On the Query Complexity of Sets

Richard Beigel*[1], William Gasarch**[2], Martin Kummer[3], Georgia Martin[2], Timothy McNicholl[4], Frank Stephan***[5]

[1] Department of Computer Science, Yale University, P.O. Box 208285, New Haven, CT 06520-8285, U.S.A. (email: `beigel@cs.yale.edu`)
[2] Department of Computer Science, University of Maryland, College Park, MD 20742, U.S.A. (email: {`gasarch; gam`}`@cs.umd.edu`)
[3] Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe, 76128 Karlsruhe, Germany (email: `kummer@ira.uka.de`)
[4] Department of Mathematics, Ottawa University, Ottawa, KS 66067, U.S.A. (email: `mcnichol@iso.ott.edu`)
[5] Mathematisches Institut, Ruprecht-Karls-Universität, Im Neuenheimer Feld 294, 69120 Heidelberg, Germany (email: `fstephan@math.uni-heidelberg.de`)

Abstract. There has been much research over the last eleven years that considers the number of queries needed to compute a function as a measure of its complexity. We are interested in the complexity of certain sets in this context. We study the sets
$\mathrm{ODD}_n^A = \{(x_1, \ldots, x_n) : |A \cap \{x_1, \ldots, x_n\}| \text{ is odd}\}$ and
$\mathrm{WMOD}(m)_n^A = \{(x_1, \ldots, x_n) : |A \cap \{x_1, \ldots, x_n\}| \not\equiv 0 \pmod{m}\}$.
If $A = K$ or A is semirecursive, we obtain tight bounds on the query complexity of ODD_n^A and $\mathrm{WMOD}(m)_n^A$. We obtain lower bounds for A r.e. The lower bounds for A r.e. are derived from the lower bounds for A semirecursive. We obtain that every tt-degree has a set A such that ODD_n^A *requires* n parallel queries to A, and a set B such that ODD_n^B can be decided with one query to B. Hence for bounded-query complexity, how information is packaged is more important than Turing degree.
We investigate when extra queries add power. We show that, for several nonrecursive sets A, the more queries you can ask, the more sets you can decide; however, there are sets for which more queries do not help at all.

1 Introduction

One concern of theoretical computer science is the classification of problems in terms of how hard they are. The natural measure of difficulty f is the amount of *time* needed to compute $f(x)$. Other resources, such as *space*, have also been considered.

In recursion theory, by contrast, a function is considered to be 'easy' if there exists *some* algorithm that computes it. We wish to classify functions that are

* Supported in part by NSF Grant NSF CCR-8952528 and NSF Grant CCR-9415410
** Supported in part by NSF Grant CCR-9301339
*** Supported by the Deutsche Forschungsgemeinschaft (DFG) grant Me 672/4-2

'hard,' i.e., not computable, in a quantitative way. We cannot use time or space, since the functions are not even computable. We cannot use Turing degree, since this notion is not quantitative. Hence we use a notion of complexity that is quantitative like time or space, yet captures difficulty within recursion theory like Turing degree. It is for these reasons that the theory of bounded queries in recursion theory was developed [1, 2, 6].

Given that a function f is recursive in a set X, our main concern will be *the number of queries to X required to compute f*. This will be our measure of complexity. Several natural *functions* have been classified this way (see [2, 3, 7, 11, 12]). In this paper we investigate the complexity of *sets* (i.e., the complexity of their characteristic functions). We informally summarize some of our results; a more formal and complete summary is contained at the end of Section 2.

We are concerned with the following sets.

1. $\mathrm{ODD}_n^A = \{(x_1, \ldots, x_n) : |A \cap \{x_1, \ldots, x_n\}| \text{ is odd}\}$, for $n \geq 1$.
2. $\mathrm{WMOD}(m)_n^A = \{(x_1, \ldots, x_n) : |A \cap \{x_1, \ldots, x_n\}| \not\equiv 0 \pmod{m}\}$, for $m \geq 2, n \geq 1$.

The set ODD_n^A is similar to the PARITY problem, which has been well studied [5, 9, 16]. In addition, Yao has proved [8, 13] (in a different framework) that if a set A is unpredictable, then ODD_n^A is also unpredictable. Hence the set ODD_n^A is of interest. $\mathrm{WMOD}(m)_n^A$ is a generalization of ODD_n^A. The acronym WMODm stands for 'Weak-MODm.' MODm is the numeric function that, given x, returns $y \in \{0, \ldots, m-1\}$ such that $x \equiv y \pmod{m}$. Weak-MODm yields less information: just the answer to the question "Is $\mathrm{MODm}(x) = 0$?".

Clearly, ODD_n^A can be decided with n parallel queries to A. We show that if A is semirecursive or r.e., this is optimal (unless A is recursive). Since there is a semirecursive set in every tt-degree, this shows that every nonrecursive tt-degree has a set A such that ODD_n^A *requires* n parallel queries to A. By contrast, we also show that every tt-degree has a set B such that ODD_n^B can be decided with just *one* query to B. This contrast bolsters a recurring theme in bounded-query complexity: How information is packaged (e.g., as a semirecursive set) will affect the complexity, but the information itself (e.g., the tt-degree) will not.

We also obtain upper and lower bounds on the complexity of ODD_n^A using sequential queries, and for $\mathrm{WMOD}(m)_n^A$ using both parallel and sequential queries. These results, and others, are summarized at the end of Section 3. We have other results on bounded-query complexity that are also summarized there.

