Terse, Superterse, and Verbose Sets[1] [2]

by

Richard Beigel[3]
Yale University

William I. Gasarch[4]
University of Maryland
Inst. for Adv. Comp. Studies

John Gill[5]
Stanford University

James C. Owings[6]
University of Maryland

This paper is dedicated to the memory of Louise E. Hay, 1935-1989.

ABSTRACT

Let A be a subset of the natural numbers, and let $F_n^A(x_1, \ldots, x_n) = \langle \chi_A(x_1), \ldots, \chi_A(x_n) \rangle$, where χ_A is the characteristic function of A. An oracle Turing machine with oracle A could certainly compute F_n^A with n queries to A. There are some sets A (e.g., the halting set) for which F_n^A can be computed with substantially fewer than n queries. One key reason for this is that the questions asked to the oracle can depend on previous answers, i.e., the questions are *adaptive*. We examine when it is possible to save queries. A set A is *terse* if the computation of F_n^A from A *requires* n queries. A set A is *superterse* if the computation of F_n^A from *any* set requires n queries. A set A is *verbose* if $F_{2^n-1}^A$ can be computed with n queries to A. The range of possible query savings is limited by the following theorem: F_n^A cannot be computed with only $\lfloor \log n \rfloor$ queries to a set X unless A is recursive. In addition we produce the following: (1) a verbose set in each truth-table degree and a superterse set in each nonzero truth-table degree; and (2) an r.e. verbose set in each r.e. truth-table degree and an r.e. terse set in each nonzero r.e. Turing degree.

[1]Some of this research has appeared in the first author's doctoral dissertation, which was prepared at Stanford University, with support from an NSF fellowship, a Hertz Foundation fellowship, and a General Electric Corporation forgivable loan.

[2]Although much of our research on bounded query classes in a complexity-theoretic framework has already appeared in print, this recursion-theoretic research actually preceded it.

[3]Dept. of Computer Science, P.O. Box 2158, Yale Station, New Haven, CT 06520-2158. Supported by NSF grants CCR-8808949 and CCR-8958528. This paper was prepared while this author was at the Johns Hopkins University.

[4]Dept. of Computer Science, University of Maryland, College Park, MD, 20742. Supported by NSF grant CCR-8803641.

[5]Dept. of Electrical Engineering, Stanford University, Stanford, CA, 94305.

[6]Dept. of Mathematics, University of Maryland, College Park, MD, 20742.

1. Introduction

Consider K, the halting set. If one is presented with n numbers $x_1, \ldots, x_n$, and asked to determine which of these numbers are in K, this can clearly be done using n queries to K, namely the x_i's themselves. Yet one can be far more economical in the questions one asks to K. With just $\lceil \log_2(n+1) \rceil$ queries to K one can determine *how many* of the x_i's are in K, and then enumerate K until that many elements have appeared. (This observation appears in [7] but was known to Louise Hay before 1980.)

The ability to answer n queries by using fewer than n queries may appear contradictory, but in fact there is a very important distinction between the n queries being simulated and the $\lceil \log_2(n+1) \rceil$ queries used in the simulation. The initial n queries are presented in *parallel*, that is, they are all specified in advance. The $\lceil \log_2(n+1) \rceil$ queries in the simulation are *sequential*, that is, each query depends on the answers to all queries preceding it. Thus, sequential queries can be exponentially more powerful than parallel queries. (In another terminology [8,16] sequential queries are called "adaptive" and parallel queries are called "nonadaptive.")

There are sets for which the difference between parallel queries and sequential queries is nonexistent. Consider a suitably random set R. It is intuitively obvious that n parallel queries to R cannot be answered with $n-1$ sequential queries to R (or to any other set).

We now formalize our notions.

Definition: If $A \subseteq \mathrm{N}$ (the natural numbers) and $x_1, \ldots, x_n \in \mathrm{N}$ then let

$$F_n^A(x_1, \ldots, x_n) = \langle \chi_A(x_1), \ldots, \chi_A(x_n) \rangle$$

where χ_A is the characteristic function of A.

Definition: If $A \subseteq \mathrm{N}$ then a *query to A* is a question of the form "$x \in A$?" If f is a function from N to N then a *query to f* is a question of the form "What is the value of $f(x)$?" (In both cases, we usually identify the natural number x with the query.)

Definition: Let $n \in \mathrm{N}$ and let f be a function mapping N to N. Then $FQ(n,f)$ is the collection of all functions g such that g is recursive in f via an algorithm that makes at most n sequential queries to f. If $A \subseteq \mathrm{N}$ then $FQ(n,\chi_A)$ is denoted $FQ(n,A)$, and $A \in FQ(n,f)$ means $\chi_A \in FQ(n,f)$.

Definition: A set A is *terse* if for all $n > 0$, $F_n^A \notin FQ(n-1,A)$. A is *superterse* if, for all $n > 0$ and for all $X \subseteq \mathrm{N}$, $F_n^A \notin FQ(n-1,X)$. A is *verbose* if, for all $n > 0$, $F_{2^n-1}^A \in FQ(n,A)$ (equivalently, for all $n > 0$, $F_n^A \in FQ(\lceil \log_2(n+1) \rceil, A)$).

In Section 2 we show that nonrecursive verbose sets exist and, in fact, that every truth-table (tt) degree contains a verbose set. In Section 3 we show that if A is a nonrecursive set then $F_{2^n}^A \notin FQ(n,A)$, which motivates our definition of verboseness: only recursive sets are "less terse" then a verbose set. In Sections 4 and 5 we explore which kinds of sets are terse. In particular we show that 1-generic sets and the jumps of nonrecursive sets are terse, that every nonzero r.e. Turing (T) degree contains an r.e. terse set, and that every nonzero tt-degree contains a superterse set.

