Journal of Computer and System Sciences **64**, 311–328 (2002)
doi:10.1006/jcss.2001.1815, available online at http://www.idealibrary.com on IDEAL®

Reducing the Number of Solutions of NP Functions[1]

Department of Computer Science, University of Rochester, Rochester, New York 14627
E-mail: lane@cs.rochester.edu, ogihara@cs.rochester.edu

and

Gerd Wechsung

Institut für Informatik, Friedrich-Schiller-Universität Jena, 07743 Jena, Germany
E-mail: wechsung@informatik.uni-jena.de

Received March 24, 2000; revised October 10, 2001

We study whether one can prune solutions from NP functions. Though it is known that, unless surprising complexity class collapses occur, one cannot reduce the number of accepting paths of NP machines, we nonetheless show that it often is possible to reduce the number of solutions of NP functions. For finite cardinality types, we give a sufficient condition for such solution reduction. We also give absolute and conditional necessary conditions for solution reduction, and in particular we show that in many cases solution reduction is impossible unless the polynomial hierarchy collapses.

Key Words: reducing solutions; solution-pruning algorithms; NP functions; multivalued functions; NPMV; function refinement; cardinality types; the Narrowing-Gap Condition; semi-feasible computation; selectivity theory; computational complexity.

1. INTRODUCTION AND DISCUSSION

Let $\mathrm{NP}_{\mathbb{N}^+}\mathrm{V}$ denote the set of all (possibly partial, possibly multivalued) functions computable by nondeterministic polynomial-time Turing machines. That is, such a function f will map from strings x to the set $\{z \mid$ some accepting path of $M(x)$ has

[1] Supported in part by Grants DARPA-F30602-98-2-0133, NSF-CCR-9322513, NSF-CCR-9701911, NSF-INT-9726724, NSF-INT-9815095/DAAD-315-PPP-gü-ab, NSF-DUE-9980943, NIH-R01-AG18231, and NIH-P30-AG18254. Work done in part while the first author was visiting Friedrich-Schiller-Universität Jena and Julius-Maximilians-Universität Würzburg, and while the third author was visiting RIT.

0022-0000/02 $35.00

z as its output (i.e., as a *solution*)}. $\mathrm{NP}_{\mathbb{N}^+}\mathrm{V}$ functions, known in the literature as NPMV (nondeterministic polynomial-time (potentially) multivalued) functions, have been extensively studied since they were introduced in the 1970s by Book *et al.* ([1, 2], see also the excellent survey by Selman [25]).

Much of this study has recently focused on the issue of whether even NP functions can prune solutions away from NP functions. As Naik *et al.* [20] have elegantly pointed out, the motivation for this is multifold: in the broadest sense this addresses the central complexity-theoretic notion of measuring how resources (such as allowed output cardinality) enable computation, more specifically this addresses the power of nondeterminism, and more specifically still this issue is deeply tied ([15, 26], see also [9, 14]) to NP-search functions and the complexity of function inversion. Also worth contrasting with this paper's proof that number of solutions of NP functions can be reduced in various ways is the fact, due to Ogiwara and Hemachandra [22], that (unless surprising complexity class collapses occur) one cannot in general reduce even by one (proper decrement) the number of accepting paths of NP machines.

To discuss rigorously whether NP functions can prune solutions from NP functions, we need a formal way to capture this. The notion of refinement exactly captures this, and is used in the literature for exactly this purpose. Given (possibly partial, possibly multivalued) functions f and f', we say that f' is a refinement (see for example the excellent survey by Selman [26]) of f if for each $x \in \Sigma^*$,

1. $f'(x)$ has at least one solution iff $f(x)$ has at least one solution, and
2. each solution of $f'(x)$ is a solution of $f(x)$.

Given any two function classes $\mathscr{C}_1$ and $\mathscr{C}_2$, we say that $\mathscr{C}_1 \subseteq_c \mathscr{C}_2$ ("$\mathscr{C}_1$ functions always have $\mathscr{C}_2$ refinements") if for each function $f \in \mathscr{C}_1$ there is a function $f' \in \mathscr{C}_2$ such that f' is a refinement of f.

For any $A \subseteq \mathbb{N}^+$, $\mathrm{NP}_A\mathrm{V}$ denotes the class of all $\mathrm{NP}_{\mathbb{N}^+}\mathrm{V}$ functions f satisfying $(\forall x \in \Sigma^*)$ [the number of solutions of $f(x)$ is an element of $\{0\} \cup A$].

Surprisingly, for the first 20 years after the classes $\mathrm{NP}_{\mathbb{N}^+}\mathrm{V}$ and $\mathrm{NP}_{\{1\}}\mathrm{V}$ (referred to in the literature respectively as NPMV and NPSV: nondeterministic polynomial-time {multivalued, single-valued} functions) were defined, there was no evidence against the dramatic possibility that $\mathrm{NP}_{\mathbb{N}^+}\mathrm{V} \subseteq_c \mathrm{NP}_{\{1\}}\mathrm{V}$, i.e., that all multivalued NP functions have single-valued NP refinements. (This is known to be equivalent to the claim that there is an NP function that on each satisfiable boolean formula as input finds exactly one satisfying assignment.) In the 1990s, Hemaspaandra, Naik, Ogihara, and Selman [13] finally gave concrete evidence against this by proving the following result.[2]

THEOREM 1.1 [13]. *If* $\mathrm{NP}_{\mathbb{N}^+}\mathrm{V} \subseteq_c \mathrm{NP}_{\{1\}}\mathrm{V}$ *then* $\mathrm{PH} = \mathrm{ZPP}^{\mathrm{NP}}$.

[2] Regarding both Theorems 1.1 and 1.2, the actual proofs of Hemaspaandra *et al.* [13] obtain the $\mathrm{ZPP}^{\mathrm{NP}}$ conclusion via proving that the hypothesis implies $\mathrm{NP} \subseteq (\mathrm{NP} \cap \mathrm{coNP})/\mathrm{poly}$, which itself by a result of Köbler and Watanabe [17] implies that $\mathrm{PH} = \mathrm{ZPP}^{\mathrm{NP}}$. Let S_2 denote the symmetric alternation class of Canetti [7] and Russell and Sundaram [23]. We mention in passing that new work of Cai *et al.* [5], which builds on the work of Cai [4], shows that $\mathrm{NP} \subseteq (\mathrm{NP} \cap \mathrm{coNP})/\mathrm{poly}$ implies $\mathrm{PH} = \mathrm{S}_2{}^{\mathrm{NP} \cap \mathrm{coNP}}$ and also shows that $\mathrm{S}_2{}^{\mathrm{NP} \cap \mathrm{coNP}} \subseteq \mathrm{ZPP}^{\mathrm{NP}}$. Thus, for both Theorems 1.1 and 1.2, one can in fact reach the stronger conclusion that $\mathrm{PH} = \mathrm{S}_2{}^{\mathrm{NP} \cap \mathrm{coNP}}$.

Thus, if the polynomial hierarchy does not collapse, the following remarkable state holds: NP functions can find all satisfying assignments of boolean formulas but cannot find (exactly) one satisfying assignment to boolean formulas [13]. (Though it would be impossible for such a claim to hold for deterministic computation, as there finding one solution is provably no harder than finding all solutions, for nondeterministic computation this state is neither impossible nor paradoxical, though the fact that finding all solutions is simpler than finding one solution may at first be disconcerting.)

In fact, Hemaspaandra *et al.* proved something a bit stronger than Theorem 1.1.

THEOREM 1.2 [13]. $\mathrm{NP}_{\{1,2\}}\mathrm{V} \subseteq_c \mathrm{NP}_{\{1\}}\mathrm{V}$ *only if* $\mathrm{PH} = \mathrm{ZPP}^{\mathrm{NP}}$.

Building on this, Ogihara [21] and Naik *et al.* [20] showed that from weaker hypotheses one could reach weaker conclusions that nonetheless are strong enough to cast strong doubt on their hypotheses.

THEOREM 1.3 [21]. *For each* α, $0 < \alpha < 1$, *it holds that if* $\mathrm{NP}_{\mathbb{N}^+}\mathrm{V} \subseteq_c \mathrm{NP}_{\{1, \ldots, \max\{1, \lfloor n^\alpha \rfloor\}\}}\mathrm{V}$ *(where n represents the length of the input) then* $\mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}$.

THEOREM 1.4 [20]. *For each* $k \geqslant 1$, *if* $\mathrm{NP}_{\{1, \ldots, k+1\}}\mathrm{V} \subseteq_c \mathrm{NP}_{\{1, \ldots, k\}}\mathrm{V}$ *then* $\mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}$.

Theorems 1.1, 1.2, 1.3, and 1.4 say that, for the cases they cover, one cannot prune solutions unless the polynomial hierarchy collapses.

