On Checking Versus Evaluation of Multiple Queries[1]

William I. Gasarch[2] Lane A. Hemachandra[3] Albrecht Hoene[4]

January 10, 1990

[1]This work was done in part while Hemachandra and Hoene visited Gerd Wechsung in Jena.

[2]Institute for Advanced Computer Studies, Department of Computer Science, University of Maryland, College Park, MD 20742. Research supported in part by the National Science Foundation under grant CCR-8803641.

[3]University of Rochester, Department of Computer Science, Rochester, NY 14627. Research supported in part by the National Science Foundation under grant CCR-8809174/CCR-8996198 and a Presidential Young Investigator Award.

[4]Technische Universität Berlin, Fachbereich Informatik, Franklinstr. 28-29, FR 6-2, D-1000 Berlin 10.

Abstract

The distinction between computing answers and checking answers is fundamental to computational complexity theory, and is reflected in the relationship of NP to P. The plausibility of *computing* the answers to many membership queries to a hard set with few queries is the subject of the theory of terseness. In this paper, we develop companion theories—both complexity-theoretic and recursion-theoretic—of *characteristic vector terseness*, which ask whether the answers to many membership queries to a hard set can be *checked* with fewer queries.

1 Introduction

The distinction between checking and evaluating is fundamental in computational complexity theory. In its most popular version it emerges as the P = NP question, but it also reappears in other problems of the field [Val76,HIS85]. We introduce this distinction between checking and evaluating to the theory of terseness.

The theory of terseness studies whether it is possible to compute the answers to multiple queries to an oracle by fewer queries. Pioneering work in this area was done by Barzdin' [Bar68]—and, more recently, by Beigel, Gasarch, Gill, and Owings [BGGO87]—for a recursion-theoretic context, and by Amir, Beigel, and Gasarch [Bei87,AG88] for a polynomial-time framework. Their approach to certain structural properties of hard sets studies the complexity of *evaluating* the values of queries—that is, of *computing* the answers to the queries.

In this paper we develop a theory of characteristic vector terseness, the 'checking' counterpart of the theory of terseness: Given a vector of queries to an oracle and answers for them, we want to know if all the answers are correct. That is, we wish to determine whether the vector of answers is the characteristic vector of the queries relative to the considered language. The minimal number of queries that suffice for this purpose is called the *characteristic vector cost* of the language considered.

In Section 2, we study the problem for nonrecursive sets. In this context the appropriate question for characteristic vector cost is: What is the minimum number j such that the answers to n queries to A can be checked with j queries to A? We show that from a result of [Rog67] it follows that the characteristic vector cost of K is exactly two. This strengthens the result of ([BGGO87], see also [Bar68]) that *computing* the solutions to two queries to K requires two queries to K. Other results on the characteristic vector cost of nonrecursive languages are also proven.

In the resource-bounded world of feasible computations, Section 3 studies the analogous question for polynomial-time oracle machines, and finds that the characteristic vector cost of a language is closely related to its basic set-theoretic properties. In this context we permit the use of any oracle that belongs to the same class as the language we are interested in. Our results show a connection between characteristic vector costs and Boolean hierarchies [Hau14,Wec85,CGH$^+$88]. For reasonably well-behaved classes C, the characteristic vectors of C can be checked with one query if and only if the Boolean hierarchy over C collapses in a certain way.

We next consider the characteristic vector cost of NP. If characteristic vectors of NP languages are recognizable *uniformly* (that is, via a single polynomial-time oracle machine) with one query to a language of the same class, then we can derive a strong conclusion: all languages polynomial-time truth-table reducible to a language L of the considered class are in fact 1-truth-table reducible to L. Finally, we investigate some specific classes within the Boolean hierarchy over NP and show how their characteristic vector costs relate to the existence of a proper Boolean hierarchy over NP.

Before presenting our results, let us introduce some notation. $\langle \cdot, \cdot, \ldots, \cdot \rangle$ applied to k arguments ($k \geq 2$) denotes the polynomial-time computable k-ary pairing function from $N^k \to N$ used in [Rog67]. We will make use of this functions for strings as well by identifying $i \in N$ with $bin(i)$, which stands for the i-th binary string in lexicographical order. For any language $A \subseteq \Sigma^*$, we denote by $\overline{A}$ its complement and by A^k the set $\{\langle x_1, \ldots, x_k \rangle | \, x_i \in A \text{ for } 1 \leq i \leq k\}$. For sets $A_i \subseteq \Sigma^*$ the k-fold disjoint union $A_1 \oplus A_2 \oplus \ldots A_k$ is defined by $\cup_{i=1}^{k} \{\langle bin(i), x \rangle | \, x \in A_i\}$.

P (NP) denotes the class of languages that can be accepted by deterministic (nondeterministic) polynomial-time Turing machines [HU79]. P^C and NP^C refer to the analogous classes relative to an oracle from the class C [BGS75]. If during such a computation only a limited number of oracles queries is permitted (say k) we indicate that by the additional superscript $[k]$, for example, $\mathrm{P}^{\mathrm{NP}[k]}$ [PZ83]. In general, the classes dealt with are either well-known or will be defined explicitly as needed.

