Computing Solutions Uniquely Collapses the Polynomial Hierarchy

Lane A. Hemaspaandra*
Dept. of Computer Science
University of Rochester
Rochester, NY 14627, USA

Ashish V. Naik†
Dept. of Computer Science
SUNY–Buffalo
Buffalo, NY 14260, USA

Mitsunori Ogiwara‡
Dept. of Comp. Sci. & Inf. Math.
University of Electro-communications
Tokyo 182, Japan

Alan L. Selman§
Dept. of Computer Science
SUNY–Buffalo
Buffalo, NY 14260, USA

August 6, 1993. Topic: Computational Complexity.

Abstract

Is there a *single-valued* NP function that, when given a satisfiable formula as input, outputs a satisfying assignment? That is, can a nondeterministic function cull just one satisfying assignment from a possibly exponentially large collection of assignments? We show that if there is such a nondeterministic function, then the polynomial hierarchy collapses to its second level. As the existence of such a function is known to be equivalent to the statement "every multivalued NP function has a single-valued NP refinement," our result provides the strongest evidence yet that multivalued NP functions cannot be refined.

We prove our result via theorems of independent interest. We say that a set A is NPSV-selective (NPMV-selective) if there is a 2-ary partial function in NPSV (NPMV, respectively) that decides which of its inputs (if any) is "more likely" to belong to A; this is a nondeterministic analog of the recursion-theoretic notion of the semi-recursive sets and the extant complexity-theoretic notion of P-selectivity. Our hierarchy collapse follows by combining the easy observation that every set in NP is NPMV-selective with either of the following two theorems that we prove: (1) If $A \in \mathrm{NP}$ is NPSV-selective, then $A \in (\mathrm{NP} \bigcap \mathrm{coNP})/\mathrm{poly}$. (2) If $A \in \mathrm{NP}$ is NPSV-selective, then A is Low_2. To wit, either result implies that if every set in NP is NPSV-selective, then the polynomial hierarchy collapses to its second level, $\mathrm{NP}^{\mathrm{NP}}$.

*Supported in part by the NSF under grants CCR-8957604 and NSF-INT-9116781/JSPS-ENG-207. Work done in part while visiting the University of Electro-Communications–Tokyo.

†Supported in part by the NSF under grant CCR-9002292.

‡Supported in part by the NSF under grant CCR-9002292 and the JSPS under grant NSF-INT-9116781/JSPS-ENG-207. Work done in part while visiting SUNY–Buffalo.

§Supported in part by the NSF under grant CCR-9002292.

Keywords: computational complexity, complexity classes, NPSV, NPMV, refinements, selectivity, lowness, advice classes, nonuniform classes, polynomial hierarchy

ACM Computing Reviews Subject Categories: F.1.2, F.1.3

1 Introduction

Valiant and Vazirani's [VV86] result that, in their words, "NP is as easy as detecting unique solutions," has rightly been the focus of great attention. Their breakthrough—a proof that every NP set *probabilistically* reduces to "detecting unique solutions" (technically, reduces to every solution to the promise problem (1SAT,SAT))—is one of the dual pillars on which Toda's [Tod91] PH $\subseteq$ P^{PP} paper rests, as do later papers extending Toda's result [TO92], and studying the complexity of function inversion [WT93,AFW92].

Fenner et al. [FHOS93] and Selman [Sel] raised a related question that, in light of the centrality of nondeterminism in complexity theory, may be equally compelling. They asked whether the following hypothesis is true.

Hypothesis 1.1 There is a *single-valued* NP function f such that for each formula $F \in$ SAT, $f(F)$ is a satisfying assignment of F.

It is easy to see that Hypothesis 1.1 is true if NP = coNP. However, as both Fenner et al. [FHOS93] and Selman [Sel] suspected that Hypothesis 1.1 fails, perhaps a more interesting issue is that of the evidential weight in that direction. In fact, little is currently known to indicate that Hypothesis 1.1 fails. The totality of current evidence seems to be the fact that Hypothesis 1.1 fails relative to a random oracle [Roy93], and the result of Selman [Sel] that if Hypothesis 1.1 holds, then there are two disjoint NP-Turing-complete sets such that every set that separates them is NP-hard.

