$L(H^X) \overset{\text{df}}{=} \bigcup_{i=1}^{k} \left( L(N_{i,1}^X) - L(N_{i,2}^X) \right).$$

If for some fixed oracle $Y$, a potential (relativized) $E_{2k}(\text{UP})$ machine $H^Y$ has the property that each of its underlying NPOMs with oracle $Y$ is unambiguous, then $L(H^Y)$ indeed is in $E_{2k}(\text{UP}^Y)$. Clearly, our enumeration of all NPOMs induces an enumeration of all potential $E_{2k}(\text{UP})$ oracle machines. For $j \geq 1$, let $H_j$ be the $j$th machine in this enumeration. Let $p_j$ be a polynomial bounding the length of the computation paths of each of $H_j$’s underlying machines (and thus bounding the number of and length of the strings they each query). As a notational convenience, we henceforward will use $H$ and $p$ as shorthands for $H_j$ and $p_j$, and we will denote the underlying NPOMs by $N_{1,1}, \ldots, N_{k,1}, N_{1,2}, \ldots, N_{k,2}$.

The oracle $X$, where $X$ stands for $A$ or $D$, is constructed in stages, $X = \bigcup_{j \geq 1} X_j$. In stage $j$, we diagonalize against $H$ by satisfying the following requirement $R_j$ for every $j \geq 1$:

$$R_j : \text{Either there is an } n > 2 \text{ and an } i, 1 \leq i \leq k, \text{ such that one of } N_{i,1}^X_j \text{ or } N_{i,2}^X_j \text{ on input } 0^n \text{ is ambiguous (thus, } H \text{ is in fact not an } E_{2k}(\text{UP}) \text{ machine relative to } X), \text{ or } L(H^X) \neq L_X.$$ 

Let $X_j$ be the set of strings contained in $X$ by the end of stage $j$, and let $X_j'$ be the set of strings forbidden membership in $X$ during stage $j$. The restraint function $r(j)$ will satisfy the condition that at no later stage will strings of length smaller than $r(j)$ be added to $X$. Also, our construction will ensure that $r(j)$ is so large that $X_{j-1}$ contains no strings of length greater than $r(j)$. Initially, both $X_0$ and $X_0'$ are empty, and $r(1)$ is set to be 2.

We now start the proof of Part 3 of the theorem. Define the test language:

$$L_A \overset{\text{df}}{=} \{0^n | (\exists x) [x \in S^n_0 \cap A \land (\forall y) [y \not\in S^n_{10} \cap A] \land (\forall z) [z \not\in S^n_{11} \cap A]\}. $$
Clearly, $L_A$ is in $\text{NP}^A \land \text{coNP}^A \land \text{coNP}^A$. However, if we ensure in the construction that the invariant $\|S^n_v \cap A\| \leq 1$ is maintained for $v \in \{0, 10, 11\}$ and every $n \geq 2$, then $L_A$ is even in $\text{UP}^A \land \text{coUP}^A \land \text{coUP}^A$, and thus in $C_4(\text{UP}^A)$. We now describe stage $j > 0$ of the oracle construction.

**Stage $j$:** Choose $n > r(j)$ so large that $2^{n-2} > 3p(n)$.

**Case 1:** $0^n \in L(H^{A_{j-1}})$. Since $0^n \not\in L_A$, we have $L(H^A) \neq L_A$.

**Case 2:** $0^n \not\in L(H^{A_{j-1}})$. Choose some $x \in S^n_0$ and set $B_j := A_{j-1} \cup \{x\}$.

**Case 2.1:** $0^n \not\in L(H^{B_j})$. Letting $A_j := B_j$ implies $0^n \in L_A$, so $L(H^A) \neq L_A$.

**Case 2.2:** $0^n \in L(H^{B_j})$. Then there is an $i, 1 \leq i \leq k$, such that $0^n \in L(N^i_{i, 1}^{B_j})$ and $0^n \not\in L(N^i_{i, 2}^{B_j})$.

"Freeze" an accepting path of $N^i_{i, 1}^{B_j}(0^n)$ into $A_j'$; that is, add those strings queried negatively on that path to $A_j'$, thus forbidding them from $A$ for all later stages. Clearly, at most $p(n)$ strings are “frozen.”

**Case 2.2.1:** $(\exists z \in (S^n_{10} \cup S^n_{11}) - A_j') \left[0^n \not\in L(N^i_{i, 2}^{B_j} \cup \{z\})\right]$.

Choose any such $z$. Set $A_j := B_j \cup \{z\}$. We have $0^n \in L(H^A) - L_A$.

**Case 2.2.2:** $(\forall z \in (S^n_{10} \cup S^n_{11}) - A_j') \left[0^n \in L(N^i_{i, 2}^{B_j} \cup \{z\})\right]$.

To apply Lemma 3.12, define a directed bipartite graph $G = (S, T, E)$ by $S \overset{df}{=} S^n_{10} - A_j', T \overset{df}{=} S^n_{11} - A_j'$, and for each $s \in S$ and $t \in T$, $(s, t) \in E$ if and only if $N^i_{i, 2}^{B_j \cup \{s\}}$ queries $t$ along its lexicographically first accepting path, and $(t, s) \in E$ is defined analogously. The out-degree of all vertices of $G$ is bounded by $p(n)$. By our choice of $n$, $\min\{\|S\|, \|T\|\} \geq 2^{n-2} - p(n) > 2p(n)$, and thus alternative 3 of Lemma 3.12 applies. Hence, there exist strings $s \in S$ and $t \in T$ such that $N^i_{i, 2}^{B_j \cup \{s\}}(0^n)$ accepts on some path $p_s$ on which $t$ is not queried, and $N^i_{i, 2}^{B_j \cup \{t\}}(0^n)$ accepts on some path $p_t$ on which $s$ is not queried. Since $p_s$ (or $p_t$) changes from reject to accept exactly by adding $s$ (or $t$) to the oracle, $s$ (or $t$) must have been queried on $p_s$ (or $p_t$). We conclude that $p_s \neq p_t$, and thus $N^i_{i, 2}^{B_j \cup \{s, t\}}(0^n)$ has at least two accepting paths. Set $A_j := B_j \cup \{s, t\}$.