However, note that all these theorems cover cases in which the allowed nonzero solution cardinalities form a (finite or infinite) prefix of $\{1,2,3, \ldots\}$. That is, the theorems deal just with the following question: Given any NP function having on each input at most ℓ solutions (ℓ is either ∞ or an element of $\mathbb{N}^+$) will it always be the case that there exists another NP function that is a refinement of the first function and that on each input has at most ℓ' solutions (ℓ', $\ell' < \ell$, is an element of $\mathbb{N}^+$)? In fact, prior to the present paper only such "left-prefix-of-$\mathbb{N}^+$"-cardinality sets had been studied.

We introduce the notion $\mathrm{NP}_A\mathrm{V}$ and propose as natural the following general challenge.

Challenge 1.1. Completely characterize, perhaps under some complexity-theoretic assumption, the sets $A \subseteq \mathbb{N}^+$ and $B \subseteq \mathbb{N}^+$ such that $\mathrm{NP}_A\mathrm{V} \subseteq_c \mathrm{NP}_B\mathrm{V}$.

This question captures far more fully the issue of what types of cardinality reduction are generally possible via refinement of NP functions. Further, this also parallels the way language classes have been defined in complexity theory. There, notions of "acceptance types" and "promises about numbers of accepting paths" are natural. In fact, a language notion "NP_A" can be found in the literature [6], and unifies many notions of counting-based acceptance (and see more generally the notion of leaf languages [3, 27]). In our function case, we view the A of $\mathrm{NP}_A\mathrm{V}$ as a *cardinality type* since it specifies the allowed nonzero numbers of solutions.

Challenge 1.1 is very broad and ambitious, as it goes well beyond the cases considered in Theorems 1.1, 1.2, 1.3, and 1.4. The present paper focuses on the case of finite cardinality types—$\mathrm{NP}_A\mathrm{V}$ for sets $A \subseteq \mathbb{N}^+$ satisfying $\|A\| < \infty$. Section 2 presents a condition, for sets $A, B \subseteq \mathbb{N}^+$, $\|A\| < \infty$, $\|B\| < \infty$, sufficient to ensure $\mathrm{NP}_A\mathrm{V} \subseteq_{\mathrm{c}} \mathrm{NP}_B\mathrm{V}$. This condition is not a complexity-theoretic assumption but rather is a simple statement about the sets A and B. *Thus, we will see that in many cases solution reduction is possible for* NP *functions, in contrast to Theorems 1.1, 1.2, 1.3, and 1.4 and in contrast to the known result [22] that unless shocking complexity class collapses occur accepting-path-cardinality reduction is not in general possible for* NP *machines.*

We conjecture that for finite cardinality types our sufficient condition is necessary unless the polynomial hierarchy collapses. Though we cannot prove that, Section 3 establishes broad necessary conditions for solution reduction under the assumption that the polynomial hierarchy does not collapse. These conditions subsume the previously known cases obtained in Hemaspaandra *et al.* and Naik *et al.* We also prove an absolute necessary condition, but we show that proving any sufficiently broad absolute necessary condition would immediately yield a proof that $\mathrm{NP} \neq \mathrm{coNP}$.

Section 4 revisits Theorem 1.4, which says that $\mathrm{NP}_{\{1, ..., k+1\}}\mathrm{V} \subseteq_{\mathrm{c}} \mathrm{NP}_{\{1, ..., k\}}\mathrm{V}$ implies $\mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}$. Of course, most complexity researchers, deep down, believe that $\mathrm{NP}_{\{1, ..., k+1\}}\mathrm{V} \not\subseteq_{\mathrm{c}} \mathrm{NP}_{\{1, ..., k\}}\mathrm{V}$. If this belief is a correct guess about the state of the world, then Theorem 1.4 tells us nothing, as it is of the form "false $\Rightarrow \cdots$." Intuitively, one would hope that Theorem 1.4 is a reflection of some structural simplicity property of sets. Section 4 proves that this is indeed the case, via showing, along with an even broader result, that all NP sets that are $(k+1)$-selective via $\mathrm{NP}_{\{1, ..., k\}}\mathrm{V}$ functions in fact belong to the second level of Schöning's low hierarchy [24]. Section 5 provides a more unified strengthening.

2. A SUFFICIENT CONDITION

We now state our sufficient condition. Intuitively, one can think of this as a "narrowing-gap" condition as it says that the gaps[3] between the cardinalities in A and certain of the cardinalities in B have to form a (perhaps nonstrictly) decreasing sequence.

THEOREM 2.1. *For each pair of finite sets $A \subseteq \mathbb{N}^+$ and $B \subseteq \mathbb{N}^+$, $A = \{a_1, ..., a_m\}$ with $a_1 < a_2 < \cdots < a_m$, we have* $\mathrm{NP}_A\mathrm{V} \subseteq_{\mathrm{c}} \mathrm{NP}_B\mathrm{V}$ *if*

$$\|A\| = 0 \textit{ or}$$
$$(\exists b_1, ..., b_m : 0 < b_1 < \cdots < b_m)[\{b_1, ..., b_m\} \subseteq B \textit{ and}$$
$$a_1 - b_1 \geqslant \cdots \geqslant a_m - b_m \geqslant 0].$$

[3] "Gap" is used here in its common-language sense of differences between integers (that here happen to represent cardinalities of outputs) rather than in its term-of-art complexity-theoretic sense of differences between integers representing cardinalities of accepting paths. Also, the "$0 < b_1 < \cdots < b_m$" is explicitly stated in Theorem 2.1 only for clarity; even if left out it would have to hold were the condition to be satisfied.

Proof. All the functions in $\mathrm{NP}_{\varnothing}\mathrm{V}$ are undefined everywhere so they belong to $\mathrm{NP}_B\mathrm{V}$ for all $B \subseteq \mathbb{N}^+$. Let $m \geqslant 1$ and let $A = \{a_1, ..., a_m\}$ and $B \supseteq \{b_1, ..., b_m\}$ be as in the hypothesis of the theorem. Let f be a function in $\mathrm{NP}_A\mathrm{V}$ and let M be a polynomial-time nondeterministic Turing machine M witnessing that $f \in \mathrm{NP}_A\mathrm{V}$. Throughout this paper we follow the standard convention that only accepting computation paths are considered to have outputs; rejecting paths are never said to have outputs (this makes it possible for machines to have no outputs on some inputs). Let N be a nondeterministic Turing machine that, given $x \in \Sigma^*$ as input, behaves as follows:

Step 1. N nondeterministically chooses an integer k, $1 \leqslant k \leqslant m$.

Step 2. N nondeterministically simulates M on x a_k times. If on at least one simulation M fails to produce an output, N rejects x.

Step 3. For each i, $1 \leqslant i \leqslant a_k$, let y_i be the output of M on x obtained in the ith simulation. If $y_j = y_{j'}$ for some $1 \leqslant j < j' \leqslant a_k$, N rejects x.

Step 4. N nondeterministically chooses an integer j, $1 \leqslant j \leqslant b_k$, and outputs the jth smallest element of $\{y_1, ..., y_{a_k}\}$.

Since A is finite and fixed, the machine N runs in nondeterministic polynomial time. Let $x \in \Sigma^*$ be arbitrary. Clearly, if $f(x)$ is undefined (i.e., has no outputs) then N on x does not have an output. Suppose $f(x)$ has precisely a_t distinct values for some t, $1 \leqslant t \leqslant m$. Let $z_1, ..., z_{a_t}$ be the outputs of $f(x)$ enumerated in increasing order. Then a computation path of N on input x arrives at Step 4 if and only if the number k chosen by N in Step 1 is at most t and N in Step 2 obtained a_k distinct output values of N on x. Furthermore, in Step 4, the largest $a_k - b_k$ of the obtained outputs of $f(x)$ will be trashed. So,

- if $k > t$, then N on x has no outputs,
- if $k = t$, then N on x outputs precisely $z_1, ..., z_{b_t}$, and
- if $k < t$, then N on x outputs no string other than z_i's and never outputs $z_{a_t-(a_k-b_k)+1}, ..., z_{a_t}$.

Since $a_1 - b_1 \geqslant \cdots \geqslant a_t - b_t \geqslant 0$, the last condition implies that

- If $k < t$, then N on x outputs no string other than z_i's and never outputs $z_{b_t+1}, ..., z_{a_t}$.

Thus, the outputs of N on x are precisely $z_1, ..., z_{b_t}$. This implies that $f \in \mathrm{NP}_B\mathrm{V}$. Hence, as f was an arbitrary $\mathrm{NP}_A\mathrm{V}$ function, $\mathrm{NP}_A\mathrm{V} \subseteq_{\mathrm{c}} \mathrm{NP}_B\mathrm{V}$. ▮

3. NECESSARY CONDITIONS

We conjecture that for finite cardinality types the narrowing-gap sufficient condition from Theorem 2.1 is in fact necessary unless the polynomial hierarchy collapses.