Since Hypothesis 1.1 is implied by NP = coNP, one might hope that Hypothesis 1.1 also implies a collapse of the polynomial hierarchy. The main result of this paper provides strong evidence that Hypothesis 1.1 fails: Hypothesis 1.1 implies that the polynomial hierarchy collapses to its second level.

We obtain our result from a surprising and seemingly little-related direction: selectivity. Selectivity is a notion of generalized membership testing; selective sets have functions choosing which of any two input elements is the "more likely" to be in the set. Sets selective with respect to recursive selector functions were introduced by Jockush [Joc68], and are called the *semirecursive* sets. Sets selective with respect to deterministic polynomial-time selector functions were introduced by Selman [Sel79], and are called the P-selective sets; sets selective with respect to single-valued total NP functions were introduced and studied by Hemachandra et al. [HHO+]. There has been an explosion of interest in selective sets [BTvEB93,BvHT93,NOS93,HNOS93,HHO+], and advances have catalyzed further advances.

In this paper, we extend the notion of selectivity, in the natural way, to functions that may be partial and/or multivalued. Important function classes of these sorts are the single-valued partial NP functions (NPSV), the multivalued partial NP functions (NPMV), and the multivalued total NP functions (NPMV_t). Though it is easily observed that all NP sets are NPMV-selective, we will prove the following result.

> ($\star\star$) If all NP sets are NPSV-selective then the polynomial hierarchy collapses to its second level.

It follows easily that Hypothesis 1.1 implies this same collapse.

Result ($\star\star$) is proven via (either of) the following two results, each of interest in its own right: (1) the NPSV-selective sets in NP are Low_2 (that is, for each such set A, $\mathrm{NP}^{\mathrm{NP}^A} = \mathrm{NP}^{\mathrm{NP}}$), and (2) the NPSV-selective sets in NP are in $(\mathrm{NP} \bigcap \mathrm{coNP})/\mathrm{poly}$. Though NPSV functions lack the totality that seems required to invoke "strong" (i.e., $\mathrm{NP} \bigcap \mathrm{coNP}$-like [Sel78,Lon82]) computation, one can nonetheless get the *effect* of totality in the cases that count. This allows us to prove that the NPSV-selective sets in NP have lowness and advice class results just as strong as those shown by [HHO+] for the NPSV_t-selective sets in NP. This is an important advance, as results about NPSV_t-selective sets offer no help in discrediting Hypothesis 1.1.

2 Definitions

Our alphabet will be $\Sigma = \{0,1\}$. Let our pairing function $\langle \cdots \rangle$ be any "multi-arity onto," polynomial-time computable, polynomial-time invertible function.

For each partial, multivalued function f, $\textit{set-}f(x)$ denotes the set of values of f on input x. If $f(x)$ is undefined, then $\textit{set-}f(x) = \emptyset$. We will use this notation for single-valued functions also, to avoid ambiguity regarding equality tests between potentially undefined values. For any two partial, multivalued functions f and g, we say that f is a *refinement* of g if, for all x, it holds that

1. $f(x)$ is defined if and only if $g(x)$ is defined, and
2. $\textit{set-}f(x) \subseteq \textit{set-}g(x)$.

We extend previous notions of selectivity [Sel79,HHO+] to multivalued and/or partial functions.

Definition 2.1 Let $\mathcal{FC}$ be any class of functions (possibly multivalued and/or partial). A set A is $\mathcal{FC}$-selective if there is a function $f \in \mathcal{FC}$ such that for every x and y, it holds that

(i) $\textit{set-}f(x,y) \subseteq \{x,y\}$, and

(ii) if $x \in A$ or $y \in A$, then $\emptyset \neq \textit{set-}f(x,y) \subseteq A$.

