A Techniques-Oriented Survey of Bounded Queries

William Gasarch*

University of Maryland

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. For nonrecursive functions, time is not an appropriate measure. In 1985 Richard Beigel [1] and William Gasarch [14] independently hit on the idea of measuring the complexity of a nonrecursive function f by how many queries to some set X are required to compute f. (Louise Hay had similar ideas but not quite in that form [15].) There have since been many papers in the area and an upcoming book [13].

In the book and in a prior survey [12] the main theme has been the *classification of functions*: given a function, how complex is it, in this measure. In this survey we instead look at the *techniques* used to answer such questions. Hence each section of this paper focuses on a technique.

All of the results in this paper have appeared elsewhere except those in Section 8. For this reason we give sketches rather than proofs, except in Section 8.

*Dept. of C.S. and Inst. for Adv. Comp. Stud., University of MD., College Park, MD 20742, U.S.A. (Email: gasarch@cs.umd.edu.)

2 Notation, Definitions, and Useful Facts

We use standard notation from recursion theory [22, 25]. We define classes of functions that can be computed with a bound on the number of queries to an oracle.

Definition 2.1 [2] $\mathrm{FQ}(n, A)$ 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). $\mathrm{FQ}^X(n, A)$ and $\mathrm{FQ}_{||}^X(n, A)$ are similar except that we also allow unlimited queries to X.

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

Definition 2.2

- $B \in \mathrm{Q}(n, A)$ 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.

Note that if (say) $f \in \mathrm{FQ}(3, A)$ then it might be that while trying to compute (say) $f(10)$, and 3 INCORRECT answers are given, the computation might diverge. We now define the class of functions for which this does not happen.

Definition 2.3 [2] $\mathrm{FQC}(n, A)$ is the collection of all total functions f such that f is recursive in A via an oracle Turing machine $M^{()}$ that has the following property: for all x, for all X, $M^X(x)$ makes at most n sequential queries to X and $M^X(x) \downarrow$.

Note 2.4 The classes $\mathrm{FQC}_{||}(n, A)$, $\mathrm{QC}(n, A)$, and $\mathrm{QC}_{||}(n, A)$ can easily be defined.

The following notion has important connections to bounded queries which will be made explicit in Theorem 2.6.

Definition 2.5 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 $f(x) \in W_{g(x)}$. Let $n \geq 1$. A function f is *strongly n-enumerable* (denoted $f \in \mathrm{SEN}(n)$) if there exists a recursive function g such that, for all x, $|D_{g(x)}| \leq n$ and $f(x) \in D_{g(x)}$. (This concept first appeared in a recursion-theoretic framework in [1]. The name "enumerable" was coined in [6].)

Theorem 2.6 *[2] If f is any function then (1) $(\exists X)[f \in \mathrm{FQ}(n, X)] \iff f \in \mathrm{EN}(2^n)$; and (2) $(\exists X)[f \in \mathrm{FQC}(n, X)] \iff f \in \mathrm{SEN}(2^n)$.*

The following definition introduceds two functions which have been very useful in the study of bounded queries.

Definition 2.7 [2] Let $n \geq 1$.

1. $\mathrm{C}_n^A \colon \mathsf{N}^n \to \{0,1\}^n$ is defined by $\mathrm{C}_n^A(x_1, \ldots, x_n) = A(x_1) \cdots A(x_n)$.
2. $\#_n^A \colon \mathsf{N}^n \to \{0, \ldots, n\}$ is defined by $\#_n^A(x_1, \ldots, x_n) = |\{i : x_i \in A\}|$.

The following two theorems are the most fundamental in bounded queries.

Theorem 2.8 *Let $n \geq 1$.*

1. *If $\mathrm{C}_n^A \in \mathrm{EN}(n)$, then A is recursive [2].*
2. *If $\#_n^A \in \mathrm{EN}(n)$, then A is recursive [20].*

3 Using Clever Techniques

In this section we examine the complexity of the set ODD_n^A (defined below). We first prove results for A semirecursive. This proof requires some cleverness. Then we derive these same results for A r.e. *from* the result for A semirecursive. This requires a clever use of known theorems. We will only consider parallel queries since they suffice to illustrate the techniques. A fuller treatment of the results here is in [4, 5]

Definition 3.1 $\mathrm{ODD}_n^A = \{(x_1, \ldots, x_n) : \#_n^A(x_1, \ldots, x_n) \text{ is odd.}\}$

3.1 The Complexity of ODD_n^A for Semirecursive Sets A

Definition 3.2 [17] A set A is semirecursive-in-X if either one of the two equivalent conditions holds. (A set is semirecursive if it is semirecursive-in-$\emptyset$.)