2 Notation, Definitions, and Useful Lemmas

We use standard notation from recursion theory [15, 17]. $M_0, M_1, \ldots$ is an effective list of Turing machines. $M_0^{()}, M_1^{()}, \ldots$ is an effective list of oracle Turing machines. W_e is the domain of M_e. Hence $W_0, W_1, \ldots$ is an effective list of all r.e. sets. $D_e = \{i : \text{ the } i^{\text{th}} \text{ bit of } e \text{ is } 1\}$. Hence $D_0, D_1, \ldots$ is a list of all finite sets. We often use W_e and D_e to denote sets of strings. Let $e, x, s \in \mathsf{N}$. $M_e(x)\downarrow$ means that $M_e(x)$ converges. $M_e(x)\downarrow= b$ means that $M_e(x)$ converges and the output is b. $M_e(x)\uparrow$ means that $M_e(x)$ diverges. $M_{e,s}(x)\downarrow$ means that $M_e(x)$ converges within s steps. $M_{e,s}(x)\uparrow$ means that $M_e(x)$ does not converge within s steps. Let A be a set. A' is $\{e : M_e^A(e)\downarrow\}$ (i.e., A' is the halting problem relative to A). Define $A^{(0)} = A$ and $A^{(i+1)} = (A^{(i)})'$. Let A^ω be $\bigcup_{i=1}^{\omega}\{\langle i,x\rangle : x \in A^{(i)}\}$. We will be using $\emptyset^\omega$. Note that if A is in the arithmetic hierarchy, then $A \leq_{\mathrm{m}} \emptyset^\omega$.

We define several classes of functions. At the end of this section we will state a known lemma that relates these classes.

Definition 1. Let $n \in \mathsf{N}$, and let A be a set. $\mathrm{FQ}(n,A)$ is the collection of all total functions f such that f is recursive in A via an algorithm that makes at most n sequential (adaptive) queries to A. $\mathrm{FQC}(n,A)$ is the collection of all functions f such that f is recursive in A via an algorithm $M^{()}$ such that (1) for all x, $M^A(x)$ makes at most n sequential queries to A, and (2) for all x, Y, the $M^Y(x)$ computation converges. (Note that if $f \in \mathrm{FQC}(n,A)$ via M^A, then even if false answers are supplied to the queries, the $M^{()}(x)$ computation will converge, though perhaps not to $f(x)$.) $\mathrm{FQ}_{||}(n,A)$ is the collection of all total functions f such that f is recursive in A via an algorithm that makes at most n parallel (nonadaptive) queries to A.

Definition 2. Let $n \geq 1$, and let f be a total function. Then f is *n-enumerable* (*strongly n-enumerable*), denoted $f \in \mathrm{EN}(n)$ ($f \in \mathrm{SEN}(n)$), if there exists a recursive function g such that, for all x, $|W_{g(x)}| \leq n$ and $f(x) \in W_{g(x)}$ ($|D_{g(x)}| \leq n$ and $f(x) \in D_{g(x)}$). If $f \in \mathrm{EN}(n)$ then, given x, we can—*over time*—list out $\leq n$ possibilities for $f(x)$, one of which is correct. We never know that we have obtained all the possibilities (unless we eventually get n possibilities). If $f \in \mathrm{SEN}(n)$ then, given x, we can list out $\leq n$ possibilities for $f(x)$, one of which is correct; moreover, we *do* have all the possibilities—and we *know* we have them all. (This concept first appeared in a recursion-theoretic framework in [1]. Some very general theorems about $\mathrm{EN}(n)$ were proved in [12]. The term 'enumerable' is from [4], where it was defined in a polynomial-time-bounded framework.)

Definition 3. Let B be a set. $B \in \mathrm{Q}(n,A)$ if $\chi_B \in \mathrm{FQ}(n,A)$. $B \in \mathrm{Q}_{||}(n,A)$ if $\chi_B \in \mathrm{FQ}_{||}(n,A)$. $B \in \mathrm{QC}(n,A)$ if $\chi_B \in \mathrm{FQC}(n,A)$.

Definition 4. [2] Let $n \geq 1$, let $x_1,\ldots,x_n \in \mathsf{N}$, and let $A \subseteq \mathsf{N}$. Then C_n^A is the string-valued function defined by $\mathrm{C}_n^A(x_1,\ldots,x_n) = A(x_1)\cdots A(x_n)$.

Definition 5. Let A be a set. A is *supportive* if $(\forall n)[\mathrm{Q}(n,A) \subset \mathrm{Q}(n+1,A)]$. A is *parallel supportive* if $(\forall n)[\mathrm{Q}_{||}(n,A) \subset \mathrm{Q}_{||}(n+1,A)]$.

We will prove several results about semirecursive sets. Semirecursive sets are interesting because (1) there is a semirecursive set in every tt-degree, (2) there is an r.e. semirecursive set in every r.e. T-degree, and (3) we will derive results about r.e. sets *from* results about semirecursive sets.

Definition 6. [10] Let X be a set. If $\sqsubseteq$ is a linear ordering, then we write $\sqsubseteq \leq_{\mathrm{T}} X$ to indicate that the problem of determining how two elements compare is recursive in X. A set A is *semirecursive in* X if there exists a linear ordering $\sqsubseteq \leq_{\mathrm{T}} X$ such that A is closed downward under $\sqsubseteq$. This is equivalent to the existence of a function $f \leq_{\mathrm{T}} X$ such that (1) $f(x,y) \in \{x,y\}$, and (2) $A \cap \{x,y\} \neq \emptyset \Rightarrow f(x,y) \in A$. A set A is *semirecursive* if it is semirecursive in $\emptyset$.

Definition 7. If A is a set, then A^{tt} is the set of all quantifier-free statements using $\wedge, \vee, \neg, \in$ that are true of A. (Example: $((12 \in A) \wedge (14 \notin A))$.)

The following lemmas from [2] will be useful.

Lemma 8. *If* $n \in \mathsf{N}$ *then* $(\exists X)[f \in \mathrm{FQ}(n,X)]$ *iff* $f \in \mathrm{EN}(2^n)$.

Lemma 9. *If* $(\exists n)[\mathrm{C}_n^A \in \mathrm{EN}(n)]$, *then* A *is recursive.*

Lemma 10. *If* $(\exists n)[\mathrm{C}_n^A \in \mathrm{FQ}_{||}(n-1,A)]$, *then* A *is recursive.*

Lemma 11. *[2] If* $(\exists k)[\mathrm{C}_k^A \in \mathrm{SEN}(2^k-1)]$, *then* $(\exists c)(\forall n)[\mathrm{C}_n^A \in \mathrm{SEN}(cn^k)]$.