By $\mathcal{FC}$-sel we denote the class of sets that are $\mathcal{FC}$-selective.

We will be interested, in particular, in the following classes of functions.

Definition 2.2 [BLS84]

1. NPMV is the class of partial, multivalued functions f for which there is a nondeterministic polynomial-time machine N such that for every x, it holds that
   (a) $f(x)$ is defined if and only if $N(x)$ has at least one accepting computation path, and
   (b) for every y, $y \in \textit{set-}f(x)$ if and only if there is an accepting computation path of $N(x)$ that outputs y.
2. NPMV_t is the class of total, multivalued functions in NPMV.
3. NPSV is the class of partial, single-valued functions in NPMV.
4. NPSV_t is the class of total, single-valued functions in NPMV.

Karp and Lipton introduced the following notion of being computable in a class supplemented by a small amount of extra information.

Definition 2.3 [KL80] For any class of sets $\mathcal{C}$, $\mathcal{C}$/poly denotes the class of sets L for which there exist a set $A \in \mathcal{C}$ and a polynomially length-bounded function $h : \Sigma^* \rightarrow \Sigma^*$ such that for every x, it holds that

$$x \in L \text{ if and only if } \langle x, h(0^{|x|}) \rangle \in A.$$

We will be particularly interested in the advice classes NP/poly, coNP/poly, and $(\text{NP} \bigcap \text{coNP})$/poly. It is important not to confuse $\text{NP/poly} \bigcap \text{coNP/poly}$ and $(\text{NP} \bigcap \text{coNP})$/poly, as it is not known that these classes coincide (see [GB91]).

Next we define lowness and extended lowness.

Definition 2.4

1. [Sch83] For each $k \geq 1$, define $\text{Low}_k = \{L \in \text{NP} \mid \Sigma_k^{p,L} = \Sigma_k^p\}$, where the Σ_k^p are the Σ levels of the polynomial hierarchy [MS72,Sto77].
2. [LS91,BBS86] For each $k \geq 2$, define $\text{ExtendedLow}_k = \{L \mid \Sigma_k^{p,L} = \Sigma_{k-1}^{p,\text{SAT} \oplus L}\}$. For each $k \geq 3$, define $\text{ExtendedLow}\Theta_k = \{L \mid \text{P}^{(\Sigma_{k-1}^{p,L})[\mathcal{O}(\log n)]} \subseteq \text{P}^{(\Sigma_{k-2}^{p,\text{SAT} \oplus L})[\mathcal{O}(\log n)]}\}$. The $[\mathcal{O}(\log n)]$ indicates that at most $\mathcal{O}(\log n)$ queries are made to the oracle.

Hemachandra et al. [HHO+] proved that NPSV$_t$-sel $\subseteq$ (NP $\bigcap$ coNP)/poly, that NPSV$_t$-sel $\bigcap$ NP $\subseteq$Low$_2$, and that NPSV$_t$-sel $\subseteq$ ExtendedLowΘ_3.

Finally, we define "promise problems" [EY80] corresponding to selectivity. Informally, a solution to the promise problem PP-A [LS93,Sel88] will—if the promise is met that exactly one of x and y is in A—contain $\langle x, y\rangle$ exactly if $x \in A$.

Definition 2.5 ([LS93], see also [Sel88]) Given any set A, we say that a set B *is a solution to* PP-A if $\{\langle x,\, y\rangle \mid (x \in A) \oplus (y \in A)\} \bigcap B = \{\langle x,\, y\rangle \mid (x \in A)$ and $(y \notin A)\}$.