Throughout this paper $\{0\}^{()}, \{1\}^{()}, \ldots$ is a list of all oracle Turing machines. The machine $\{e\}^{()(\le i)}$ is identical to $\{e\}^{()}$, except that if it attempts to make more than i queries to the oracle then it diverges without making further queries. A subscript of s on any of these machines means that we run its computations for only s steps. Let $\{e\}$ denote $\{e\}^{\emptyset}$. We also let $\{e\}$ denote the partial function computed by Turing machine $\{e\}$; usage will be clear from context. Let W_e denote the e^{th} recursively enumerable set, i.e., the domain of $\{e\}$. Let $W_{e,s}$ be W_e after s stages, i.e., $W_{e,s} = \{0,1,2,\ldots,s\} \cap \{x : \{e\}_s(x)\downarrow\}$. Let K denote the halting set, i.e., $\{e : e \in W_e\}$.

We denote a fixed recursive bijection from $\mathrm{N} \cup (\mathrm{N} \times \mathrm{N}) \cup (\mathrm{N} \times \mathrm{N} \times \mathrm{N}) \cup \cdots$ to N by $\langle -, \cdots, - \rangle$. That is, the ordered k-tuple $(x_1, \ldots, x_k)$ is coded by the natural number $\langle x_1, \ldots, x_k \rangle$. We assume that for all tuples $(x_1, \ldots, x_k)$ and all i, we have $x_i < \langle x_1, \ldots, x_k \rangle$. Since the tupling function is a recursive bijection, it has a recursive inverse. Most of the other functions in this paper have domain N but we abbreviate "$f(\langle x_1, \ldots, x_n \rangle)$" by "$f(x_1, \ldots, x_n)$." We also abbreviate "$\{e\}(\langle x_1, \ldots, x_n \rangle)$" by "$\{e\}(x_1, \ldots, x_n)$."

A string σ is a finite sequence of 0's and 1's. The length of σ is denoted $|\sigma|$. If $0 \le i < |\sigma|$ then $\sigma(i)$ denotes the i^{th} bit of σ, where we begin counting at 0.

All logarithms in this paper are base 2.

Applications of our work to recursive graph theory appear in [5,6]. Questions concerning $Q(n, A)$, the class of sets B such that $\chi_B = \{e\}^{A(\leq n)}$ for some e, are considered in [3,4,7]. Similar questions have been considered in a polynomial time framework. See [1] for a list of references.

2. Verbose Sets

An important question is whether all nonrecursive sets are terse. They are not: in fact the halting set is verbose [7]. In this section we show that every tt-degree contains a verbose set, and that every r.e. tt-degree contains an r.e. verbose set.

For the sake of completeness, we include a proof that K is verbose.

PROPOSITION 1. [7] K is verbose.

Proof: Recall that K is recursively isomorphic to the set of all Turing machines that halt on the empty tape. Given Turing machines $x_1, \ldots, x_{2^n-1}$ we can, for any m, formulate a query to the halting set which asks if at least m of $x_1, \ldots, x_{2^n-1}$ are in K. We can use these questions to perform a binary search which will determine in n queries to K *exactly* how many of the machines halt. Once we know how many halt, we dovetail the computations of the machines until that many halt. These are the ones that belong to K; all the others do not. ☒

In the algorithm above, if incorrect answers are supplied by the oracle, then the computation may diverge. The question arises, "Is it possible to obtain $F^K_{2^n-1} \in FQ(n, K)$ — or merely $F^K_{n+1} \in FQ(n, X)$ for some X — via an oracle Turing machine such that, no matter what answers are supplied, the machine halts (possibly with the wrong answer)?" This is not the case: Using the $(n+1)$-ary recursion theorem it is easy to construct machines $x_1, \ldots, x_{n+1}$ that deterministically simulate the n-query reduction for all 2^n possible sequences of oracle answers and then halt or diverge so as to defeat the reduction.

It follows from our definitions that verboseness and terseness are invariant under complementation and under many-one equivalence (indeed, these properties are invariant under 1-tt equivalence, which is truth table equivalence with only one query allowed). In particular, all Σ_1-complete and all Π_1-complete sets are verbose.

We were able to apply binary search in proving that K is verbose because we were able to test the necessary thresholds by making a single query to K. This technique can be used more generally. Given any r.e. set A, we construct an r.e. set $B \equiv_{tt} A$ such that B is the *closure of A under thresholds.* Define by induction

$$B_0 = \{\langle\langle x_1\rangle, \ldots, \langle x_n\rangle, m\rangle : |A \cap \{x_1, \ldots, x_n\}| \geq m\}$$

$$B_{k+1} = \{\langle x_1, \ldots, x_n, m\rangle : |B_k \cap \{x_1, \ldots, x_n\}| \geq m\} \cup \bigcup_{i=0}^{k} B_i$$

Let $B = \cup_{i=0}^{\infty} B_i$. It is easily seen that B is r.e. and verbose; hence every r.e. tt-degree contains an r.e. verbose set. By considering Jockusch's semirecursive sets, we obtain another proof of this statement, and we extend the result to arbitrary tt-degrees.

A set $A \subseteq \mathrm{N}$ is *semirecursive* [14] if there exists a recursive function $f(x, y)$ such that

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

Jockusch [14] shows that every (r.e.) tt-degree contains a (r.e.) semirecursive set; he credits McLaughlin and Appel with showing that A is semirecursive iff A is an initial segment of some recursive linear ordering of N.

PROPOSITION 2. If A is semirecursive then A is verbose.

Proof: Suppose A is an initial segment of a recursive linear ordering $\prec$. Given $2^n - 1$ distinct natural numbers $x_1, \ldots, x_{2^n-1}$, first order them by $\prec$ and rename them so that $x_1 \prec x_2 \prec \cdots \prec x_{2^n-1}$. By using binary search we can, in n queries to A, locate the largest i such that $x_i \in A$. (If none of $x_1, \ldots, x_{2^n-1}$ are in A then $i = 0$.) We then output the information that $x_1, \ldots, x_i$ are in A, and $x_{i+1}, \ldots, x_{2^n-1}$ are not. ☒

(Note that, unlike the halting set, semirecursive sets are verbose via an algorithm that terminates regardless of oracle answers.) Since every (r.e.) tt-degree contains a (r.e.) semirecursive set, we obtain:

THEOREM 3. Every (r.e.) tt-degree contains a (r.e.) verbose set.