A language L is said to be *k-pterse* [AG88] if a polynomial-time oracle machine cannot in general *compute* the correct answers to an input set of k queries with less than k queries to the oracle L. If L is k-pterse for each $k \geq 2$ then it is *pterse* [AG88].

We now propose and study a "checking answers" analog of the [AG88,Bar68,BGGO87] theory of terseness.

2 Recursion-Theoretic Results on Characteristic Vectors

We study the number of queries required to verify whether a given string is a characteristic vector of a nonrecursive set.

Definition 2.1 If $A \subseteq \Sigma^*$ and $x_1, \ldots, x_i \in \Sigma^*$ then let

$$F_i^A(x_1, \ldots, x_i) = \langle \chi_A(x_1), \ldots, \chi_A(x_i) \rangle,$$

$$V_i^A = \{\langle x_1, \ldots, x_i, b_1, \ldots, b_i \rangle \mid (\forall j)[\chi_A(x_j) = b_j]\},$$

and

$$V_\omega^A = \bigcup_{i=1}^{\infty} \{\langle i, v\rangle \mid v \in V_i^A\}.$$

where χ_A is the characteristic function of A.

Definition 2.2 [BGGO87] Let $i \in N$ and let A be a set. Then $Q(i, A)$ ($Q_{||}(i, A)$) is the collection of all sets B such that $B \leq_T A$ and this Turing-reduction can be performed by a machine that makes at most i serial (parallel, i.e., truth-table) queries to A.

The following notion is new to this paper.

Definition 2.3 A set A is *vterse* if for all $i \geq 1$, $V_i^A \notin Q(i-1, A)$.

We are interested in how many queries to A are required to decide the set V_i^A (and V_ω^A). Naively it appears that for sets A that are nonrecursive, i queries might be required.

We first examine r.e. sets. How many queries are required to compute V_i^A if queries to an r.e. set other than A are allowed?

Theorem 2.4 If A is r.e. and $i \in N$, then $V_i^A \in Q_{||}(2, K)$.

Proof: Given $\langle x_1, \ldots, x_i, b_1, \ldots, b_i\rangle$ create machines M_{y_1} and M_{y_2} such that M_{y_1} enumerates A and stops only if all the elements of $\{x_j \mid b_j = 1\}$ appear, and M_{y_2} enumerates A and stops only if some element of $\{x_j \mid b_j = 0\}$ appears. It is easy to see that $\langle x_1, \ldots, x_i, b_1, \ldots, b_i\rangle \in V_i^A$ if and only if $(y_1 \in K) \wedge (y_2 \notin K)$. ∎

The above proof in fact shows that V_i^A is 2-r.e. (see [EHK81] for a definition of 2-r.e.). Theorem 2.4 is optimal for $A = K$ in that the set V_2^K requires two queries to K. The following proof of Theorem 2.5, pointed out to us by Richard Chang, is more direct than that found in earlier versions of this paper.

Theorem 2.5 $V_2^K \notin Q(1, K)$.

Proof: It is known (see [Rog67]) that the m-degree of $K \times \overline{K}$ contains the m-degree of $K \oplus \overline{K}$ properly. By the fact that $K \times \overline{K}$ many-one reduces to V_2^K, it follows that every language in the m-degree of $K \times \overline{K}$ many-one reduces to V_2^K. Since every language in $Q(1, K)$ many-one reduces to $K \oplus \overline{K}$ it follows that $V_2^K \notin Q(1, K)$.

∎

We exhibit some sets that are vterse.

Theorem 2.6 If A is 1-generic (see [Joc79]) then A is vterse.

Proof: In [BGGO87] it is shown that if A is 1-generic then for all i $A^{i+1} \notin Q(i, A)$; hence for all i it holds that $V_{i+1}^A \notin Q(i, A)$. ∎

Jockusch [Joc79] showed that the class of 1-generic sets is comeager. Hence:

Corollary 2.7 The class of vterse sets is comeager.

The following theorem shows that not all sets are vterse.

Theorem 2.8 Every truth-table degree (see [Rog67]) contains a set A such that $V_\omega^A \in Q(1, A)$.

Proof: Let D be any set. We construct $A \equiv_{tt} D$ such that $V_\omega^A \in Q(1, A)$. Let $A_0 = D$. For $i \geq 0$ let $A_{i+1} = \{\langle x_1, b_1, j_1, \ldots, x_z, b_z, j_z, z, i+1\rangle \mid z \geq 1$ and $(\forall k)[1 \leq k \leq z \Rightarrow [(0 \leq j_k \leq i)$ and $(F_1^{A_{j_k}}(x_k) = b_k)]]$. Set $A = \cup_{i=1}^\infty A_i$. It is easy to set that $A \equiv_{tt} D$ and that

$$\langle x_1, \ldots, x_i, b_1, \ldots, b_i\rangle \in V_\omega^A \text{ iff } \langle x_1, \ldots, x_i, b_1, \ldots, b_i, i, q+1\rangle \in A,$$

where q is one greater than the maximum value of the last component of any x_i (since we are adopting the recursively defined pairing functions of Rogers, this can be written $q = 1 + \max_{1 \leq k \leq i}\{q_k \mid x_k = \langle r_k, q_k\rangle\}$). ∎

We now examine vterseness for r.e. sets.