3 Unique Solutions Collapse The Polynomial Hierarchy

What lowness results hold for the three new selectivity classes we've mentioned—the NPSV-selective sets, the NPMV-selective sets, and the NPMV$_t$-selective sets? Clearly, NPMV-sel (and thus NPSV-sel) is contained in NP/poly, and NPMV$_t$-sel is contained in NP/poly $\bigcap$ coNP/poly, via using a standard divide-and-conquer approach to find an appropriate advice set, similar to the approach in Ko's proof [Ko83] that the P-selective sets are in P/poly (see also the proofs of Theorem 3.1 and Theorem 3.5). It follows immediately by known results on the extended lowness [HHO+] (respectively, lowness [Yap83]) of NP/poly $\bigcap$ coNP/poly (respectively, NP/poly $\bigcap$ coNP/poly $\bigcap$ NP = coNP/poly $\bigcap$ NP) that NPMV$_t$-sel is ExtendedLow$_3$, and that NPMV$_t$-sel $\bigcap$ NP is Low$_3$. We now turn towards our main result.

Theorem 3.1 NPSV-sel $\bigcap$ NP $\subseteq$ (NP $\bigcap$ coNP)/poly.

Proof Let $L \in$ NP be NPSV-selective, with selector function $f \in$ NPSV. Without loss of generality we assume f satisfies $(\forall x,\, y)[\mathit{set}\text{-}f(x,y) = \mathit{set}\text{-}f(y,x)]$, since if it doesn't, we can replace it with $f\text{-}\mathit{new}(a,b) = f(\min(a,b), \max(a,b))$. Recall that $\mathit{set}\text{-}f(x) = \{y \mid y$ is a value of $f(x)\}$.

For each n, there is a (possibly empty) set B_n so that:

1. $B_n \subseteq L^{=n}$,
2. B_n has at most n elements, and
3. $(\forall x \in \Sigma^n)\,[x \in L \iff (\exists y \in B_n)\,[\mathit{set}\text{-}f(x,y) = \{x\}]]$.

This is true, as in the case of Ko's proof that the P-selective sets are in P/poly [Ko83], as we may repeatedly choose to add to B_n some element of $L^{=n}$ that loses to (that is, proves less likely to be in L according to f) at least half the elements that both are not yet in B_n and don't yet beat some element in B_n.

Let

$L' = \{\langle x, \langle \alpha_1, \alpha_2, \ldots, \alpha_z, wit_1, wit_2, \ldots, wit_z\rangle\rangle \mid z \leq |x| \text{ and } (\forall i \in \{1, 2, \cdots, z\})[wit_i$ is a witness certifying (with respect to some fixed polynomial-time binary relation defining NP set L) that $\alpha_i \in L]$ and $(\exists i \in \{1, 2, \ldots, z\})[set\text{-}f(x, \alpha_i) = \{x\}]\}$.

Note that $L' \in \mathrm{NP} \bigcap \mathrm{coNP}$. The containment in NP is immediate. The containment in coNP follows from the fact that, since the α_i are certifiably in L, by the definition of NPSV-selectivity, the partial function f is guaranteed to be defined when it is invoked in the actual run of the NP machine for $\overline{L'}$; informally, one might say that f is functioning in a guardedly total manner, in this setting. Furthermore, the set $\{x \mid \langle x, \langle \alpha_1, \alpha_2, \ldots, \alpha_z, wit_1, wit_2, \ldots, wit_z\rangle\rangle \in L'\}$, is exactly L, where $B_n = \{\alpha_1, \cdots, \alpha_z\}$, and the *wit* strings are witnesses of the membership of the elements of B_n in L. This proves that $L \in (\mathrm{NP} \bigcap \mathrm{coNP})/\mathrm{poly}$. ∎

It is not hard to see that essentially the same proof as that of the above theorem also establishes the following result.

Theorem 3.2 NPSV-sel $\subseteq$ NP/poly $\bigcap$ coNP/poly,

It follows immediately that:

Corollary 3.3 The NPSV-selective sets are ExtendedLow$_3$.

What can be said about the lowness of the NPSV-selective sets in NP? Theorem 3.2 and Köbler's "(NP $\bigcap$ coNP)/poly $\bigcap$ NP is LowΘ_3" result imply a LowΘ_3 result. However, below, we establish that the NPSV-selective sets in NP are in fact Low$_2$.