3 Summary of Results

We summarize our results. Let $A = K$ or any nonrecursive semirecursive set. Let $n \geq 1$, let $m \geq 2$, and assume that m divides n. Then the following hold.

1. $\mathrm{ODD}_n^A \in \mathrm{Q}_{||}(n,A) - \mathrm{Q}_{||}(n-1,A)$.
2. $\mathrm{ODD}_{2^n}^A \in \mathrm{Q}(n+1,A) - \mathrm{Q}(n,A)$.
3. $\mathrm{WMOD}(m)_n^A \in \mathrm{Q}_{||}(\frac{2n}{m},A) - \mathrm{Q}_{||}(\frac{2n}{m}-1,A)$.
4. $\mathrm{WMOD}(m)_n^A \in \mathrm{Q}(\lceil \log(\frac{2n}{m}+1)\rceil, A) - \mathrm{Q}(\lceil \log(\frac{2n}{m}+1)\rceil - 1, A)$.
5. A is both supportive and parallel supportive.
6. All the lower bounds implied by the above items hold for any nonrecursive r.e. set. The upper bounds *do not* hold for every r.e. set [2].

There are also sets that are not supportive and/or not parallel supportive.

1. There exist sets that are neither supportive nor parallel supportive.
2. The sets K^{tt} and $\emptyset^{\omega}$ are supportive but not parallel supportive.

We also obtain results about when $\mathrm{Q}(n,A) = \mathrm{QC}(n,A)$.

1. $(\forall n)[\mathrm{Q}(n,K) = \mathrm{QC}(n,K)]$.
2. If A is a nonrecursive semirecursive r.e. set, then $\mathrm{Q}(1,A) - \bigcup_{i=0}^{\infty} \mathrm{QC}(i,A) \neq \emptyset$.

4 Supportive Sets and the Complexity of $\mathrm{ODD}_{\mathrm{n}}^{\mathrm{A}}$

The following lemma will help in proving that sets are supportive.

Lemma 12. *Let* A *be a set. If* $(\forall n)(\exists m \geq 1)[\mathrm{ODD}_m^A \notin \mathrm{Q}(n,A)]$, *then* A *is supportive and parallel supportive.*

Proof sketch. We show that A is parallel supportive. The proof that A is supportive is similar. Fix n. Note that $A \in \mathrm{Q}_{||}(1, A) - \mathrm{Q}_{||}(0, A)$, since A is not recursive, so assume that $n > 0$. Let $m \geq 1$ be the least number such that $\mathrm{ODD}_m^A \notin \mathrm{Q}_{||}(n, A)$. Note that $m > 1$ and $\mathrm{ODD}_{m-1}^A \in \mathrm{Q}_{||}(n, A)$. One can use this to show that $\mathrm{ODD}_m^A \in \mathrm{Q}_{||}(n+1, A)$. Hence $\mathrm{ODD}_m^A \in \mathrm{Q}_{||}(n+1, A) - \mathrm{Q}_{||}(n, A)$, so A is parallel supportive.

Our next two theorems pin down the complexity of ODD_n^A in terms of both parallel and sequential queries to A, for A semirecursive.

Theorem 13. *Let A be a nonrecursive semirecursive set, and let $n \geq 1$. Then $\mathrm{ODD}_n^A \in \mathrm{Q}_{||}(n, A) - \mathrm{Q}_{||}(n-1, A)$.*

Proof sketch. Clearly, $\mathrm{ODD}_n^A \in \mathrm{Q}_{||}(n, A)$.

Assume that A is semirecursive via recursive linear ordering $\sqsubseteq$. Assume that $\mathrm{ODD}_n^A \in \mathrm{Q}_{||}(n-1, A)$ via M^A. The following algorithm shows that $\mathrm{C}_{2n+1}^A \in \mathrm{FQ}_{||}(2n, A)$. Hence by Lemma 10, A is recursive. This contradicts the hypothesis. Let the input be $(x_1, \ldots, x_{2n+1})$, where $x_1 \sqsubseteq \cdots \sqsubseteq x_{2n+1}$. Note that $\mathrm{C}_{2n+1}^A(x_1, \ldots, x_{2n+1}) \in 1^*0^*$. Use M^A to obtain $z_1, \ldots, z_{n-1}$ such that we can determine $\mathrm{ODD}_n^A(x_2, x_4, \ldots, x_{2n})$ from $\mathrm{C}_{n-1}^A(z_1, \ldots, z_{n-1})$. Ask $\mathrm{C}_{2n}^A(x_1, x_3, \ldots, x_{2n+1}, z_1, z_2, \ldots, z_{n-1})$. Now compute $\mathrm{ODD}_n^A(x_2, x_4, \ldots, x_{2n})$ and $\mathrm{C}_{n+1}^A(x_1, x_3, \ldots, x_{2n+1})$. Since $\mathrm{C}_{2n+1}^A(x_1, \ldots, x_{2n+1}) \in 1^*0^*$, we can easily compute $\mathrm{C}_{2n+1}^A(x_1, \ldots, x_{2n+1})$.

Theorem 14. *Let A be a nonrecursive semirecursive set, and let $n \in \mathsf{N}$. Then $\mathrm{ODD}_{2^n}^A \in \mathrm{Q}(n+1, A) - \mathrm{Q}(n, A)$.*
Hence by Lemma 12, A is supportive and parallel supportive.

Proof sketch. Let A be semirecursive via recursive linear ordering $\sqsubseteq$. A binary-search algorithm shows that $\mathrm{C}_{2^n}^A \in \mathrm{Q}(n+1, A)$, hence $\mathrm{ODD}_{2^n}^A \in \mathrm{Q}(n+1, A)$.

Assume that $\mathrm{ODD}_{2^n}^A \in \mathrm{Q}(n, A)$ via M^A. We use $M^{()}$ in the following algorithm to show that $\mathrm{C}_{2^n}^A \in \mathrm{EN}(2^n)$. By Lemma 9, A is recursive, contrary to hypothesis.