Narrowing-Gap Conjecture. For each pair of finite sets $A \subseteq \mathbb{N}^+$ and $B \subseteq \mathbb{N}^+$ that do not satisfy the condition of Theorem 2.1, we have:

$$\mathrm{NP}_A\mathrm{V} \subseteq_c \mathrm{NP}_B\mathrm{V} \Rightarrow \mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}.$$

Why did we not make an even stronger version of the conjecture that asserts that for finite cardinality types the condition of Theorem 2.1 is (unconditionally) necessary? After all, certain finite cardinality types violating the condition of Theorem 2.1 trivially do not allow solution reduction, as the following result shows.

PROPOSITION 3.1. *Let* $A \subseteq \mathbb{N}^+$, $B \subseteq \mathbb{N}^+$, $A \neq \emptyset$, *and* $B \neq \emptyset$. *If* $\min\{i \mid i \in A\} < \min\{i \mid i \in B\}$ *then* $\mathrm{NP}_A\mathrm{V} \not\subseteq_c \mathrm{NP}_B\mathrm{V}$.

Proof. Let A and B be as in the hypothesis. Let $m = \min\{i \mid i \in A\}$. Let f be the function that maps from each $x \in \Sigma^*$ to the numbers $\{1, ..., m\}$. Clearly, $f \in \mathrm{NP}_A\mathrm{V}$. Since f has exactly m outputs on each input, for any function g to be a refinement of f, $g(x)$ has to have output cardinality between 1 and m for every $x \in \Sigma^*$. However, this is not possible for any function in $\mathrm{NP}_B\mathrm{V}$ since $\min\{i \mid i \in B\} > m$. ▮

Nonetheless, unconditionally showing that the condition of Theorem 2.1 is necessary for finite accepting types seems out of reach. The reason is that, due to the following result, showing the condition to be necessary would in fact prove that $\mathrm{NP} \neq \mathrm{coNP}$.

THEOREM 3.1. *If* $\mathrm{NP} = \mathrm{coNP}$ *then for any set* $A \subseteq \mathbb{N}^+$ *it holds that* $\mathrm{NP}_A\mathrm{V} \subseteq_c \mathrm{NP}_{\{1\}}\mathrm{V}$.

In contrast, the Narrowing-Gap Conjecture does not seem to imply $\mathrm{NP} \neq \mathrm{coNP}$, or any other unexpected fact, in any obvious way. We suggest that the Narrowing-Gap Conjecture is a plausible long-term goal.

The following result shows that, if $\mathrm{PH} \neq \mathrm{NP}^{\mathrm{NP}}$, then a wide range of cardinality types A and B do not have solution reduction.

THEOREM 3.2. *Let* $A, B \subseteq \mathbb{N}^+$ *be nonempty. Suppose there exist four integers* $c > 0$, $d > 0$, $e \geqslant 0$, *and* $\delta \geqslant 0$ *satisfying the following conditions:*

- $d \leqslant c \leqslant 2d$ *and* $\delta < 2d - c$,
- $c, 2d + e \in A$,
- $c - \delta \leqslant \min\{i \mid i \in B\} \leqslant c$, *and*
- $2d - (2\delta + 1) \geqslant \max\{i \in B \mid i \leqslant 2d + e\}$.

Then $\mathrm{NP}_A\mathrm{V} \subseteq_c \mathrm{NP}_B\mathrm{V}$ *implies* $\mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}$.

The statement of Theorem 3.2 is not simple. So, we give an informal explanation of what the theorem is expressing.

Let $A, B \subseteq \mathbb{N}^+$ be finite, let f be any function in $\mathrm{NP}_A\mathrm{V}$, and let $g \in \mathrm{NP}_B\mathrm{V}$ be a refinement of f. Let x be an input to f and g on which f has at least one output. Suppose the output cardinality of $f(x)$, say α, is not a member of B (of course, by definition, $\alpha \in A$). Since g is a refinement of f, $g(x)$ has to have some output values, which must be selected from those of $f(x)$. Then $\alpha \notin B$ indicates that g has to exclude some of the output values of $f(x)$ from its output list. How many does $g(x)$ have to exclude? Let $\mu = \max\{i \in B \mid i \leqslant \alpha\}$. Then the number of output values $g(x)$ has to exclude is at least $\alpha - \mu$. On the other hand, suppose the output cardinality of $f(x)$ is equal to $\alpha' < \alpha$ and let $\nu = \min(B)$. Then the maximum number of output values that $g(x)$ can legally exclude, if necessary at all, is at most $\alpha' - \nu$. In Theorem 2.1 we observed that such reduction in the number of output values is essentially possible if $\alpha' - \nu$ is at least $\alpha - \mu$ (and this holds for all the other combinations of α and α' in A). Now we ask: What if $\alpha' - \nu < \alpha - \mu$? Theorem 3.2 essentially shows that, with some additional requirements, if $2(\alpha' - \nu) < \alpha - \mu$, then $\mathrm{NP}_A\mathrm{V}$ being refined to $\mathrm{NP}_B\mathrm{V}$ collapses the polynomial hierarchy.

We will prove this theorem in Section 5.

Theorem 3.2 is a rather complex necessary[4] condition, as it is loaded with degrees of freedom to let it be broad. Nonetheless, there are some cases it misses, for example due to the fact that the $d \leqslant c \leqslant 2d$ clause can limit us when dealing with certain cardinality types with widely varying values. For example, regarding cardinality-2 cardinality types, Theorem 3.2 yields as a corollary result 3.1 below. However, we can also prove Theorem 3.3, which is another necessary-condition theorem and which seemingly does not follow (in any obvious way) from Theorem 3.2. We will prove Theorem 3.3 in Section 5.

Corollary 3.1. *For any integers $k > 0$, $k' > 0$, $k' \geqslant k$, we have the following: If* $\mathrm{NP}_{\{k-1, k'\}}\mathrm{V} \subseteq_{\mathrm{c}} \mathrm{NP}_{\{k-1\}}\mathrm{V}$ *then* $\mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}$.

Theorem 3.3. *Let $k \geqslant 2$ and d, $1 \leqslant d \leqslant k-1$, be integers. Let $A, B \subseteq \mathbb{N}^+$ be such that $\binom{k-1}{k-d} \in A$, $\binom{k}{k-d} \in A$, and* $\max\{i \mid i \in B$ *and* $i \leqslant \binom{k}{k-d}\} \leqslant \lceil \frac{k}{d} \rceil - 1$. *Then* $\mathrm{NP}_A\mathrm{V} \subseteq_{\mathrm{c}} \mathrm{NP}_B\mathrm{V}$ *implies* $\mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}$.

In fact, Theorem 3.3, a necessary-condition theorem quite different from Theorem 3.2, has some very useful corollaries. For example, the necessary condition of Naik *et al.* (Theorem 1.4) follows immediately from Theorem 3.3 by plugging $d = 1$ into the above; in fact, doing so gives the statement

(★★) For each $k \geqslant 1$, if $\mathrm{NP}_{\{1, k+1\}}\mathrm{V} \subseteq_{\mathrm{c}} \mathrm{NP}_{\{1, \ldots, k\}}\mathrm{V}$ then $\mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}$,

which is even stronger than the Naik *et al.* result. However, we note that if one closely examines the proof of Naik *et al.* one can in fact see that their *proof* establishes (★★).

Theorem 3.3 yields other interesting necessary conditions. As an example, from the $d = 2$ case we can certainly conclude the following result.

Corollary 3.2. *For each $k > 2$, if* $\mathrm{NP}_{\{k-1, \binom{k}{2}\}}\mathrm{V} \subseteq_{\mathrm{c}} \mathrm{NP}_{\{1, \ldots, \lceil k/2 \rceil - 1\}}\mathrm{V}$ *then* $\mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}$.

[4] Recall that by this somewhat unusual use of "necessary," and the later uses in the same way, we mean exactly what was discussed at the start of this section; i.e., we are dealing with theorems that move toward the issue of, for finite cardinality types, whether the narrowing-gap sufficient condition from Theorem 2.1 is in fact necessary unless the polynomial hierarchy collapses.

Here we give another example of using Theorem 3.2 to prove necessary conditions. Setting the d and c of Theorem 3.2 to k, the δ to δ, and the e to 0, we obtain the following.