Lemma 3.4 [LS93] If A is in Σ_i^p and B is a solution of PP-A, then $\Sigma_{i+1}^{p,A} \subseteq \Sigma_{i+1}^{p,B}$.

Theorem 3.5 If $A \in$ NPSV-sel $\bigcap$ NP, then PP-A has a solution L that is Low$_2$.

Corollary 3.6 NPSV-sel $\bigcap$ NP $\subseteq$ Low$_2$.

Proof of Corollary 3.6 Follows immediately from Theorem 3.5 via Lemma 3.4. ∎

The proof of Theorem 3.5 is related to that of Theorem 3.1; we defer it to Section 4.

Note that every NP set is NPMV-selective. Is this also true for NPSV-selectivity? We have the following result.

Theorem 3.7 If NP $\subseteq$ NPSV-sel, then $\mathrm{NP}^{\mathrm{NP}} = \mathrm{PH}$.

Proof This is a corollary of Theorem 3.1, since it is well-known [Käm91,AFK89] that $\mathrm{NP} \subseteq (\mathrm{NP} \bigcap \mathrm{coNP})/\mathrm{poly} \Rightarrow \mathrm{NP}^{\mathrm{NP}} = \mathrm{coNP}^{\mathrm{NP}}$. Alternatively, it follows from Corollary 3.6. ∎

We now have implicitly proved our main result.

Theorem 3.8 If Hypothesis 1.1 is true (equivalently, if every NPMV function has an NPSV refinement [Sel]), then $\mathrm{NP}^{\mathrm{NP}} = \mathrm{PH}$.

Proof The reader may easily observe that every set in NP is NPMV-selective. It is not hard to see that any NPSV refinement of an NPMV-selector for a set is itself an NPSV-selector for the set. Thus, all NP sets are NPSV-selective, so by Theorem 3.7, $\mathrm{NP}^{\mathrm{NP}} = \mathrm{PH}$. ∎

4 Proof of Theorem 3.5

Let $A \in \mathrm{NPSV\text{-}sel} \cap \mathrm{NP}$, with selector function $f \in \mathrm{NPSV}$. Define $L_1 = \{\langle x, y\rangle \mid \textit{set-}f(x,y) = \{x\} \text{ and } x \in A\}$ and $L_2 = \{\langle x, y\rangle \mid \textit{set-}f(x,y) = \{y\}\}$. Note that L_1 and L_2 are in NP. Since L_1 is a solution of PP-A, it suffices to show that $\Sigma_2^{p,L_1} \subseteq \Sigma_2^p$. Let $C \in \Sigma_2^{p,L_1}$. There exist NP machines N_1 and N_2 such that $C = L(N_1^{L(N_2^{L_1})})$. We will describe an NP machine F and define sets D and E in NP so that $C = L(F^{D\oplus E})$. Recall that, as in the proof of Theorem 3.1, for each n, there exists a set B_n such that:

1. $B_n \subseteq A \cap \Sigma^n$,
2. $||\, B_n \,|| \leq n$, and
3. $(\forall x \in \Sigma^n)[x \in A \iff (\exists y \in B_n)[\textit{set-}f(x,y) = \{x\}]]$.

Let p be a polynomial such that for every x and for every query string y of N_1 on input x, the query strings of N_2 on input y are of length at most $p(|x|)$; such a polynomial exists as both N_1 and N_2 are polynomial-time (nondeterministic) machines. Let x be a string whose membership in C we are testing. On input x, the machine F behaves as follows:

(a) For each $i, 1 \leq i \leq p(|x|)$, F guesses $B_i \subseteq \Sigma^i$ with $||\, B_i \,|| \leq i$.

For each $i, 1 \leq i \leq p(|x|)$, and for each $y \in B_i$, F tests whether $y \in A$ by simulating an NP machine witnessing $A \in \mathrm{NP}$. If the test is not successful for some i and y, F rejects x.