Input $(x_1, \ldots, x_{2^n})$, where $x_1 \sqsubseteq \cdots \sqsubseteq x_{2^n}$. The algorithm will search (forever, if need be) for evidence that a possibility for $\mathrm{C}_{2^n}^A(x_1, \ldots, x_{2^n})$ deserves to be output. It outputs the possibility $1^i 0^{2^n - i}$ if it finds $\sigma_i \in \{0,1\}^n$ and $y_1, \ldots, y_n \in \mathsf{N}$ such that the following occur.

1. If $M^{()}(x_1, \ldots, x_{2^n})$ is run and the j^{th} query is answered with $\sigma_i(j)$ (for $j = 1, \ldots, n$), then the computation converges to $b \in \{0,1\}$, where $b = 1$ iff i is odd.
2. The queries made are (in order) $y_1, \ldots, y_n$.
3. The assignment $\mathrm{C}_{2^n+n}^A(x_1, \ldots, x_{2^n}, y_1, \ldots, y_n) = 1^i 0^{2^n - i}\sigma_i$ is *consistent* with $\sqsubseteq$. (E.g., if $x_2 \sqsubseteq y_7 \sqsubseteq x_3$ and x_2, x_3 are both assigned to 1, then y_7 must be assigned to 1.)

Clearly, the value of $\mathrm{C}^A_{2^n}(x_1,\ldots,x_{2^n})$ will be enumerated. We need to show that $\le 2^n$ possibilities are enumerated. Let Y be the set of all queries that could be asked on any converging query path of the computation. Note that $|Y| \le 2^n - 1$, hence there are at most 2^n possible assignments with domain Y that are consistent with $\sqsubseteq$.

Assume, by way of contradiction, that all 2^n+1 elements of the form $1^i0^{2^n-i}$ are enumerated as possibilities for $\mathrm{C}^A_{2^n}(x_1,\ldots,x_{2^n})$. We show that there are 2^n+1 consistent assignments with domain Y, a contradiction.

Let $i \le 2^n$. Since $1^i0^{2^n-i}$ is enumerated, we know that σ_i is an assignment with domain a *subset* of Y that is compatible with $\mathrm{C}^A_{2^n}(x_1,\ldots,x_{2^n}) = 1^i0^{2^n-i}$. Let ρ_i be the minimal assignment with domain Y that is consistent with $\sqsubseteq$ and extends σ_i in such a way that ρ_i is compatible with $\mathrm{C}^A_{2^n}(x_1,\ldots,x_{2^n}) = 1^i0^{2^n-i}$ (minimal in the number of elements of Y that are assigned to 1). Then the number of elements of Y that are assigned to 1 by ρ_i is a nondecreasing function of i.

If $i < 2^n$, then $\rho_i \ne \rho_{i+1}$, since ρ_i and ρ_{i+1} lead to different answers about the value of $\mathrm{ODD}^A_{2^n}(x_1,\ldots,x_{2^n})$. Therefore ρ_i is a strictly increasing function of i. Hence $\rho_0,\ldots,\rho_{2^n}$ are distinct.

We now show that if A is a nonrecursive r.e. set, then $\mathrm{ODD}^A_n \notin \mathrm{Q}_{||}(n-1, A)$ and $\mathrm{ODD}^A_{2^n} \notin \mathrm{Q}(n, A)$. Our plan is to make r.e. sets 'look like' semirecursive sets and then apply a relativized version of Theorems 13 and 14.

Definition 15. A set X is *nice* if every partial recursive 0,1-valued function g has a total extension $h \le_{\mathrm{T}} X$.

Definition 16. A set X is *recursively bounded* (abbreviated r.b.) if, for every (total) function $f \le_{\mathrm{T}} X$, there exists a recursive function g such that $(\forall x)[f(x) < g(x)]$. A Turing degree is *r.b.* if it has an r.b. set. Note that all the sets in an r.b. degree are r.b. (In the literature, r.b. degrees are called 'hyperimmune free.' This term comes from an equivalent definition that we will not be using. See [14] or [15, 17].)

Lemma 17. *Let A, X be sets, and assume that A is r.e. and X is nice. Then A is semirecursive in X. (Hence r.e. sets 'look' semirecursive.)*

Proof. We will show that A is semirecursive in X by exhibiting a function $f \le_{\mathrm{T}} X$ such that $f(x,y) \in \{x,y\}$ and $A \cap \{x,y\} \ne \emptyset \Rightarrow f(x,y) \in A$.

Let g be the partial recursive 0,1-valued function defined by

$$g(x,y) = \begin{cases} 1, & \text{if } (\exists s)[x \in A_s \wedge y \notin A_s]; \\ 0, & \text{if } (\exists s)[y \in A_s \wedge x \notin A_s]; \\ \uparrow, & \text{otherwise.} \end{cases}$$

Since X is nice, there is a 0,1-valued total function $h \le_{\mathrm{T}} X$ such that h extends g. A is semirecursive in X via the function f defined by

$$f(x,y) = \begin{cases} x, & \text{if } h(x,y) = 1; \\ y, & \text{if } h(x,y) = 0. \end{cases}$$

Lemma 18. *Let A, X be sets, and assume that A is r.e., X is r.b., and $A \leq_{\mathrm{T}} X$. Then A is recursive.*

Proof. Let f be the function defined by

$$f(x) = \begin{cases} \mu s[x \in A_s], & \text{if } x \in A; \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, $f \leq_{\mathrm{T}} A \leq_{\mathrm{T}} X$. Since X is r.b., there is a recursive function g such that $(\forall x)[f(x) < g(x)]$. Note that, for all x, $x \in A$ iff $x \in A_{g(x)}$. Hence A is recursive.

The following result is known.