1. There exists a linear ordering $\sqsubseteq$ on N such that $\sqsubseteq$ is recursive in X and A is closed downward under $\sqsubseteq$.

2. There exists $f \leq_{\mathrm{T}} X$ such that, for all x, y, (1) $f(x, y) \in \{x, y\}$, and (2) $A \cap \{x, y\} \neq \emptyset \Rightarrow f(x, y) \in A$.

Theorem 3.3 *If A is semirecursive, $m \in N$, and $\#_{m+1}^A \in \mathrm{FQ}(1, \mathrm{C}_m^A)$ then A is recursive.*

Proof sketch: Note that for A semirecursive $\mathrm{C}_m^A \in \mathrm{FQ}(1, \mathrm{C}_m^A)$, hence the premise can be restated as $\mathrm{C}_{m+1}^A \in \mathrm{FQ}(1, \mathrm{C}_m^A)$. An easy induction shows that $(\forall n)[\mathrm{C}_n^A \in \mathrm{FQ}(1, \mathrm{C}_m^A)]$. In particular $\mathrm{C}_{2^m}^A \in \mathrm{FQ}(1, \mathrm{C}_m^A) \subseteq \mathrm{FQ}(m, A) \subseteq \mathrm{EN}(2^m)$ (The last inclusion is from Theorem 2.6.) By Theorem 2.8.a A is recursive. ∎

Theorem 3.4 *Let $n \geq 1$, and let A be a semirecursive set such that* $\mathrm{ODD}_n^A \in \mathrm{Q}_{||}(n-1, A)$. *Then A is recursive.*

Proof sketch: Clearly A is recursive if $n = 1$, since $(\forall x)[\mathrm{ODD}_1^A(x) = A(x)]$, so assume that $n > 1$. Choose a recursive linear ordering $\sqsubseteq$ so that A is semirecursive via $\sqsubseteq$, and choose an oracle Turing machine $M^{()}$ so that $\mathrm{ODD}_n^A \in \mathrm{Q}_{||}(n-1, A)$ via M^A.

The following algorithm shows that $\#_{2n+1}^A \in \mathrm{FQ}(1, \mathrm{C}_{2n}^A)$. Hence A is recursive, by Theorem 3.3.

ALGORITHM

1. Input $(x_1, \ldots, x_{2n+1})$, where we can assume $x_1 \sqsubseteq \cdots \sqsubseteq x_{2n+1}$. Note that $\mathrm{C}_{2n+1}^A(x_1, \ldots, x_{2n+1}) \in 1^*0^*$.

2. Simulate the computation of $M^{()}(x_2, x_4, x_6, \ldots, x_{2n})$ to obtain numbers $z_1, \ldots, z_{n-1}$ such that the (parallel) queries made in the computation $M^A(x_2, x_4, x_6, \ldots, x_{2n})$ are "$z_1 \in A$?", ..., "$z_{n-1} \in A$?". (We do not make these queries at this point; we will make them in parallel with others in step 3.)

3. Ask: "What is $C^A_{2n}(x_1, x_3, x_5 \ldots, x_{2n+1}, z_1, z_2, z_3, \ldots, z_{n-1})$?" Using this information we can find $C^A_n(x_1, x_3, x_5, \ldots, x_{2n+1})$ and $\mathrm{ODD}^A_n(x_2, x_4, \ldots, x_{2n})$. From this information, together with the ordering on A, we can compute $C^A_{2n+1}(x_1, \ldots, x_{2n+1})$.

END OF ALGORITHM

∎

The above theorem easily relativizes to yield the following.

Theorem 3.5 *Let $n \geq 1$, and let A, X be such that A is semirecursive-in-X and $\mathrm{ODD}^A_n \in \mathrm{Q}^X_{||}(n-1, A)$. Then $A \leq_{\mathrm{T}} X$.*

3.2 The Complexity of ODD^A_n for R.E. Sets A

Definition 3.6 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.