(b) F asks whether $\langle 0^{p(|x|)}, B_1, \cdots, B_{p(|x|)}\rangle \in D$, where D is a set in NP defined by

$$D = \{\langle 0^n, B_1, \cdots, B_n\rangle \mid (\exists i : 1 \leq i \leq n)(\exists y \in A^{=i} - B_i)(\forall z \in B_i)[\textit{set-}f(y,z) = \{z\}]\}.$$

If the query is answered affirmatively, then F rejects x. Note that on questions to D asked during an actual run of F, the B_i are all subsets of A, so f will never be undefined during such runs.

(c) F simulates N_1 on x by substituting query r of N_1 to $L(N_2^{L_1})$ with query $\langle r, B_1, \cdots, B_{p(|x|)}\rangle$ to E.

F accepts x if and only if N_1 on x accepts in the above simulation.

The set E above is recognized by the following NP machine H.

(i) On input $\langle r, B_1, \cdots, B_n\rangle$, H simulates N_2 on r. When N_2 asks L_1 about $\langle y_1, y_2\rangle$, H nondeterministically chooses one of the following three tests:

(A) Nondeterministically verify that $\langle y_1, y_2\rangle \in L_1$.

(B) Nondeterministically verify that $\langle y_1, y_2\rangle \in L_2$.

(C) Check that $y_1 \notin B_{|y_1|}$ and $y_2 \notin B_{|y_2|}$, and nondeterministically verify that $(\forall z \in B_{|y_1|})[set\text{-}f(y_1, z) = \{z\}]$ and that $(\forall z \in B_{|y_2|})[set\text{-}f(y_2, z) = \{z\}]$.

Again, on actual calls to this machine made during actual runs of F, the B_i will be subsets of A, so f will always be defined.

(ii) If the chosen test above is not successful, then H rejects. If (A) was chosen and successful, then H assumes the answer to $\langle y_1, y_2\rangle$ to be affirmative. If either (B) or (C) was chosen and successful, then H assumes the answer to $\langle y_1, y_2\rangle$ to be negative.

H continues the simulation of N_2.

(iii) H accepts if and only if its simulation of N_2 leads to acceptance.

We claim that $F^{D\oplus E}$ accepts x if and only if $x \in C$. Note for any $B_1, \cdots, B_{p(|x|)}$ satisfying (1) and (2) above, it holds that

$$(\forall i : 1 \le i \le p(|x|))[B_i \text{ satisfies } (3)\ ] \iff \langle 0^{p(|x|)}, B_1, \cdots, B_{p(|x|)}\rangle \notin D.$$

So, for some computation path, F enters step (c), and if it enters that step, the sets $B_1, \cdots, B_{p(|x|)}$ will satisfy (1), (2), and (3). Thus, in order to prove the claim, we have only to show that for any query $\langle y_1, y_2\rangle$ of N_2 on r in H's simulation, it holds that

$$\langle y_1, y_2\rangle \in L_1 \iff \text{neither (B) nor (C) is successful for } \langle y_1, y_2\rangle.$$

Suppose that $\langle y_1, y_2\rangle \in L_1$. Since $L_1 \cap L_2 = \emptyset$, (B) is not successful for $\langle y_1, y_2\rangle$. Since $\langle y_1, y_2\rangle \in L_1$, $y_1 \in A$. This implies $(\exists z \in B_{|y_1|})[set\text{-}f(y_1, z) \neq \{z\}]$. For such z, $set\text{-}f(y_1, z) = \{y_1\}$, as it cannot be undefined in this case. Thus, (C) is not successful.