In each case, requirement $R_j$ is fulfilled. Let $r(j + 1) = \max\{n, w_j\}$, where $w_j$ is the length of the largest string queried through stage $j$.

**End of stage $j$.**
We now turn to the proof of Part 2 of the theorem. The test language here, \( L_D \), is defined by:

\[
L_D \overset{\text{df}}{=} \left\{ 0^n \mid ((\exists x) [x \in S_0^n \cap D] \land (\exists y) [y \in S_0^n \cap D] \land (\exists z) [z \in S_0^n \cap D]) \lor ((\forall x) [x \notin S_0^n \cap D] \land (\forall y) [y \notin S_0^n \cap D] \land (\forall z) [z \in S_0^n \cap D]) \lor ((\exists x) [x \in S_0^n \cap D] \land (\forall y) [y \notin S_0^n \cap D] \land (\forall z) [z \notin S_0^n \cap D]) \lor ((\forall x) [x \notin S_0^n \cap D] \land (\exists y) [y \in S_0^n \cap D] \land (\forall z) [z \notin S_0^n \cap D]) \right\}.
\]

Again, provided that the invariant \( ||S_0^n \cap D|| \leq 1 \) is maintained for \( v \in \{0, 10, 11\} \) and every \( n \geq 2 \) throughout the construction, \( L_D \) is clearly in \( \text{SD}_3(\text{UP}^D) \), as for all sets \( A, B, \) and \( C \),

\[
A \Delta B \Delta C = (A \cap B \cap C) \cup (A \cap \overline{B} \cap C) \cup (A \cap B \cap \overline{C}) \cup (A \cap \overline{B} \cap \overline{C}).
\]

Stage \( j > 0 \) of the construction of \( D \) is as follows.

**Stage \( j \):** Choose \( n > r(j) \) so large that \( 2^{n-2} > 3p(n) \).

1. **Case 1:** \( 0^n \in L(H^{D_{j-1}}) \). Since \( 0^n \notin L_D \), we have \( L(H^D) \neq L_D \).

2. **Case 2:** \( 0^n \notin L(H^{D_{j-1}}) \). Choose some \( x \in S_0^n \) and set \( E_j := D_{j-1} \cup \{x\} \).

   **Case 2.1:** \( 0^n \notin L(H^{E_j}) \). Letting \( D_j := E_j \) implies \( 0^n \in L_D \), so \( L(H^D) \neq L_D \).

   **Case 2.2:** \( 0^n \in L(H^{E_j}) \). Then, there is an \( i, \, 1 \leq i \leq k \), such that \( 0^n \in L(N_{i,1}^{E_j}) \) and \( 0^n \notin L(N_{i,2}^{E_j}) \). “Freeze” an accepting path of \( N_{i,1}^{E_j}(0^n) \) into \( D_j' \). Again, at most \( p(n) \) strings are “frozen.”

   **Case 2.2.1:** \( \exists w \in (S_0^n \cup S_1^n) - D_j' \left[ 0^n \notin L(N_{i,2}^{E_j}(w)) \right] \).

   Choose any such \( w \) and set \( D_j := E_j \cup \{w\} \). We have \( 0^n \in L(H^D) - L_D \).

   **Case 2.2.2:** \( \forall w \in (S_0^n \cup S_1^n) - D_j' \left[ 0^n \in L(N_{i,2}^{E_j}(w)) \right] \).

   As before, Lemma 3.12 yields two strings \( s \in S_0^n - D_j' \) and \( t \in S_1^n - D_j' \) such that \( N_{i,2}^{E_j}(s,t)(0^n) \) is ambiguous. Set \( D_j := E_j \cup \{s, t\} \).

Again, \( R_j \) is always fulfilled. Define \( r(j + 1) \) as before.

**End of stage \( j \).**

Finally, we note that a slight modification of the above proof establishes the analogous result (of Theorem 3.13) for the case of \( \text{US} \) [BG82] (which is denoted 1NP in [GW87, Cro94]).

## 4 Sparse Turing-complete and Turing-hard Sets for UP

In this section, we show some consequences of the existence of sparse Turing-complete and Turing-hard sets for UP. This question has been carefully investigated for the class NP [KL80, H88].

