The Complexity of ODD_n^A

Richard Beigel*
University of Illinois at Chicago

William Gasarch†
University of Maryland

Martin Kummer‡
Universität Chemnitz-Zwickau

Georgia Martin§
Wheaton, Maryland

Timothy McNicholl¶
University of Dallas

Frank Stephan‖
Universität Heidelberg

*Research performed at Yale University, the University of Maryland, and Lehigh University. Address: Dept. of EECS (m/c 154), University of Illinois at Chicago, 851 S. Morgan St. - 1120 SEO, Chicago, IL 60607-7053, U.S.A. Supported in part by the National Science Foundation under grants CCR–8958528 and CCR–9415410. Also supported through the Human–Computer Interaction Laboratory by NASA grant NAG 52895. (Email: `beigel@uic.edu`.)

†Dept. of C.S. and Inst. for Adv. Comp. Stud., University of MD, College Park, MD 20742, U.S.A. Supported in part by NSF grants CCR–9301339 and CCR–9732692. (Email: `gasarch@cs.umd.edu`.)

‡Technische Universität Chemnitz-Zwickau, Fakultät für Informatik, Straße der Nationen 62, 09107 Chemnitz, Germany, EU. (Email: `martin.kummer@informatik.tu-chemnitz.de`.)

§12602 Goodhill Road, Wheaton, MD 20906, U.S.A.

¶Dept. of Math., University of Dallas, Irving, TX 75062, U.S.A. (Email: `tmcnicho@acad.udallas.edu`.)

‖Mathematisches Institut, Universität Heidelberg, Im Neuenheimer Feld 294, 69120 Heidelberg, Germany, EU. Supported by the Deutsche Forschungsgemeinschaft (DFG) grant Am 60/9-2. (Email: `fstephan@math.uni-heidelberg.de`.)

Abstract

For a fixed set A, the number of queries to A needed in order to decide a set S is a measure of S's complexity. We consider the complexity of certain sets defined in terms of A:

$$\mathrm{ODD}_n^A = \{(x_1, \ldots, x_n) : \#_n^A(x_1, \ldots, x_n) \text{ is odd}\}$$

and, for $m \geq 2$,

$$\mathrm{MOD}m_n^A = \{(x_1, \ldots, x_n) : \#_n^A(x_1, \ldots, x_n) \not\equiv 0 \pmod{m}\},$$

where $\#_n^A(x_1, \ldots, x_n) = A(x_1) + \cdots + A(x_n)$. (We identify $A(x)$ with $\chi_A(x)$, where χ_A is the characteristic function of A.)

If A is a nonrecursive semirecursive set or if A is a jump, we give tight bounds on the number of queries needed in order to decide ODD_n^A and $\mathrm{MOD}m_n^A$:

- ODD_n^A can be decided with n parallel queries to A, but not with $n-1$.
- ODD_n^A can be decided with $\lceil \log(n+1) \rceil$ sequential queries to A but not with $\lceil \log(n+1) \rceil - 1$.
- $\mathrm{MOD}m_n^A$ can be decided with $\lceil n/m \rceil + \lfloor n/m \rfloor$ parallel queries to A but not with $\lceil n/m \rceil + \lfloor n/m \rfloor - 1$.
- $\mathrm{MOD}m_n^A$ can be decided with $\lceil \log(\lceil n/m \rceil + \lfloor n/m \rfloor + 1) \rceil$ sequential queries to A but not with $\lceil \log(\lceil n/m \rceil + \lfloor n/m \rfloor + 1) \rceil - 1$.

The lower bounds above hold for nonrecursive r.e. sets A as well. (Interestingly, the lower bounds for r.e. sets follow by a general result from the lower bounds for semirecursive sets.)

In particular, every nonzero truth-table degree contains a set A such that ODD_n^A cannot be decided with $n-1$ parallel queries to A. Since every truth-table degree also contains a set B such that ODD_n^B can be decided with one query to B, a set's query complexity depends more on its structure than on its degree.

For a fixed set A,

$$\begin{aligned} \mathrm{Q}(n, A) &= \{S : S \text{ can be decided with } n \text{ sequential queries to } A\}, \\ \mathrm{Q}_{\|}(n, A) &= \{S : S \text{ can be decided with } n \text{ parallel queries to } A\}. \end{aligned}$$

We show that if A is semirecursive or r.e., but is not recursive, then these classes form non-collapsing hierarchies:

- $\mathrm{Q}(0, A) \subset \mathrm{Q}(1, A) \subset \mathrm{Q}(2, A) \subset \cdots$
- $\mathrm{Q}_{\|}(0, A) \subset \mathrm{Q}_{\|}(1, A) \subset \mathrm{Q}_{\|}(2, A) \subset \cdots$

The same is true if A is a jump.

1 Introduction

One paradigm in computational complexity theory is the classification of recursive functions according to their difficulty. Time is the most common complexity measure for recursive functions. In this paper we will consider functions that are recursive in some fixed non-recursive set A. One of the earliest measures considered for a function f recursive in A is the running time of the fastest algorithm with oracle A that computes f. However, we find that measure unsatisfying, because it is highly dependent on the computational model. Instead, we define the complexity of a function f (relative to A) as the minimum number of queries to A needed by an algorithm with oracle A that computes f. Thus we measure the hardness of f directly in terms of the hard aspect of its computation, namely accessing the nonrecursive oracle A.

Several natural examples of *functions* have been classified under the query complexity measure [BGGO93, BGK96a, Gas91, Kum92, KS94]. In this paper we investigate the complexity of the following *sets*:

Definition 1 Let A be a set of natural numbers. Let m and n be natural numbers such that $m \geq 2$ and $n \geq 1$.

- $\mathrm{ODD}_n^A = \{(x_1, \ldots, x_n) : \#_n^A(x_1, \ldots, x_n) \equiv 1 \pmod 2\}$
- $\mathrm{MOD}m_n^A = \{(x_1, \ldots, x_n) : \#_n^A(x_1, \ldots, x_n) \not\equiv 0 \pmod m\}$

where $\#_n^A(x_1, \ldots, x_n) = A(x_1) + \cdots + A(x_n)$. (We identify $A(x)$ with $\chi_A(x)$, where χ_A is the characteristic function of A.)

Sets like ODD_n^A and $\mathrm{MOD}m_n^A$ are clearly computable with one query to a good oracle, namely themselves. We do not want to measure the difficulty relative to an arbitrary oracle, but rather relative to a natural one: A itself. This idea has already been employed in various cases. For example, a "terse" set [BGGO93] is a set A such that, for every $n \geq 1$, computation of the n-fold characteristic function of A (i.e., the string-valued function $C_n^A(x_1, \ldots, x_n) = A(x_1) \cdots A(x_n)$) requires n queries to A. (Note that if A is any set such that, for every $n \geq 1$, ODD_n^A cannot be decided with $n-1$ queries to A, then A is terse.)

The set ODD_n^A is similar to the PARITY function, which has been well studied in the contexts of circuit complexity [FSS84, Yao85, BS90, Hås87, Smo87] and of pseudorandomness [GNW95, Lev87].

Clearly, ODD_n^A can be decided with n parallel queries to A. We show that if A is semirecursive or if A is Σ_i- or Π_i-complete for some i, then $n-1$ parallel queries do not suffice to decide ODD_n^A (unless A is recursive). Because every truth-table (tt) degree contains a semirecursive set, it follows that every nonzero tt-degree contains a set A such that ODD_n^A requires n parallel queries to A. On the other hand, it is known that every tt-degree contains a natural set A such that ODD_n^A can be decided with just one query to A. This contrast illustrates a theme in query complexity: a set's complexity depends more on its structure than on its degree.

We also obtain upper and lower bounds on the complexity of ODD_n^A using sequential queries, and on $\mathrm{MOD}m_n^A$ using both parallel and sequential queries. If A is nonrecursive, but semirecursive or r.e., the results on ODD_n^A imply that more queries to A allow you to decide more sets, i.e., the query-complexity hierarchies over nonrecursive semirecursive sets and over nonrecursive r.e. sets do not collapse. Interestingly, we prove a general implication from lower bounds for semirecursive sets to lower bounds for r.e. sets.

Although query complexity is a useful measure, the dependence on a fixed oracle may be unsatisfying. Since there is a semirecursive set in every tt-degree, it is also possible to define the "semirecursive query complexity" of a set S as the minimum, over all semirecursive sets B, of the number of queries to B needed in order to decide S. For each S, this complexity is a well-defined function of the input. However, because semirecursive sets can have arbitrarily high degree, you might suspect that every set has semirecursive query complexity 1, rendering the measure meaningless. In fact, if A is semirecursive then the semirecursive parallel-query complexity of ODD_n^A is always n, unless A is recursive. Thus, semirecursive query complexity depends on the structure of the problem and of the oracle, but not on the degree of the problem or of the oracle.