Note 3.7 The degrees of extensive sets are the same as the degrees of complete consistent models of Peano Arithemetic [18]. Hence they have been called PA sets. They have also been called DNR_2 sets. We hope the terminology 'extensive' is the one that will survive.

Proposition 3.8 ([19]) *There is a minimal pair of extensive sets.*

The next lemma will make an r.e. sets A "look semirecursive."

Lemma 3.9 *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.

■

The following lemma follows from Theorem 3.5 and Lemma 3.9.

Lemma 3.10 *Let $n \geq 1$, and let A, X be sets such that A is r.e., X is extensive, and $\mathrm{ODD}_n^A \in \mathrm{Q}_{||}^X(n-1, A)$. Then $A \leq_{\mathrm{T}} X$.*

Theorem 3.11 *Let A be an r.e. set, and let $n \geq 1$. If $\mathrm{ODD}_n^A \in \mathrm{Q}_{||}(n-1, A)$, then A is recursive.*

Proof: Suppose $\mathrm{ODD}_n^A \in \mathrm{Q}_{||}(n-1, A)$. Note $\mathrm{ODD}_n^A \in \mathrm{Q}_{||}^X(n-1, A)$ for every set X. Choose sets X_1 and X_2 that are extensive and form a minimal pair (such X_1, X_2 exists, by Proposition 3.8). Since X_1 and X_2 are extensive, it follows from Lemma 3.10 that $A \leq_{\mathrm{T}} X_1$ and $A \leq_{\mathrm{T}} X_2$. Since X_1, X_2 form a minimal pair we have A recursive. ■

4 Using Coding Theory

In this section we will state theorems about the function $freq_{m,n}^A$, which is an 'approximations' of C_n^A. In order to even state our theorems we need some terminology from coding theory. A full treatment of this material can be found in [3].

Definition 4.1 Let A be a set, and let $m, n, i \geq 1$ $(m, i \leq n)$. $freq_{m,n}^A$ is the class of functions f such that $f(x_1, \ldots, x_n)$ and $\mathrm{C}_n^A(x_1, \ldots, x_n)$ agree in at least m places. ($freq$ stands for 'frequency,' i.e., the frequency of agreement, in terms of number of components, between $f(x_1, \ldots, x_n)$ and $\mathrm{C}_n^A(x_1, \ldots, x_n)$.)

Definition 4.2 Let $a, r \in \mathsf{N}$. Let $z \in \{0,1\}^a$. The *closed ball of radius* r *centered at* z is the set $B(z, r) = \{y \in \{0,1\}^a : y =^r z\}$. If $D \subseteq \{0,1\}^a$ then D *is covered by* k *balls of radius* r means that there exist $z_1, \ldots, z_k \in \{0,1\}^a$ such that $D \subseteq \bigcup_{i=1}^{k} B(z_i, r)$.

Definition 4.3 Let $a, r \in \mathsf{N}$ and $D \subseteq \{0,1\}^a$. Define $k(D, r)$ to be the minimal number j such that D can be covered by j balls of radius r. The quantity $k(\{0,1\}^a, r)$ is denoted by $k(a, r)$.

The quantity $k(a, r)$ is known as *the covering number*. It has been studied extensively (see [7, 8, 9, 16, 26]). No exact formula is known for it.

Definition 4.4 Let $a, r \in \mathsf{N}$ and $\mathcal{D} \subseteq 2^{\{0,1\}^a}$. We define $k(\mathcal{D}, r)$ to be $\max\{k(D, r) : D \in \mathcal{D}\}$.

We now define the notions of $\mathcal{D}$-verbose and strongly $\mathcal{D}$-verbose in order to state a very general result. Note that every set is strongly $2^{\{0,1\}^a}$-verbose.

Definition 4.5 Let $a \in \mathsf{N}$. Let $\mathcal{D} \subseteq 2^{\{0,1\}^a}$. A set A is $\mathcal{D}$-*verbose* if there is a recursive function g such that, for all $x_1, \ldots, x_a$, $W_{g(x_1,\ldots,x_a)} \in \mathcal{D}$ and $\mathrm{C}_a^A(x_1, \ldots, x_a) \in W_{g(x_1,\ldots,x_a)}$. A set A is *strongly* $\mathcal{D}$-*verbose* if there is a recursive function g such that, for all $x_1, \ldots, x_a$, $D_{g(x_1,\ldots,x_a)} \in \mathcal{D}$ and $\mathrm{C}_a^A(x_1, \ldots, x_a) \in D_{g(x_1,\ldots,x_a)}$.