14
Kadin showed that if there is a sparse \( \leq^p_{L} \)-complete set in NP, then the polynomial hierarchy collapses to \( \mathsf{P}^{\mathsf{NP}^{[\log]}} \) [Kad89]. Due to the promise nature of UP (in particular, UP probably lacks complete sets [HH88]), Kadin’s proof does not seem to apply here. But does the existence of a sparse Turing-complete set in UP cause at least some collapse of the unambiguous polynomial hierarchy (which was introduced recently in [NR93])?\footnote{For reductions less flexible than Turing reductions (e.g., \( \leq^p_{L}, \leq^p_{LT} \), etc.), this issue has been studied even more intensely (see, e.g., the surveys [You92, HOW92]).}

Cai, Hemachandra, and Vyskoč [CHV93] observe that ordinary Turing access to UP, as formalized by \( \mathsf{P}^{\mathsf{UP}} \), may be too restrictive a notion to capture adequately one’s intuition of Turing access to unambiguous computation, since in that model the oracle machine has to be unambiguously every input—even those the base DPOM never asks (on any of its inputs). To relax that unnaturally strong uniformity requirement they introduce the class denoted \( \mathsf{P}^{\mathsf{UP}^0} \), in which NP oracles are accessed in a \textit{guardedly} unambiguous manner, a natural notion of access to unambiguous computation—suggested in the rather analogous case of \( \mathsf{NP} \cap \mathsf{coNP} \) by Grollmann and Selman [GS88]—in which \textit{only computations actually executed need be unambiguous}. Lange, Niedermeier, and Rossmanith [LR94] [NR93, p. 483] generalize this approach to build up an entire hierarchy of unambiguous computations in which the oracle levels are guardedly accessed (Definition 4.1, Part 3)—the \textit{promise unambiguous polynomial hierarchy}.

**Definition 4.1**

1. The \textit{polynomial hierarchy} [MS72, Sto77] is defined as follows:
   \[
   \Sigma^p_0 \equiv \mathsf{P}, \quad \Delta^p_0 \equiv \mathsf{P}, \quad \Sigma^p_k \equiv \mathsf{NP}^{\Sigma^p_{k-1}}, \quad \Pi^p_k \equiv \mathsf{coNP}^p_k, \quad \Delta^p_k \equiv \mathsf{P}^{\Sigma^p_{k-1}}, \quad k \geq 1, \quad \text{and} \quad \mathsf{PH} \equiv \bigcup_{k \geq 0} \Sigma^p_k.
   \]

2. The \textit{unambiguous polynomial hierarchy} [NR93] is defined as follows:
   \[
   \mathsf{U} \Sigma^p_0 \equiv \mathsf{P}, \quad \mathsf{U} \Delta^p_0 \equiv \mathsf{P}, \quad \mathsf{U} \Sigma^p_k \equiv \mathsf{UP}^{\Sigma^p_{k-1}}, \quad \mathsf{U} \Pi^p_k \equiv \mathsf{coUP}^p_k, \quad \mathsf{U} \Delta^p_k \equiv \mathsf{P}^{\Sigma^p_{k-1}}, \quad k \geq 1, \quad \text{and} \quad \mathsf{UPH} \equiv \bigcup_{k \geq 0} \mathsf{U} \Sigma^p_k.
   \]

3. The \textit{promise unambiguous polynomial hierarchy} ([LR94] [NR93, p. 483]) is defined as follows:
   \[
   \mathsf{U} \Sigma^p_0 \equiv \mathsf{P}, \quad \mathsf{U} \Delta^p_1 \equiv \mathsf{UP}, \quad \text{and for } k \geq 2, \quad \mathsf{L} \Sigma^p_k \text{ if and only if } \mathsf{L} \in \mathsf{U} \Sigma^p_k \text{ via \textsf{NPOMs}} N_1, \ldots, N_k \text{ satisfying for all inputs } x \text{ and every } i, 1 \leq i \leq k-1, \text{ that if } N_i \text{ asks some query } q \text{ during the computation of } N_1(x), \text{ then } N_{i+1}(q) \text{ with oracle } L(N_{i+1}^{L(N_i^{L(N_{i+2}^{N_k})})}) \text{ has at most one accepting path.} \quad \mathsf{UPH} \equiv \bigcup_{k \geq 0} \mathsf{U} \Sigma^p_k. \quad \text{The classes } \mathsf{U} \Delta^p_k \text{ and } \mathsf{U} \Pi^p_k, \quad k \geq 0, \quad \text{are defined analogously. As a notational shorthand, we often use } \mathsf{P}^{\mathsf{UP}} \text{ to represent } \mathsf{U} \Delta^p_2; \text{ we stress that}
   \]

\footnote{Note that it is not known whether such a collapse implies a collapse of \( \mathsf{PH} \). Note also that Toda’s [Tod93] result on whether \( \mathsf{P} \)-selective sets can be truth-table-hard for UP does not imply such a collapse, as truth-table reductions are less flexible than Turing reductions.}
both notations are used here to represent the class of sets accepted via guardedly unambiguous access to an NP oracle (that is, the class of sets accepted by some P machine with an NP machine’s language as its oracle such that on no input does the P machine ask its oracle machine any question on which the oracle machine has more than one accepting path).

4. For each of the above hierarchies, we use \(\Sigma_p^A\) (respectively, \(U\Sigma_p^A\) and \(U\Sigma_p^A\)) to denote that the \(\Sigma_p^A\) (respectively, \(U\Sigma_p^A\) and \(U\Sigma_p^A\)) computation is performed relative to oracle \(A\); similar notation is used for the \(\Pi\) and \(\Delta\) classes of the hierarchies.

The following facts follow from the definition (see also \[NR93\]) or can easily be shown.