2 Notation, Definitions, and Useful Facts

Throughout this paper, all lower-case italic letters denote natural numbers, M denotes a Turing machine, and all other upper-case italic letters denote subsets of N. $\{A_i\}_{i\in\mathsf{N}}$ denotes the infinite sequence $A_0, A_1, \ldots$. Let $\langle\rangle$ be a computable, 1-1, onto mapping from the set of all sequences $x_1, \ldots, x_n$ such that $n \geq 1$ into N.

We use standard notation from recursion theory [Odi89, Soa87]. Let $M_0, M_1, \ldots$ be an effective list of all Turing machines, and let $M_0^{()}, M_1^{()}, \ldots$ be an effective list of all oracle Turing machines. Let W_e denote the domain of M_e. Hence $W_0, W_1, \ldots$ is an effective list of all r.e. sets of natural numbers. We sometimes write W_e to denote the corresponding set of strings. We write $M_e(x)\downarrow$ if $M_e(x)$ converges, $M_e(x)\downarrow= b$ if $M_e(x)$ converges and the output is b, $M_e(x)\uparrow$ if $M_e(x)$ diverges, $M_{e,s}(x)\downarrow$ if $M_e(x)$ converges within s steps, and $M_{e,s}(x)\uparrow$ if $M_e(x)$ does not converge within s steps.

Define A' to be $\{e : M_e^A(e)\downarrow\}$, i.e., A' is the halting problem relative to A. Define $A^{(0)} = A$, and $A^{(i+1)} = \left(A^{(i)}\right)'$. For all $i \geq 1$, the set $\emptyset^{(i)}$ is Σ_i-complete, and the set $\overline{\emptyset^{(i)}}$ is Π_i-complete. Note that $K \equiv_{\mathrm{m}} \emptyset'$.

Definition 2 (Join) $A \oplus B = \{2x : x \in A\} \cup \{2x+1 : x \in B\}$.

When an oracle Turing machine with oracle $A \oplus B$ queries an even number, we say that it queries A; when it queries an odd number, we say that it queries B.

Definition 3 Let A be r.e. in Z.

1. $\{A_s\}_{s\in\mathsf{N}}$ is a *recursive-in-Z enumeration* of A if
   - $A = \bigcup_{s\in\mathsf{N}} A_s$,
   - $A_0 = \emptyset$,
   - $(\forall s)[A_s$ is finite$]$,
   - $(\forall s)[A_s \subseteq A_{s+1}]$, and
   - the function f defined by $f(x,s) = A_s(x)$ is recursive in Z.

2. A *recursive enumeration* of A is a recursive-in-$\emptyset$ enumeration of A.

We define classes of functions that can be computed with a bound on the number of queries to an oracle.

Definition 4 ([BGGO93]) $\mathrm{FQ}(n, A)$ is the collection of all total functions f such that f is recursive in A via an oracle Turing machine that makes at most n sequential (i.e., adaptive) queries to A. $\mathrm{FQ}_{||}(n, A)$ is the collection of all total functions f such that f is recursive in A via an oracle Turing machine that makes at most n parallel (i.e., nonadaptive) queries to A (as in a weak truth-table reduction).

Definition 5 ([BGGO93]) $\mathrm{FQ}^X(n, A)$ is the collection of all total functions f such that f is recursive in $A \oplus X$ via an oracle Turing machine that makes at most n sequential (i.e., adaptive) queries to A and an unlimited (though finite) number of queries to X. $\mathrm{FQ}_{||}^X(n, A)$ is the collection of all total functions f such that f is recursive in $A \oplus X$ via an oracle Turing machine that makes at most n parallel (i.e., nonadaptive) queries to A and an unlimited number of queries to X. The queries to X can be made before, simultaneously with, or after the queries to A.

Correspondingly, we define classes of sets that can be decided with a bound on the number of queries.

Definition 6

- $B \in \mathrm{Q}(n, A)$ if $\chi_B \in \mathrm{FQ}(n, A)$.
- $B \in \mathrm{Q}_{||}(n, A)$ if $\chi_B \in \mathrm{FQ}_{||}(n, A)$.
- $B \in \mathrm{Q}^X(n, A)$ if $\chi_B \in \mathrm{FQ}^X(n, A)$.
- $B \in \mathrm{Q}_{||}^X(n, A)$ if $\chi_B \in \mathrm{FQ}_{||}^X(n, A)$.

If the oracle is a function g rather than a set A, complexity classes $\mathrm{FQ}(n, g)$, $\mathrm{FQ}_{||}(n, g)$, $\mathrm{FQ}^X(n, g)$, $\mathrm{FQ}_{||}^X(n, g)$, $\mathrm{Q}(n, g)$, $\mathrm{Q}_{||}(n, g)$, $\mathrm{Q}^X(n, g)$, and $\mathrm{Q}_{||}^X(n, g)$ are defined similarly to $\mathrm{FQ}(n, A)$ etc. For a class of sets $\mathcal{C}$, we define $\mathrm{FQ}(n, \mathcal{C}) = \bigcup_{A \in \mathcal{C}} \mathrm{FQ}(n, A)$, and we define $\mathrm{FQ}_{||}(n, \mathcal{C})$ etc. similarly.

We define sets whose query-complexity hierarchy does not collapse.

Definition 7

- A is *supportive* if $(\forall n)[\mathrm{Q}(n, A) \subset \mathrm{Q}(n + 1, A)]$.
- A is *parallel supportive* if $(\forall n)[\mathrm{Q}_{||}(n, A) \subset \mathrm{Q}_{||}(n + 1, A)]$.

Fact 8 *Let* $n \geq 1$.

- $\mathrm{Q}(i, A) = \mathrm{Q}(i, \overline{A})$
- $\mathrm{Q}_{||}(i, A) = \mathrm{Q}_{||}(i, \overline{A})$

- $\mathrm{ODD}_n^A \in \mathrm{Q}(i, A) \iff \mathrm{ODD}_n^{\overline{A}} \in \mathrm{Q}(i, A)$
- $\mathrm{ODD}_n^A \in \mathrm{Q}_{\|}(i, A) \iff \mathrm{ODD}_n^{\overline{A}} \in \mathrm{Q}_{\|}(i, A)$
- *A is supportive iff $\overline{A}$ is supportive.*
- *A is parallel supportive iff $\overline{A}$ is parallel supportive.*

Definition 9 Let $n \geq 1$. A function f is *n-enumerable* (denoted $f \in \mathrm{EN}(n)$) if there exists a recursive function g such that, for all x, $|W_{g(x)}| \leq n$ and $f(x) \in W_{g(x)}$.

Suppose that $f \in \mathrm{EN}(n)$. Then, given x, we can enumerate over time at most n possibilities for $f(x)$, one of which is correct, although we might never know that we have enumerated all the possibilities. (This concept first appeared in a recursion-theoretic framework in [Bei87b]. Some very general theorems about $\mathrm{EN}(n)$ were proved in [KS94]. The name "enumerable" was coined in [CH89].)

Definition 10 ([BGGO93]) Let $n \geq 1$. C_n^A is the n-fold characteristic function of A, i.e., the string-valued function defined by $\mathrm{C}_n^A(x_1, \ldots, x_n) = A(x_1) \cdots A(x_n)$.

(In the definition above, C stands for characteristic. In [BGGO93] and most of the literature, C_n^A is denoted F_n^A; however, the notation C_n^A is used in a recent book on bounded queries [GM99].) Note that $\mathrm{FQ}_{\|}(n, A) = \mathrm{FQ}(1, \mathrm{C}_n^A)$.

Theorem 11 ([BGGO93]) *Let f be a function.*

$$(\exists X)[f \in \mathrm{FQ}(n, X)] \iff f \in \mathrm{EN}(2^n).$$

Theorem 12 *Let $n \geq 1$.*

1. **(Nonspeedup [Bei87b, BGGO93])** *If $\mathrm{C}_n^A \in \mathrm{EN}(n)$, then A is recursive.*
2. **(Cardinality [Kum92])** *If $\#_n^A \in \mathrm{EN}(n)$, then A is recursive.*

An easy corollary to the Nonspeedup theorem shows that extra queries allow you to compute more functions.