COROLLARY 3.3. *Let k and δ be integers such that $k \geqslant 1$ and $0 \leqslant \delta \leqslant k-1$. Let $A, B \subseteq \mathbb{N}^+$ be nonempty sets such that $\{k, 2k\} \subseteq A$, $\min\{i \in B \mid i \leqslant 2k\} = k-\delta$, and $\max\{i \in B \mid i \leqslant 2k\} \leqslant 2k-2\delta-1$. Then $\mathrm{NP}_A\mathrm{V} \subseteq_c \mathrm{NP}_B\mathrm{V}$ implies $\mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}$.*

4. LOWNESS RESULTS

We now prove another strengthening of the result of Naik *et al.* [20] stated here as Theorem 1.4. Namely, we show a lowness result—a general result about the simpleness of sets having certain properties—from which Naik *et al.*'s Theorem 1.4 is a consequence. Informally, lowness captures the level of the polynomial hierarchy, if any, at which a given NP set becomes worthless as an oracle—the level at which it gives that level no more additional information than would the empty set. Of interest to us will be the class of sets for which this level is two.

DEFINITION 4.1 [20]. For any integer $k > 0$ and any function class $\mathscr{FC}$ we say that a set A is $\mathscr{FC}$-k-selective if there is a function $f \in \mathscr{FC}$ such that for every k distinct strings $b_1, ..., b_k$,

1. every output of $f(b_1, ..., b_k)$ is a cardinality $k-1$ subset of $\{b_1, ..., b_k\}$ and
2. if $\|\{b_1, ..., b_k\} \cap A\| \geqslant k-1$, then $f(b_1, ..., b_k)$ has at least one output and each set output by $f(b_1, ..., b_k)$ is a subset of A.

DEFINITION 4.2 [24]. $\mathrm{Low}_2 = \{A \in \mathrm{NP} \mid \mathrm{NP}^{\mathrm{NP}^A} = \mathrm{NP}^{\mathrm{NP}}\}$.

THEOREM 4.1. *For each $k \in \{2, 3, ...\}$, it holds that every $\mathrm{NP}_{\{1, ..., k-1\}}\mathrm{V}$-$k$-selective NP set belongs to Low_2.*

We will postpone proving this theorem until Section 5.

Theorem 1.4 certainly follows from Theorem 4.1. In fact, recall the stronger form of Theorem 1.4 that we noted in Section 3 (marked (★★)). Since SAT is $\mathrm{NP}_{\{1, ..., k-1\}}\mathrm{V}$-$k$-selective and even $\mathrm{NP}_{\{1, k-1\}}\mathrm{V}$-$k$-selective [20], even this stronger form of Theorem 1.4 follows immediately from Theorem 4.1. In fact, note that Theorem 4.1 establishes, for example, the simpleness of NP's $\mathrm{NP}_{\{1, ..., k-1\}}\mathrm{V}$-$k$-selective sets.

5. A UNIFIED STRENGTHENING

Note that in the previous sections we have stated extensions of the work of Naik *et al.* (Theorem 1.4) in two incomparable ways, namely providing as Theorem 3.3 a broader necessary condition and as Theorem 4.1 a general lowness theorem that implied the Naik *et al.* result. It is very natural to ask whether our two results can be unified, via proving a lowness result that itself implies not just Theorem 1.4 but all the necessary conditions we identify in this paper. In fact, the answer is yes. We have the following result, which provides exactly such a unification.

DEFINITION 5.1. Let $k \geqslant 2$ be an integer. A parameter tuple for input size k is a $k+3$ tuple $\Lambda = \langle \ell_0, ..., \ell_{k-1}, \alpha, \beta, \gamma \rangle$ of nonnegative integers such that

- at least one of $\ell_1, \ldots, \ell_{k-1}$ is positive,
- $0 \leqslant \alpha \leqslant \sum_{1 \leqslant i \leqslant k-1} \binom{k-1}{i} \ell_i$,
- $0 \leqslant \beta \leqslant \sum_{1 \leqslant i \leqslant k-1} (\binom{k}{i} - \binom{k-1}{i}) \ell_i$, and
- $0 \leqslant \gamma \leqslant \ell_0$.

Definition 5.2. Let $k \geqslant 2$ be an integer. Let Λ be a parameter tuple for input size k. Let $\mathscr{F}\mathscr{C}$ be a class of multivalued functions. A language A is $\mathscr{F}\mathscr{C}$-(k, Λ)-selective if there is some $f \in \mathscr{F}\mathscr{C}$ such that, for every set X of k distinct strings $x_1, \ldots, x_k$, the following properties hold:

1. Each output value of $f(X)$ belongs to the union of the following three classes of strings:

Class A $\{\langle i, j, W \rangle \mid 1 \leqslant i \leqslant k-1$ and $1 \leqslant j \leqslant \ell_i$ and W is a cardinality i subset of $X \cap A\}$;

Class B $\{\langle i, j, W \rangle \mid 1 \leqslant i \leqslant k-1$ and $1 \leqslant j \leqslant \ell_i$ and W is a cardinality i subset of X containing at least one member of $\bar{A}\}$; and

Class C $\{\langle 0, j \rangle \mid 0 \leqslant j \leqslant \ell_0\}$;

2. If $\|X \cap A\| = k-1$, then $f(X)$ should output no more than α Class A strings, no more than β Class B strings, and no more than γ Class C strings.

Definition 5.3. Let $k \geqslant 2$ and $\Lambda = \langle \ell_0, \ldots, \ell_{k-1}, \alpha, \beta, \gamma \rangle$ be a parameter tuple for input size k. Let $B \subseteq \mathbb{N}^+$ be nonempty and finite. Define the predicate

$$Q(k, \Lambda, B) = [\alpha + \beta + \gamma \geqslant \min\{i \mid i \in B\} \text{ and } k\Delta < \sum_{1 \leqslant i \leqslant k-1} (k-i)\, t_i],$$

where Δ and $t_1, \ldots, t_{k-1}$ are defined as follows:

- $s_k = 0$, $s'_k = 0$, and for each d, $1 \leqslant d \leqslant k-1$, $s_d = \sum_{d \leqslant i \leqslant k-1} \binom{k-1}{i} \ell_i$ and $s'_d = \sum_{d \leqslant i \leqslant k-1} \binom{k}{i} \ell_i$.
- $\Delta = (s_1 + \beta + \gamma) - \min\{i \mid i \in B\}$.
- $\Delta' = \max\{0, s'_1 - \max\{i \in B \mid i \leqslant s'_1 + \ell_0\}\}$.
- For each d, $1 \leqslant d \leqslant k-1$, $t_d = \max\{0, \min\{\Delta' - s'_{d+1}, s'_d - s'_{d+1}\}\}$.

Note that in the above definition the $\max\{i \in B \mid i \leqslant s'_1 + \ell_0\}$, which is used to define the second conjunct of Q, is well-defined conditionally upon the first conjunct holding, as in that case we have $\min\{i \mid i \in B\} \leqslant \beta + \gamma \leqslant (\sum_{1 \leqslant i \leqslant k-1} (\binom{k}{i} - \binom{k-1}{i}) \ell_i) + \ell_0 \leqslant s'_1 + \ell_0$.

Theorem 5.1. *Let $k \geqslant 2$ be an integer, let Λ be a parameter tuple for input size k, and let B be a nonempty finite set of positive integers such that $Q(k, \Lambda, B)$ holds. Then every* $\mathrm{NP}_B\mathrm{V}$-(k, Λ)*-selective set in* NP *belongs to* Low_2.

Here we give a brief, informal overview of the proof of Theorem 5.1. Let k, Λ, and B be as in the hypothesis of the theorem. Let L be an NP set that is $\mathrm{NP}_B\mathrm{V}$-(k, Λ)-selective. Let f be an $\mathrm{NP}_B\mathrm{V}$-(k, Λ)-selector for L. Let X be a $(k+1)$-tuple consisting of $k+1$ distinct strings $w_1, \ldots, w_{k+1}$. Suppose that all these

w's are members of L. Since the property $Q(k, \Delta, B)$ holds there are some Class A or Class B strings that $f(X)$ does not output. Let U be the set of all such strings. For each i, $1 \leqslant i \leqslant k+1$, and each $u = \langle W, j \rangle$ that $f(X)$ does not output, we say that u misses w_i if $w_i \notin W$. For each i, $1 \leqslant i \leqslant k+1$, we count the number of strings in U that miss w_i. Then we show that by the Pigeonhole Principle there is some i, $1 \leqslant i \leqslant k+1$, such that more than Δ strings in U miss w_i.

On the other hand, suppose that exactly k of the $k+1$ strings in X are members of L. Let us say that w_1 is not a member. The property $Q(k, \Delta, B)$ guarantees that for no i, $2 \leqslant i \leqslant k+1$, is it the case that the number of strings in U that miss w_i is greater than Δ.

By these two properties we establish that if $w_1, \ldots, w_{k+1}$ are $k+1$ distinct strings, $w_2, \ldots, w_{k+1}$ are members of L, and the number of strings in U that miss w_1 is at most Δ, then w_1 belongs to L (see Claims 5.1, 5.2, and 5.4). We use a divide-and-conquer argument (inspired by the work of Ko [16]) to show that for each n there is a polynomial collection of strings in $L^{\leqslant n}$ with which membership of all strings having length at most n can be tested by a coNP language (see Claim 5.3). Now the Σ_2^p-lowness of L can be shown using this observation (Claim 5.5).