**Fact 4.2** For every \(k \geq 1\),

1. \(U\Sigma_k^P \subseteq U\Sigma_k^A \subseteq \Sigma_k^P\) and \(U\Delta_k^P \subseteq U\Delta_k^A \subseteq \Delta_k^P\).
2. If \(U\Sigma_k^P = U\Pi_k^P\), then \(UPH = U\Sigma_k^P\).
3. If \(U\Sigma_k^P = U\Sigma_{k-1}^P\), then \(UPH = U\Sigma_{k-1}^P\).
4. \(U\Sigma_k^{UP, coUP} = U\Sigma_k^P\) and \(U\Sigma_k^{UP, coUP} = U\Pi_k^P\).

The classes “\(UP_{\leq k}\)”, the analogs of \(UP\) in which up to \(k\) accepting paths are allowed, have been studied in various contexts \[Wat88, Hem87, Bei89, CHV93, HH94, HZ93\]. One motivation for \(U\Sigma_k^P\) is that, for each \(k \geq 3\), the \(k\)th level of \(UPH\) is contained in the \((k - 1)\)st level of \(UPH\).

**Theorem 4.3** If there exists a sparse Turing-complete set for \(UP\), then

1. \(UP^{UP} \subseteq P^{UP}\), and
2. \(U\Sigma_k^P \subseteq U\Sigma_{k-1}^P\) for every \(k \geq 3\).

**Proof.** For the first statement, let \(L\) be any set in \(UP^{UP}\). By assumption, \(L \in UP^{S} = UP^{S}\) for some sparse set \(S \in UP\). Let \(q\) be a polynomial bounding the density of \(S\), that is, \(|S^{\leq m}| \leq q(m)\) for every \(m \geq 0\), and let \(N_S\) be a UPM for \(S\). Let \(N_L\) be a UPOM witnessing that \(L \in UP^{S}\), that is,
Let \( L = L(N_L^S) \). Let \( p(n) \) be a polynomial bounding the length of all query strings that can be asked during the computation of \( N_L \) on inputs of length \( n \). Define the polynomial \( r(n) \equiv q(p(n)) \) that bounds the number of strings in \( S \) that can be queried in the run of \( N_L \) on inputs of length \( n \).

To show that \( L \in P^{UP} \), we shall construct a DPOM \( M \) that may access its \( UP \) oracle \( D \) in a guarded manner (more formally, “may access its NP oracle \( D \) in a guardedly unambiguous manner,” but we will henceforward use \( UP \) and other \( U \cdots \) notations in this informal manner).

Before formally describing machine \( M \) (Figure [1]), we give some informal explanations. \( M \) will proceed in three basic steps: First, \( M \) determines the exact census of that part of \( S \) that is relevant for the given input length, \( |S|^{\leq p(n)} \). Knowing the exact census, \( M \) can construct (by prefix search) a table \( T \) of all strings in \( S^{\leq p(n)} \) without asking queries that make its oracle’s machine ambiguous, so the \( P^{UP} \)-like behavior is guaranteed. Finally, \( M \) asks its oracle \( D \) to simulate the computation of \( N_L \) on input \( x \) (answering \( N_L \)'s oracle queries by table-lookup using table \( T \)), and accepts accordingly.

In the formal description of machine \( M \) (given in Figure [1]), three oracle sets \( A, B, \) and \( C \) are used. Since \( M \) has only one \( UP \) oracle, the actual set to be used is \( D = A \oplus B \oplus C \) (with suitably modified queries to \( D \)). \( A, B, \) and \( C \) are defined as follows (we assume the set \( T \) below is coded in some standard reasonable way):

\[
A \overset{df}{=} \left\{ \langle 1^n, k \rangle \mid \begin{array}{l}
  n \geq 0 \land 0 \leq k \leq r(n) \land (\forall \ell : 1 \leq \ell \leq k) \left[ |c_\ell| \leq p(n) \land N_S(c_\ell) \text{ accepts} \right]
\end{array} \right\},
\]

\[
B \overset{df}{=} \left\{ \langle 1^n, i, j, k, b \rangle \mid \begin{array}{l}
  n \geq 0 \land 1 \leq j \leq k \land 0 \leq k \leq r(n) \land \\
  (\exists c_1 <_{\text{lex}} c_2 <_{\text{lex}} \cdots <_{\text{lex}} c_k) (\forall \ell : 1 \leq \ell \leq k) \\
  |c_\ell| \leq p(n) \land N_S(c_\ell) \text{ accepts} \land \text{the } i^{\text{th}} \text{ bit of } c_j \text{ is } b
\end{array} \right\},
\]

\[
C \overset{df}{=} \left\{ \langle x, T \rangle \mid ||T|| \leq r(|x|) \land N_L^T(x) \text{ accepts} \right\}.
\]

It is easy to see that \( M \) runs deterministically in polynomial time. This proves that \( L \in P^{UP} \).

In order to prove the second statement, let \( L \) be a set in \( \Sigma_k^p \) for any fixed \( k \geq 3 \). By assumption, there exists a sparse set \( S \) in \( UP \) such that \( L \in \Sigma_k^p \cup \Sigma_k^{-p} \); let \( N_1, N_2, \ldots, N_{k-1} \) be the UPOMs that witness this fact, that is, \( L = L(N_1^{L(N_2^{L(N_3^{\cdots^{L(N_{k-1}^S)}})}}) \).