Corollary 13 ([Bei87a])

1. *If $\mathrm{FQ}(n+1, A) = \mathrm{FQ}(n, A)$, then A is recursive.*
2. *If $\mathrm{FQ}_{\|}(n+1, A) = \mathrm{FQ}_{\|}(n, A)$, then A is recursive.*

Proof: 1. Assume that $\mathrm{FQ}(n+1, A) = \mathrm{FQ}(n, A)$. Then $\mathrm{C}_{n+1}^A \in \mathrm{FQ}(n, A)$, since $\mathrm{C}_{n+1}^A \in \mathrm{FQ}(n+1, A)$. By an easy induction, $\mathrm{C}_k^A \in \mathrm{FQ}(n, A)$ for all $k \geq 1$. In particular, $\mathrm{C}_{2^n}^A \in \mathrm{FQ}(n, A) \subseteq \mathrm{EN}(2^n)$ by Theorem 11. By Theorem 12.1, A is recursive.
2. Same as 1. Just change $\mathrm{FQ}(\cdot, A)$ to $\mathrm{FQ}_{\|}(\cdot, A)$ throughout. ▮

In contrast, we will see in Section 8 that extra queries do not always allow you to decide more sets.

We will prove several results about semirecursive sets. Semirecursive sets are abundant: there is a semirecursive set in every tt-degree, and there is an r.e., semirecursive set in every r.e. tt-degree. Furthermore, some deep recursion-theoretic techniques show that query-complexity lower bounds concerning semirecursive sets carry over automatically to r.e. sets.

Definition 14 ([Joc68])

1. A linear ordering $\sqsubseteq$ on N is *recursive in* X if the set of ordered pairs $\{(x, y) : x \sqsubseteq y\}$ is recursive in X.

2. A is *semirecursive in* X if there exists a linear ordering $\sqsubseteq$ on N such that $\sqsubseteq$ is recursive in X and A is closed downward under $\sqsubseteq$.

3. A is *semirecursive* if A is semirecursive in $\emptyset$.

Note that A is semirecursive in X iff $\overline{A}$ is semirecursive in X.

Lemma 15 ([Joc68]) *A is semirecursive in X iff there exists $f \leq_{\mathrm{T}} X$ such that, for all x, y,*

- *$f(x, y) \in \{x, y\}$, and*
- *$A \cap \{x, y\} \neq \emptyset \Rightarrow f(x, y) \in A$.*

Theorem 16 ([Joc68])

1. *Every tt-degree contains a semirecursive set.*

2. *Every r.e. tt-degree contains an r.e., semirecursive set.*

Definition 17 Fix a computable, 1-1, onto mapping $\langle\langle\rangle\rangle$ from the set of all sequences $x_1, \ldots, x_n, j$ such that $1 \leq j \leq n$ into N.

$$\mathrm{GE}^A = \{\langle\langle x_1, \ldots, x_n, j\rangle\rangle : \#_n^A(x_1, \ldots, x_n) \geq j\}.$$

Definition 18 ([Hay78]) A is a *p-cylinder* if $A \times A \leq_{\mathrm{m}} A$ and $\overline{A} \times \overline{A} \leq_{\mathrm{m}} \overline{A}$.

(In the definition above, p stands for positive.)

Fact 19 *Let $n \geq 1$.*

1. **[Hay78]** *A is a p-cylinder iff the set $\{B : B \leq_{\mathrm{m}} A\}$ is closed under positive truth-table reductions.*

2. *A is a p-cylinder iff $\mathrm{GE}^A \leq_{\mathrm{m}} A$.*

3. *If A is a p-cylinder then $\#_{2^n-1}^A \in \mathrm{FQ}(n, A)$.*

4. *If A is semirecursive, $A = B'$, or $A = \overline{B'}$, then A is a p-cylinder.*

Proof: 1. Every positive truth table can be written as an OR of ANDs.
2. GE^A is positive truth-table reducible to A.
3. Use binary search with GE^A as an oracle.
4. If A is semirecursive via $\sqsubseteq$, define min and max according to the order $\sqsubseteq$. Then $(x, y) \in A \times A$ iff $\max(x, y) \in A$, and $(x, y) \in \overline{A} \times \overline{A}$ iff $\min(x, y) \in \overline{A}$. The conclusion for jumps is easy and well known. ∎

The following is practically folklore in the bounded-query community.

Fact 20 ([Bei87b],[GM99, Page 84]) *Let A be semirecursive.*

1. $(\forall n \geq 1)[\mathrm{C}_n^A \in \mathrm{FQ}(1, \#_n^A) \subseteq \mathrm{FQ}(\lceil \log(n+1) \rceil, A)]$.

2. *If A is not recursive, then* $(\forall n \geq 1)[\#_n^A \notin \mathrm{FQ}_{||}(n-1, A)]$.

3. $(\forall n \geq 1)[\mathrm{C}_n^A \in \mathrm{EN}(n+1)]$.

Proof: Let A be semirecursive via $\sqsubseteq$.

1. Without loss of generality, assume that $x_1 \sqsubseteq \cdots \sqsubseteq x_n$. If $\#_n^A(x_1, \ldots, x_n) = i$, then $\mathrm{C}_n^A(x_1, \ldots, x_n) = 1^i 0^{n-i}$. By Facts 19.3 and 19.4, $\#_n^A \in \mathrm{FQ}(\lceil \log(n+1) \rceil, A)$.

2. Since A is not recursive, $\mathrm{C}_n^A \notin \mathrm{FQ}_{||}(n-1, A)$, by Corollary 13.2; therefore, $\#_n^A \notin \mathrm{FQ}_{||}(n-1, A)$, by part 1.

3. For $x_1, \ldots, x_n$ with $x_1 \sqsubseteq \cdots \sqsubseteq x_n$, enumerate the strings 0^n, 10^{n-1}, $\ldots$, $1^{n-1}0$, 1^n as possibilities for $\mathrm{C}_n^A(x_1, \ldots, x_n)$. ∎

Let REC denote the class of recursive sets, RE denote the class of r.e. sets, and SEMI denote the class of semirecursive sets. Let REC^X, RE^X, and SEMI^X denote those classes relativized to the oracle X.

3 Summary of Results

Let i, n, m be integers such that $m \geq 2$, $n \geq 1$, and $i \geq 1$. Let A be a Σ_i- or Π_i-complete set or a nonrecursive semirecursive set. Then

1. $\mathrm{ODD}_n^A \in \mathrm{Q}_{||}(n, A) - \mathrm{Q}_{||}(n-1, A)$.

2. $\mathrm{ODD}_n^A \in \mathrm{Q}(\lceil \log{(n+1)} \rceil, A) - \mathrm{Q}(\lceil \log{(n+1)} \rceil - 1, A)$.

3. $\mathrm{MOD}m_n^A \in \mathrm{Q}_{||}(\lceil n/m \rceil + \lfloor n/m \rfloor, A) - \mathrm{Q}_{||}(\lceil n/m \rceil + \lfloor n/m \rfloor - 1, A)$.

4. $\mathrm{MOD}m_n^A \in \mathrm{Q}(\lceil \log(\lceil n/m \rceil + \lfloor n/m \rfloor + 1) \rceil, A) - \mathrm{Q}(\lceil \log(\lceil n/m \rceil + \lfloor n/m \rfloor + 1) \rceil - 1, A)$.

5. A is supportive and parallel supportive.

Statement 5 and the lower bounds in statements 1–4 above hold as well for nonrecursive r.e. sets. But the upper bounds in 2–4 do not, because every nonzero r.e. T-degree contains an r.e. set B such that, for all $n \geq 1$, $\mathrm{ODD}_n^B \notin \mathrm{Q}(n-1, B)$. A proof of this result appears in [GM99, Section 4.8.3]; this proof can easily be modified to show that every nonzero r.e. T-degree contains an r.e. set B such that, for all $m \geq 2$ and $n \geq 1$, $\mathrm{MOD}m_n^B \notin \mathrm{Q}(n-1, B)$.

By statement 1 above and Theorem 16, every nonzero tt-degree contains a set A such that $\mathrm{ODD}_n^A \notin \mathrm{Q}_{||}(n-1, A)$ for all $n \geq 1$. On the other hand, it is known that every tt-degree contains a natural set A such that $\mathrm{ODD}_n^A \in \mathrm{Q}(1, A)$ for all $n \geq 1$, and we will present other examples of natural sets with this property (but not in every degree).

4 The Complexity of ODD_n^A for Semirecursive A

Let A be a nonrecursive semirecursive set. We know that, for all $n \geq 1$, ODD_n^A can be decided with n parallel queries or $\lceil \log (n+1) \rceil$ sequential queries to A. In this section, we prove that both of those bounds are tight, even if we allow queries to an arbitrary semirecursive set instead of A. This legitimizes the definition of semirecursive query complexity that we mentioned in Section 1.