Proof of Theorem 5.1. Let k, $\Delta = \langle \ell_0, \ldots, \ell_{k-1}, \alpha, \beta, \gamma \rangle$, and B be as in the hypothesis of the theorem. We henceforth view them as fixed so, for example, we will not explicitly parametrize T_{loser} below with k. Let Δ, Δ', $s_1, \ldots, s_k$, $s'_1, \ldots, s'_k$, and $t_1, \ldots, t_{k-1}$ be as given in Definition 5.3. Let $A \in \mathrm{NP}$ be an arbitrary $\mathrm{NP}_B\mathrm{V}$-(k, Δ)-selective NP set and let M be a polynomial-time nondeterministic Turing machine accepting A. Let $f \in \mathrm{NP}_B\mathrm{V}$ be a function witnessing the $\mathrm{NP}_B\mathrm{V}$-(k, Δ)-selectivity of A. Let Y be a set of $k-1$ distinct strings and let z be a string not in Y. Let $u = \langle i, j, W \rangle$ be an output value of $f(Y \cup \{z\})$, where $1 \leqslant i \leqslant k-1$, $1 \leqslant j \leqslant \ell_i$, $W \subseteq Y \cup \{z\}$, and $\|W\| = i$. We say that u *hits* z if $z \in W$ and *misses* z otherwise. For every Class C output u of $f(Y \cup \{z\})$, we say that u neither hits nor misses z. We say that z *loses* to Y if out of all possible output strings of $f(Y \cup \{z\})$ that miss z, there are more than Δ such strings that are not outputs of $f(Y \cup \{z\})$. Define $T_{loser} = \{\langle z, Y \rangle \mid z \in \Sigma^*, \|Y\| = k-1$, and either $z \in Y$ or ($z \notin Y$ and z loses to Y)$\}$.

Claim 5.1. Let Y be a cardinality $k-1$ subset of A and z be a member of $\bar{A}$. Then z does not lose to Y.

Proof of Claim 5.1. Let Y and z be as in the hypothesis of the claim. Since $z \in \bar{A}$, $z \notin Y$. Let $X = Y \cup \{z\}$. By Property 2 of Definition 5.2, there are at most s_1 Class A output strings of $f(X)$, at most β Class B output strings of $f(X)$, and at most γ Class C output strings of $f(X)$. Since $\Delta = s_1 + \beta + \gamma - \min\{i \mid i \in B\}$, $f(X)$ has at least $s_1 + \beta + \gamma - \Delta$ output values. So, it has at least $s_1 - \Delta$ Class A output strings. Since only Class A output strings of $f(X)$ miss z, the number of potential output strings of $f(X)$ missing z that are not outputs of $f(X)$ is at most Δ. Then, by definition, z does not lose to Y. ▌

Claim 5.2. Let X be a set of k distinct strings in A. Then there exists an element $z \in X$ such that z loses to $X - \{z\}$.

Proof of Claim 5.2. Let $X=\{x_1, ..., x_k\}$ be a set of k distinct strings in A. Let $\mathscr{R}$ be the set of all possible Class A strings that $f(X)$ does not output. Since $X \subseteq A$, there are no Class B outputs of $f(X)$, so there are $s'_1+\ell_0$ potential output strings of $f(X)$, out of which ℓ_0 are Class C strings. Since $\Delta'=\max\{0, s'_1-\max\{i \in B \mid i \leqslant s'_1+\ell_0\}\}$, there are at least Δ' Class A strings that $f(X)$ does not output, and thus, $\|\mathscr{R}\| \geqslant \Delta'$. For each i, $1 \leqslant i \leqslant k$, let θ_i be the number of strings in $\mathscr{R}$ that miss x_i. Then the statement of the claim we are proving is equivalent to:

$$\max\{\theta_1, ..., \theta_k\} > \Delta.$$

By the Pigeonhole Principle, it suffices to show that:

$$\sum_{1 \leqslant i \leqslant k} \theta_i > k\Delta.$$

In other words, the total number of "misses" (with multiple counting) in the set $\mathscr{R}$ is more than $k\Delta$.

Let $u=\langle i, j, W\rangle$ be an arbitrary Class A output string of $f(X)$. Since $\|W\|=i$, there are precisely $k-i$ elements that u misses. Then the total number of misses (with multiple counting) in $\mathscr{R}$ is minimized by selecting $\mathscr{R}$'s elements $\langle i, j, W\rangle$ *greedily* from the pool of potential Class A output strings without replacement; i.e., we will always select an output string with the largest first component amongst all the remaining output strings in the pool. For every i, $1 \leqslant i \leqslant k-1$, there are exactly $\binom{k}{k-i}\ell_{k-i}$ output strings in $\mathscr{R}$ with exactly i misses. So, for every d, $1 \leqslant d \leqslant k-1$, s'_d is the number of output strings in $\mathscr{R}$ with at most $k-d$ misses. Noting that we need to put Δ' elements in $\mathscr{R}$, we observe that, with the above greedy strategy, for every d, $1 \leqslant d \leqslant k-1$, the total number of output strings that are selected whose number of misses is precisely $k-d$ is determined as follows:

- If $\Delta' \geqslant s'_{d+1}$, then the number of output strings whose number of misses is less than $k-d$ is at least Δ', so there is no need to put elements in $\mathscr{R}$ whose number of misses is $\geqslant k-d$. Thus, the number in question is 0.
- If $\Delta' < s'_{d+1}$, then the number of output strings whose number of misses is less than $k-d$ is smaller than Δ', so at least one output string whose number of misses is precisely $k-d$ has to be put into $\mathscr{R}$. The exact number of such output strings is $s'_d-s'_{d+1}$ if $\Delta' \geqslant s'_d$ and $\Delta'-s'_{d+1}$ otherwise.

Combining the above two we observe that, for every d, $1 \leqslant d \leqslant k-1$, the total number of output strings that are put in $\mathscr{R}$ whose number of misses is precisely $k-d$ is $\max\{0, \min\{s'_d-s'_{d+1}, \Delta'-s'_{d+1}\}\}$, and thus is equal to t_d. Hence, the total number of misses (with multiple counting) is $\sum_{1 \leqslant i \leqslant k-1} (k-i)\, t_i$, and by our supposition this amount is greater than $k\Delta$. This proves the claim. ∎

Claim 5.3. $T_{loser} \in$ coNP.

Proof of Claim 5.3. Define T'_{loser} to be the set of all tuples $\langle z, Y, u_1, ..., u_{\Delta+1}\rangle$ such that

- $z \in \Sigma^*$,
- $\|Y\| = k-1$,
- $u_1, \ldots, u_{\Delta+1}$ are distinct strings, each of the form $\langle i, j, W\rangle$, for some $1 \leqslant i \leqslant k-1$, $1 \leqslant j \leqslant \ell_i$, $\|W\| = i$, and $W \subseteq Y$, and
- $f(Y \cup \{z\})$ outputs none of $u_1, \ldots, u_{\Delta+1}$.

Then T'_{loser} is in coNP. Since there are only a constant number of possible outputs of $f(Y \cup \{z\})$, T_{loser} is polynomial-time disjunctive reducible to T'_{loser}—namely, via a machine that rejects if $\|Y\| \neq k-1$, else accepts if $z \in Y$, else it brute-force disjunctively reduces to T'_{loser}. So, $T_{loser} \in$ coNP. ▮

Claim 5.4. $\{z \mid (\exists Y \subseteq A)[\|Y\| = k-1 \text{ and } z \text{ loses to } Y]\} \subseteq A$.

Proof of Claim 5.4. Let Y be an arbitrary cardinality $k-1$ subset of A and z be an arbitrary string. If $z \in Y$, clearly $z \in A$. If $z \notin Y$ and z loses to Y, then by Claim 5.1, $z \in A$. ▮

Recall that we throughout this paper use $\Sigma^{\leqslant n}$ as a shorthand for $(\Sigma^*)^{\leqslant n}$. Let n_0 be the smallest n such that $A \cap \Sigma^{\leqslant n}$ has at least $k-1$ elements. For each $n \geqslant n_0$, we say that a string W is a *valid advice string* for $\Sigma^{\leqslant n}$ if W is of the form

$$\langle\langle Z_1, \ldots, Z_m\rangle, \langle \pi_{1,1}, \ldots, \pi_{1,k-1}, \ldots, \pi_{m,1}, \ldots, \pi_{m,k-1}\rangle\rangle,$$

where (a) $m \geqslant 1$, (b) for every i, $1 \leqslant i \leqslant m$, $\|Z_i\| = k-1$, and (c) for every i, $1 \leqslant i \leqslant m$, and every j, $1 \leqslant j \leqslant k-1$, π_{ij} is an accepting computation path of M on the jth element of Z_i. Whether W is a valid advice string for $\Sigma^{\leqslant n}$ can be tested in time polynomial in $|W| + n$. Furthermore, we say that W is a *good advice string* for $\Sigma^{\leqslant n}$ if W is a valid advice string and for every $x \in A$ of length at most n, there is some i, $1 \leqslant i \leqslant m$, such that $\langle x, Z_i\rangle \in T_{loser}$, where $\langle Z_1, \ldots, Z_m\rangle$ is the Z-component of W.