Except for the halting set K, the verbose sets we have presented so far are arguably contrived examples. Lest the reader think that verbose sets are somehow "unnatural," we point out that Jockusch's work shows that the set of truth stages of an r.e. set (defined below) is semirecursive, and hence verbose. Truth stages are of importance in recursion theory. They were first defined in [10], and were used in showing that every r.e. T-degree contains a hypersimple set. They were later used in infinite injury priority arguments (see [19]).

Definition: Let A be an r.e. set, let $a_1, a_2, a_3, \ldots$ be a recursive enumeration of A, and let $A_s = \{a_1, a_2, \ldots, a_s\}$. The set of *truth stages* of A is $\{s : A_s[a_s] = A[a_s]\}$, where $S[n]$ denotes the first n bits of the characteristic sequence of the set S. The complement of the set of truth stages is called the set of *deficiency stages*.

Definition: A set A is *retraceable* [11] if there exists a total recursive function f such that if $x \in A$ and x is not the smallest element of A then $f(x)$ is the largest element of A that is less than x.

The set of truth stages and the set of deficiency stages are Turing-equivalent to A, and furthermore the set of deficiency stages is r.e. Jockusch has shown that the set of truth stages is retraceable, and that every co-r.e. retraceable set is semirecursive. Hence the set of truth stages is semirecursive, therefore so is its complement, and consequently both sets are verbose.

COROLLARY 4. If A is an r.e. set then the set of deficiency stages of A is verbose.

Thus every r.e. T-degree contains a "natural" r.e. verbose set.

Even if A is terse it need not be superterse, e.g., it may be possible that $F^A_{2^n-1}$ can be computed with a small number of queries to some other set B. The next two results (which follow from results in [7]) are simple examples of this phenomenon. We include self-contained proofs for the sake of completeness.

PROPOSITION 5. If A is r.e. then for all n, $F^A_{2^n-1} \in FQ(n, K)$.

Proof: Since $A \leq_m K$ this follows immediately from the fact that K is verbose. ⊠

In Section 4, we will construct an r.e. terse set. Thus it is possible to have for all n, $F_n^A \in FQ(\lceil \log(n+1) \rceil, K)$ but $F_n^A \notin FQ(n-1, A)$.

PROPOSITION 6. If A is weakly m-r.e. [12,13] then $F_{2^n}^A \in FQ(n + \lceil \log(m+1) \rceil, K)$.

Proof: The characteristic function of a weakly m-r.e. set is the limit of a recursive function that changes its mind at most m times [12]. Given $\langle x_1, \ldots, x_{2^n} \rangle$ we formulate (but do not ask) the following questions to K: "Does f change its mind on x_i at least j times?" $(1 \leq i \leq 2^n, 1 \leq j \leq m)$. Since K is verbose, these $2^n m$ queries can be answered with $\lceil \log(2^n m + 1) \rceil = n + \lceil \log(m + 2^{-n}) \rceil = n + \lceil \log(m+1) \rceil$ queries to K. The answers to these queries immediately determine whether each x_i belongs to A. ⊠

The reader may verify that the preceding result is in fact tight, by considering a weakly $2^n m$-r.e. set that is not weakly $(2^n m - 1)$-r.e. and using Theorems 6.9(ii-iii), 8.2, and 6.11(i) of [7].

3. Lower bound: For all nonrecursive A and all B, $F_{2^n}^A \notin FQ(n, B)$

We have shown that verbose sets exist in abundance. The main theorem of this section shows that verbose sets are as non-terse as possible: it is impossible to obtain a nonrecursive set A such that $F_{2^n}^A \in FQ(n, A)$.

To prove this we need to look at $FQ(n, A)$ in a different light:

Definition: A function f is *computable by a set of partial functions* S if for all z there is a $\varphi \in S$ such that $\varphi(z) = f(z)$. If $|S| = n$ and every element of S is a partial recursive function, then we say that f is computable by a set of n partial recursive functions.

LEMMA 7. If a function f is in $FQ(n, B)$ then f is computable by a set of 2^n partial recursive functions. Conversely, if a function f is computed by a set of 2^n partial recursive functions then there exists an oracle $X \equiv_T f$ such that $f \in FQ(1, F_n^X)$.

Proof: Assume $f \in FQ(n, B)$. Let $\{e\}^{()(\leq n)}$ be the oracle machine such that $\{e\}^{B(\leq n)}$ computes f. Let $w_0, w_1, \ldots, w_{2^n-1}$ be the elements of $\{0,1\}^n$. For $0 \leq i \leq 2^n - 1$ we define a partial recursive function φ_i as follows: $\varphi_i(x)$ is computed by running $\{e\}^{()(\leq n)}(x)$ and using the bits of w_i consecutively for the query answers. Since one of the sequences is correct (i.e., would be the sequence of answers if B was used for the oracle) $\varphi_i(x)$ is equal to $f(x)$ for some i.

Conversely, assume that f is computed by a set of 2^n partial recursive functions. Let the functions be $\varphi_0, \varphi_1, \ldots, \varphi_{2^n-1}$. Let

$$\varphi(x) = \text{the first } i \text{ found (by dovetailing) such that } \varphi_i(x) = f(x),$$

$$X = \{\langle x, j\rangle : \text{the } j^{\text{th}} \text{ bit of the binary representation of } i \text{ is } 1\}.$$

The function f is in $FQ(1, F_n^X)$ since to compute $f(x)$ one need only know the answers to the questions "$\langle x, 1\rangle \in X$?", "$\langle x, 2\rangle \in X$?",…,"$\langle x, n\rangle \in X$?" ☒

We are trying to show that $F_{2^n}^A$ is not in $FQ(n, B)$ for any B. By the above lemma, that is the same as showing that $F_{2^n}^A$ cannot be computed by a set of 2^n partial recursive functions. More generally, we will show that F_n^A cannot be computed by a set of n partial recursive functions for any n. The next lemma is the key to the proof of this section's main theorem.