Lemma 21 *Let $n \geq 1$. If A is a p-cylinder then $\#_{2n+1}^A$ can be computed with two parallel queries, one to $\#_{n+1}^A$ and one to ODD_n^A.*

Proof: We show how to compute $\#_{2n+1}^A(\vec{z})$. Since A is a p-cylinder, $\mathrm{GE}^A \leq_{\mathrm{m}} A$ by Fact 19.2. Therefore, for every i with $1 \leq i \leq 2n+1$, we can compute x_i such that $x_i \in A \iff \#_{2n+1}^A(\vec{z}) \geq i$. Thus $\#_{2n+1}^A(x_1, \ldots, x_{2n+1}) = \#_{2n+1}^A(\vec{z})$ and $A(x_1) \geq \cdots \geq A(x_{2n+1})$, so

$$\left\lceil \tfrac{1}{2} \#_{2n+1}^A(x_1, x_2, x_3, \ldots, x_{2n+1}) \right\rceil = \#_{n+1}^A(x_1, x_3, x_5, \ldots, x_{2n+1}).$$

By the definition of ODD_{2n+1}^A,

$$\begin{aligned} \#_{2n+1}^A(x_1, x_2, x_3, \ldots, x_{2n+1}) \bmod 2 &= \mathrm{ODD}_{2n+1}^A(x_1, x_2, x_3, \ldots, x_{2n+1}) \\ &= \mathrm{ODD}_{n+1}^A(x_1, x_3, x_5, \ldots, x_{2n+1}) \oplus \mathrm{ODD}_n^A(x_2, x_4, x_6, \ldots, x_{2n}), \end{aligned}$$

where $\oplus$ denotes addition modulo 2, not the join of two sets. Therefore,

$$\begin{aligned} &\#_{2n+1}^A(x_1, x_2, x_3, \ldots, x_{2n+1}) \\ &= \ 2\#_{n+1}^A(x_1, x_3, x_5, \ldots, x_{2n+1}) - (\mathrm{ODD}_{n+1}^A(x_1, x_3, x_5, \ldots, x_{2n+1}) \oplus \mathrm{ODD}_n^A(x_2, x_4, x_6, \ldots, x_{2n})). \end{aligned}$$

Thus $\#_{2n+1}^A$ can be computed with two parallel queries, one to $\#_{n+1}^A$ and one to ODD_n^A, completing the proof. ∎

The following definition and lemma are needed to prove the lower bound stated in Theorem 25, which is then used to establish lower bounds for semirecursive sets through Theorems 26 and 27.

Definition 22 Let $m \in \mathsf{N}$ and $B \subseteq \mathsf{N}$.

1. A function f is in $\mathrm{TQ}(m, B)$ if there exist an oracle Turing machine $M^{()}$ and a recursive function g such that
   - f is computed by M^B,
   - for every $x \in \mathrm{domain}(f)$, the set of queries made in the $M^B(x)$ computation is contained in $W_{g(x)}$, and
   - $|W_{g(x)}| \leq m$.
2. A set S is in $\mathrm{TQ}(m, B)$ if $\chi_S \in \mathrm{TQ}(m, B)$.

The following facts about $\mathrm{TQ}(m, B)$ are left as easy exercises for the reader.

Fact 23 *Let $k, p \geq 1$, let $m, m_1, \ldots, m_p \in \mathsf{N}$, and let $A, B \subseteq \mathsf{N}$.*

1. $\mathrm{FQ}_{\|}(m, B) \subseteq \mathrm{TQ}(m, B) \subseteq \mathrm{FQ}(m, B)$.
2. $\mathrm{FQ}(m, B) \subseteq \mathrm{TQ}(2^m - 1, B)$.
3. *Let g be a p-ary recursive function, and let $f, f_1, \ldots, f_p$ be total functions such that*
   - *for every x, $f(x) = g(f_1(x), \ldots, f_p(x))$, and*
   - *for every i with $1 \leq i \leq p$, $f_i \in \mathrm{TQ}(m_i, B)$.*

   Then $g \in \mathrm{TQ}(\sum_{i=1}^{p} m_i, B)$.
4. *If $A \in \mathrm{TQ}(m, B)$ then, for all $n \geq 1$, $\mathrm{C}_n^A \in \mathrm{TQ}(nm, B)$ and $\#_n^A \in \mathrm{TQ}(nm, B)$.*
5. *If $\mathrm{ODD}_k^A \in \mathrm{TQ}(m, B)$ then, for all $n \geq 1$,*
   (a) $\mathrm{ODD}_{nk}^A \in \mathrm{TQ}(nm, B)$, *and*
   (b) $n < k \Rightarrow \mathrm{ODD}_n^A \in \mathrm{TQ}(m, B)$.

Lemma 24 *Let $m \in \mathsf{N}$. If $\mathrm{C}_2^B \in \mathrm{EN}(3)$, then $\mathrm{TQ}(m, B) \subseteq \mathrm{EN}(m+1)$.*

Proof: Suppose that $\mathrm{C}_2^B \in \mathrm{EN}(3)$ via h, and let $f \in \mathrm{TQ}(m, B)$ via $M^{()}$ and g. Let $x \in \mathrm{domain}(f)$. We describe a partial recursive function E_x such that

- $\mathrm{domain}(E_x) \subseteq \{0,1\}^{\leq m}$,
- $f(x) \in \mathrm{image}(E_x)$, and
- $|\mathrm{image}(E_x)| \leq m + 1$.

The process of going from x to the code for E_x will be recursive; hence, using E_x, one can easily show that $f \in \mathrm{EN}(m+1)$.

ALGORITHM For E_x

1. Input $\sigma \in \{0,1\}^{\leq m}$. Let $p = |\sigma|$, and let $b_1, \ldots, b_p \in \{0,1\}$ such that $\sigma = b_1 \cdots b_p$.
2. Enumerate $W_{g(x)}$ until p elements appear. (If $|W_{g(x)}| < p$, then $E_x(\sigma)\uparrow$.)
3. Let $y_1, \ldots, y_p$ be the first p elements of $W_{g(x)}$, in order of enumeration. For all i, j such that $1 \leq i < j \leq p$, enumerate $W_{h(y_i, y_j)}$ until $b_i b_j$ appears. (If for some i, j this never occurs, then $E_x(\sigma)\uparrow$.)
4. (We now know that the statement $\mathrm{C}_p^B(y_1, \ldots, y_p) = b_1 \cdots b_p$ is consistent with the fact that $\mathrm{C}_2^B \in \mathrm{EN}(3)$ via h.) Simulate the computation $M^{()}(x)$. For every i with $1 \leq i \leq p$, if y_i is queried, answer with b_i. If a number $z \notin \{y_1, \ldots, y_p\}$ is queried, diverge.

5. If the simulation in step 4 terminates, then output its answer and halt.

END OF ALGORITHM

Let $W_{g(x)} = \{y_1, y_2, \ldots\}$, a finite set of size $\leq m$. For every $\sigma \in \{0,1\}^{\leq m}$, the computation of $E_x(\sigma)$ is guessing that $\mathrm{C}^B_{|\sigma|}(y_1, \ldots, y_{|\sigma|}) = \sigma$. Note that the different σ's are coordinated, in that they all use their i^{th} bit as a guess for $B(y_i)$ (unless $i > |\sigma|$). Also note that if $E_x(\sigma)\downarrow$ and $\sigma \prec \tau$, then either $E_x(\tau)\downarrow = E_x(\sigma)$ or $E_x(\tau)\uparrow$.

Clearly, $f(x) \in \text{image}(E_x)$. We show that $|\text{image}(E_x)| \leq m+1$. Let

$$C = \{\sigma \in \{0,1\}^{\leq m} : E_x(\sigma)\downarrow \wedge (\forall \sigma' \prec \sigma)[E_x(\sigma')\uparrow]\}.$$

For every $k \leq m$, let

$$\begin{aligned} G_k = &\quad C \cap \{0,1\}^{\leq k} \\ H_k = &\quad \{\tau \in \{0,1\}^k : (\exists \sigma)[\sigma \in C \wedge \tau \prec \sigma]\} \\ C_k = &\quad G_k \cup H_k \end{aligned}$$

Clearly, the following hold.

- $|\text{image}(E_x)| \leq |C|$.
- $C_m = C$.
- $(\forall k \leq m)[G_k \cap H_k = \emptyset]$.
- $(\forall k < m)[G_k \subseteq G_{k+1}]$.