Now we describe the computation of a \( \Sigma^p_k \) machine \( N \) recognizing \( L \). As before, \( N \) on input \( x \) computes in \( P^{UP} \) its table of advice strings, \( T = S^{\leq p(|x|)} \), and then simulates the \( \Sigma^p_k \)

computation of \( N_i^{L(N_{i+1}^{L(N_{i+2}^{\cdots^{L(N_{k-1}^S)}})}})(x) \) except with \( N_1, N_2, \ldots, N_{k-1} \) modified as follows. If in the simulation some machine \( N_i, 1 \leq i \leq k-2 \), consults its original oracle \( L(N_i^{(\cdots)} \) about some string, say \( z \), then the modified machine \( N_i' \) queries the modified machine at the next level, \( N_{i+1}' \), about the string \( (z, T) \) instead. Finally, the advice table \( T \), which has been “passed up” in this manner, is used to correctly answer all queries of \( N_{k-1} \).
Description of DPOM $M$.

input $x$;
begin
\[ n := |x|; \]
\[ k := r(n); \]
loop
\[ \text{if } \langle 1^n, k \rangle \in A \text{ then exit loop} \]
\[ \text{else } k := k - 1 \]
end loop
*T collects the strings of $S^{\leq p(n)}$ *)
\[ T := \emptyset; \]
* k is now the exact census of $S^{\leq p(n)} *$
for $j = 1$ to $k$ do
\[ c_j := \epsilon; \]
\[ i := 1; \]
repeat
\[ \text{if } \langle 1^n, i, j, k, 0 \rangle \in B \text{ then } c_j := c_j 0; i := i + 1 \]
\[ \text{else if } \langle 1^n, i, j, k, 1 \rangle \in B \text{ then } c_j := c_j 1; i := i + 1 \]
\[ \text{else } i := 0 \]
until $i = 0$;
\[ T := T \cup \{c_j\} \]
end for
if $\langle x, T \rangle \in C$ then accept
else reject
end

End of Description of DPOM $M$.

Figure 1: DPOM $M$ guardedly unambiguously accessing an NP oracle to accept a set in $\text{UP}^{\text{UP}}$. 
Note that \(N\)'s oracle in this simulation, \(L(N_2^\prime L(N_3^\prime L(N_{k-1}^\prime)))\), is not in general a \(U^{\Sigma^p_{k-2}}\) set (and \(L\) is thus not in \(U^{\Sigma^p_{k-1}}\) in general), as the above-described computation depends on the advice table \(T\), and so, for some bad advice \(T\), the unambiguity of the modified machines \(N_1^\prime, N_2^\prime, \ldots, N_{k-1}^\prime\) is no longer guaranteed. But since our base machine \(N\) is able to provide correct advice \(T\), we have indeed shown that \(L \in U^{\Sigma^p_{k-1}}\).

In the above proof, the assumption that the sparse set \(S\) is in \(UP\) is needed to determine the exact census of \(S\) using the \(UPM\) for \(S\). Let us now consider the weaker assumption that \(UP\) has only a Turing-hard sparse set. Karp and Lipton have shown that if there is a sparse Turing-hard set for \(NP\), then the polynomial hierarchy collapses to its second level [KL80]. Hopcroft [Hop81] dramatically simplified their proof, and Balcázar, Book, and Schöning [BBS86, Sch86] generalized, as Theorem 4.6, the Karp-Lipton result; the general approach of Hopcroft and Balcázar, Book, and Schöning will be central to our upcoming proof of Theorem 4.7. Schöning’s low hierarchy [Sch83] gives a way of classifying the complexity of \(NP\) sets that seem to be neither in \(P\) nor \(NP\)-complete. Of particular interest to us is the class \(\text{Low}_2 \overset{df}{=} \{A \mid A \in NP \text{ and } NP \overset{\text{poly}}{\subseteq} NP \}\). Note that for the special case \(k = 0\), Theorem 4.6 below says that \(\text{Low}_2 \supseteq NP \cap P/\text{poly} \cap \{L \mid L \text{ is self-reducible}\}\).

**Definition 4.4 [MP79]**

1. A partial order \(\prec_{\text{pw1}}\) on \(\Sigma^*\) is *polynomially well-founded and length-related* if and only if (a) every strictly decreasing chain is finite and there is a polynomial \(p\) such that every finite \(\prec_{\text{pw1}}\)-decreasing chain is shorter than \(p\) of the length of its maximum element, and (b) \(\exists q : q\) polynomial \((\forall x, y \in \Sigma^*) [x \prec_{\text{pw1}} y \Rightarrow |x| \leq q(|y|)]\).

2. A set \(A\) is *self-reducible* if and only if there exist a polynomially well-founded and length-related order \(\prec_{\text{pw1}}\) on \(\Sigma^*\) and a DPOM \(M\) such that \(A = L(M^A)\) and on any input \(x \in \Sigma^*\), \(M\) queries only strings \(y\) with \(y \prec_{\text{pw1}} x\).

**Lemma 4.5 [BBS86]** Let \(A\) be a self-reducible set and let \(M\) witness \(A\)'s self-reducibility. For any set \(B\) and any \(n\), if \((L(M^B))^\leq n = B^\leq n\), then \(A^\leq n = B^\leq n\).  

**Theorem 4.6 [BBS86]** If \(A\) is a self-reducible set and there is a \(k \geq 0\) and a sparse set \(S\) such that \(A \in \Sigma^p_k\), then \(\Sigma^p_{2^k} \subseteq \Sigma^p_{k+2}\).  

---

6Very recently, Köbler and Watanabe [KW94] have improved this collapse to \(ZPP^{NP}\), and have also obtained new consequences from the assumption that \(UP \subseteq (NP \cap \text{coNP})/\text{poly}\), whereas we obtain different consequences from the assumption that \(UP \subseteq P/\text{poly}\) (see [KW94] for the notations not defined in this footnote).