Claim 5.5.

1. There exists a polynomial q such that, for every $n \geqslant n_0$, there is a good advice string for $\Sigma^{\leqslant n}$ of length at most $q(n)$.

2. The set $T_{good} = \{\langle 0^n, W\rangle \mid W \text{ is a good advice string for } \Sigma^{\leqslant n}\}$ belongs to coNP.

Proof of Claim 5.5. We first prove part 1. Let Y be any subset of A of cardinality at least k. By Claim 5.2, for each cardinality k subset X of Y, there is at least one $y \in X$ such that y loses to $X - \{y\}$. There are $\binom{\|Y\|}{k}$ cardinality k subsets of Y and there are $\binom{\|Y\|}{k-1}$ cardinality $k-1$ subsets of Y. By standard counting (namely, a version of the Pigeonhole Principle) there is some $Z \subseteq Y$ of cardinality $k-1$ such that at least $\frac{\|Y\|-k+1}{k} \geqslant \frac{\|Y\|}{k} - 1$ members of Y lose to Z.

Let $n \geqslant n_0$. Let $Y_0 = A \cap \Sigma^{\leqslant n}$. Define the sequences $Z_1, \ldots, Z_m$ and $Y_1, \ldots, Y_m$ as follows with the variable i being 0 at the beginning:

Step 1. If $\|Y_i\| = 0$, then set m to i and quit the loop.

Step 2. Otherwise, if $1 \leqslant \|Y_i\| \leqslant k-2$, then set m to $i+1$, set Z_m to any subset of Y_0 of cardinality $k-1$ that includes W_i, set Y_m to $\varnothing$, and quit the loop.

Step 3. Otherwise, set Z_{i+1} to a $Z \subseteq Y_i$ of cardinality $k-1$ that maximizes $\|\{z \in Y_i - Z \mid z \text{ loses to } Z\}\|$ and set Y_{i+1} to $\{z \in Y_i - Z_{i+1} \mid z \text{ does not lose to } Z_{i+1}\}$. Increment i by 1 and go back to Step 1.

By definition, for every $x \in Y_0$, if $x \notin Z_1 \cup \cdots \cup Z_m$, then there is some i, $1 \leqslant i \leqslant m$, such that x loses to Z_i. Thus, $Y_0 \subseteq \{x \mid |x| \leqslant n$ and, for some i, $1 \leqslant i \leqslant m$, $\langle x, Z_i \rangle \in T_{loser}\}$. On the other hand if $x \notin Y_0$, then $x \notin Z_1 \cup \cdots \cup Z_m$ and, by Claim 5.4, x cannot lose to any of $Z_1, ..., Z_m$. So, Y_0 is actually $\{x \mid |x| \leqslant n$ and, for some i, $1 \leqslant i \leqslant m$, $\langle x, Z_i \rangle \in T_{loser}\}$.

For every i, $1 \leqslant i \leqslant m$, $\|Y_i\| \leqslant \|Y_{i-1}\| - ((\|Y_{i-1}\|/k) - 1) - \|Z_i\| = \|Y_{i-1}\| - ((\|Y_{i-1}\|/k) - 1) - (k-1) \leqslant \frac{k-1}{k} \|Y_{i-1}\|$, so $\|Y_i\|/\|Y_{i-1}\| < \frac{k-1}{k}$. Since $\|Y_0\| \leqslant 2^{n+1}$, $m \leqslant cn$ for some constant $c > 0$. Since $A \in \mathrm{NP}$ via M, for each i, $1 \leqslant i \leqslant m$, and each j, $1 \leqslant j \leqslant k-1$, there is an accepting computation path of M on z_{ij}, where z_{ij} is the jth element of Z_i. So, for each i, $1 \leqslant i \leqslant m$, and j, $1 \leqslant j \leqslant k-1$, pick an accepting computation path π_{ij}. By appending $\langle \pi_{1,1}, \dots, \pi_{1,k-1}, \dots, \pi_{m,1}, \dots, \pi_{m,k-1} \rangle$ to the Z-part we get a valid advice string. Since M is polynomial time-bounded—let p_A denote that polynomial—and $m \leqslant cn$, the total length of the advice string will be bounded by $c'np_A(n)$ for some constant $c' > 0$. So, let $q(n) = c'np_A(n)$. So, part 1 of our claim holds.

Now we prove part 2. For each $n \geqslant n_0$ and each valid advice string W for $\Sigma^{\leqslant n}$, let $R(n, W)$ be the set of all strings $x \in \Sigma^{\leqslant n}$ such that $\langle x, Z_i \rangle \in T_{loser}$ for some i, $1 \leqslant i \leqslant m$, where $\langle Z_1, ..., Z_m \rangle$ is the Z-part of W. If W is not a valid advice string for $\Sigma^{\leqslant n}$, $R(n, W) = \emptyset$. By Claim 5.3, then $R(n, W) \subseteq A$. So, W is a good advice string for $\Sigma^{\leqslant n}$ if and only if $R(n, W) \cup \bar{A} \supseteq \Sigma^{\leqslant n}$. Since $T_{loser} \in \mathrm{coNP}$ and $\bar{A} \in \mathrm{coNP}$, the latter condition can be tested in coNP. Thus, part 2 holds. This proves the claim. ▮

Now suppose $L \in \mathrm{NP}^{\mathrm{NP}^A}$ via a pair of nondeterministic polynomial-time Turing machines N_1 and N_2 such that $L = L(N_1^{L(N_2^A)})$. Let p_1 and p_2 be the polynomials bounding the running times of N_1 and N_2, respectively. Let D be the set of all strings $\langle 0^h, W, y \rangle$ satisfying the following conditions:

1. $h = p_2(p_1(n))$ for some n.
2. W is a valid advice string for $\Sigma^{\leqslant h}$.
3. $|y| \leqslant p_1(n)$.
4. There is an accepting computation path ρ of N_2 on input y such that for each query v made and the answer b obtained for v along ρ, it holds that:

   (i) if $b = 1$ (the query is a member of the oracle), then $v \in A$; and

   (ii) if $b = 0$ (the query is not a member of the oracle), then $v \notin R(h, W)$.

Since $A \in \mathrm{NP}$ and R is a coNP-predicate, D is in NP. Furthermore, for every n such that $h = p_2(p_1(n)) \geqslant n_0$, every good advice string W for $\Sigma^{\leqslant h}$, and every $y \in \Sigma^{\leqslant p_1(n)}$, the condition $y \in L(N_2)$ is equivalent to $\langle 0^h, W, y \rangle \in D$. This is because $\Sigma^{\leqslant h}$ is partitioned between A and $R(0^h, W)$ if W is a good advice string.

Now consider the following machine S that, on input $x \in \Sigma^*$ of length at least n_0, behaves as follows:

Step 1. Let $h = p_2(p_1(|x|))$. S guesses a string W of length at most $q(h)$ and asks its oracle whether $\langle 0^h, W\rangle \notin T_{good}$. If the answer is affirmative, S rejects immediately.

Step 2. Let $Z_1, ..., Z_m$ be the Z-part of W. S simulates N_1 on input x, except S replaces each query y of N_1 by the query $\langle 0^h, W, y\rangle \in D$.

Note that S is simulating N_1 and so S accepts exactly if N_1 accepts.

We observed in the above that if $\langle 0^h, W\rangle \in T_{good}$ then for every string y of length at most $p_1(|x|)$, $y \in L(N_1)$ if and only if $\langle 0^h, W, y\rangle \in D$. Thus, S correctly accepts L nondeterministically with two oracles, T_{good} and D. Since $T_{good} \in$ coNP and $D \in$ NP, we can replace the two oracles by a single oracle $E = \{0w \mid w \notin T_{good}\} \cup \{1w \mid w \in D\}$, which is in NP. It is clear that S runs in nondeterministic polynomial time. Thus, $L \in \mathrm{NP}^{\mathrm{NP}}$. Hence, A is in Low_2. This concludes the proof of Theorem 5.1. ▌

PROPOSITION 5.1. *Let $k \geqslant 2$ be an integer and $\Lambda = \langle \ell_0, ..., \ell_{k-1}, \alpha, \beta, \gamma\rangle$ be a parameter tuple for input size k. Let $H \subset \mathbb{N}^+$ be any finite set such that $\{\alpha+\beta+\gamma, \sum_{0 \leqslant i \leqslant k-1} \binom{k}{i}\ell_i\} \subseteq H$. Then every language $A \in$ NP is $\mathrm{NP}_H\mathrm{V}$-(k, Λ)-selective.*

Proof. Let k, $\Lambda = \langle \ell_0, ..., \ell_{k-1}, \alpha, \beta, \gamma\rangle$, σ, and H be as in the hypothesis. Set $\sigma = \sum_{0 \leqslant i \leqslant k-1} \binom{k}{i}\ell_i$. Let A be any language in NP. Define f to be the function defined by the procedure that works as follows on an input $X \in \Sigma^*$:

1. If either X is not a k-tuple or X is a k-tuple but some elements are equal to each other, then reject X. Nondeterministically select and execute one of Steps 2 and 3 below.