Furthermore, if $\sigma \in C_k$ and $1 \leq i < j \leq |\sigma|$, then $\sigma(i)\sigma(j) \in W_{h(y_i,y_j)}$. We show that $|C| \leq m+1$, which implies that $|\text{image}(E_x)| \leq m+1$.

Claim: If $1 \leq k < m$ and σ, τ are strings in C_{k-1} with $\sigma \neq \tau$ and $|\sigma| = |\tau| = k-1$, then $|C_k \cap \{\sigma 0, \sigma 1, \tau 0, \tau 1\}| \leq 3$. Hence there is at most one string in C_{k-1} with both 1-bit extensions in C_k. Since $H_{k-1} \subseteq C_{k-1}$, there is at most one string in H_{k-1} with both 1-bit extensions in C_k.

Proof of Claim: Suppose that σ and τ differ on the i^{th} bit, and that $\sigma(i) = 0$ and $\tau(i) = 1$. If $\sigma 0 \in C_k$, then $00 \in W_{h(y_i,y_k)}$. If $\sigma 1 \in C_k$, then $01 \in W_{h(y_i,y_k)}$. If $\tau 0 \in C_k$, then $10 \in W_{h(y_i,y_k)}$. If $\tau 1 \in C_k$, then $11 \in W_{h(y_i,y_k)}$. Since $|W_{h(y_i,y_k)}| \leq 3$, we know that $|C_k \cap \{\sigma 0, \sigma 1, \tau 0, \tau 1\}| \leq 3$.

End of Proof of Claim

We prove, by induction on $k \leq m$, that $|C_k| \leq k+1$. For $k = 0$, we have $C_0 = \{\lambda\}$, where λ is the empty string, so $|C_0| \leq 1$. Assume that $k \geq 1$, and that $|C_{k-1}| \leq k$. Note that

$$|C_k| = |G_k - G_{k-1}| + |H_k| + |G_{k-1}|.$$

Every string in $(G_k - G_{k-1}) \cup H_k$ is a 1-bit extension of some string in H_{k-1}. Since, by the claim above, at most one string in H_{k-1} has both 1-bit extensions in C_k, we have $|G_k - G_{k-1}| + |H_k| \leq |H_{k-1}| + 1$. Hence

$$|C_k| = |G_k - G_{k-1}| + |H_k| + |G_{k-1}| \leq |H_{k-1}| + 1 + |G_{k-1}| = |C_{k-1}| + 1 \leq k+1.$$

Since $|C_m| \leq m+1$, we have $|\text{image}(E_x)| \leq m+1$. ∎

Theorem 25 *Let A be a p-cylinder, and let B be such that $\mathrm{C}_2^B \in \mathrm{EN}(3)$. Assume there exists $k \geq 1$ such that $\mathrm{ODD}_k^A \in \mathrm{TQ}(k-1, B)$. Then A is recursive.*

Proof:

We show that there exists $n_0 \geq 1$ such that $\#_{n_0}^A \in \mathrm{TQ}(n_0 - 1, B)$. By Lemma 24, $\mathrm{TQ}(n_0 - 1, B) \subseteq \mathrm{EN}(n_0)$; hence we will have $\#_{n_0}^A \in \mathrm{EN}(n_0)$. By Theorem 12.2, A is recursive.

For $n \in \mathsf{N}$, define $T(n)$ to be the least m such that $\#_{2^n+1}^A \in \mathrm{TQ}(m, B)$. Since $\mathrm{ODD}_k^A \in \mathrm{TQ}(k-1, B)$, we have $\mathrm{ODD}_{\lceil \frac{2^k}{k} \rceil (k)}^A \in \mathrm{TQ}(\lceil \frac{2^k}{k} \rceil (k-1), B)$ by Fact 23.5. Now $\lceil \frac{2^k}{k} \rceil (k) \geq 2^k$, and it can be shown that $\lceil \frac{2^k}{k} \rceil (k-1) \leq 2^k - 1$. Combining these inequalities with Fact 23.5, we have that $\mathrm{ODD}_{2^k}^A \in \mathrm{TQ}(2^k - 1, B)$ and $A \in \mathrm{TQ}(2^k - 1, B)$. By Fact 23.4, for every n, $\#_{2^n+1}^A \in \mathrm{TQ}((2^n + 1)(2^k - 1), B)$, so $T(n) \leq (2^n + 1)(2^k - 1)$. In particular, $T(k) \leq (2^k + 1)(2^k - 1) = 2^{2k} - 1$.

By Lemma 21, for $n \geq 1$, $\#_{2^n+1}^A$ is computable with two parallel queries, one to $\#_{2^{n-1}+1}^A$ and one to $\mathrm{ODD}_{2^{n-1}}^A$. Since $\mathrm{ODD}_{2^k}^A \in \mathrm{TQ}(2^k - 1, B)$, by Fact 23.5 we have that, for $n > k$, $\mathrm{ODD}_{2^{n-1}}^A \in \mathrm{TQ}((2^{n-1}/2^k)(2^k - 1), B)$; hence, by Fact 23.3,

$$T(n) \leq T(n-1) + (2^{n-1}/2^k)(2^k - 1).$$

By a simple induction we have that, for $n \geq k$,

$$T(n) \leq 2^n(1 - 1/2^k) + 2^{2k} - 2^k.$$

In particular, $T(3k) \leq 2^{3k} - 2^k < 2^{3k}$, so $\#_{2^{3k}+1}^A \in \mathrm{TQ}(2^{3k}, B)$. Hence it suffices to let $n_0 = 2^{3k} + 1$. ∎

Theorem 26 *Let A be a p-cylinder, and let B be such that $\mathrm{C}_2^B \in \mathrm{EN}(3)$.*

1. *If $(\exists k \geq 1)[\mathrm{ODD}_k^A \in \mathrm{Q}_{||}(k-1, B)]$, then A is recursive.*

2. *If $(\exists k \geq 1)[\mathrm{ODD}_{2^k}^A \in \mathrm{Q}(k, B)]$, then A is recursive.*

Proof:

1) If $\mathrm{ODD}_k^A \in \mathrm{Q}_{||}(k-1, B)$ then, by Fact 23.1, $\mathrm{ODD}_k^A \in \mathrm{TQ}(k-1, B)$. By Theorem 25, A is recursive.

2) If $\mathrm{ODD}_{2^k}^A \in \mathrm{Q}(k, B)$ then, by Fact 23.2, $\mathrm{ODD}_{2^k}^A \in \mathrm{TQ}(2^k - 1, B)$. By Theorem 25, A is recursive.

∎

Theorem 27 *If A is a nonrecursive semirecursive set, then for all $n \geq 1$,*

$$\mathrm{ODD}_n^A \in \mathrm{Q}_{||}(n, A) - \mathrm{Q}_{||}(n-1, \mathrm{SEMI})$$

and

$$\mathrm{ODD}_n^A \in \mathrm{Q}(\lceil \log (n+1) \rceil, A) - \mathrm{Q}(\lceil \log (n+1) \rceil - 1, \mathrm{SEMI}).$$

In particular, every nonzero tt-degree contains a set A with this property, and every nonzero r.e. tt-degree contains an r.e. set A with this property.

Proof: Let A be a nonrecursive semirecursive set. The nontrivial upper bound follows from Fact 20.1. The lower bounds follow from Fact 20.3, Fact 19.4, and Theorem 26.

The existence of such sets in every nonzero tt-degree (and of such sets in every nonzero r.e. tt-degree which are also r.e.) follows from Theorem 16. ∎

5 The Complexity of ODD_n^A for R.E. A

In this section we will prove a deep connection between recursive enumerability and semirecursiveness. The key to our results is the existence of a set X such that (1) every r.e. set is semirecursive in X and (2) every r.e. set that is recursive in X is in fact recursive. Our proofs are based on two important concepts from recursion theory: recursively bounded sets and extensive sets.[1]

Definition 28 ([Joc89]) A set X is *extensive* if, for every 0,1-valued partial recursive function g, there is a 0,1-valued total function $h \leq_{\mathrm{T}} X$ such that h extends g.

(The Turing degrees of the extensive sets are the same as the Turing degrees of the consistent extensions of Peano arithmetic [Joc89], but this is not important for our purposes.)

Lemma 29 *If A is r.e. and X is extensive, then A is semirecursive in X.*

Proof: Assume that A is r.e. and X is extensive. Let $\{A_s\}_{s\in\mathsf{N}}$ be a recursive enumeration of A. Define a 0,1-valued partial recursive function g 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 extensive, there is a 0,1-valued total function $h \leq_{\mathrm{T}} X$ such that h extends g. Let

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

Then $f \leq_{\mathrm{T}} X$, $(\forall x,y)[f(x,y) \in \{x,y\}]$, and

$$A \cap \{x,y\} \neq \emptyset \Rightarrow f(x,y) \in A,$$

so A is semirecursive in X by Lemma 15. ∎

Definition 30

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

[1] The degrees of recursively bounded sets and extensive sets have been well studied. The former are called *hyperimmune free* [MM68, Odi89, Soa87], the latter are called *DNR_2* [Joc89] or *PA* [Kuč85].