Conversely, suppose that $\langle y_1, y_2 \rangle \notin L_1$. If $\langle y_1, y_2 \rangle \in L_2$, then (B) is successful and we are done. Suppose that $\langle y_1, y_2 \rangle \notin L_2$. Then $y_1 \notin A$ and *set-*$f(y_1, y_2)$ is either $\emptyset$ or $\{y_1\}$. This implies $y_2 \notin A$. So for any $z \in B_{|y_1|}$, *set-*$f(y_1, z) = \{z\}$ because $B_{|y_1|} \subseteq A$, and for much the same reason, for any $z \in B_{|y_2|}$, *set-*$f(y_2, z) = \{z\}$. Thus, (C) is successful. This completes the proof.

5 Conclusion and Open Questions

This paper has studied the complexity of computing a *single* satisfying assignment of an input satisfiable formula. Previously, it was (trivially) known that satisfying assignments could be found by polynomial-time functions if and only if P=NP. It was also (trivially) known that an assignment could be found via a polynomial-time machine using an NP oracle (and it was known, not trivially, that finding the lexicographically largest assignment was the hardest of all problems solvable in that class [Kre88]).

But what about function classes between FP and $\mathrm{FP}^{\mathrm{NP}}$? In this paper, we proved that the NPSV functions are unlikely to have the power to find satisfying assignments; they can do so only if the polynomial hierarchy collapses to $\mathrm{NP}^{\mathrm{NP}}$. There remains an important function class intermediate in power between the NPSV functions (shown by this paper to be unlikely to have the power of finding satisfying assignments) and the functions computable via Turing access to an NP oracle (which clearly can find satisfying assignments). This class is the class of (partial) functions computable via *parallel* (that is, truth-table) access to an NP oracle. Clearly, NPSV is a subset of this class (cf. [Sel]). The key open issue is distilled in the following hypothesis (see [WT93,AFW92,HNOS93,NOS93,Sel] for background and discussion).

Hypothesis 5.1 Every NPMV function has a (single-valued) refinement in $\mathrm{FP}^{\mathrm{NP}}_{tt}$.

The above can be equivalently phrased as: There is partial function f computable by a polynomial-time Turing machine making parallel queries to NP such that for each formula $F \in \mathrm{SAT}$, $f(F)$ is a satisfying assignment of F. Does Hypothesis 5.1 imply a collapse of the polynomial hierarchy? It seems that such a result would require techniques substantially different than those of this paper. In particular, our result that "NPMV has NPSV refinements only if $\mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}$" itself relativizes. However, any relativizable proof of "Hypothesis 5.1 implies collapse of the polynomial hierarchy" would immediately imply—due to the result of Watanabe and Toda [WT93] that Hypothesis 5.1 holds relative to a random oracle—that the polynomial hierarchy collapses relative to a random oracle. Though it remains plausible that the polynomial hierarchy collapses relative to a random oracle, the conventional wisdom is that the hierarchy does not collapse relative to a random oracle (see [Cai86] for some indication of the difficulties surrounding this issue). The main result of the present paper does not similarly imply that the polynomial hierarchy collapses relative to a random oracle, as Hypothesis 1.1 is known to fail relative to a random oracle [Roy93].

Acknowledgment

We thank Edith Hemaspaandra, Marius Zimand, and Yenjo Han for many helpful comments and suggestions.

References

[AFK89] M. Abadi, J. Feigenbaum, and J. Killian. On hiding information from an oracle. *Journal of Computer and System Sciences*, 39:21–50, 1989.

[AFW92] K. Abrahamson, M. Fellows, and C. Wilson. Parallel self-reducibility. In *Proceedings of the 4th International Conference on Computing and Information*, pages 67–70. IEEE Computer Society Press, May 1992.

[BBS86] J. Balcázar, R. Book, and U. Schöning. Sparse sets, lowness and highness. *SIAM Journal on Computing*, 15(3):739–746, 1986.

[BLS84] R. Book, T. Long, and A. Selman. Quantitative relativizations of complexity classes. *SIAM Journal on Computing*, 13(3):461–487, 1984.

[BTvEB93] H. Buhrman, L. Torenvliet, and P. van Emde Boas. Twenty questions to a P-selector. Manuscript, 1993.