LEMMA 8. Let g and h be any total functions from N to N and let $f(x, y) = \langle g(x), h(y)\rangle$. If f is computable by a set of $p+1$ partial recursive functions then g is computable by a set of p partial recursive functions or else h is recursive.

Proof: Assume that f is computed by a set $S = \{\varphi_1, \ldots, \varphi_{p+1}\}$ consisting of $p+1$ partial recursive functions. Define $\pi(\langle u, v\rangle) = u$. We consider two cases:

Case 1: For all x there exists y such that two of the $p+1$ partial functions in S converge and the outputs agree on the first component. Formally:

$$(\forall x)(\exists y, j, k)_{j \neq k} \, [(\langle x, y\rangle \in \text{dom}(\varphi_j) \cap \text{dom}(\varphi_k)) \;\wedge\; (\pi(\varphi_j(x, y)) = \pi(\varphi_k(x, y)))].$$

Intuitively, we can save one function since two of the φ_i's agreed. Formally, we describe a set $T = \{\psi_1, \ldots, \psi_p\}$ consisting of p partial recursive functions of one variable. On input

x, the computation of ψ_i first searches for y, j, k that have the above property. Then ψ_i outputs the value of

$$\begin{cases} \pi(\varphi_i(x,y)) & \text{if } i < k \\ \pi(\varphi_{i+1}(x,y)) & \text{if } k \le i \le p. \end{cases}$$

On any input x one of the φ_i's is correct, so one of the ψ_i's must be correct. Hence the set of functions T computes g. (Since j and k may depend on the input, T is not necessarily obtained by choosing p functions in S and restricting them to their first component.)

Case 2 (the negation of Case 1): There exists x such that for every y all of the functions in S either diverge or disagree on the first component when evaluated at $\langle x, y\rangle$. Formally:

$$(\exists x)(\forall y, j, k)_{j \neq k}[(\langle x, y\rangle \in \mathrm{dom}(\varphi_j) \cap \mathrm{dom}(\varphi_k)) \Rightarrow \pi(\varphi_j(x,y)) \neq \pi(\varphi_k(x,y))]$$

We show that h is recursive. We may encode into our algorithm the number x, mentioned in the condition of Case 2, and the value of $g(x)$. Intuitively, only one of the φ_i's gives the right answer for x; it must also give the right answer for y. To compute $h(y)$, search for a j such that $\langle x, y\rangle \in \mathrm{dom}(\varphi_j)$ and $\pi(\varphi_j(x,y)) = g(x)$. Then $h(y)$ is the second coordinate of $\varphi_j(x,y)$. ☒

THEOREM 9. (Nonspeedup Theorem) If $A, B \subseteq \mathrm{N}$, $n \in \mathrm{N}$, and $F^A_{2^n} \in FQ(n, B)$, then A is recursive.

Proof: If $F^A_{2^n} \in FQ(n, B)$ then, by Lemma 7, $F^A_{2^n}$ is computable by a set of 2^n partial recursive functions. Applying Lemma 8 with $g = F^A_{2^n-1}$ and $h = \chi_A$, we find that $F^A_{2^n-1}$ is computable by a set of $2^n - 1$ partial recursive functions or χ_A is recursive. In either case $F^A_{2^n-1}$ is computable by a set of $2^n - 1$ partial recursive functions. Repeating this argument $2^n - 2$ more times, we find that F^A_1 is computable by a set containing 1 partial recursive function. This function is total because F^A_1 is total. But $F^A_1 = \chi_A$. Thus A is recursive. ☒

The proof of the Nonspeedup Theorem is nonuniform in that one cannot use it to recursively compute an index of a machine that decides A from an index for a set of 2^n machines that compute $F^A_{2^n}$ (in the manner that a set of partial functions computes a function as discussed above). In fact no such construction is possible:

PROPOSITION 10. There is no partial recursive function f that takes as input indices e_1 and e_2 for total recursive functions and produces an index for the characteristic function of a set A, if one exists, such that F_2^A is computed by the set of functions $\{\{e_1\}, \{e_2\}\}$. (If no such set A exists then f may produce any answer or may diverge. If several such sets A exist then f may produce an index for any one of them.[7])

Proof: In fact we prove that there is no partial recursive function g with inputs e_1 and e_2 that has the following properties:

- If e_1 and e_2 are not indices for total functions or if there is no set A such that F_2^A is computed by $\{\{e_1\}, \{e_2\}\}$, then g may produce an arbitrary result or may diverge.
- If e_1 and e_2 are indices for total functions and there is a unique set A such that F_2^A is computed by $\{\{e_1\}, \{e_2\}\}$, then (1) if $A = \emptyset$ then $g(e_1, e_2) = 0$, and (2) if $A = \mathrm{N}$ then $g(e_1, e_2) = 1$.
- In all other cases, g may produce an arbitrary result, but must converge.

For the sake of contradiction, suppose that there exists an f as in the statement of this proposition. Then there exists a g with the above properties: let $g(e_1, e_2) = \{f(e_1, e_2)\}(0)$. Now let W_x and W_y be any pair of recursively inseparable sets. We will use g to recursively separate W_x and W_y, a contradiction.

Define total recursive functions h_1 and h_2 such that for all z

$$\{h_1(z)\}(u, v) = \begin{cases} \langle 0, 0\rangle & \text{if } z \in W_{x,u+v} \\ \langle 1, 1\rangle & \text{otherwise.} \end{cases}$$

$$\{h_2(z)\}(u, v) = \begin{cases} \langle 1, 1\rangle & \text{if } z \in W_{y,u+v} \\ \langle 0, 0\rangle & \text{otherwise.} \end{cases}$$

If $z \in W_x$ (hence $z \notin W_y$) then $\emptyset$ is the unique set A such that F_2^A computed by $\{\{h_1(z)\}, \{h_2(z)\}\}$. If $z \in W_y$ (hence $z \notin W_x$) then N is the unique set A such that F_2^A

[7] Although not relevant to the current proposition, the reader may verify that for fixed $e_1, \ldots, e_n$ there are at most n sets A such that F_n^A is computed by $\{\{e_1\}, \ldots, \{e_n\}\}$, by the separation lemma in [18]. Obviously this is tight.

is computed by $\{\{h_1(z)\},\{h_2(z)\}\}$. If $z \notin W_x \cup W_y$ then $\emptyset$ and N are both computed by $\{\{h_1(z)\},\{h_2(z)\}\}$. Let

$$R = \{z : g(h_1(z), h_2(z)) = 0\}$$

It is easy to see that R is a recursive set that contains W_x and is disjoint from W_y. ⊠

4. Terse Sets and Degrees

We are interested in determining which sets are terse and which degrees contain terse sets. In this section we show that all 1-generic sets (defined in [15]) are terse and that every nonzero r.e. T-degree contains an r.e. terse set. In Section 5, we will see that every nonzero tt-degree contains a terse set.