- A set X is *recursively bounded* (r.b.) if every total function f such that $f \leq_{\mathrm{T}} X$ is recursively dominated.[2]

(The r.b. sets are the same as the sets of hyperimmune-free degree [MM68].)

Lemma 31 ([MM68]) *Let X be r.b. If A is r.e. and $A \leq_{\mathrm{T}} X$, then A is recursive.*

Proof: Assume that A is r.e. and $A \leq_{\mathrm{T}} X$. Let $\{A_s\}_{s\in\mathsf{N}}$ be a recursive enumeration of A, and let

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

Then $f \leq_{\mathrm{T}} A \leq_{\mathrm{T}} X$. Since X is r.b., there is a recursive function g such that $(\forall x)[f(x) < g(x)]$. But then, for all x,

$$x \in A \iff x \in A_{g(x)},$$

so A is recursive. ∎

The following lemma is implicit in [JS72].

Lemma 32 *There exists a set that is r.b. and extensive.*

Proof sketch: There exists an infinite, recursive binary tree T such that every infinite branch of T is extensive: Given a finite string of bits σ,

$$\sigma \in T \iff (\forall e, x < |\sigma|)(\forall b \in \{0,1\})[(\langle e,x\rangle < |\sigma| \wedge M_{e,|\sigma|}(x) \downarrow= b) \Rightarrow \sigma(\langle e,x\rangle) = b].$$

A is an infinite branch of T unless there are $b \in \{0,1\}$ and e, s, x such that $e, x, \langle e,x\rangle < s$ and $M_{e,s}(x) \downarrow= b \neq A(\langle e,x\rangle)$, in which case the string $A(0)\cdots A(s-1)$ is not in T. By the Hyperimmune-Free Basis Theorem [JS72] (see also [Soa87, Page 109]), T has an infinite branch A which is of hyperimmune-free degree; such A is r.b. [MM68]. ∎

Lemma 33 *There exists X such that*

- $\mathrm{RE} \subseteq \mathrm{SEMI}^X$, *and*
- $\mathrm{RE} - \mathrm{REC} \subseteq \mathrm{SEMI}^X - \mathrm{REC}^X$.

Proof: Let X be an r.b., extensive set, which exists by Lemma 32. By Lemma 29, $\mathrm{RE} \subseteq \mathrm{SEMI}^X$. By Lemma 31, $\mathrm{RE} \cap \mathrm{REC}^X \subseteq \mathrm{REC}$, so $\mathrm{RE} - \mathrm{REC} \subseteq \overline{\mathrm{REC}^X}$. Thus $\mathrm{RE} - \mathrm{REC} \subseteq \mathrm{SEMI}^X - \mathrm{REC}^X$. ∎

The following lemma lets us turn lower bounds for semirecursive sets into lower bounds for r.e. sets.

Lemma 34 *For each set A, let f^A be a function.*

[2] The term *recursively bounded* as used here is not to be confused with the same term used elsewhere to define other entities. It is common (see [JS72], for example) to say that a tree T is recursively bounded if there is a recursive function f such that, for every node σ of T and every $x < |\sigma|$, $\sigma(x) \leq f(x)$. (Note that, by that definition, every binary tree is recursively bounded.)

1. $(\forall X)(\forall A \in \mathrm{SEMI}^X - \mathrm{REC}^X)[f^A \notin \mathrm{FQ}(k, \mathrm{SEMI}^X)] \Rightarrow$
   $(\forall A \in \mathrm{RE} - \mathrm{REC})[f^A \notin \mathrm{FQ}(k, \mathrm{RE})]$.
2. $(\forall X)(\forall A \in \mathrm{SEMI}^X - \mathrm{REC}^X)[f^A \notin \mathrm{FQ}_{||}(k, \mathrm{SEMI}^X)] \Rightarrow$
   $(\forall A \in \mathrm{RE} - \mathrm{REC})[f^A \notin \mathrm{FQ}_{||}(k, \mathrm{RE})]$.

Proof: 1. We prove the contrapositive. Assume that there exists A with

$$A \in \mathrm{RE} - \mathrm{REC} \text{ and } f^A \in \mathrm{FQ}(k, \mathrm{RE}).$$

By Lemma 33, there exists X such that $\mathrm{RE} - \mathrm{REC} \subseteq \mathrm{SEMI}^X - \mathrm{REC}^X$ and $\mathrm{RE} \subseteq \mathrm{SEMI}^X$. Then

$$A \in \mathrm{SEMI}^X - \mathrm{REC}^X \text{ and } f^A \in \mathrm{FQ}(k, \mathrm{SEMI}^X).$$

2. Similar. Replace $\mathrm{FQ}(k, \cdot)$ by $\mathrm{FQ}_{||}(k, \cdot)$ in the proof of part 1 above. ∎

Theorem 35 *Let A be a nonrecursive r.e. set.*

1. $(\forall n \geq 1)[\mathrm{ODD}_n^A \notin \mathrm{Q}_{||}(n-1, \mathrm{RE})]$.
2. $(\forall n \geq 1)[\mathrm{ODD}_{2^n}^A \notin \mathrm{Q}(n, \mathrm{RE})]$.

Proof: By the relativization of the lower bounds in Theorem 27 (which can be done by relativizing the proofs of Theorems 25 and 26 and the results used therein, including the Cardinality Theorem), we obtain the following for all X and all $A \in \mathrm{SEMI}^X - \mathrm{REC}^X$:

1. $(\forall n \geq 1)[\mathrm{ODD}_n^A \notin \mathrm{Q}_{||}^X(n-1, \mathrm{SEMI}^X)]$.
2. $(\forall n \geq 1)[\mathrm{ODD}_{2^n}^A \notin \mathrm{Q}^X(n, \mathrm{SEMI}^X)]$.

The result now follows by Lemma 34. ∎

For other applications of the Hyperimmune-Free Basis Theorem relevant to the study of query complexity, see [KS94, Section 6]. For a self-contained proof of Theorem 35 that does not use the Hyperimmune-Free Basis Theorem, see [GM99, Section 6.2.2].

6 The Complexity of ODD_n^A for Σ_i-complete A

The following theorem is obtained by relativizing the proof of Theorem 35.

Theorem 36 *Let A be r.e. in Z, but not recursive in Z.*

1. $(\forall n \geq 1)[\mathrm{ODD}_n^A \notin \mathrm{Q}_{||}^Z(n-1, \mathrm{RE}^Z)]$.
2. $(\forall n \geq 1)[\mathrm{ODD}_{2^n}^A \notin \mathrm{Q}^Z(n, \mathrm{RE}^Z)]$.

Theorem 37 *Let $A = Z'$ or $A = \overline{Z'}$.*

1. $(\forall n \geq 1)[\mathrm{ODD}_n^A \in \mathrm{Q}_{||}(n, A) - \mathrm{Q}_{||}^Z(n-1, A)]$.

2. $(\forall n \geq 1)[\mathrm{ODD}_n^A \in \mathrm{Q}(\lceil \log (n+1) \rceil, A) - \mathrm{Q}^Z(\lceil \log (n+1) \rceil - 1, A)]$.

Proof: The first upper bound is obvious. The second upper bound follows from Fact 19, parts 3 and 4. If $A = Z'$, then A is r.e. in Z but not recursive in Z, so in this case the lower bounds follow from Theorem 36. By Fact 8, the same lower bounds apply to $\overline{Z'}$ as to Z'. ∎

The following theorem is immediate.

Theorem 38 *Let A be Σ_i-complete or Π_i-complete for some $i \geq 1$.*

1. $(\forall n \geq 1)[\mathrm{ODD}_n^A \in \mathrm{Q}_{\|}(n, A) - \mathrm{Q}_{\|}(n-1, A)]$.
2. $(\forall n \geq 1)[\mathrm{ODD}_n^A \in \mathrm{Q}(\lceil \log (n+1) \rceil, A) - \mathrm{Q}(\lceil \log (n+1) \rceil - 1, A)]$.

Note: The special case $A = K$ appears in [BGH89].

7 The Complexity of $\mathrm{MOD}m_n^A$

For a broad class of sets A, including jumps and semirecursive sets, we show that $\mathrm{MOD}m_n^A$ is m-equivalent to ODD_k^A where $k = \lceil n/m \rceil + \lfloor n/m \rfloor$. Thus our bounds for ODD_k^A translate into bounds for $\mathrm{MOD}m_n^A$.