[BvHT93] H. Buhrman, P. van Helden, and L. Torenvliet. P-selective self-reducible sets: A new characterization of P. In *Proceedings of the 8th Structure in Complexity Theory Conference*, pages 44–51. IEEE Computer Society Press, May 1993.

[Cai86] J. Cai. *On Some Most Probable Separations of Complexity Classes*. PhD thesis, Cornell University, Ithaca, NY, 1986.

[EY80] S. Even and Y. Yacobi. Cryptocomplexity and NP-completeness. In *Proceedings of the 7th International Colloquium on Automata, Languages, and Programming*, pages 195–207. Springer-Verlag *Lecture Notes in Computer Science*, 1980.

[FHOS93] S. Fenner, S. Homer, M. Ogiwara, and A. Selman. On using oracles that compute values. In *Proceedings of the 10th Annual Symposium on Theoretical Aspects of Computer Science*, pages 398–407. Springer-Verlag *Lecture Notes in Computer Science #665*, February 1993.

[GB91] R. Gavaldà and J. Balcázar. Strong and robustly strong polynomial time reducibilities to sparse sets. *Theoretical Computer Science*, 88(1):1–14, 1991.

[HHO+] L. Hemachandra, A. Hoene, M. Ogiwara, A. Selman, T. Thierauf, and J. Wang. Selectivity. In *Proceedings of the 5th International Conference on Computing and Information*. IEEE Computer Society Press. To appear.

[HNOS93] E. Hemaspaandra, A. Naik, M. Ogiwara, and A. Selman. P-selective sets, and reducing search to decision vs. self-reducibility. Technical Report 93-21, State University of New York at Buffalo, Department of Computer Science, Buffalo, NY, 1993.

[Joc68] C. Jockusch. Semirecursive sets and positive reducibility. *Transactions of the AMS*, 131(2):420–436, 1968.

[Käm91] J. Kämper. Non-uniform proof systems: a new framework to describe non-uniform and probabilistic complexity classes. *Theoretical Computer Science*, 85(2):305–331, 1991.

[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.

[Ko83] K. Ko. On self-reducibility and weak P-selectivity. *Journal of Computer and System Sciences*, 26:209–221, 1983.

[Kre88] M. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36:490–509, 1988.

[Lon82] T. Long. Strong nondeterministic polynomial-time reducibilities. *Theoretical Computer Science*, 21:1–25, 1982.

[LS91] T. Long and M. Sheu. A refinement of the low and high hierarchies. Technical Report OSU-CISRC-2/91-TR6, Ohio State University, Department of Computer Science, Columbus, Ohio, February 1991.

[LS93] L. Longpré and A. Selman. Hard promise problems and nonuniform complexity. *Theoretical Computer Science*, 115(3):277–290, 1993.

[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.

[NOS93] A. Naik, M. Ogiwara, and A. Selman. P-selective sets, and reducing search to decision vs. self-reducibility. In *Proceedings of the 8th Structure in Complexity Theory Conference*, pages 52–64. IEEE Computer Society Press, May 1993.

[Roy93] J. Royer, August 1993. Personal Communication.

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

[Sel] A. Selman. A taxonomy of complexity classes of functions. *Journal of Computer and System Sciences*. In press.

[Sel78] A. Selman. Polynomial time enumeration reducibility. *SIAM Journal on Computing*, 7(4):440–457, 1978.

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

[Sel88] A. Selman. Promise problems complete for complexity classes. *Information and Computation*, 78:87–98, 1988.

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

[TO92] S. Toda and M. Ogiwara. Counting classes are at least as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 21(2):316–328, 1992.

[Tod91] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.

[VV86] L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.

[WT93] O. Watanabe and S. Toda. Structural analysis of the complexity of inverse functions. *Mathematical Systems Theory*, 26:203–214, 1993.

[Yap83] C. Yap. Some consequences of non-uniform conditions on uniform classes. *Theoretical Computer Science*, 26:287–300, 1983.