2. (i) Nondeterministically select a size $k-1$ subset Y of X.

(ii) Nondeterministically verify, for all elements y of Y, that $y \in A$. If unsuccessful, reject X.

(iii) Nondeterministically select and execute one of the following three tasks:

(I) Nondeterministically select an integer j between 1 and α, and output $\langle 0, j\rangle$.

(II) Nondeterministically select an integer j between 1 and β, and output the jth smallest string (in the lexicographic order) in $\{\langle i, j, W\rangle \mid 1 \leqslant i \leqslant k-1$ and $1 \leqslant j \leqslant \ell_i$ and $\|W\| = i$ and $W \subseteq Y\}$.

(III) Nondeterministically select an integer j between 1 and γ, and output the jth smallest string (in the lexicographic order) in $\{\langle i, j, W\rangle \mid 1 \leqslant i \leqslant k-1$ and $1 \leqslant j \leqslant \ell_i$ and $\|W\| = i$ and $W \nsubseteq Y$ and $W \subseteq X\}$.

3. (i) Nondeterministically verify, for all elements y of X, that $y \in A$. If unsuccessful, reject X.

(ii) Nondeterministically select and execute one of the following two tasks:

(I) Nondeterministically select an integer j between 1 and ℓ_0, and output $\langle 0, j\rangle$.

(II) Nondeterministically select an element u from $\{\langle i, j, W\rangle \mid 1 \leqslant i \leqslant k-1$ and $1 \leqslant j \leqslant \ell_i$ and $\|W\| = i$ and $W \subseteq X\}$ and output u.

Note that Step 3 will produce no outputs in the case when the input X has at most $k-1$ strings in A, and that the strings output in Step 2 will also be output in Step 3 when X is a k-tuple of distinct strings in A. There are exactly $\alpha+\beta+\gamma$ strings that can be output in Step 2 and σ many such in Step 3. Thus A is $\mathrm{NP}_H\mathrm{V}$-(k, Λ)-selective. ▮

From Theorem 5.1 and Proposition 5.1, we have the following corollary.

COROLLARY 5.1. *Let $k \geqslant 2$ be an integer and $\Lambda = \langle \ell_0, ..., \ell_{k-1}, \alpha, \beta, \gamma \rangle$ a parameter tuple for input size k. Let $H \subset \mathbb{N}^+$ be any finite set such that $\{\alpha+\beta+\gamma, \sum_{0 \leqslant i \leqslant k-1} \binom{k}{i} \ell_i\} \subseteq H$. Let B be a finite set of positive integers such that $Q(k, \Lambda, B)$ holds and such that $\min\{i \mid i \in B\} \leqslant \min\{i \mid i \in H\}$. Then $\mathrm{NP}_H\mathrm{V} \subseteq_{\mathrm{c}} \mathrm{NP}_B\mathrm{V}$ implies $\mathrm{PH} = \Sigma_2^p$.*

Now we turn to proving Theorems 3.2, 3.3, and 4.1 using the unified lowness theorem (Theorem 5.1).

Proof of Theorem 3.2. Let A, B, c, d, e, and δ be as in the hypothesis of the theorem. Let $k = 2$ and define a parameter tuple for size 2, $\Lambda = \langle \ell_0, \ell_1, \alpha, \beta, \gamma \rangle$ by:

- $\ell_0 = e$, $\ell_1 = d$,
- $\alpha = d$, $\beta = c-d$, and $\gamma = 0$.

Let L be an arbitrary language in NP. By Proposition 5.1, L is $\mathrm{NP}_\Lambda\mathrm{V}$-$(2, \Lambda)$-selective. Now suppose $\mathrm{NP}_\Lambda\mathrm{V} \subseteq_{\mathrm{c}} \mathrm{NP}_B\mathrm{V}$. Then $s_1 = \alpha$ and $c = \alpha+\beta+\gamma$. Since $c-\delta \leqslant \min\{i \mid i \in B\}$, $\Delta \leqslant \delta$. Also, $2d+e = s_1' + \ell_0$. Thus, $2d-(2\delta+1) \geqslant \max\{i \in B \mid i \leqslant 2d+e\}$ implies $\Delta' \geqslant 2\delta+1$. Thus, $t_1 = \Delta'$. So, we have $\sum_{1 \leqslant i \leqslant k} t_i\,(k-i) = t_1 \cdot 1 = t_1 = \Delta' > 2\delta \geqslant 2\Delta$, and so, $Q(2, \Lambda, B)$ holds. Hence, L is Low_2. Since L is an arbitrary NP language, this implies $\mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}$.

Proof of Theorem 3.3. Let k, d, A, and B be as in the hypothesis of the theorem. By Proposition 3.1, we can assume that $\min\{i \mid i \in B\} \leqslant \binom{k-1}{k-d}$; otherwise, the statement of the theorem is *false* $\Rightarrow \mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}$. Define a parameter tuple for size k, $\Lambda = \langle \ell_0, ..., \ell_{k-1}, \alpha, \beta, \gamma \rangle$ by:

- $\ell_0 = \cdots = \ell_{k-d-1} = \ell_{k-d+1} = \cdots = \ell_{k-1} = 0$, $\ell_{k-d} = 1$,
- $\alpha = \binom{k-1}{k-d}$, and $\beta = \gamma = 0$.

Let L be an arbitrary language in NP. By Proposition 5.1, L is $\mathrm{NP}_\Lambda\mathrm{V}$-$(k, \Lambda)$-selective. Now $\mathrm{NP}_\Lambda\mathrm{V} \subseteq_{\mathrm{c}} \mathrm{NP}_B\mathrm{V}$ implies $\Delta \leqslant \binom{k-1}{k-d} - 1$ and $\Delta' \geqslant \binom{k}{k-d} - \lceil \frac{k}{d} \rceil + 1$. Since $\Delta' < \binom{k}{k-d}$ and $\ell_i \neq 0$ holds only for $i = k-d$, we have that $t_{k-d} = \Delta'$ and that, for every i, $0 \leqslant i \leqslant k-1$, such that $i \neq k-d$, $t_i = 0$. So, $\sum_{0 \leqslant i \leqslant k-i} t_i\,(k-i) = k\Delta' \geqslant d(\binom{k}{k-d} - \lceil \frac{k}{d} \rceil + 1) > d(\binom{k}{k-d} - \frac{k}{d}) = k(\binom{k-1}{k-d} - 1) = k\Delta$. This implies that $Q(k, \Lambda, B)$ holds, so by Theorem 5.1 L is in Low_2. Since L is an arbitrary NP set, this implies $\mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}$.

Proof of Theorem 4.1. Let $k \geqslant 2$. Define a parameter tuple for size k, $\Lambda = \langle \ell_0, ..., \ell_{k-1}, \alpha, \beta, \gamma \rangle$ by:

- $\ell_0 = \cdots = \ell_{k-2} = 0$, $\ell_{k-1} = 1$,
- $\alpha = \binom{k-1}{k-1} = 1$, and $\beta = \gamma = 0$.

Let $B = \{1, ..., k-1\}$. The family of all $\mathrm{NP}_B\mathrm{V}$-k-selective sets is precisely that of all $\mathrm{NP}_B\mathrm{V}$-(k, Δ)-selective sets. Since $\alpha + \beta + \gamma = 1$, Δ is necessarily 0. On the other hand, the hypothesis of the theorem implies $\Delta' > 0$. So, Q holds. The rest of the proof is the same.

Here we give another example of the use of Theorem 5.1.