Lemma 39 *Let $m \geq 2$ and $n \geq 1$. If A is a p-cylinder then*

$$\mathrm{MOD}m_n^A \equiv_{\mathrm{m}} \mathrm{ODD}^A_{\lceil n/m \rceil + \lfloor n/m \rfloor}.$$

Proof: Since A is a p-cylinder, $\mathrm{GE}^A \leq_{\mathrm{m}} A$ by Fact 19.2. Let $k = \lceil n/m \rceil + \lfloor n/m \rfloor$.

First, we show that $\mathrm{MOD}m_n^A \leq_{\mathrm{m}} \mathrm{ODD}_k^A$. Consider an input $x_1, \ldots, x_n$. Since $\mathrm{GE}^A \leq_{\mathrm{m}} A$, we can compute $y_1, \ldots, y_n$ such that $y_i \in A \iff \#_n^A(x_1, \ldots, x_n) \geq i$. Then $\#_n^A(x_1, \ldots, x_n) = \#_n^A(y_1, \ldots, y_n)$ and $A(y_1) \geq \cdots \geq A(y_n)$, so

$$(x_1, \ldots, x_n) \in \mathrm{MOD}m_n^A \iff (y_1, y_m, y_{m+1}, y_{2m}, y_{2m+1}, y_{3m}, \ldots) \in \mathrm{ODD}_k^A.$$

(If m divides n, the last argument is y_n; otherwise, the last argument is $y_{\lfloor \frac{n}{m} \rfloor m+1}$.)

Second, we show that $\mathrm{ODD}_k^A \leq_{\mathrm{m}} \mathrm{MOD}m_n^A$. Consider an input $x_1, \ldots, x_k$. Since $\mathrm{GE}^A \leq_{\mathrm{m}} A$, we can compute $y_1, \ldots, y_k$ such that $y_i \in A \iff \#_k^A(x_1, \ldots, x_k) \geq i$. Then $\#_k^A(x_1, \ldots, x_k) = \#_k^A(y_1, \ldots, y_k)$ and $A(y_1) \geq \cdots \geq A(y_k)$, so

$$(x_1, \ldots, x_k) \in \mathrm{ODD}_k^A \iff (y_1, \ldots, y_1, y_2, y_3, \ldots, y_3, y_4, \ldots) \in \mathrm{MOD}m_n^A.$$

(The argument y_i appears $m-1$ times if i is odd and $i < k$. The argument y_k appears $n \bmod m$ times if k is odd.) ∎

The following theorem is immediate from the results in this paper for ODD_n^A.

Theorem 40

1. *If A is a nonrecursive semirecursive set, then for all $m \geq 2$ and $n \geq 1$,* $\mathrm{MOD}m_n^A \in \mathrm{Q}_{\|}(\lceil n/m \rceil + \lfloor n/m \rfloor, A) - \mathrm{Q}_{\|}(\lceil n/m \rceil + \lfloor n/m \rfloor - 1, \mathrm{SEMI})$ *and* $\mathrm{MOD}m_n^A \in \mathrm{Q}(\lceil \log{(\lceil n/m \rceil + \lfloor n/m \rfloor + 1)} \rceil, A) - \mathrm{Q}(\lceil \log{(\lceil n/m \rceil + \lfloor n/m \rfloor + 1)} \rceil - 1, \mathrm{SEMI})$. *In particular, every nonzero tt-degree contains a set A with this property, and every nonzero r.e. tt-degree contains an r.e. set A with this property.*

2. *If A is r.e. but not recursive, then for all $m \geq 2$ and $n \geq 1$,*

$$\mathrm{MOD}m_n^A \notin \mathrm{Q}_{\|}(\lceil n/m \rceil + \lfloor n/m \rfloor - 1, \mathrm{RE}) \cup \mathrm{Q}(\lceil \log{(\lceil n/m \rceil + \lfloor n/m \rfloor + 1)} \rceil - 1, \mathrm{RE}).$$

3. *If $A = B'$ or $A = \overline{B'}$ for some set B, then for all $m \geq 2$ and $n \geq 1$,* $\mathrm{MOD}m_n^A \in \mathrm{Q}_{\|}(\lceil n/m \rceil + \lfloor n/m \rfloor, A) - \mathrm{Q}_{\|}(\lceil n/m \rceil + \lfloor n/m \rfloor - 1, A)$ *and* $\mathrm{MOD}m_n^A \in \mathrm{Q}(\lceil \log{(\lceil n/m \rceil + \lfloor n/m \rfloor + 1)} \rceil, A) - \mathrm{Q}(\lceil \log{(\lceil n/m \rceil + \lfloor n/m \rfloor + 1)} \rceil - 1, A)$.

The conclusion of part 3 applies, in particular, to Σ_i- and Π_i-complete sets for $i \geq 1$.

8 Sets A with $\mathrm{ODD}_n^A \in \mathrm{Q}(1, A)$ and $\mathrm{MOD}m_n^A \in \mathrm{Q}(1, A)$

It is known [Rog67, Page 112] that every tt-degree contains a natural set A such that, for all B, $B \leq_{\mathrm{tt}} A \iff B \leq_{\mathrm{m}} A$. In particular, for all $m \geq 2$ and $n \geq 1$, ODD_n^A and $\mathrm{MOD}m_n^A$ are decidable with just one query to A. We construct other examples of natural sets A with these properties (but not in every degree).

Definition 41 ([Rog67, Page 112]) Fix an effective encoding for Boolean functions of finitely many variables.

$$\{B^{\mathrm{tt}} = \langle h, x_1, \ldots, x_n \rangle : n \geq 1,\ h \text{ is an } n\text{-ary Boolean function},\ h(B(x_1), \ldots, B(x_n)) = 1\}.$$

Proposition 42 ([Rog67, Page 112],[Odi89, Page 593]) *Every tt-degree contains a set A such that, for all $m \geq 2$ and $n \geq 1$,*

- $S \leq_{\mathrm{tt}} A \Rightarrow S \leq_{\mathrm{m}} A$,
- $\mathrm{ODD}_n^A \leq_{\mathrm{m}} A$, *and*
- $\mathrm{MOD}m_n^A \leq_{\mathrm{m}} A$.

Proof: Let $A = B^{\mathrm{tt}}$ for some B in the given degree. ∎

We now exhibit other natural sets A such that ODD_n^A and $\mathrm{MOD}m_n^A$ can be decided with just one query to A.

Definition 43 $B^{(\omega)} = \{\langle x, i \rangle : x \in B^{(i)}\}$.

(It is well known that the set $\emptyset^{(\omega)}$ is recursively isomorphic to the set of all true first-order statements of arithmetic, suitably encoded [Rog67, Page 318].)

Proposition 44 *Let $A = B^{(\omega)}$. Then, for all $m \geq 2$ and $n \geq 1$,*

1. $S \leq_{\rm wtt} A \Rightarrow S \leq_{\rm m} A$,
2. $\mathrm{ODD}_n^A \leq_{\rm m} A$,
3. $\mathrm{MOD}m_n^A \leq_{\rm m} A$, *and*
4. $\mathrm{Q}_{\|}(n, A) = \mathrm{Q}(1, A)$.

Proof: Part 1. Assume that $S \leq_{\rm wtt} B^{(\omega)}$. On input x, compute the list of queries, and determine the maximum i such that $\langle y, i\rangle$ is on the list of queries for some y. For this particular x, the remaining part of the computation is recursive in $B^{(i)}$, hence r.e. in $B^{(i)}$. Thus we can compute a single number z such that $S(x) = B^{(i+1)}(z) = B^{(\omega)}(\langle z, i+1\rangle)$. Therefore $S \leq_{\rm m} B^{(\omega)}$.

Parts 2–4 follow immediately from part 1. ▮

Note the contrast between Proposition 44 and Theorem 38. The following generalization of those two results is left as an exercise for readers familiar with recursive ordinals and the extension of the jump operator to them.

Proposition 45 *Let α be a recursive ordinal and $A = B^{(\alpha)}$.*

1. *If α is a successor ordinal, then* $(\forall n \geq 1)[\mathrm{ODD}_n^A \in \mathrm{Q}_{\|}(n, A) - \mathrm{Q}_{\|}(n-1, A)]$ *and* $(\forall n \geq 1)[\mathrm{ODD}_{2^n}^A \in \mathrm{Q}(n+1, A) - \mathrm{Q}(n, A)]$.
2. *If α is a limit ordinal, then* $(\forall n \geq 1)[\mathrm{ODD}_n^A \in \mathrm{Q}(1, A)]$.