First we sketch a proof that all 1-generic sets are terse. We actually prove that if A is 1-generic then $A^{n+1} \notin FQ(n, A)$ for all n. This is not surprising, because a set A with that property can be constructed by an initial segment argument. Formally, we want to show that every 1-generic set A satisfies, for every $\langle e, n\rangle$, the requirement

$$R_{\langle e,n\rangle} \;:\; A^{n+1} \neq \{e\}^{A(\leq n)}.$$

By standard techniques described in [15], it suffices to show that for every string σ and every $\langle e, n\rangle$, there exists a string τ such that any set whose characteristic sequence is an extension of $\sigma\tau$ satisfies $R_{\langle e,n\rangle}$. For $1 \leq i \leq n+1$, let $x_i = |\sigma| - 1 + i$. Simulate $\{e\}^{()(\leq n)}(x_1, \ldots, x_{n+1})$, using $\sigma(q)$ as the answer to any query q that is numerically less than the length of σ and using 1 as the answer to all other queries. Let $y_1, \ldots, y_k$ $(k \leq n)$ be all the queries that are numerically greater than or equal to the length of σ. Let τ be such that

(1) For $1 \leq i \leq k$, $\sigma\tau(y_i) = 1$.

(2) If the computation rejects then $\sigma\tau(x_1) = \sigma\tau(x_2) = \cdots = \sigma\tau(x_{n+1}) = 1$.

(3) For all numbers x that are not determined by (1) or (2) and are less than or equal to $\max(M, x_{n+1})$, where M is the largest number queried, let $\sigma\tau(x) = 0$.

Note that at least one of the numbers $x_1, \ldots, x_{n+1}$ was not queried. Then it is easy to see that any set that extends $\sigma\tau$ will satisfy requirement $R_{\langle e,n\rangle}$. By standard techniques, as described in [15], it follows now that every 1-generic set A satisfies $A^{n+1} \notin FQ(n, A)$

for all n. Consequently, all 1-generic sets are terse. Jockusch [15] has shown that the set of 1-generic sets is co-meager. (See [20] for more on co-meager sets. The intuition is that co-meager sets are topologically large.) Therefore the set of terse sets is co-meager. Intuitively, this means that most sets are terse.

Similarly, one may show that all 1-generic sets are in fact superterse (use the requirements $R_{\langle e,n\rangle} : F^A_{n+1}$ is not computed by the e^{th} set of 2^n partial recursive functions). The proof is left to the reader.

We now show that every r.e. T-degree contains an r.e. terse set. In fact, we prove slightly more.

Definition: Let $n \in \mathrm{N}$, and let A be any set.

$$\mathrm{PARITY}^A_n(x_1, \ldots, x_n) = \sum_{1\le i\le n} \chi_A(x_i) \bmod 2.$$

THEOREM 11. Every nonrecursive r.e. T-degree contains an r.e. set A such that, for all i, $\mathrm{PARITY}^A_{i+1} \notin FQ(i, A)$.

Proof: Let **d** be a nonrecursive r.e. T-degree and let D be an r.e. set in **d**. We use a finite injury priority argument to construct the desired r.e. set $A \equiv_T D$. We obtain $A \le_T D$ by a permitting argument. We obtain $D \le_T A$ by coding D into A via the even numbers in a manner to be described later.

To ensure for all i that $\mathrm{PARITY}^A_{i+1} \notin FQ(i, A)$, we construct A to satisfy the following requirements:

$$R_{\langle e,i\rangle} : \{e\}^{A(\le i)} \text{ total } \Rightarrow \{e\}^{A(\le i)} \ne \mathrm{PARITY}^A_{i+1}.$$

With every $R_{\langle e,i\rangle}$, we associate an infinite set of sets of odd numbers

$$\{z(e,i,k) : k \in \mathrm{N}\}$$

such that $|z(e,i,k)| = i+1$ and the sets

$$\{z(e,i,k) : e,i,k \in \mathrm{N}\}$$

form a recursive partition of the odd numbers. If $z(e,i,k)$ appears as the argument of a function then the intended argument is the $(i+1)$-tuple of elements that is formed by taking the elements of $z(e,i,k)$ in increasing order. We intend to satisfy $R_{\langle e,i\rangle}$ by making $\{e\}^A(z(e,i,k)) \neq \text{PARITY}^A_{i+1}(z(e,i,k))$ for some k.

We use the even numbers to code D into A. Informally, we have to satisfy the requirements

$$P_j: \text{ if } j \in D \text{ then code this fact into } A.$$

We will explain forthwith precisely how we intend to accomplish this. We use a priority argument with priority ordering $R_0, P_0, R_1, P_1, R_2, P_2, \ldots$.