Lemma 19. *There exist sets which are both r.b. and nice.*

Theorem 20. *Let A be an r.e. set, and let $n \geq 1$. If $\mathrm{ODD}_n^A \in \mathrm{Q}_{||}(n-1, A)$ or $\mathrm{ODD}_{2^n}^A \in \mathrm{Q}(n, A)$, then A is recursive.*

Proof. Let X be a nice r.b. set (such exist, by Lemma 19). A is semirecursive in X, by Lemma 17. If $\mathrm{ODD}_n^A \in \mathrm{Q}_{||}(n-1, A)$ or $\mathrm{ODD}_{2^n}^A \in \mathrm{Q}(n, A)$ then, by a relativized version of Theorems 13 and 14, $A \leq_{\mathrm{T}} X$. By Lemma 18, A is recursive.

Note 21. We have shown, by other techniques, that if A and B are r.e. and either $\mathrm{ODD}_n^A \in \mathrm{Q}_{||}(n-1, B)$ or $\mathrm{ODD}_{2^n}^A \in \mathrm{Q}(n, B)$, then A is recursive.

5 The Complexity of $\mathrm{WMOD}(m)_{\mathrm{n}}^{\mathrm{A}}$

We prove a very general theorem which applies to semirecursive sets and K.

Definition 22. Let A be a set. GE^A is the set of $(x_1, \ldots, x_n, j)$ such that $n \geq 1$, $j \in \mathsf{N}$, and at least j of $x_1, \ldots, x_n$ are in A. (GE is a mnemonic for 'Greater than or Equal to.')

Theorem 23. *Let A be a set, and let m, n be such that $2 \leq m \leq n$ and m divides n.*

1. *If $\mathrm{GE}^A \in \mathrm{Q}(1, A)$, then $\mathrm{WMOD}(m)_n^A \in \mathrm{Q}_{||}(\frac{2n}{m}, A) \cap \mathrm{Q}(\lceil \log(\frac{2n}{m}+1) \rceil, A)$.*
2. *If $\mathrm{GE}^A \leq_{\mathrm{m}} A$ and $\mathrm{ODD}_{2n/m}^A \notin \mathrm{Q}_{||}(\frac{2n}{m}-1, A) \cup \mathrm{Q}(\lceil \log(\frac{2n}{m}+1) \rceil - 1, A)$, then $\mathrm{WMOD}(m)_n^A \notin \mathrm{Q}_{||}(\frac{2n}{m}-1, A) \cup \mathrm{Q}(\lceil \log(\frac{2n}{m}+1) \rceil - 1, A)$.*

Proof sketch. We prove the statements about $\mathrm{Q}_{||}(-, A)$. The ones about $\mathrm{Q}(-, A)$ are similar.

(1) Assume that $\mathrm{GE}^A \in \mathrm{Q}(1, A)$. The following algorithm shows that $\mathrm{WMOD}(m)_n^A \in \mathrm{Q}_{||}(\frac{2n}{m}, A)$. We use the premise $\mathrm{GE}^A \in \mathrm{Q}(1, A)$ implicitly.

On input $(x_1, \ldots, x_n)$, make the queries

$$\begin{array}{rr}
 & \text{“}(x_1, \ldots, x_n, 1) \in \mathrm{GE}^A?\text{”}, \\
\text{“}(x_1, \ldots, x_n, m) \in \mathrm{GE}^A?\text{”}, & \text{“}(x_1, \ldots, x_n, m+1) \in \mathrm{GE}^A?\text{”}, \\
\text{“}(x_1, \ldots, x_n, 2m) \in \mathrm{GE}^A?\text{”}, & \text{“}(x_1, \ldots, x_n, 2m+1) \in \mathrm{GE}^A?\text{”}, \\
\vdots & \vdots \\
\text{“}(x_1, \ldots, x_n, (\frac{n}{m}-1)m) \in \mathrm{GE}^A?\text{”}, & \text{“}(x_1, \ldots, x_n, (\frac{n}{m}-1)m+1) \in \mathrm{GE}^A?\text{”}, \\
\text{“}(x_1, \ldots, x_n, \frac{n}{m}m) \in \mathrm{GE}^A?\text{”}. &
\end{array}$$

From this information, one can determine $\mathrm{WMOD}(m)_n^A(x_1, \ldots, x_n)$.
(2) Assume that $\mathrm{GE}^A \leq_{\mathrm{m}} A$ and $\mathrm{ODD}_{2n/m}^A \notin \mathrm{Q}_{||}(\frac{2n}{m}-1, A)$. The following algorithm shows that $\mathrm{ODD}_{2n/m}^A \in \mathrm{Q}(1, \mathrm{WMOD}(m)_n^A)$.
This implies that $\mathrm{WMOD}(m)_n^A \notin \mathrm{Q}_{||}(\frac{2n}{m}-1, A)$, since otherwise $\mathrm{ODD}_{2n/m}^A \in \mathrm{Q}_{||}(\frac{2n}{m}-1, A)$, which contradicts our assumption.

Let the input be $(x_1, \ldots, x_{2n/m})$. Using the fact that $\mathrm{GE}^A \leq_{\mathrm{m}} A$, obtain numbers $z_1, \ldots, z_{2n/m}$ such that, for all i with $1 \leq i \leq 2n/m$,

$$(x_1, \ldots, x_{2n/m}, i) \in \mathrm{GE}^A \text{ iff } z_i \in A.$$

Ask whether

$$(z_1, z_2, \ldots, z_{2n/m}, z_2, \ldots, z_2, z_4, \ldots, z_4, \ldots, z_{2n/m}, \ldots, z_{2n/m}) \in \mathrm{WMOD}(m)_n^A.$$

(The expression $z_i, \ldots, z_i$ means that the element z_i appears as a component $m-2$ times in succession.) If the answer is YES, then output 1; otherwise, output 0.

Clearly, this algorithm makes 1 query to $\mathrm{WMOD}(m)_n^A$. One can show that it decides $\mathrm{ODD}_{2n/m}^A$.

If $A = K$ or if A is a nonrecursive semirecursive set, then $\mathrm{GE}^A \leq_{\mathrm{m}} A$. Combining this fact with Theorems 13, 14, and 20, we obtain the following corollary.