7\(A\) can be viewed as a “fixed point” of \(M\).
We now state and prove our results regarding sparse Turing-hard sets for UP.

**Theorem 4.7** If there exists a sparse Turing-hard set for UP, then

1. UP \( \subseteq \text{Low}_2 \), and
2. \( \bigcup_{k=0}^{\Sigma^p_k} \subseteq \bigcup_{j=0}^{\Sigma^p_{k-3}} \bigcup_{j=0}^{\Sigma^p_{k-j}} \bigcup_{j=0}^{\Sigma^p_{k-j-3}} \bigcup_{k=3}^{\Sigma^p} \) for every \( k \geq 3 \) and every \( j \), with \( 0 \leq j \leq k - 3 \).

**Proof.**

1. Let \( L \in \Sigma^p_{A} \), where \( A \in \text{UP} \) via \( \text{UPM} \) \( N_A \) and polynomial-time bound \( t \) (we assume that each step is nondeterministic—one can require this, without loss of generality, while maintaining categoricity). Our proof uses the well-known fact that the “left set” \( \{x \mid \exists y \in \Sigma^{\leq(|x|)} : x <_{\text{pw1}} y \} \) of any UP set is self-reducible and is in UP. More precisely, to apply Theorem 4.6 we would need \( A \) to be self-reducible. Although that can’t be assumed in general of an arbitrary UP set, the left set of \( A \), i.e., the set of prefixes of witnesses for elements in \( A \) defined by

\[
B = \{ (x, y) \mid (\exists z) \ |yz| = t(|x|) \land N_A(x) \text{ accepts on path } yz \},
\]

does have this property and is also in UP. A self-reducing machine \( M_{\text{self}} \) for \( B \) is given in Figure 3.

Note that the queries asked in the self-reduction are strictly less than the input with respect to a polynomially well-founded and length-related partial order \( <_{\text{pw1}} \) defined by: For fixed \( x \) and all strings \( y_1, y_2 \in \Sigma^{\leq(|x|)} \), \( \langle x, y_1 \rangle <_{\text{pw1}} \langle x, y_2 \rangle \) if and only if \( y_2 \) is prefix of \( y_1 \).

By assumption, since \( B \) is a UP set, \( B \in \text{P}^{S} \) for some sparse set \( S \), so Theorem 4.6 with \( k = 0 \) applies to \( B \). Furthermore, \( A \) is in \( \text{P}^{B} \), via prefix search by DPOM \( M_A \) (Figure 3). Thus, \( L \in \Sigma^p_A \subseteq \Sigma^p_B \subseteq \Sigma^p \), which shows that \( A \in \text{Low}_2 \).

2. For \( k = 3 \) (thus \( j = 0 \)), both inclusions have already been shown in Part 1, as \( \Sigma^p_k \subseteq \Delta^p_3 \). Now fix any \( k > 3 \), and let \( L \in \bigcup_{k=0}^{\Sigma^p_k} = \bigcup_{k=1}^{\Sigma^p_{k-1}} \) be witnessed by UPOMs \( N_1, N_2, \ldots, N_{k-1} \) and \( A \in \text{UP} \). Define \( B \) to be the left set of \( A \) as in Part 1, so \( A \in \text{P}^{B} \) via DPOM \( M_A \) (see Figure 3), \( B \) is self-reducible via \( M_{\text{self}} \) (see Figure 2), and \( B \) is in UP. By hypothesis, \( B \in \text{P}^{S} \) for some sparse set \( S \); let \( M_B \) be the reducing machine, that is \( B = L(M_B^S) \), and let \( m \) be a polynomial bound on the runtime of \( M_B \). Let \( q \) be a polynomial such that \( \|S^{\leq m}\| \leq q(m) \) for every \( m \geq 0 \). Let \( p(n) \) be a polynomial bounding the length of all query strings whose membership in the oracle set \( B \) can be asked in the run of \( N_1 \) (with oracle machines \( N_2, N_3, \ldots, N_{k-1}, M_B^S \)) on inputs of length \( n \). Define the polynomials \( r(n) \equiv m(p(n)) \) and \( s(n) \equiv q(r(n)) \).

To show that \( L \in \text{P}^{U\Sigma^p_{k-1}} = \bigcup_{k=3}^{\Sigma^p} \), we will describe a DPOM \( M \) that on input \( x \), \( |x| = n \), using the \( \Sigma^p_2 \) part \( D \) (defined below) of its oracle, performs a prefix search to extract the lexicographically smallest of all “good” advice sets (this informal term will be formally defined in the next paragraph), say \( T \), and then calls the \( U\Sigma^p_{k-1} \) part of its oracle to simulate the \( U\Sigma^p_{k-1} \) computation of
Description of Self-reducer $M_{\text{self}}$ for $B$.

\begin{verbatim}
input $\langle x, y \rangle$;
begin
if $|y| > t(|x|)$ then reject;
if $N_A(x)$ accepts on path $y$ then accept
else
  if $\langle x, y0 \rangle \in B$ or $\langle x, y1 \rangle \in B$ then accept
  else reject
end
End of Description of Self-reducer $M_{\text{self}}$ for $B$.
\end{verbatim}

Figure 2: A self-reducing machine for the left set of a UP set.