Let $nr(j)$ be the function such that $nr(\langle e,i\rangle) = i$. Let

$$NR(j) = \sum_{0 \leq k \leq j} nr(k).$$

The symbol "nr" stands for "number restrained." We will later see that during any single stage, R_j cannot restrain more than $nr(j)$ numbers from being in A. Hence the number of elements restrained by requirements of priority higher than P_j is at most $NR(j)$. Therefore if we give P_j more than $NR(j)$ numbers to work with, then P_j will always be able to choose one number that is not restrained by requirements of higher priority. Let

$$\{code(j) : j \in \mathrm{N}\}$$

be a recursive partition of the even numbers such that $|code(j)| = NR(j) + 1$ for each j. Now we can formally state requirements P_j for $j = 1, 2, \ldots$.

$$P_j : \text{if } j \in D \text{ then } A \cap code(j) \neq \emptyset.$$

During the construction only P_j will be able to place elements of $code(j)$ into A (although other requirements will be able to *restrain* elements of $code(j)$ from entering A). Hence, if P_j is satisfied then we know that

$$j \in D \text{ iff } A \cap code(j) \neq \emptyset.$$

The construction proceeds in stages. Let A_s denote A at the end of stage s. Let $res(\langle e,i\rangle, s)$ be the set of numbers that $R_{\langle e,i\rangle}$ wants to restrain from A at the end of stage s. Let

$$RES(\langle e,i\rangle, s) = \bigcup_{j \leq \langle e,i\rangle} res(j,s).$$

We assume that we are given a recursive enumeration $d_1, d_2, d_3, \ldots$ of D. Let $D_s = \{d_1, \ldots, d_s\}$. We construct A as follows:

CONSTRUCTION

Stage 0: Set $A_0 = \emptyset$. For all $\langle e,i \rangle$ set $res(\langle e,i\rangle, 0) = \emptyset$ and mark requirement $R_{\langle e,i\rangle}$ as unsatisfied.

Stage $s+1$:

I) (First we code D into A.) Let $j = d_s$. Let x be the least element in $code(j)$ that is not in $RES(j,s)$. (We prove later that such an x exists.) We enumerate x into A; declare P_j satisfied; and for every $\langle e,i\rangle > j$ such that $R_{\langle e,i\rangle}$ has been previously satisfied, declare $R_{\langle e,i\rangle}$ not satisfied and set $res(\langle e,i\rangle, s+1) = \emptyset$.

II) (Second, we attempt to satisfy some $R_{\langle e,i\rangle}$.) Find the least number $\langle e,i\rangle \le s$ such that the following are all true (or go to stage $s+2$ if no such number exists):

a) $R_{\langle e,i\rangle}$ is not satisfied.

b) For some k the computation of $\{e\}_s^{A_s(\le i)}(z(e,i,k))$ converges in such a way that we can attempt to satisfy $R_{\langle e,i\rangle}$. Formally there exists $k \le s$ such that

$$\{e\}_s^{A_s(\le i)}(z(e,i,k)) \downarrow = b \text{ where } b \in \{0,1\}, \text{ and}$$

$$z(e,i,k) \cap RES(\langle e,i\rangle - 1, s) = z(e,i,k) \cap A_s = \emptyset.$$

c) Let y be the least element of $z(e,i,k)$ such that y is not queried in the $\{e\}_s^{A_s(\le i)}(z(e,i,k))$ computation. Such a y exists since the $\{e\}_s^{A_s(\le i)}(z(e,i,k))$ computation makes at most i queries and $|z(e,i,k)| = i+1$. We require $y > d_s$. (This is the permitting strategy that will make $A \le_T D$.)

If such an $\langle e,i\rangle$ exists then we handle two cases:

$b=0$: Enumerate y into A;

$b=1$: We need not take any action since $z(e,i,k) \cap A = \emptyset$ and no other requirement can place any element of $z(e,i,k)$ into A.

In either case, we declare $R_{\langle e,i\rangle}$ satisfied and set

$$res(\langle e,i\rangle, s+1) = \{x : x \text{ is queried in the } \{e\}_s^{A_s(\le i)} \text{ computation }\};$$

and for all $\langle e',i'\rangle > \langle e,i\rangle$ declare $R_{\langle e',i'\rangle}$ unsatisfied and set $res(\langle e',i'\rangle, s+1) = \emptyset$. For all

$\langle e', i'\rangle < \langle e, i\rangle$ set $res(\langle e', i'\rangle, s+1) = res(\langle e', i'\rangle, s)$.

END OF CONSTRUCTION

We show that each P_j is satisfied. The requirement $R_{\langle e,i\rangle}$ restrains at most i elements at any stage, since it only restrains numbers from A that are queried by *one* computation of the form $\{e\}_s^{A_s(\le i)}(z(e,i,k))$. Therefore $|res(j,s)| \le nr(j)$, $|RES(j,s)| \le NR(j) < |code(j)|$, and part I of the construction can always be executed. Hence every P_j is satisfied. By the comments made about the coding, we have $D \le_{\rm tt} A$. By the usual permitting argument we have $A \le_T D$. Standard finite injury techniques yield that each R_j is satisfied. ☒

In the proof, the permitting ensures that A is weak truth-table reducible to D. (Weak truth-table reductions are defined in [18].) Therefore the preceding result holds for weak truth-table degrees as well. Furthermore, if $\mathrm{PARITY}_{i+1}^A \notin FQ(i,A)$ then certainly $F_{i+1}^A \notin FQ(i,A)$. Hence we have the following corollary:

COROLLARY 12. Every nonrecursive r.e. weak truth-table degree contains an r.e. terse set.

5. Superterse Sets

Recall that a set A is superterse if, for all n and all B, $F_{n+1}^A \notin FQ(n,B)$. Using the Nonspeedup Theorem, we can show that every nonzero tt-degree contains a superterse set. The following lemma is central to the proof.

LEMMA 13. Let A be a nonrecursive set. If $F_{2^n-1}^A \in FQ(n,B)$ via a fixed algorithm $\mathcal{A}$ that works for all n, then there exists a superterse set $C \le_T B$.

Proof: Let

$C = \{\langle x, i\rangle$: on input x, $\mathcal{A}$ makes at least i queries to B and the ith oracle answer is "yes"$\}$.

Clearly $C \le_T B$. Furthermore, the oracle answers required by $\mathcal{A}$ on input x can be determined by making the single query $F_n^C(\langle x,1\rangle, \ldots, \langle x,n\rangle)$ instead of making n serial queries to B. Therefore $F_{2^n-1}^A \in FQ(1, F_n^C)$. If C were not superterse then we would have $F_n^C \in FQ(n-1, D)$ for some n and D, but then $F_{2^n-1}^A \in FQ(n-1,D)$, which violates the Nonspeedup Theorem because $2^n - 1 \ge 2^{n-1}$. ☒

THEOREM 14. Every nonzero tt-degree contains a superterse set.