Theorem 2.9 Every r.e. Turing degree contains r.e. sets A and B such that $V_\omega^A \in Q_{||}(2, A)$ and B is vterse.

Proof: Every r.e. Turing degree contains an r.e. set that is semirecursive, i.e., a lower cut of a recursive linear ordering [Joc68]. Let A be that set, and let $<$ denote the recursive linear ordering. We show that $V_\omega^A \in Q_{||}(2, A)$.

Given $\langle x_1, \ldots, x_i, b_1, \ldots, b_i\rangle$, rename the arguments such that $\hat{x}_1 < \hat{x}_2 < \cdots < \hat{x}_i$, and rename the b_i's via the same permutation. If there is an i such that $\hat{b}_i = 0$ and $\hat{b}_{i+1} = 1$, then the given characteristic vector is not in V_ω^A. If no such i exists, then there exists a j such that $\hat{b}_1 = \hat{b}_2 = \cdots = \hat{b}_j = 1$ and $\hat{b}_{j+1} = \cdots = \hat{b}_i = 0$. It is easy to see that the given characteristic vector is in V_ω^A if and only if $(\hat{x}_j \in A) \wedge (\hat{x}_{j+1} \notin A)$.

In [BGGO87] it was shown that every r.e. Turing degree contains an r.e. set B such that for all i the set $\text{PARITY}_{i+1}^B = \{\langle x_1, \ldots, x_{i+1}\rangle : |B \cap \{x_1, \ldots, x_{i+1}\}|$ is even$\}$ is not in

$Q(i,B)$. A careful examination of the proof reveals that $\overline{B}^{i+1}$ is not in $Q(i,B)$. Hence for all i, $V^B_{i+1} \notin Q(i,B)$, so B is vterse. ∎

By way of contrast, it follows easily from Theorem 2.5 that some m-degrees—e.g., that of the r.e. complete sets—contain no sets A for which $V^A_\omega \in Q(1,A)$. Though it follows from Theorem 2.8 that every r.e. Turing degree contains a set A such that $V^A_\omega \in Q(1,A)$, it remains an open question whether every r.e. Turing degree contains a *recursively enumerable* set A such that $V^A_\omega \in Q(1,A)$.

3 Characteristic Vector Complexity and Boolean Hierarchies

To what extent do the results of the last section hold in a resource-bounded framework? The last section's questions about characteristic vector complexity become, in a time-bounded world: Given a vector of queries and a vector of purported answers to the queries, how many queries does a polynomial-time oracle machine need to check whether all the answers are correct? We are primarily interested in classes that lie above P and are reasonably well-behaved.

Definition 3.1 We call a class C *interesting* if:

1. $\mathrm{P} \subseteq C$,
2. C is closed under disjoint union, i.e., if $L_1, \ldots L_k \in C$ then $L_1 \oplus L_2 \oplus \ldots \oplus L_k \in C$,
3. C is closed under trivial pairing, i.e., for any language $L \in C$ the sets $\{\langle x, y\rangle \mid x \in L, y \in \Sigma^*\}$ and $\{\langle x, y\rangle \mid x \in \Sigma^*, y \in L\}$ are in C, and
4. C is closed downwards under $\leq^p_m$ reductions, i.e., $L' \in C$ and $L \leq^p_m L'$ implies $L \in C$.

Furthermore, we are interested mainly in the use of oracles of approximately the same power as the considered language. Hence we will define characteristic vector cost for classes, rather than for sets.

Definition 3.2 Let C be any class of sets.

1. The *characteristic vector cost* of a class C is:
$$CV_C(k) = \min_{i \geq 0} \{\, i : \text{for all } L \in C,\, V^L_k \in \mathrm{P}^{C[i]} \,\}.$$

2. $CV_C^+(k) = \min_{i \geq 0} \{ i$: for all $L \in C$, $L^k \in \mathrm{P}^{C[i]} \}$.

3. $CV_C^-(k) = \min_{i \geq 0} \{ i$: for all $L \in C$, $\overline{L}^k \in \mathrm{P}^{C[i]} \}$.

4. $CV_C = \min_{i \geq 0} \{ i$: for all $L \in C$, $V_\omega^L \in \mathrm{P}^{C[i]} \}$.

We observe that the behavior of a class under basic set-theoretic operations is closely related to its characteristic vector cost.

Proposition 3.3 Let C be an interesting class other than P. If C is closed under:

1. union (intersection) then for all k: $CV_C^-(k) = 1$ ($CV_C^+(k) = 1$).

2. complement then for all k: $CV_C^+(k) = CV_C^-(k) = CV_C(k)$.

3. union and complement (equivalently, intersection and complement) then for all k: $CV_C(k) = 1$.

4. union and intersection then for all k: $CV_C(k) \leq 2$.