Corollary 24. *Let m, n be such that $2 \leq m \leq n$ and m divides n.*

1. *If a set A is nonrecursive and semirecursive, then*
   $\mathrm{WMOD}(m)_n^A \in \mathrm{Q}_{||}(\frac{2n}{m}, A) - \mathrm{Q}_{||}(\frac{2n}{m}-1, A)$, *and*
   $\mathrm{WMOD}(m)_n^A \in \mathrm{Q}(\lceil\log(\frac{2n}{m}+1)\rceil, A) - \mathrm{Q}(\lceil\log(\frac{2n}{m}+1)\rceil - 1, A)$.
2. $\mathrm{WMOD}(m)_n^K \in \mathrm{Q}_{||}(\frac{2n}{m}, K) - \mathrm{Q}_{||}(\frac{2n}{m}-1, K)$, *and*
   $\mathrm{WMOD}(m)_n^K \in \mathrm{Q}(\lceil\log(\frac{2n}{m}+1)\rceil, K) - \mathrm{Q}(\lceil\log(\frac{2n}{m}+1)\rceil - 1, K)$.
3. *Every nonzero r.e. degree has an r.e. set A with*
   $\mathrm{WMOD}(m)_n^A \in \mathrm{Q}_{||}(\frac{2n}{m}, A) - \mathrm{Q}_{||}(\frac{2n}{m}-1, A)$, *and*
   $\mathrm{WMOD}(m)_n^A \in \mathrm{Q}(\lceil\log(\frac{2n}{m}+1)\rceil, A) - \mathrm{Q}(\lceil\log(\frac{2n}{m}+1)\rceil - 1, A)$.

Theorem 25. *Every tt-degree has a set B such that $(\forall n \geq 1)[\mathrm{ODD}_n^B \in \mathrm{Q}(1, B)]$.*

Proof. Let A be some set in the tt-degree. Let $B = A^{\mathrm{tt}}$, and let $n \geq 1$. Clearly, $\mathrm{ODD}_n^B \in \mathrm{Q}(1, B)$, by making a query involving an exponentially large Boolean combination constructed from the n statements in the input.

6 Supportive and Non-supportive Sets

We show that there exists a set that is neither supportive nor parallel supportive.

Theorem 26. *Let A be an r.b. set. Then (1)* $(\forall n)[\mathrm{Q}(n,A)=\mathrm{QC}(n,A)]$*, and (2)* A^{tt} *is neither supportive nor parallel supportive.*

Proof sketch. Let $n \in \mathsf{N}$. Assume that $B \in \mathrm{Q}(n,A)$ via M^A. We show that $B \in \mathrm{QC}(n,A)$. Let f be the function $f(x)=\mu s[M^A(x)$ halts in s steps]. Clearly, $f \leq_{\mathrm{T}} A$. By the definition of an r.b. set, there exists a recursive function g such that $(\forall x)[f(x)<g(x)]$. We use g to define an oracle Turing machine $N^{()}$ such that $B \in \mathrm{QC}(n,A)$ via N^A.

$N^{()}$ is defined as follows: On input x and with X as oracle, simulate $M^X(x)$ for $g(x)$ steps. If the computation halts within $g(x)$ steps, then output the result; else output 0. Clearly, $B \in \mathrm{QC}(n,A)$ via N^A.

Note that A^{tt} is r.b., hence $\mathrm{Q}(n,A^{\mathrm{tt}})=\mathrm{QC}(n,A^{\mathrm{tt}})$. It is straightforward to show that $\mathrm{QC}(n,A^{\mathrm{tt}}) \subseteq \mathrm{QC}(1,A^{\mathrm{tt}})$ (hence that $\mathrm{QC}(n,A^{\mathrm{tt}})=\mathrm{QC}(1,A^{\mathrm{tt}})$), by making a query involving an exponentially large Boolean combination of statements used as queries in the $\mathrm{QC}(n,A^{\mathrm{tt}})$ algorithm. Thus $\mathrm{Q}(n,A^{\mathrm{tt}})=\mathrm{QC}(1,A^{\mathrm{tt}})$.

Theorem 27. K^{tt} *and* $\emptyset^{\omega}$ *are supportive but not parallel supportive.*

Proof sketch. We prove this for K^{tt}; the proof for $\emptyset^{\omega}$ is similar.

K^{tt} not parallel supportive: Let $n \in \mathsf{N}$, and let $A \in \mathrm{Q}_{||}(n,K^{\mathrm{tt}})$. We can assume that $n \geq 1$. On input x, we can find $x_1,\ldots,x_n$ such that knowing $\mathrm{C}_n^{K^{\mathrm{tt}}}(x_1,\ldots,x_n)$ would yield $A(x)$. We can easily formulate a query y to K^{tt}, using $x_1,\ldots,x_n$, such that $x \in A$ iff $y \in K^{\mathrm{tt}}$.

K^{tt} supportive: For all n and all sets X, let

$$H(n,X)=\{e : M_e^X(e)\downarrow \text{ and asks } \leq n \text{ queries}\}.$$

It is easy to show that, for all n,X, $H(n,X) \notin \mathrm{Q}(n,X)$. One can also show that $H(n,K^{\mathrm{tt}}) \in \mathrm{Q}(n+1,K^{\mathrm{tt}})$.

7 The Power of Not Converging

The class $\mathrm{Q}(n,A)$ seems to be more powerful than $\mathrm{QC}(n,A)$, since the machine used to decide a set $B \in \mathrm{Q}(n,A)$ need not converge on every query path. In this section we examine which sets have $(\forall n)[\mathrm{Q}(n,A)=\mathrm{QC}(n,A)]$. Most of our theorems involve sets that are extreme: For some A, $(\forall n)[\mathrm{Q}(n,A)=\mathrm{QC}(n,A)]$; for other A, $\mathrm{Q}(1,A)-\bigcup_{n=0}^{\infty}\mathrm{QC}(n,A) \neq \emptyset$. Our definitions reflect this behavior.