Description of DPOM $M_A$.

\begin{verbatim}
input $x$;
begin
  $y := \epsilon$;
  while $|y| < t(|x|)$ do
    if $\langle x, y0 \rangle \in B$ then accept
    else $y := y1$
  end while
  if $\langle x, y \rangle \in B$ then accept
  else reject
end
End of Description of DPOM $M_A$.
\end{verbatim}

Figure 3: A Turing reduction from a UP set $A$ to its left set $B$ via prefix search.
Corollary 4.8

We are very grateful to Gerd Wechsung for his help in bringing about this collaboration, and for his kind and insightful advice over many years. We thank Marius Zimand for proofreading, and Nikolai Vereshchagin for helpful discussions during his visit to Rochester. We thank Osamu Watanabe for discussing with us his results joint with Johannes Köbler, and we thank Osamu Watanabe and Johannes Köbler for providing us with copies of their paper [KW94].
References

[BBJ+89] A. Bertoni, D. Bruschi, D. Joseph, M. Sitharam, and P. Young. Generalized Boolean hierarchies and Boolean hierarchies over RP. In Proceedings of the 7th Conference on Fundamentals of Computation Theory, pages 35–46. Springer-Verlag Lecture Notes in Computer Science #380, August 1989.

[BBS86] J. Balcázar, R. Book, and U. Schöning. The polynomial-time hierarchy and sparse oracles. Journal of the ACM, 33(3):603–617, 1986.

[BCO93] R. Beigel, R. Chang, and M. Ogiwara. A relationship between difference hierarchies and relativized polynomial hierarchies. Mathematical Systems Theory, 26:293–310, 1993.

[Bei89] R. Beigel. On the relativized power of additional accepting paths. In Proceedings of the 4th Structure in Complexity Theory Conference, pages 216–224. IEEE Computer Society Press, June 1989.

[BG82] A. Blass and Y. Gurevich. On the unique satisfiability problem. Information and Control, 55:80–88, 1982.

[BG94] R. Beigel and J. Goldsmith. Downward separation fails catastrophically for limited nondeterminism classes. In Proceedings of the 9th Structure in Complexity Theory Conference, pages 134–138. IEEE Computer Society Press, June/July 1994.

[BJY90] D. Bruschi, D. Joseph, and P. Young. Strong separations for the Boolean hierarchy over RP. International Journal of Foundations of Computer Science, 1(3):201–218, 1990.

[Cai87] J. Cai. Probability one separation of the Boolean hierarchy. In Proceedings of the 4th Annual Symposium on Theoretical Aspects of Computer Science, pages 148–158. Springer-Verlag Lecture Notes in Computer Science #247, 1987.

[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.

[CGH+89] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The Boolean hierarchy II: Applications. SIAM Journal on Computing, 18(1):95–111, 1989.

[CH85] J. Cai and L. Hemachandra. The Boolean hierarchy: Hardware over NP. Technical Report 85-724, Cornell University, Department of Computer Science, Ithaca, NY, December 1985.

[Cha91] R. Chang. On the Structure of NP Computations under Boolean Operators. PhD thesis, Cornell University, Ithaca, NY, 1991.
[CHV93] J. Cai, L. Hemachandra, and J. Vyskoč. Promises and fault-tolerant database access. In K. Ambos-Spies, S. Homer, and U. Schöning, editors, Complexity Theory, pages 101–146. Cambridge University Press, 1993.

[CK90a] R. Chang and J. Kadin. The Boolean hierarchy and the polynomial hierarchy: A closer connection. In Proceedings of the 5th Structure in Complexity Theory Conference, pages 169–178. IEEE Computer Society Press, July 1990.

[CK90b] R. Chang and J. Kadin. On computing Boolean connectives of characteristic functions. Technical Report TR 90-1118, Department of Computer Science, Cornell University, Ithaca, NY, May 1990. To appear in Mathematical Systems Theory.

[CM87] J. Cai and G. Meyer. Graph minimal uncolorability is DP-complete. SIAM Journal on Computing, 16(2):259–277, 1987.

[Coo71] S. Cook. The complexity of theorem-proving procedures. In Proceedings of the 3rd ACM Symposium on Theory of Computing, pages 151–158, 1971.

[Cro94] K. Cronauer. A criterion to separate complexity classes by oracles. Technical Report 76, Universität Würzburg, Institut für Informatik, Würzburg, Germany, January 1994.

[GNW90] T. Gundermann, N. Nasser, and G. Wechsung. A survey on counting classes. In Proceedings of the 5th Structure in Complexity Theory Conference, pages 140–153. IEEE Computer Society Press, July 1990.

[GS88] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. SIAM Journal on Computing, 17(2):309–335, 1988.

[GW87] T. Gundermann and G. Wechsung. Counting classes with finite acceptance types. Computers and Artificial Intelligence, 6(5):395–409, 1987.

[Hau14] F. Hausdorff. Grundzüge der Mengenlehre. Leipzig, 1914.

[Hem87] L. Hemachandra. Counting in Structural Complexity Theory. PhD thesis, Cornell University, Ithaca, NY, May 1987. Available as Cornell Department of Computer Science Technical Report TR87-840.

[HH88] J. Hartmanis and L. Hemachandra. Complexity classes without machines: On complete languages for UP. Theoretical Computer Science, 58:129–142, 1988.

[HH90] J. Hartmanis and L. Hemachandra. Robust machines accept easy sets. Theoretical Computer Science, 74(2):217–226, 1990.

[HH94] E. Hemaspaandra and L. Hemaspaandra. Quasi-injective reductions. Theoretical Computer Science, 123:407–413, 1994.