Proof:
1. By the assumption that C is different from P, we know that $CV_C^+(k)$ and $CV_C^-(k)$ are greater than 0 if $k \geq 1$. Let L be a language in C. Since C is closed under trivial pairing the sets, $L' = \{\langle x, y\rangle \mid x \in L, y \in \Sigma^* \}$ and $L'' = \{\langle x, y\rangle \mid x \in \Sigma^*, y \in L \}$ are in C. By the assumption that C is closed under union the set $L' \cup L'' = \{\langle x, y\rangle \mid x \in L$ or $y \in L\}$ belongs to C. On an input $\langle x, y\rangle$ an oracle machine can check with one query to this oracle if both x and y do not belong to L. This is easily seen to work for arbitrary k. For intersection the proof is analogous.
2. It suffices to show that $CV_C(k) \leq CV_C^+(k)$. Let $L \in C$, C closed under complement. Clearly $V_k^L \leq_m^p (L \oplus \overline{L})^k \in \mathrm{P}^{C[CV_C^+(k)]}$. Thus $CV_C(k) \leq CV_C^+(k)$.
3. and 4. are immediate by 1. and 2. ∎

The proposition indicates that the characteristic vector costs of a class reflect its closure properties under the corresponding operations.

Definition 3.4 We will call an interesting class C:

1. *pseudo-closed under intersection* if for all $k \geq 1 : CV_C^+(k) = 1$,

2. *pseudo-closed under union* if for all $k \geq 1 : CV_C^-(k) = 1$, and

3. *pseudo-closed under complement* if for all $k \geq 1 : CV_C^+(k) = CV_C^-(k) = CV_C(k)$.

Definition 3.5 A class C has *wee* characteristic vector cost if, for all $k \geq 1$, $CV_C(k) = 1$.

A justification of these names is given by Theorem 3.9, which shows that pseudo-closure and closure under corresponding set-theoretic operations are closely related. For example, it holds that classes that are pseudo-closed under union and complement have no infinite Boolean hierarchies.

From Proposition 3.3 it follows that among the classes with wee characteristic vector cost are all interesting deterministic classes that are defined by time or space bounds, parity polynomial time ($\oplus$P, defined in [PZ83,GP86]), ZPP [Gil77], the Δ_k^p-classes of the polynomial time hierarchy, and so on, because all of them are closed under complement and union.

Also by Proposition 3.3 the characteristic vector cost is not higher than two for classes such as NP, the Σ_k^p and Π_k^p levels of the polynomial time hierarchy [Sto77], and FewP ([All86,AR88], see also [Rub88,CH]), since they are closed under union and intersection. Among the classes for which the number of queries can not be reduced straightforwardly are US (which tests for unique solutions, [BG82], see also [GW87]) and the classes of the Boolean hierarchy over NP [Wec85,CGH$^+$88,CGH$^+$89]. Their vector checking cost will be studied in Theorem 3.14 and its corollaries.

We now turn to Boolean hierarchies over complexity classes. Among them, the Boolean hierarchy over NP has received the most attention. However, Boolean hierarchies can be defined over arbitrary classes [Hau14,Wag88,HH88,BBJ$^+$].

Definition 3.6 [Hau14] For a class C, the Boolean hierarchy over C is the collection of classes defined by:

$\mathrm{BH}_C(1) = C$.

$\mathrm{BH}_C(k) = \{ L_1 - L_2 \mid L_1 \in C, L_2 \in \mathrm{BH}_C(k-1)\}$ for $k > 1$.

$\mathrm{BH}_C = \bigcup_k \mathrm{BH}_C(k)$.

We use the the following observation.

Proposition 3.7 ([BBJ$^+$,Bei88a]) Let C be any class of sets. For all $k \geq 1 : \mathrm{P}^{C[k]} \subseteq \mathrm{BH}_C$.

We observe that if there exists any $k > 1$ such that one can recognize characteristic vectors of length k with one query then one can do the same for every length.

Proposition 3.8 Let C be an interesting class. If there is a $k > 1$ with $CV_C(k) = 1$, then for all $i \geq 1$ it holds that $CV_C(i) = 1$.

Proof: Suppose $CV_C(j) = 1$. Let M be the machine that checks characteristic vectors of length j for a language $L \in C$ with one query to an oracle $L' \in C$. Now vectors $(x_1, \ldots, x_{j+1}, b_1, \ldots, b_{j+1})$ of L can be checked the following way: Run M on the vector $v = (x_1, \ldots, x_j, b_1, \ldots, b_j)$ pursuing both outcoming paths of the query but without actually asking the query to the oracle. If both outcomes of the query $q(v)$ are different (otherwise the case is trivial), we know that v is a characteristic vector if and only if M accepts on the 'yes' path and $q(v) \in L'$ or M accepts on the 'no' path and $q(v) \notin L'$. Since C is closed under disjoint union there is a machine M' checking vectors of $L \oplus L'$ with one query to a language $L'' \in C$. We run M' on $(1q(v), 0x_{j+1}, 1, b_{j+1})$ if M accepts v on the 'yes' path, and on $(1q(v), 0x_{j+1}, 0, b_{j+1})$ if it accepts v on the 'no' path. Thus one query to L'' suffices to check vectors of length $j + 1$, and the claim follows. ∎

The following theorem explores the consequences for these Boolean hierarchies of the pseudo-closures of their basic classes.

Theorem 3.9 Let C be an interesting class that contains P properly:
There is a $k > 1$ with $CV_C(k) = 1$ if and only if $\mathrm{BH}_C = \mathrm{P}^{C[1]}$.