Definition 28. A set A is *oracle convergent* (abbreviated o.c.) if $(\forall n)[\mathrm{Q}(n,A)=\mathrm{QC}(n,A)]$. A (T-, tt-) degree is *o.c.* if it has an o.c. set. A (T-, tt-) degree is *completely o.c.* if every set in it is o.c. A is *oracle non-convergent* (abbreviated o.n.c.) if $\mathrm{Q}(1,A)-\bigcup_{n=0}^{\infty}\mathrm{QC}(n,A) \neq \emptyset$. A (T-, tt-) degree is *o.n.c.* if it has an o.n.c. set. A (T-, tt-) degree is *completely o.n.c.* if every set in it is o.n.c. (By Theorem 26, any r.b. degree is completely o.c.)

Definition 29. Let A be an r.e. set, and let $e, x, m \in \mathsf{N}$. *$M_e^A(x)$ changes its mind at least m times* if there exist $s_0, \ldots, s_m$ with $s_0 < \cdots < s_m$ (if $m > 0$) such that, for all $i \le m$, $M_{e,s_i}^{A_{s_i}}(x) \downarrow\in \{0,1\}$ and (if $i < m$) $M_{e,s_i}^{A_{s_i}}(x) \ne M_{e,s_{i+1}}^{A_{s_{i+1}}}(x)$.

Theorem 30. *K is o.c.*

Proof sketch. Let $X \in \mathrm{Q}(n, K)$, and assume that $X \in \mathrm{Q}(n, K)$ via M_e^K. We show that $X \in \mathrm{QC}(n, K)$. Let

$$B = \{(x, m) : M_e^K(x) \text{ changes its mind at least } m \text{ times}\}.$$

B is r.e., hence $B \le_{\mathrm{m}} K$. Given x, we can perform a binary search, with n queries to K, to determine how many times $M_e^K(x)$ changes its mind. Then determine the least s such that $M_{e,s}^{K_s}(x) \downarrow\in \{0,1\}$, and let $b = M_{e,s}^{K_s}(x)$. Output b if the number of mindchanges is even; otherwise, output $1 - b$. Note that even if the answers supplied were incorrect, the algorithm converges.

We will show that every nonrecursive semirecursive r.e. set is o.n.c.

Lemma 31. *Let A, B be sets, and assume that $(\exists k \ge 1)[\mathrm{C}_k^A \in \mathrm{SEN}(2^k - 1)]$ and $(\forall k \ge 1)[\mathrm{C}_k^B \notin \mathrm{SEN}(2^k - 1)]$. Then $B \notin \bigcup_{n=0}^{\infty} \mathrm{QC}(n, A)$. (The intuition is that A is 'easy' to decide and B is 'hard,' so it is reasonable that B cannot be reduced to A in certain ways.)*

Proof. We show that $B \notin \bigcup_{n=1}^{\infty} \mathrm{QC}(1, \mathrm{C}_n^A)$. It is easy to show that $\bigcup_{n=1}^{\infty} \mathrm{QC}(1, \mathrm{C}_n^A) = \bigcup_{n=0}^{\infty} \mathrm{QC}(n, A)$; hence we obtain our result.

Assume, by way of contradiction, that $(\exists n_0 \ge 1)[B \in \mathrm{QC}(1, \mathrm{C}_{n_0}^A)]$. By making queries in parallel, we have $(\forall n \ge 1)[\mathrm{C}_n^B \in \mathrm{FQC}(1, \mathrm{C}_{n_0 \cdot n}^A)]$. Since $(\exists k \ge 1)[\mathrm{C}_k^A \in \mathrm{SEN}(2^k - 1)]$ and n_0 is a constant, it follows (by Lemma 11) that there exists c such that $(\forall n)[\mathrm{C}_{n_0 \cdot n}^A \in \mathrm{SEN}(cn^k)]$. Hence for large enough n, $\mathrm{C}_n^B \in \mathrm{FQC}(1, \mathrm{C}_{n_0 \cdot n}^A) \subseteq \mathrm{SEN}(cn^k) \subseteq \mathrm{SEN}(2^n - 1)$. This contradicts the hypothesis on B.

Definition 32. A function f is *dominated* if there is a recursive function g such that $(\forall x)[f(x) < g(x)]$.

Lemma 33. *Let A be a set such that (1) $(\exists k \ge 1)[\mathrm{C}_k^A \in \mathrm{SEN}(2^k - 1)]$ and (2) there exists a function $f \in \mathrm{FQ}(1, A)$ that is not dominated. Then A is o.n.c.*

Proof sketch. We construct a set B so that $(\forall k \ge 1)[\mathrm{C}_k^B \notin \mathrm{SEN}(2^k - 1)]$ and $B \in \mathrm{Q}(1, A)$, and then apply Lemma 31 to obtain that $B \notin \bigcup_{n=0}^{\infty} \mathrm{QC}(n, A)$.

Let the sets $X_{\langle e,i,k\rangle}$ (for $e, i, k \in \mathsf{N}$) be a recursive partition of N so that $|X_{\langle e,i,k\rangle}| = k$. Let $\mathbf{x}_{\langle e,i,k\rangle}$ be the k-tuple of numbers formed by taking the elements of $X_{\langle e,i,k\rangle}$ in increasing order. We use the $\mathbf{x}_{\langle e,i,k\rangle}$ to make sure that C_k^B is not strongly $(2^k - 1)$-enumerable via M_e. In particular, we construct B so that, for all e, k, we satisfy requirement

$$R_{\langle e,k\rangle} : M_e \text{ total } \Rightarrow (\exists i)[|D_{M_e(\mathbf{x}_{\langle e,i,k\rangle})}| \ge 2^k \vee \mathrm{C}_k^B(\mathbf{x}_{\langle e,i,k\rangle}) \notin D_{M_e(\mathbf{x}_{\langle e,i,k\rangle})}].$$

We construct $B \in \mathrm{Q}(1, A)$ via the following algorithm for it. We will use f, a function in $\mathrm{FQ}(1, A)$ that is not dominated, in the algorithm.