Proof: Every nonzero tt-degree contains a semirecursive set A. The proof of Proposition 2 provides an algorithm $\mathcal{A}$ such that $F^A_{2^n-1} \in FQ(n, A)$ via $\mathcal{A}$, and $\mathcal{A}$ terminates regardless of oracle answers. Let C be the superterse set constructed in Lemma 13 (with $B = A$). Since $\mathcal{A}$ terminates regardless of oracle answers, it follows from the definition of C that $C \leq_{tt} B = A$. By looking at the case $n = 1$ it is easy to see that $A \leq_m C$ (alternatively, note that $A \oplus C$ has the desired properties). ☒

Degtev [9] has shown that every nonzero truth-table degree contains at least two bounded truth-table degrees. We extend this somewhat.

Definition: A set B is *bounded weak truth-table reducible* to a set A if $B \in FQ(1, F^A_k)$ for some k.

Note that truth-table reductions must converge even if incorrect oracle answers are supplied; hence a bounded weak truth-table reduction need not even be a truth-table reduction [17].

COROLLARY 15. Every nonzero truth-table degree contains two sets that are inequivalent under bounded weak truth-table reductions.

Proof: By Theorem 3 and Theorem 14, every non-zero truth-table degree contains a verbose set A and a superterse set B. Suppose that C is some set that is bounded weak truth-table reducible to A, i.e., that $C \in FQ(1, F^A_k)$ for some k. Then $F^C_n \in FQ(1, F^A_{kn}) \subseteq FQ(\lceil \log(kn+1) \rceil, A)$, so C is not superterse. In particular, B must not be bounded weak truth-table reducible to A. ☒

We have natural examples of superterse sets, because the jump of every nonrecursive set is superterse.

THEOREM 16. If A is nonrecursive then A' is superterse.

Proof: Assume that A' is not superterse. Then there exists a natural number n and a set B such that $F^{A'}_{n+1} \in FQ(n, B)$. Since $A \leq_m A'$, we have $F^A_{2^n} \in FQ(2^n, B \oplus A')$. We show that $F^A_{2^n} \in FQ(n, B \oplus A')$, contradicting the Nonspeedup Theorem. Suppose

$m > n$ and $F_{2^n}^A \in FQ(m, B \oplus A')$. Then $F_{2^n}^A$ is computable by a set of 2^m partial recursive functions by Lemma 7. Therefore, by the same lemma, there exists an oracle $X \equiv_T A$ such that $F_{2^n}^A \in FQ(1, F_m^X)$. Since $X \leq_T A$, we have $X \leq_m A'$. Therefore $F_{2^n}^A \in FQ(1, F_m^{A'}) \subseteq FQ(m-1, B \oplus A')$ because $m > n$ and $F_{n+1}^{A'} \in FQ(n, B)$. Now a simple induction shows that $F_{2^n}^A \in FQ(n, B \oplus A')$. ☒

COROLLARY 17. Let $n > 1$. If A is Σ_n-complete or Π_n-complete then A is superterse.

Since K is verbose, we have the following corollary:

COROLLARY 18. A is recursive if and only if A' is verbose. A is nonrecursive if and only if A' is superterse.

6. Amplifying Non-superterseness

In this section we show that if a set A is non-superterse, then A is "very" non-superterse. That is, if $F_{k+1}^A \in FQ(k, B)$ for some k and B then $F_n^A \in FQ(O(\log n), C)$ for some C. Intuitively this means that every non-superterse set is "nearly" verbose. By defining variants of the semirecursive sets, we show that this result is tight. As a corollary, we show that all Kolmogorov-random sets are superterse, and hence that almost all sets are superterse.

Let $\binom{n}{i}$ denote the binomial coefficient "n choose i" and let

$$S(n,k) = \sum_{0 \leq i \leq k-1} \binom{n}{i}.$$

The following lemma has appeared in [4]. A complexity-theoretic variant with essentially the same proof has appeared in [2].

LEMMA 19. [4] If F_k^A is computable by a set of $2^k - 1$ partial recursive functions then, for all n, F_n^A is computable by a set of $S(n,k)$ partial recursive functions.

We note that the set of $S(n,k)$ functions obtained in [4] is in fact obtained uniformly in n.

If A is not superterse then there exists a natural number k and a set B such that $F_k^A \in FQ(k-1, B)$, so F_k^A is computable by a set of 2^{k-1} partial recursive functions by Lemma 7. Applying Lemma 8 with $g = F_{k-1}^A$ and $h = \chi_A$ we find that F_{k-1}^A is computable by a set of $2^{k-1} - 1$ partial recursive functions. Thus by the preceding lemma, for all n, F_n^A is computable by a set of $S(n, k-1)$ partial recursive functions. Therefore for all n, there exists $X_n \equiv_T A$ such that $F_n^A \in FQ(1, F^{X_n}_{(k-2)\log n + o(\log n)})$, by Lemma 7. Since the set of $S(n,k)$ functions in the preceding lemma is obtained uniformly in n, the reduction from X_n to A is uniform in n. Let X be the recursive join of all the X_i's. Then $X \equiv_T A$ and $F_n^A \in FQ(1, F^X_{(k-2)\log n + o(\log n)})$. In particular, we have

THEOREM 20. If A is not superterse then there exists a set $X \equiv_T A$ such that $F_n^A \in FQ(1, F^X_{O(\log n)})$.

Except for superterse sets, we have not given examples of sets A for which F_n^A requires more than $\log n$ plus a constant number of queries. However, by analogy to semirecursive sets, we can define sets that require $\log n$ *times* a constant number of queries. For any natural number k we define a set A that is an initial segment of a recursive partial ordering of N with k incomparable chains. Then $F_{kn}^A \in FQ(\lceil k \log(n+1) \rceil, A)$, and it is not hard to construct A so that no smaller number of queries to any oracle is sufficient. Thus Theorem 20 is tight.