Proof:
If $\mathrm{BH}_C = \mathrm{P}^{C[1]}$ then by Proposition 3.7 we get $V_k^L \in P^{C[1]}$, which means that we can check characteristic vectors of length k with one query to a language in C.
Now assume the left-hand side of the equivalence holds.
'$\supseteq$': holds independently of the assumption by Proposition 3.7.
'$\subseteq$': Fix $S \in \mathrm{BH}_C$. We will show that $S \in \mathrm{P}^{C[1]}$. By definition there is a k such that $S = S_1 - (S_2 - (\cdots S_{k-1} - S_k)\cdots)$, with $S_i \in C$ for all $i \leq k$. This gives rise to a Boolean formula with

$$x \in S \iff S_1(x) \wedge \neg(S_2(x) \wedge \neg(\cdots \neg(S_{k-1}(x) \wedge \neg S_k(x))\cdots)),$$

where $S_i(x)$ stands for $x \in S_i$. The conjunctive normal form of this formula is of the form $cl_1 \wedge cl_2 \wedge \cdots \wedge cl_j$, where j and the number of literals in each clause cl_i are constant, i.e., they depend only on S but not on the individual input string x. Thus it is clear that there is a machine that evaluates each clause $cl_i = (S_{i_1}(x), \ldots, S_{i_s}(x), \neg S_{i_{s+1}}(x), \ldots, \neg S_{i_r}(x))$ with one query to a C language: Since C is interesting $S_1 \oplus \cdots \oplus S_k \in C$ and by Proposition

3.8 there is machine M that checks $S_1 \oplus \cdots \oplus S_k$-vectors of the maximal length of all clauses cl_i with one query to a language $L' \in C$. We run M on the vector $v_i = (\langle bin(i_1), S_{i_1}(x)\rangle, \ldots, \langle bin(i_r), S_{i_r}(x)\rangle, 0, \ldots, 0, 1, \ldots, 1)$ with 0 repeated s times and 1 repeated $r - s$ times—exploiting the fact that v_i is a characteristic vector of $S_1 \oplus \cdots \oplus S_k$ if and only if the value of the clause cl_i is false.

Without loss of generality we can assume that for any vector v_i this machine queries $q(v_i)$, accepts on one outcome of the query, and rejects on the other one. Thus the value of cl_i is true if and only if either M accepts on the 'yes' path and $q(v_i) \in L'$, or M accepts on the 'no' path and $q(v_i) \notin L'$. In polynomial-time we can precompute which queries M asks for the different clauses, and which of the 'yes' or on the 'no' paths it accepts on.
This gives rise to the following procedure:
On input x with $F(x) = cl_1 \wedge cl_2 \wedge \cdots \wedge cl_j$ being the corresponding CNF formula (which is fixed for each S).

- For each clause cl_i compute the vector v_i and the query $q(v_i)$ the machine M asks. If M accepts v_i on the 'yes' path then $b_i := 0$ else $b_i := 1$.
- Check the vector $(q(v_1), q(v_2), \ldots, q(v_j), b_1, \ldots, b_j)$.

By assumption the last part is possible by running a polynomial-time oracle machine, with an oracle $L \in C$ that is queried only once. From the construction it follows that $x \in S$ if and only if $F(x) = cl_1 \wedge cl_2 \wedge \cdots \wedge cl_j$ is true if and only if for all $i \leq j$ it holds that $\chi_L(q(v_i)) = b_i$.

∎

It follows that for classes with wee characteristic vector cost it is possible for deterministic oracle machines to query the oracle only once, instead of a constant number of times, without loss of power.

Corollary 3.10 Let C be an interesting class.

1. There is a $k > 1$ with $CV_C(k) = 1$ if and only if $\mathrm{P}^{C[i]} = \mathrm{P}^{C[1]}$ for all $i \geq 1$.
2. There is a $k > 1$ with $CV_C(k) = 1$ if and only if every language that is polynomial-time truth-table reducible to a set in C is indeed polynomial-time one-truth-table reducible to a set in C.

By a result of Papadimitriou and Zachos ([PZ83], see also [KSW87,AG88]), for each i, $\mathrm{P}^{\mathrm{NP}[i]}$ is contained in the Boolean hierarchy. This also holds for arbitrary interesting

classes (Proposition 3.7), and the same applies for bounded-truth-table degrees of interesting classes. Thus Corollary 3.10 follows from Theorem 3.9.

Corollary 3.10 contrasts with a result of Chang and Kadin ([CK89], correcting [Kad88]) and Beigel [Bei88b]: They gave a criterion that is equivalent to the collapse of the Boolean hierarchy of a class to its closure under polynomial many-one reductions. For $C = \mathrm{NP}$, Corollary 3.13 will provide a criterion that is equivalent to the coincidence of the closure of NP under polynomial-time truth-table reductions with its closure under one-truth-table reductions.

Theorem 3.9 immediately yields many results about well-known interesting complexity classes (in the technical sense defined previously). We will illustrate and extend them in the case of NP and some classes that are in its Boolean hierarchy. The Boolean hierarchy over NP arose from the study of sets that are the difference of two NP languages [PZ83, PY84], and has been investigated in detail by [Wec85], [CGH+88], and [CGH+89]. For NP itself we obtain Corollary 3.11, which follows from the fact the NP is closed under union and intersection Theorem 3.9, and the result of Chang and Kadin result that a collapse of the Boolean hierarchy implies one of the polynomial-time hierarchy [CK89].