A similar proposition holds for $\mathrm{MOD}m_n^A$.

9 Supportive Sets and ODD_n^A

Recall that A is supportive if $(\forall n)[\mathrm{Q}(n, A) \subset \mathrm{Q}(n+1, A)]$, and A is parallel supportive if $(\forall n)[\mathrm{Q}_{\|}(n, A) \subset \mathrm{Q}_{\|}(n+1, A)]$. By Theorems 27 and 37, jumps and nonrecursive semirecursive sets are supportive and parallel supportive. By Theorem 35, nonrecursive r.e. sets are parallel supportive. We will show that nonrecursive r.e. sets are supportive as well. As in the previous results, the separation is witnessed by ODD_k^A for some k, but unlike the previous results, we do not know which value of k.

Lemma 46 *If $(\forall n)(\exists k \geq 1)[\mathrm{ODD}_k^A \notin \mathrm{Q}(n, A)]$, then A is supportive and parallel supportive.*

Proof: Assume that $(\forall n)(\exists k \geq 1)[\mathrm{ODD}_k^A \notin \mathrm{Q}(n, A)]$. Then A is nonrecursive, so $A \in \mathrm{Q}(1, A) - \mathrm{Q}(0, A)$. Fix $n \geq 1$. The set $\{k \geq 1 : \mathrm{ODD}_k^A \notin \mathrm{Q}(n, A)\}$ is nonempty. Let m be its least element. Then $m > n \geq 1$ and $\mathrm{ODD}_{m-1}^A \in \mathrm{Q}(n, A)$, so

$$\mathrm{ODD}_m^A \in \mathrm{Q}(n+1, A) - \mathrm{Q}(n, A).$$

Thus A is supportive. The proof that A is parallel supportive is similar. ▮

Corollary 47

1. *Let A be nonrecursive. If A is semirecursive, r.e., or in the range of the jump operator, then A is supportive and parallel supportive.*

2. *Every nonzero tt-degree contains a set that is both supportive and parallel supportive.*

Proof: 1. By Theorems 27, 35, and 37, $(\forall n)(\exists k \geq 1)[\mathrm{ODD}_k^A \notin \mathrm{Q}(n, A)]$. Therefore, by Lemma 46, A is supportive and parallel supportive.

2. By Theorem 16, every tt-degree contains a semirecursive set, so we are done by part 1. ∎

The proofs of the following two theorems are left as exercises for readers familiar with random sets, 1-generic sets, and/or autoreducible sets.

Theorem 48 *If A is random or 1-generic, then $(\forall n \geq 1)[\mathrm{ODD}_n^A \in \mathrm{Q}_{\|}(n, A) - \mathrm{Q}(n-1, A)]$; hence A is both supportive and parallel supportive, by Lemma 46.*

Theorem 49 *If $(\exists n \geq 1)[\mathrm{ODD}_n^A \in \mathrm{Q}(n-1, A)]$, then A is autoreducible.*

Note 50 It is well known (see [Odi89, Page 588], for example) that if A is r.b. (i.e., of hyperimmune-free degree) then $(\forall S)[S \leq_{\mathrm{T}} A \Rightarrow S \leq_{\mathrm{tt}} A]$. By the proof of Proposition 42, $(\forall S)[S \leq_{\mathrm{T}} A^{\mathrm{tt}} \Rightarrow S \leq_{\mathrm{m}} A^{\mathrm{tt}}]$. Thus A^{tt} is neither supportive nor parallel supportive.

This fact, together with Corollary 47 and Theorem 48, might give the impression that a set is supportive iff it is parallel supportive. This is not the case, however, since K^{tt} and $\emptyset^{(\omega)}$ are examples of sets that are supportive but not parallel supportive. Proof sketches of these results appeared in [BGK+96b]; full proofs appear in [GM99, Section 8.2].

Some earlier work on supportive sets and the like appears in [Bei87a, Bei88].

References

[Bei87a] Richard Beigel. Functionally supportive sets. Technical Report 87–10, The Johns Hopkins University, Dept. of Computer Science, 1987.

[Bei87b] Richard Beigel. *Query-Limited Reducibilities.* PhD thesis, Stanford University, 1987. Also available as Report No. STAN-CS-88–1221.

[Bei88] Richard Beigel. When are $k+1$ queries better than k? Technical Report 88-06, The Johns Hopkins University, Dept. of Computer Science, 1988.

[BGGO93] Richard Beigel, William I. Gasarch, John T. Gill, and James C. Owings. Terse, superterse, and verbose sets. *Information and Computation*, 103(1):68–85, March 1993.

[BGH89] Richard Beigel, William I. Gasarch, and Louise Hay. Bounded query classes and the difference hierarchy. *Archive for Mathematical Logic*, 29(2):69–84, December 1989.

[BGK96a] Richard Beigel, William Gasarch, and Efim Kinber. Frequency computation and bounded queries. *Theoretical Computer Science*, 163:177–192, 1996.

[BGK+96b] Richard Beigel, William Gasarch, Martin Kummer, Georgia Martin, Timothy McNicholl, and Frank Stephan. On the query complexity of sets. In *21st International Symposium on Mathematical Foundations of Computer Science (MFCS '96), Cracow, Poland*, 1996.

[BS90] Ravi Boppana and Michael Sipser. The complexity of finite functions. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, pages 757–804. MIT Press and Elsevier, The Netherlands, 1990.

[CH89] Jin-yi Cai and Lane A. Hemachandra. Enumerative counting is hard. *Information and Computation*, 82(1):34–44, July 1989.

[FSS84] Merrick Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, April 1984.

[Gas91] William Gasarch. Bounded queries in recursion theory: A survey. In *Proc. of the 6th Annu. Conference on Structure in Complexity Theory*, pages 62–78. IEEE Computer Society Press, June 1991.

[GM99] William I. Gasarch and Georgia A. Martin. *Bounded Queries in Recursion Theory*. Birkhäuser, Boston, 1999.

[GNW95] Oded Goldreich, Noam Nisan, and Avi Wigderson. On Yao's XOR-lemma. Technical Report TR95-050, Electronic Colloquium on Computational Complexity, 1995.

[Hås87] Johan Håstad. *Computational Limitations of Small-Depth Circuits*. MIT Press, Cambridge, MA, 1987.

[Hay78] Louise Hay. Convex subsets of 2^n and bounded truth-table reducibility. *Discrete Mathematics*, 21(1):31–46, January 1978.

[Joc68] Carl G. Jockusch. Semirecursive sets and positive reducibility. *Transactions of the AMS*, 131:420–436, May 1968.

[Joc89] Carl G. Jockusch. Degrees of functions with no fixed points. In J.E. Fenstad, I. Frolov, and R. Hilpinen, editors, *Logic, Methodology, and Philosophy of Science VIII*, pages 191–201. North Holland, 1989.

[JS72] Carl G. Jockusch, Jr. and Robert I. Soare. Π^0_1 classes and degrees of theories. *Transactions of the AMS*, 173:33–56, 1972.

[KS94] Martin Kummer and Frank Stephan. Effective search problems. *Mathematical Logic Quarterly*, 40:224–236, 1994.

[Kuč85] Antonin Kučera. Measure of Π^0_1 classes and complete extensions of PA. In *Recursion Theory Week at Oberwolfach*, volume 1141 of *Lecture Notes in Mathematics*, pages 245–259, Berlin, 1985. Springer-Verlag.

[Kum92] Martin Kummer. A proof of Beigel's cardinality conjecture. *Journal of Symbolic Logic*, 57(2):677–681, June 1992.

[Lev87] Leonid A. Levin. One way functions and pseudorandom generators. *Combinatorica*, 7:357–363, 1987.

[MM68] Webb Miller and Donald A. Martin. The degree of hyperimmune sets. *Zeitsch. f. math. Logik und Grundlagen d. Math.*, 14:159–166, 1968.

[Odi89] Piergiorgio Odifreddi. *Classical Recursion Theory (Volume I)*. North-Holland, Amsterdam, 1989.

[Rog67] Hartley Rogers, Jr. *Theory of Recursive Functions and Effective Computability*. McGraw Hill, New York, 1967.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proc. of the 19th ACM Sym. on Theory of Computing*, pages 77–82, 1987.

[Soa87] Robert I. Soare. *Recursively Enumerable Sets and Degrees*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1987.

[Yao85] Andrew C. Yao. Separating the polynomial-time hierarchy by oracles. In *Proc. of the 26th IEEE Sym. on Found. of Comp. Sci.*, pages 1–10, 1985.