In [4] there appears a result of some interest that is similar to Theorem 20, but which does not require a change of oracle: if A is not terse then there is a real number $r < 1$ such that for all n, $F_n^A \in FQ(n^r, A)$. A complexity-theoretic version of that theorem is also proved in [4].

For a fixed universal Turing machine, the *Kolmogorov complexity* of a string x, denoted $K(x)$, is the length of the shortest program that outputs x on empty input. A set A is *Kolmogorov-random* if there exists a c such that for infinitely many n, $K(A[n]) \geq n - c$, where $A[n]$ denotes the first n bits of the characteristic sequence of A.

COROLLARY 21. All Kolmogorov-random sets are superterse.

Proof: Suppose that A is not superterse. Then by Theorem 20 there exists a set X and an oracle Turing machine M such that $F_n^A \in FQ(1, F^X_{O(\log n)})$ via M^X. $A[n]$ can be encoded by specifying n, the machine M, and the answers to the $O(\log n)$ queries to X that are asked. Hence $A[n]$ can be encoded with $O(\log n)$ bits. Therefore A is not Kolmogorov-random.$\boxtimes$

In particular, this implies that almost all sets are superterse. The converse of the preceding corollary is easily seen to be false: not every superterse set is algorithmically random. For example, if A is superterse, then so is $Q = \{x^2 : x \in A\}$. But Q is somewhat sparse and is therefore a nonrandom set; the first n^2 bits of Q can be described by an input of length $n + O(1)$.

7. Open Problems

There are many open questions about terseness and verboseness not touched upon in this paper. One interesting question is "does every nonzero 2-r.e. (m-r.e.) T-degree contain a terse 2-r.e. (m-r.e.) set?" The proof we use for r.e. sets involves permitting, which does not seem to work with 2-r.e. sets. Another question is whether every nonzero r.e. *tt*-degree contains an r.e. terse set.

Recall that the Nonspeedup Theorem says we cannot determine which of 2^n numbers belong to a nonrecursive set A by performing an $FQ(n, X)$ computation. Is there a nonrecursive set A for which we can determine *how many* of 2^n numbers belong to A by performing an $FQ(n, X)$ computation? We conjecture that no such A exists. Along these lines, Owings [18] has shown that necessarily $n > 1$ and $A \leq_T K$.

8. Acknowledgments

We would like to thank Carl Smith for suggesting the name "terse," Larry Herman and Mark Pleszkoch for proofreading, and Stuart Kurtz for asking whether the proof of the Nonspeedup Theorem is necessarily nonuniform.

REFERENCES

1. Amir, A., Beigel, R., and Gasarch, W. I. Some Connections between Bounded Query Classes and Non-Uniform Complexity. *Proceedings of the Fifth Annual Conference on Structure in Complexity Theory,* IEEE Computer Society Press (1990).

2. Beigel, R. A Structural Theorem that Depends Quantitatively on the Complexity of SAT. *Proceedings of the Second Annual Conference on Structure in Complexity Theory* , IEEE Computer Society Press (1987), 28-32.

3. Beigel, R. *When are $k+1$ Queries Better Than k?* The Johns Hopkins University, Department of Computer Science, TR-6 (1988).

4. Beigel, R. *Query Limited Reducibilities.* Ph.D thesis, Stanford University (1987).

5. Beigel, R. and Gasarch, W. I. On the Complexity of Finding the Chromatic Number of a Recursive Graph I: The Bounded Case. *Annals of Pure and Applied Logic 45* (November 1989), 1-38.

6. Beigel, R. and Gasarch, W. I. On the Complexity of Finding the Chromatic Number of a Recursive Graph II: The Unbounded Case. *Annals of Pure and Applied Logic 45* (December 1989), 227-247.

7. Beigel, R., Gasarch, W. I., and Hay, L. Bounded Queries Classes and the Difference Hierarchy. *Archive for Math. Logic 29* (December 1989), 69-84.

8. Book, R. V. and Ko, Ker-I. On Sets Truth-Table Reducible to Sparse Sets. *SIAM Journal of Computing, 17* (1988), 903-919.

9. Degtev, A. N. Three Theorems on tt-degrees. *Algebra and Logic 17* (1978), 187-194.

10. Dekker, J. C. E. A theorem on hypersimple sets. *Proceedings American Math Society 5* (1954), 791-796.

11. Dekker, J. C. E. and Myhill, J. Retraceable Sets. *Canadian Journal of Mathematics 10* (1958), 357-373.

12. Epstein, R. L., Haas, R., and Kramer, R. L. Logic Year 1979-80. *Hierarchies of sets and degrees below $\emptyset'$* (1981), 32-48. Lecture Notes in Mathematics 859, Springer-Verlag, Berlin.

13. Ershov, A. A Hierarchy of Sets I,II,III. *Algebra and Logic* (1968a, 1968b, 1970). Vol. 7 (1) 25-43, Vol. 7 (4) 212-232, Vol. 9 (1) 20-31.

14. Jockusch, C. G. Semirecursive sets and positive reducibility. *Transactions of the AMS 131* (May 1968), 420-436.

15. JOCKUSCH, C. G. Degrees of Generic Sets. *Recursion Theory: its Generalisations and Applications, Proceedings of Logic Colloquium 1979, Leeds* (August 1979), 140-157.

16. KO, KER-I. On Adaptive Versus Nonadaptive Bounded Query Machines. *Theoretical Computer Science* (To appear in 1991).

17. LACHLAN, A. H. Some Notions of Reducibility and Productiveness. *Zeitsch. f. math. Logik und Grundlagen d. Math. 11* (1965), 17-44.

18. OWINGS, J. C. A Cardinality Version of Beigel's Nonspeedup Theorem. *Journal of Symbolic Logic 54* (September 1989), 761-767.

19. SOARE, R. I. *Recursively Enumerable Sets and Degrees.* Springer Verlag (Omega Series), Berlin, 1987.

20. STROMBERG, K. R. *An Introduction to Classical Real Analysis.* Wadsworth International Mathematics Series, Belmont, California, 1981.