The following theorem provides for any $A \subseteq \mathsf{N}$ (1) matching upper and lower bounds for the strong enumerability of $freq_{b,a}^A$, and (2) lower bounds for the enumerability of $freq_{b,a}^A$.

Theorem 4.6 *Assume* $1 \le b \le a$ *and* $A \subseteq \mathsf{N}$. *For all* k *the following hold.*

1. *The following are equivalent.*
   - *(a) There exists* $\mathcal{D} \subseteq 2^{\{0,1\}^a}$ *such that* A *is strongly* $\mathcal{D}$-*verbose and* $k(\mathcal{D}, a-b) \le k$.
   - *(b)* $freq_{b,a}^A \cap \mathrm{SEN}(k) \ne \emptyset$.
2. *If* $freq_{b,a}^A \cap \mathrm{EN}(k) \ne \emptyset$ *then there exists* $\mathcal{D} \subseteq 2^{\{0,1\}^a}$ *such that* A *is* $\mathcal{D}$-*verbose and* $k \ge k(\mathcal{D}, a-b)$.

Theorem 4.6 yields matching upper and lower bounds; however they are not readily computable. We state theorems about two cases where it can be computed.

Theorem 4.7 *Assume* $1 \le b \le a$, *A is a semirecursive set that is not recursive, and* $k = \left\lceil \frac{a+1}{2(a-b)+1} \right\rceil$. *Then* $freq^A_{b,a} \cap \mathrm{SEN}(k) \neq \emptyset$ *but* $freq^A_{b,a} \cap \mathrm{SEN}(k-1) = \emptyset$. *Note that if* $\frac{b}{a} \le \frac{1}{2}$ *then* $k = 1$ *so* $freq^A_{b,a} \cap \mathrm{EN}(1) \neq \emptyset$, *hence some function in* $freq^A_{b,a}$ *is recursive.*

Definition 4.8 [2] A set A is *superterse* if $(\forall n)(\forall X)[\mathrm{C}^A_n \notin \mathrm{FQ}(n-1, X)]$. A set A is *weakly superterse* if $(\forall n)(\forall X)[\mathrm{C}^A_n \notin \mathrm{FQC}(n-1, X)]$.

Theorem 4.9 *Assume* $1 \le b \le a$, $\frac{b}{a} > \frac{1}{2}$, *and* $A \subseteq \mathsf{N}$.

1. $freq^A_{b,a} \cap \mathrm{SEN}(k(a, a-b)) \neq \emptyset$. *The algorithm that achieves this does not look at the input and runs in constant time.*
2. *If A is superterse then* $freq^A_{b,a} \cap \mathrm{EN}(k(a, a-b) - 1) = \emptyset$.
3. *If A is weakly superterse then* $freq^A_{b,a} \cap \mathrm{SEN}(k(a, a-b) - 1) = \emptyset$.

5 Using the Recursion Theorem

In this section we show two uses of the recursion theorem within bounded queries.

5.1 The Complexity of C^K_m

It is well known that if A is any r.e. set then $\mathrm{C}^A_m \in \mathrm{EN}(m+1)$: given $(x_1, \ldots, x_m)$, first output 0^n. Enumerate A until an element appears. If this happens then output the appropriate $0^m 1 0^{n-m-1}$. Then enumerate A some more until the next element appears, etc. The proof does not seem to yield $\mathrm{C}^A_m \in \mathrm{SEN}(m+1)$. The next theorem shows that, in fact, $\mathrm{C}^K_m \notin \mathrm{SEN}(2^m - 1)$.

Theorem 5.1 $\mathrm{C}^K_m \notin \mathrm{SEN}(2^m - 1)$.

Proof: Assume $C_m^K \in \mathrm{SEN}(2^m - 1)$ via f. We construct $x_1, \ldots, x_m$ such that $C_m^K(x_1, \ldots, x_m) \notin D_{f(x_1,\ldots,x_m)}$. Our construction of $x_1, \ldots, x_m$ uses the m-ary recursion theorem [23, 24] so program x_i knows the programs $x_1, \ldots, x_m$.