On input x, find e, i, k such that $x \in X_{\langle e,i,k\rangle}$. Compute $t = f(i)$. (This will require at most one query to A.) If $M_{e,t}(\mathbf{x}_{\langle e,i,k\rangle}) \uparrow$, then output 0 and halt; else compute $z = M_{e,t}(\mathbf{x}_{\langle e,i,k\rangle})$. (Note that if C_k^B is strongly $(2^k - 1)$-enumerable via M_e and $M_{e,t}(\mathbf{x}_{\langle e,i,k\rangle}) \downarrow$, then $\mathrm{C}_k^B(\mathbf{x}_{\langle e,i,k\rangle}) \in D_z$ and $|D_z| \le 2^k - 1$. We ensure that this does not happen.) If $|D_z| \ge 2^k$, then output 0 and halt. Otherwise, let $b_1 \cdots b_k$ be the lexicographically least element of $\{0,1\}^k - D_z$, and let j be such that x is the j^{th} component of $\mathbf{x}_{\langle e,i,k\rangle}$. Output b_j and halt.

Let $e, k \in \mathsf{N}$. We show that $R_{\langle e,k\rangle}$ is satisfied. Assume, by way of contradiction, that $R_{\langle e,k\rangle}$ is not satisfied. We obtain a recursive function g that dominates f, contrary to our assumption about f. Since $R_{\langle e,k\rangle}$ is not satisfied, we know that M_e is total and, for all i, the number of steps required for the computation of $M_e(\mathbf{x}_{\langle e,i,k\rangle})$ to halt is strictly greater than $f(i)$ (otherwise, $R_{\langle e,k\rangle}$ would have been satisfied when the elements of $X_{\langle e,i,k\rangle}$ were input to the algorithm). Hence the recursive function $g(i) = \mu t[M_{e,t}(\mathbf{x}_{\langle e,i,k\rangle}) \downarrow]$ dominates f.

Corollary 34. *Let A be a nonrecursive r.e. set which is semirecursive. Then A is o.n.c.*

Proof sketch. We show that A satisfies the two premises of Lemma 33.

$(\forall k)[\mathrm{C}_k^A \in \mathrm{SEN}(k+1)]$: Let A be semirecursive via recursive linear ordering $\sqsubseteq$. On input $(x_1, \ldots, x_k)$, we enumerate exactly $k+1$ candidates for the value of $\mathrm{C}_k^A(x_1, \ldots, x_k)$. We can assume that $x_1 \sqsubseteq \cdots \sqsubseteq x_k$. Enumerate the candidates $\{1^i 0^{k-i} : 0 \le i \le k\}$.

We exhibit a function $f \in \mathrm{FQ}(1, A)$ that is not dominated. Define f as follows. If $x \in A$, then $f(x) = \mu s[x \in A_s]$, else $f(x) = 0$. Clearly, $f \in \mathrm{FQ}(1, A)$. If f is dominated by a recursive function g, then $x \in A$ iff $x \in A_{g(x)}$; hence A is recursive, contrary to hypothesis.

We state three theorems without proof.

Theorem 35. *There exists an r.e. degree none of whose r.e. sets is o.c.*

Theorem 36. *There exist r.e. sets A such that $\emptyset <_{\mathrm{T}} A <_{\mathrm{T}} K$ and A is o.c.*

Theorem 37. *Let A be a set. The tt-degree of A is completely o.c. iff $(\forall f)[f \le_{\mathrm{wtt}} A \Rightarrow f$ is dominated$]$.*
The tt-degree of A is o.n.c. iff $(\exists f)[f \le_{\mathrm{wtt}} A$ and f is not dominated$]$. Hence every tt-degree is either o.n.c. or completely o.c. ($f \le_{\mathrm{wtt}} B$ if there exists an oracle Turing machine $M^{()}$ and a recursive function h such that (1) M^B computes f and (2) for all x, $h(x)$ is an upper bound on the queries made in the $M^B(x)$ computation.)

8 Acknowledgments

We would like to thank Richard Chang for proofreading.

References

1. R. Beigel. *Query-Limited Reducibilities.* PhD thesis, Stanford University, 1987. Also available as Report No. STAN-CS-88-1221.
2. R. Beigel, W. Gasarch, J. T. Gill, and J. C. Owings. Terse, superterse, and verbose sets. *Information and Computation*, 103:68–85, 1993.
3. R. Beigel, W. Gasarch, and E. Kinber. Frequency computation and bounded queries. *Theoretical Computer Science*, 1996 (to appear).
4. J. Cai and L. A. Hemachandra. Enumerative counting is hard. *Information and Computation*, 82(1):34–44, July 1989.
5. M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, April 1984.
6. W. I. Gasarch. A hierarchy of functions with applications to recursive graph theory. Technical Report 1651, University of Maryland, Dept. of Computer Science, 1985.
7. W. Gasarch. Bounded queries in recursion theory: A survey. In *Proc. of the 6th Ann. Conf. on Structure in Comp. Theory*, IEEE Comp. Soc. Press, June 1991.
8. O. Goldreich, N. Nisan, and A. Wigderson. On Yao's xor-lemma. Technical Report TR95-050, Electronic Colloquium on Computational Complexity, 1995.
9. J. Hastad. *Computational Limitations of Small-Depth Circuits.* MIT Press, 1987.
10. C. G. Jockusch. Semirecursive sets and positive reducibility. *Transactions of the AMS*, 131:420–436, May 1968.
11. M. Kummer. A proof of Beigel's cardinality conjecture. *Journal of Symbolic Logic*, 57(2):677–681, June 1992.
12. M. Kummer and F. Stephan. Effective search problems. *Math. Logic Quarterly*, 40, 1994.
13. L. Levin. One way fns and pseudorandom generators. *Combinatorica*, 7, 1987.
14. W. Miller and D. A. Martin. The degree of hyperimmune sets. *Zeitsch. f. math. Logik und Grundlagen d. Math.*, 14:159–166, 1968.
15. P. Odifreddi. *Classical Recursion Theory (Volume I).* North-Holland, 1989.
16. R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proc. of the 19th ACM Sym. on Theory of Comp.*, 1987.
17. R. I. Soare. *Recursively Enumerable Sets and Degrees.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1987.

This article was processed using the LaTeX macro package with LLNCS style