Corollary 3.11

1. For all $k \geq 1$, $CV_{\mathrm{NP}}(k) \leq 2$.
2. There is a $k > 1$ such that $CV_{\mathrm{NP}}(k) = 1$ if and only if $\mathrm{BH}_{\mathrm{NP}} = \mathrm{P}^{\mathrm{NP}[1]}$.
3. If there is a $k > 1$ such that $CV_{\mathrm{NP}}(k) = 1$, then the polynomial-time hierarchy collapses.

Up to here we studied the consequences of being able to recognize characteristic vectors of a fixed constant length with one query to an oracle of the same class. The following theorem explores the consequences of the stronger assumption that characteristic vectors of variable length are uniformly checkable with one query.

Theorem 3.12 $CV_{\mathrm{NP}} = 1$ if and only if $\mathrm{P}^{\mathrm{NP}[\log n]} = \mathrm{P}^{\mathrm{NP}[1]}$.

Proof: The idea of the proof (left to right, the other direction is immediate by the observation that $V_{\omega}^{SAT} \in \mathrm{P}^{SAT[2]}$) is similar to that of Theorem 3.9: Run the machine M, which queries its oracle $L \in \mathrm{NP}$ at most $\log n$ times, along every possible path arising from all answers that might be given by the oracle (without querying the oracle). Each accepting

path corresponds to a candidate for a characteristic vector of at most log n components. M accepts the input if and only if one of them is a characteristic vector. By assumption we can recognize these vectors with a machine M' with one query to a C language.

Now we run M' on each of those vectors without querying the oracle. M' asks a query q for such a vector, and accepts it on the 'yes' path and rejects on the 'no' path or vice versa (otherwise the case is trivial). The vector is a characteristic vector if and only if q is in the oracle and the 'yes' path of M' accepts or q is not in the oracle if the 'no' path accepts. Thus by checking the outcomes of M' on the different input vectors we obtain a vector of new queries and new answers, knowing that M accepts if and only if at least one of the answers is correct. Thus it follows that M accepts if and only if the vector consisting of the same queries, and having exchanged 'yes' and 'no' answers is not a characteristic vector, which again by assumption we can check with one query. ∎

The fact that the class of languages polynomial-time truth-table reducible to an NP language equals $\mathrm{P}^{\mathrm{NP}[\log n]}$ [Hem89,Wag87] yields the following corollary, which is interesting when compared with Corollary 3.10, part 2.

Corollary 3.13 $CV_{\mathrm{NP}} = 1$ if and only if all languages that are polynomial-time truth-table reducible to an NP language are indeed polynomial-time one-truth-table reducible to an NP language.

Now let's turn our attention to some classes that are located within the Boolean hierarchy over NP. Its levels are neither known nor believed to be closed under complement, intersection, or union. In fact, by a result of Chang and Kadin, [CK89], for any $k \geq 3$ the closure of $\mathrm{BH}_{NP}(k)$ under one of those operations implies its collapse on the same level.

What are the consequences if vectors of languages of some level there can be checked with one query?

Theorem 3.14 For all $k > 1$ it holds that:

$$\begin{aligned} \exists i > 1 \; CV_{\mathrm{BH_{NP}}(2^k-1)}(i) = 1 \quad &\Longrightarrow \\ \mathrm{BH_{NP}} = \mathrm{P}^{\mathrm{NP}[k]} \quad &\Longrightarrow \\ \forall i \geq 1 \; CV_{\mathrm{BH_{NP}}(2^k)}(i) = 1 & \end{aligned}$$

Proof: For the first part let $L \in \mathrm{BH_{NP}}(2^k)$ and $L = L_1 - L_2$ with $L_1 \in \mathrm{NP}$ and $L_2 \in \mathrm{BH_{NP}}(2^k - 1)$. Furthermore let $\hat{L}$ be any $\mathrm{BH_{NP}}(2^k - 1)$-complete problem (in [CGH+88]

it was shown that such problems exist) and f_1, f_2 be the corresponding reductions from L_1 and L_2 to it. Then $x \in L$ if and only if $(f_1(x), f_2(x), 1, 0)$ is a characteristic vector of $\hat{L}$, which can be checked with one query to $\hat{L}$ itself, since $\hat{L}$ is complete in $\mathrm{BH_{NP}}(2^k - 1)$. Using the result of [AG88] that $\mathrm{BH_{NP}}(2^k - 1) \subseteq \mathrm{P}^{\mathrm{NP}[k]}$ and thus $\mathrm{P}^{\mathrm{BH_{NP}}(2^k-1)[1]} \subseteq \mathrm{P}^{\mathrm{NP}[k]}$ it follows that $\mathrm{BH_{NP}}(2^k) \subseteq \mathrm{P}^{\mathrm{NP}[k]}$. Since $\mathrm{BH_{NP}}(2^k) \supseteq \mathrm{P}^{\mathrm{NP}[k]}$ it follows that $\mathrm{BH_{NP}}(2^k) = \mathrm{P}^{\mathrm{NP}[k]}$ and the Boolean hierarchy collapses to that class, since in that case $\mathrm{BH_{NP}}(2^k)$ is closed under complement.