Corollary 5.2. *Let $k \geqslant 4$. Let $A, B \subseteq \mathbb{N}^+$ be nonempty sets such that $\{3k, 2k^2 - k\} \subseteq A$, $\min\{i \in B \mid i \leqslant 2k^2 - k\} = 2k$ and $\max\{i \in B \mid i \leqslant 2k^2 - k\} \leqslant k^2 - k - 1$. Then $\mathrm{NP}_A\mathrm{V} \subseteq_c \mathrm{NP}_B\mathrm{V}$ implies $\mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}$.*

Proof. Let $k \geqslant 4$. Define $\ell_0 = 2$, $\ell_1 = \cdots = \ell_{k-3} = 0$, $\ell_{k-2} = 2$, $\ell_{k-1} = k$, $\alpha = 3k-2$, $\beta = 0$, and $\gamma = 2$. Then $s_1 = k\binom{k-1}{k-1} + 2\binom{k-1}{k-2} = 3k-2$ and $s'_1 = k\binom{k}{k-1} + 2\binom{k}{k-2} = 2k^2 - k$. Also, $\Delta = (3k-2) + 0 + 2 - 2k = k$ and $\Delta' \geqslant (2k^2 - k) - (k^2 - k - 1) = k^2 + 1$. Then $t_{k-1} = k^2$ and $t_{k-2} \geqslant 1$. Thus, $\sum_{0 \leqslant i \leqslant k-1} (k-i)\, t_i \geqslant k^2 + 2 > k\Delta$. This implies that $Q(k, \Delta, B)$ holds. The rest of the proof is the same as before. ▮

Although Theorem 5.1 is relatively broad, there exist cases to which the theorem does not speak. For example, let $k \geqslant 2$ and d, $1 \leqslant d \leqslant k-1$, be integers. Let $A, B \subseteq \mathbb{N}^+$ be nonempty sets such that $\{k, 2k\} \subseteq A$, $\max\{i \in B \mid i \leqslant 2k\} \geqslant 2k - 2d$, and $\min\{i \in B \mid i \leqslant 2k\} \leqslant k - d$. Theorem 5.1 does not speak to, for this case, the issue of whether $\mathrm{NP}_A\mathrm{V} \subseteq_c \mathrm{NP}_B\mathrm{V}$.

6. CONCLUSION

In this paper we gave a condition that we conjecture is, assuming that the polynomial hierarchy does not collapse, necessary and sufficient for determining for finite cardinality types A and B whether $\mathrm{NP}_A\mathrm{V} \subseteq_c \mathrm{NP}_B\mathrm{V}$, i.e., informally, for determining the ways in which solution cardinality can be pruned. We proved our condition to be (unconditionally) sufficient. We also established a necessary condition assuming that the polynomial hierarchy does not collapse. However, our necessary-condition theorem is not yet strong enough to match our sufficient condition.

Nonetheless, we hope that in time this will be established and we recommend as an interesting open problem the issue of proving the Narrowing-Gap Conjecture. Certainly, there are many similar cases in the literature where similar complete characterizations have been completed or interestingly posed. Hemaspaandra *et al.* [12] have under the assumption that the polynomial hierarchy does not collapse, completely characterized (for pairs of levels of the boolean hierarchy) when "query order" (see [10, 12], see also [11, 28]) matters. Kosub and Wagner [19] have posed, and made powerful progress toward, a complete characterization regarding their boolean hierarchy of NP-partitions. Also, Kosub [18] has proven that for each cardinality-type pair A and B violating the Narrowing-Gap Conjecture, there is an oracle $W(A, B)$ such that $\mathrm{NP}_A\mathrm{V}^{W(A,B)} \not\subseteq_c \mathrm{NP}_B\mathrm{V}^{W(A,B)}$. Finally, we commend to the reader's attention the work—which in contrast to the work of this paper is about accepting-path counts rather than solution cardinalities—of Ogiwara and Hemachandra [22] and Durand *et al.* [8].

ACKNOWLEDGMENTS

We are grateful to the anonymous referees for their helpful comments, and we thank A. Durand, M. Hermann, P. Kolaitis, S. Kosub, and K. Wagner for many enjoyable conversations and emails.

REFERENCES

1. R. Book, T. Long, and A. Selman, Quantitative relativizations of complexity classes, *SIAM J. Comput.* **13** (1984), 461–487.
2. R. Book, T. Long, and A. Selman, Qualitative relativizations of complexity classes. *J. Comput. System Sci.* **30** (1985), 395–413.
3. D. Bovet, P. Crescenzi, and R. Silvestri, A uniform approach to define complexity classes, *Theoret. Comput. Sci.* **104** (1992), 263–283.
4. J. Cai, $S_2^p \subseteq \mathrm{ZPP}^{\mathrm{NP}}$, *in* "Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science," pp. 620–628, IEEE Comput. Soc. Press, Los Alamitos, CA, 2001.
5. J. Cai, V. Chakaravarthy, L. Hemaspaandra, and M. Ogihara, "Some Karp–Lipton Type Theorems Based on S_2," Technical Report TR759, Department of Computer Science, University of Rochester, Rochester, NY, September 2001.
6. J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung, The boolean hierarchy I: Structural properties, *SIAM J. Comput.* **17** (1988), 1232–1252.
7. R. Canetti, More on BPP and the polynomial-time hierarchy, *Inform. Process. Lett.* **57** (1996), 237–241.
8. A. Durand, M. Hermann, and P. Kolaitis, Subtractive reductions and complete problems for counting complexity classes, *in* "Proceedings of the 25th International Symposium on Mathematical Foundations of Computer Science," Lecture Notes in Computer Science, Vol. 1893, pp. 323–332, Springer-Verlag, Berlin/New York, August/September 2000.
9. S. Fenner, L. Fortnow, A. Naik, and J. Rogers, Inverting onto functions, *in* "Proceedings of the 11th Annual IEEE Conference on Computational Complexity," pp. 213–222, IEEE Comput. Soc. Press, Los Alamitos, CA, May 1996.
10. E. Hemaspaandra, L. Hemaspaandra, and H. Hempel, An introduction to query order, *Bull. EATCS* **63** (1997), 93–107.
11. E. Hemaspaandra, L. Hemaspaandra, and H. Hempel, Query order in the polynomial hierarchy, *J. Univ. Comput. Sci.* **4** (1998), 574–588.
12. L. Hemaspaandra, H. Hempel, and G. Wechsung, Query order, *SIAM J. Comput.* **28** (1999), 637–651.
13. L. Hemaspaandra, A. Naik, M. Ogihara, and A. Selman, Computing solutions uniquely collapses the polynomial hierarchy, *SIAM J. Comput.* **25** (1996), 697–708.
14. L. Hemaspaandra, J. Rothe, and G. Wechsung, Easy sets and hard certificate schemes, *Acta Inform.* **34** (1997), 859–879.
15. B. Jenner and J. Torán, The complexity of obtaining solutions for problems in NP, *in* "Complexity Theory Retrospective II" (L. Hemaspaandra and A. Selman, Eds.), Springer-Verlag, Berlin/New York, 1997.
16. K. Ko, On self-reducibility and weak P-selectivity, *J. Comput. System Sci.* **26** (1983), 209–221.
17. J. Köbler and O. Watanabe, New collapse consequences of NP having small circuits, *SIAM J. Comput.* **28** (1998), 311–324.
18. S. Kosub, On NP-partitions over posets with an application to reducing the set of solutions of NP problems, *in* "Proceedings of the 25th International Symposium on Mathematical Foundations of Computer Science," Lecture Notes in Computer Science, Vol. 1893, pp. 467–476, Springer-Verlag, Berlin/New York, August/September 2000.

19. S. Kosub and K. Wagner, The boolean hierarchy of NP-partitions, *in* "Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science," Lecture Notes in Computer Science, Vol. 1770, pp. 157–168, Springer-Verlag, Berlin/New York, February 2000.

20. A. Naik, J. Rogers, J. Royer, and A. Selman, A hierarchy based on output multiplicity, *Theoret. Comput. Sci.* **207** (1998), 131–157.

21. M. Ogihara, Functions computable with limited access to NP, *Inform. Process. Lett.* **58** (1996), 35–38.

22. M. Ogiwara and L. Hemachandra, A complexity theory for feasible closure properties, *J. Comput. System Sci.* **46** (1993), 295–325.

23. A. Russell and R. Sundaram, Symmetric alternation captures BPP, *Computational Complexity* **7** (1998), 152–162.

24. U. Schöning, A low and a high hierarchy within NP, *Journal of Computer and System Sciences* **27** (1983), 14–28.

25. A. Selman, A taxonomy of complexity classes of functions, *J. Comput. System Sci.* **48** (1994), 357–381.

26. A. Selman, Much ado about functions, *in* "Proceedings of the 11th Annual IEEE Conference on Computational Complexity," pp. 198–212, IEEE Comput. Soc. Press, Los Alamitos, CA, May 1996.

27. N. Vereshchagin, Relativizable and nonrelativizable theorems in the polynomial theory of algorithms, *Russian Acad. Sci. Izv. Math.* **42** (1994), 261–298.

28. K. Wagner, A note on parallel queries and the symmetric-difference hierarchy, *Inform. Process. Lett.* **66** (1998), 13–20.