[HIS85] J. Hartmanis, N. Immerman, and V. Sewelson. Sparse sets in NP–P: EXPTIME versus NEXPTIME. Information and Control, 65(2/3):159–181, 1985.
[HJ93] L. Hemachandra and S. Jha. Defying upward and downward separation. In Proceedings of the 10th Annual Symposium on Theoretical Aspects of Computer Science, pages 185–195. Springer-Verlag Lecture Notes in Computer Science #665, February 1993.

[HJV93] L. Hemaspaandra, S. Jain, and N. Vereshchagin. Banishing robust Turing completeness. International Journal of Foundations of Computer Science, 4(3):245–265, 1993.

[Hop81] J. Hopcroft. Recent directions in algorithmic research. In Proceedings 5th GI Conference on Theoretical Computer Science, pages 123–134. Springer-Verlag Lecture Notes in Computer Science #104, 1981.

[HOW92] L. Hemachandra, M. Ogiwara, and O. Watanabe. How hard are sparse sets? In Proceedings of the 7th Structure in Complexity Theory Conference, pages 222–238. IEEE Computer Society Press, June 1992.

[HR92] L. Hemachandra and R. Rubinstein. Separating complexity classes with tally oracles. Theoretical Computer Science, 92(2):309–318, 1992.

[HS65] J. Hartmanis and R. Stearns. On the computational complexity of algorithms. Transactions of the American Mathematical Society, 117:285–306, 1965.

[HU79] J. Hopcroft and J. Ullman. Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, 1979.

[HZ93] L. Hemaspaandra and M. Zimand. Strong forms of balanced immunity. Technical Report TR 480, Department of Computer Science, University of Rochester, Rochester, NY, December 1993.

[IT89] R. Impagliazzo and G. Tardos. Decision versus search problems in super-polynomial time. In Proceedings of the 30th IEEE Symposium on Foundations of Computer Science, pages 222–227. IEEE Computer Society Press, October/November 1989.

[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.

[Kad89] J. Kadin. $P^{NP[\log n]}$ and sparse Turing-complete sets for NP. Journal of Computer and System Sciences, 39(3):282–298, 1989.

[KL80] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In Proceedings of the 12th ACM Symposium on Theory of Computing, pages 302–309, April 1980. An extended version has also appeared as: Turing machines that take advice, L’Enseignement Mathématique, 2nd series 28, 1982, pages 191–209.

[KS85] K. Ko and U. Schöning. On circuit-size complexity and the low hierarchy in NP. SIAM Journal on Computing, 14(1):41–51, 1985.
[KSW87] J. Köbler, U. Schöning, and K. Wagner. The difference and truth-table hierarchies for NP. *R.A.I.R.O. Informatique théorique et Applications*, 21:419–435, 1987.

[KW94] J. Köbler and O. Watanabe. New collapse consequences of NP having small circuits. Technical Report 94-11, Universität Ulm, Institut für Informatik, Ulm, Germany, November 1994.

[Lev73] L. Levin. Universal sorting problems. *Problems of Information Transmission*, 9:265–266, 1973.

[LR94] K.-J. Lange and P. Rossmanith. Unambiguous polynomial hierarchies and exponential size. In *Proceedings of the 9th Structure in Complexity Theory Conference*, pages 106–115. IEEE Computer Society Press, June/July 1994.

[LS86] T. Long and A. Selman. Relativizing complexity classes with sparse oracles. *Journal of the ACM*, 33(3):618–627, 1986.

[MP79] A. Meyer and M. Paterson. With what frequency are apparently intractable problems difficult? Technical Report MIT/LCS/TM-126, MIT Laboratory for Computer Science, Cambridge, MA, 1979.

[MS72] A. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pages 125–129, 1972.

[NR93] R. Niedermeier and P. Rossmanith. Extended locally definable acceptance types. In *Proceedings of the 10th Annual Symposium on Theoretical Aspects of Computer Science*, pages 473–483. Springer-Verlag *Lecture Notes in Computer Science* #665, February 1993.

[OW91] M. Ogiwara and O. Watanabe. On polynomial-time bounded truth-table reducibility of NP sets to sparse sets. *SIAM Journal on Computing*, 20(3):471–483, 1991.

[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.

[Reg89] K. Regan. Provable complexity properties and constructive reasoning. Manuscript, April 1989.

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

[Sch83] U. Schöning. A low and a high hierarchy within NP. *Journal of Computer and System Sciences*, 27:14–28, 1983.

[Sch86] U. Schöning. *Complexity and Structure*. Springer Verlag *Lecture Notes in Computer Science* #211, 1986.
[Sel88] A. Selman. Natural self-reducible sets. *SIAM Journal on Computing*, 17(5):989–996, 1988.

[SL92] M. Sheu and T. Long. UP and the low and high hierarchies: a relativized separation. To appear in *Mathematical Systems Theory*.

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

[Tod91] S. Toda. On polynomial-time truth-table reducibilities of intractable sets to P-selective sets. *Mathematical Systems Theory*, 24:69–82, 1991.

[Val76] L. Valiant. The relative complexity of checking and evaluating. *Information Processing Letters*, 5:20–23, 1976.

[Wat88] O. Watanabe. On hardness of one-way functions. *Information Processing Letters*, 27:151–157, 1988.

[You92] P. Young. How reductions to sparse sets collapse the polynomial-time hierarchy: A primer. *SIGACT News*, 1992. Part I (#3, pages 107–117), Part II (#4, pages 83–94), and Corrigendum to Part I (#4, page 94).