For the second implication let $L \in \mathrm{BH_{NP}}(2^k)$. How many queries to a language $L' \in \mathrm{BH_{NP}}(2^k)$ do we need to check i membership queries to L? Clearly, i queries to a $\mathrm{BH_{NP}}(2^k)$ language are sufficient. By the assumption that $\mathrm{BH_{NP}}(2^k) \subseteq \mathrm{P}^{\mathrm{NP}[k]}$ it follows that $V_i^L \in \mathrm{P}^{\mathrm{NP}[i \cdot k]}$, which again by the assumption is equal to $\mathrm{P}^{\mathrm{NP}[k]}$ and $\mathrm{BH_{NP}}(2^k)$. Thus one query to a $\mathrm{BH_{NP}}(2^k)$-language suffices. ∎

We note that for almost all oracles, characteristic vectors of any class of the Boolean hierarchy cannot be checked with one query to an oracle of the same power. In particular, it follows that with respect to a random oracle the bound of part 1 of Corollary 3.11 is optimal.

Corollary 3.15 With probability one relative to a random oracle A,

$$(\forall k \geq 1)[CV_{\mathrm{BH_{NP}}^A(k)} \neq 1].$$

This follows from the facts that the proof of Theorem 3.14 relativizes and the Boolean hierarchy over NP is infinite with respect to a random oracle [Cai89].

Another class located in the Boolean hierarchy over NP is US: A language belongs to US if it is accepted in polynomial time by a nondeterministic machine that, by definition, accepts if and only if it has exactly one accepting path [BG82,GW87]. US is known to be closed under intersection, but does not seem to be closed under union or complement. For this class Theorem 3.9 yields the following corollary:

Corollary 3.16

1. US is pseudo-closed under complement if and only if $\mathrm{BH_{NP}} = \mathrm{P}^{\mathrm{US}[1]}$.
2. If US is pseudo-closed under complement then the polynomial-time hierarchy collapses.

To see why this is so, first note that $\mathrm{BH_{NP}} = \mathrm{BH_{US}}$, since US contains coNP [BG82], so $\mathrm{BH_{NP}} \subseteq \mathrm{BH_{US}}$, and thus, since $\mathrm{US} \subseteq \mathrm{BH_{NP}}(2)$ the two hierarchies are equal. Part 1

follows from this observation and the fact the US is closed under intersection. Part 2 is a consequence of the result of Chang and Kadin that a collapse of the Boolean hierarchy over NP implies a collapse of the polynomial-time hierarchy.

Finally we observe that wee characteristic vector cost for a class within NP implies that it is low. Low classes, intuitively, are those sets carrying far less information than NP-complete sets. In particular, a set L is in $\widehat{low_3}$ if $\mathrm{P}^{\mathrm{NP}^{\mathrm{NP}^L}} = \mathrm{P}^{\mathrm{NP}^{\mathrm{NP}}}$ ($\widehat{low_3}$ was first defined in [KS85] as a generalization of the work of [Sch83]). Following Schöning's seminal paper on the low hierarchy, a number of papers have explored and refined its structure [KS85,BBS86, K̈88], culminating in the essentially optimal placement of classes within the low hierarchy [AH89]. The following result shows that classes with wee characteristic vector cost are simple in the sense of lowness.

Theorem 3.17 All classes $C \subseteq \mathrm{NP}$ with wee characteristic vector cost are $\widehat{low_3}$.

Proof: A sufficient condition for C to be $\widehat{low_3}$ is that for any $L \in C$ there is an $L' \in C$ such that $\{(x,y)|x \in L \wedge y \notin L\} \leq_m^p \{(x,y)|x \notin L' \vee y \in L'\}$ [Cha89]. For $L \in C$ with wee characteristic vector cost it holds that $\{(x,y)|x \in L \wedge y \notin L\} \in \mathrm{P}^{L'[1]}$ for some $L' \in C$. As in the proof of Theorem 3.9, one can easily show that $\mathrm{P}^{L'[1]} \subseteq \{(x,y)|x \notin L' \vee y \in L'\}$, and thus the condition is fulfilled. ∎

Acknowledgements

We wish to thank Gerd Wechsung, Garry Benson, Gerhard Buntrock, Dirk Siefkes, and Hubert Wagener for reading early versions of the paper, and to Paul Vitanyi for helpful conversations. We are especially grateful to Richard Chang for pointing out the modified proof of Theorem 2.5.

References

[AG88] A. Amir and W. Gasarch. Polynomial terse sets. *Information and Computation*, 77:37–56, 1988.

[AH89] E. Allender and L. Hemachandra. Lower bounds for the low hierarchy. In *Automata, Languages, and Programming (ICALP 1989)*, pages 31–45. Springer-Verlag *Lecture Notes in Computer Science #372* , July 1989.

[All86] E. Allender. The complexity of sparse sets in P. In *Proceedings 1st Structure in Complexity Theory Conference*, pages 1–11. Springer-Verlag *Lecture Notes in Computer Science #223*, June 1986.

[AR88] E. Allender and R. Rubinstein. P-printable sets. *SIAM Journal on Computing*, 17(6):1193–1202, 1988.

[Bar68] Y. Barzdin'. Complexity of programs to determine whether natural numbers not greater than n belong to a recursively enumerable set. *Soviet Math. Dokl.*, 9:1251–1254, 1968.

[BBJ+] A. Bertoni, D. Bruschi, D. Joseph, M. Sitharam, and P. Young. Generalized boolean hierarchies and boolean hierarchies over RP. Manuscript, 1989. Preliminary version appears in *Proceedings Fundamentals of Computation Theory*, Springer-Verlag Lecture Notes in Computer Science.

[BBS86] J. Balcázar, R. Book, and U. Schöning. Sparse sets, lowness, and highness. *SIAM Journal on Computing*, 15:739–747, 1986.

[Bei87] R. Beigel. A structural theorem that depends quantitatively on the complexity of SAT. In *Proceedings of the 2nd Annual Conference on Structure in Complexity Theory*, pages 28–32. IEEE Computer Society Press, June 1987.

[Bei88a] R. Beigel. Bounded queries to SAT and the Boolean hierarchy. Unpublished manuscript, August 1988.

[Bei88b] R. Beigel. NP-hard sets are P-superterse unless R=NP. Technical Report 88-04, Johns Hopkins Department of Computer Science, August 1988.

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

[BGGO87] R. Beigel, W. Gasarch, J. Gill, and J. Owings. Terse, superterse, and verbose sets. Technical Report TR-1806, University of Maryland, Department of Computer Science, College Park, Maryland, 1987.

[BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the P=?NP question. *SIAM Journal on Computing*, 4(4):431–442, 1975.

[Cai89] J. Cai. With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy. *Journal of Computer and System Sciences*, 38(1):68–85, 1989.

[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, December 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, February 1989.

[CH] J. Cai and L. Hemachandra. On the power of parity polynomial time. *Mathematical Systems Theory.* To appear.

[Cha89] R. Chang. On the structure of bounded queries to arbitrary NP sets. In *Proceedings of the 4th Conference on Structure in Complexity Theory*, pages 250–258. IEEE Computer Science Press, June 1989.

[CK89] R. Chang and J. Kadin. The boolean hierarchy and the polynomial hierarchy: A closer connection. Technical Report TR 89-1008, Department of Computer Science, Cornell University, Ithaca, NY, May 1989.

[EHK81] R. Epstein, R. Haas, and R. Kramer. Hierarchies of sets and degrees below 0'. In *Logic Year 1979-80, The University of Connecticut, Lecture Notes in Mathematics #859*, pages 32–47. Springer-Verlag, Berlin, 1981.

[Gil77] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4):675–695, December 1977.

[GP86] L. Goldschlager and I. Parberry. On the construction of parallel computers from various bases of Boolean functions. *Theoretical Computer Science*, 43:43–58, 1986.

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

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

[Hem89] L. Hemachandra. The strong exponential hierarchy collapses. *Journal of Computer and System Sciences*, 39(3):299–322, 1989.

[HH88] L. Hemachandra and A. Hoene. On checking versus evaluation of multiple queries: Characteristic vector terseness. Technical Report No. 88-21, Technische Universität, Berlin, October 1988.

[HIS85] J. Hartmanis, N. Immerman, and V. Sewelson. Sparse sets in NP-P: EXPTIME versus NEXPTIME. *Information and Control*, 65(2/3):159–181, May/June 1985.

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

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

[Joc79] C. Jockusch. Recursion theory: Its generalizations and applications. In *Proccedings of the Logic Colloquium, Leeds*, pages 140–157. Cambridge University Press, 1979.

[Kö88] J. Kämper. Non-uniform proof systems: a new framework to describe non-uniform and probabilistic complexity classes. In *8th Conference on Foundations of Software Technology and Theoretical Computer Science (FST-TCS 1988)*, pages 193–210. Springer-Verlag *Lecture Notes in Computer Science #338*, December 1988.

[Kad88] J. Kadin. The polynomial time hierarchy collapses if the boolean hierarchy collapses. *SIAM Journal on Computing*, 17(6):1263–1282, 1988.

[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.

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

[PZ83] C. Papadimitriou and S. Zachos. Two remarks on the power of counting. In *Proceedings 6th GI Conference on Theoretical Computer Science*, pages 269–276. Springer-Verlag Lecture Notes in Computer Science #145, 1983.

[Rog67] H. Rogers, Jr. *The Theory of Recursive Functions and Effective Computability.* McGraw-Hill, 1967.

[Rub88] R. Rubinstein. *Structural Complexity Classes of Sparse Sets: Intractability, Data Compression and Printability.* PhD thesis, Northeastern University, Boston, MA, August 1988.

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

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

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

[Wag87] K. Wagner. Bounded query classes. Institut für Mathematik 157, Universität Augsburg, Augsburg, W. Germany, October 1987.

[Wag88] K. Wagner. Bounded query computation. In *Proceedings 3rd Structure in Complexity Theory Conference*, pages 260–277. IEEE Computer Society Press, June 1988.

[Wec85] G. Wechsung. On the boolean closure of NP. In *Proccedings of the International Conference on Fundamentals of Computation Theory*, pages 485–493. Springer-Verlag, Lecture Notes in Computer Science, 1985.