Our algorithm for x_i is as follows. On any input, x_i first finds $D_{f(x_1,\ldots,x_m)} = \{w_1, \ldots, w_{2^m-1}\}$. Let w be the least element (lexicographically) of $\{0,1\}^m$ that is not in $\{w_1, \ldots, w_{2^{m-1}}\}$. If the ith bit of w is 0 then x_i diverges, else x_i converges.

For all x_i the same w is found. The x_i's conspire to make $C_m^K(x_1, \ldots, x_m) = w$. But $w \notin D_{f(x_1,\ldots,x_m)}$ Hence $C_m^K \notin \mathrm{SEN}(2^m - 1)$ via f. ∎

5.2 The Complexity of $freq_{b,a}^K$

In Section 4 we determined the complexity of $freq_{b,a}^A$ for several A using notions from coding theory. In this section we determine the complexity of $freq_{b,a}^K$. The lower bound uses the recursion theorem. The upper bound does not use the recursion theorem; however, it is included for completeness.

Theorem 5.2 *If* $1 \le b \le a$ *then* $freq_{b,a}^K \cap \mathrm{EN}(\lceil \frac{a+1}{(a-b)+1} \rceil) \ne \emptyset$.

Proof: Given $(x_1, \ldots, x_a)$ we show how to enumerate $\le \lceil \frac{a+1}{(a-b)+1} \rceil$ possibilities such that one of them agrees with $C_a^K(x_1, \ldots, x_a)$ on at least b positions.

Let $k = \lceil \frac{a+1}{(a-b)+1} \rceil$, and let $I_1, \ldots, I_k$ be intervals of length at most $a-b+1$ that partition $\{0, \ldots, a\}$. (Notice that $k > 1$ because $b \ge 1$.) For each interval $I = [c, d]$ we enumerate a possibility that is based on the belief that $\#_a^K(x_1, \ldots, x_a) \in [c, d]$. By dovetailing these computations we enumerate at most k possibilities.

For interval $I = [c, d]$ we do the following. If $c = 0$ then output $(0, \ldots, 0)$. If $c > 0$ then simultaneously run all of $M_{x_1}(x_1), \ldots, M_{x_a}(x_a)$ until exactly c of them halt (this need not happen). Output a string that indicates that these c programs are in K and no other programs are in K.

We show that if $\#_a^K(x_1, \ldots, x_a) \in I = [c, d]$ then the possibility associated to I is correct. Clearly the c 1's are correct. Since there are at most d programs in K, at least $a-d$ of the 0's are correct. Hence at least $c+a-d = a + (c-d) = a + 1 - |I| \ge a + 1 - (a - b + 1) = b$ bits are correct. ∎

We show that the above bound is tight. For this we need the a-ary recursion theorem.

Theorem 5.3 *If* $1 \le b \le a$ *then* $freq^K_{b,a} \cap \mathrm{EN}(\lceil \frac{a+1}{(a-b)+1} \rceil - 1) = \emptyset$.

Proof: Assume, by way of contradiction, that there exists $f \in freq^K_{b,a} \cap \mathrm{EN}(\lceil \frac{a+1}{(a-b)+1} \rceil - 1)$. Assume that $f \in \mathrm{EN}(\lceil \frac{a+1}{(a-b)+1} \rceil - 1)$ via g. We create programs $x_1, \ldots, x_a$ that conspire to cause

$$(\forall \vec{v} \in W_{g(x_1,\ldots,x_a)})[\neg(\vec{v} =^{a-b} \mathrm{C}^K_a(x_1, \ldots, x_a))].$$

We plan to have different blocks of programs invalidate different elements of $W_{g(x_1,\ldots,x_a)}$. Let $k = \lceil \frac{a+1}{(a-b)+1} \rceil - 1$. Since $b \ge 1$ we have $k \ge 1$. Let $J_1, \ldots, J_k$ be intervals of length $\ge a - b + 1$ that partition $\{0, \ldots, a\}$.

By the a-ary recursion theorem we can assume that x_i has access to the numbers $\{x_1, \ldots, x_a\}$.

ALGORITHM FOR x_i

1. Let j be such that $i \in J_j$ (if no such j exists then diverge).
2. Enumerate $W_{g(x_1,\ldots,x_a)}$ until j elements appear (this step might not terminate). Let that jth element be $\vec{v} = b_1 \cdots b_a$.
3. If $b_i = 0$ then converge. If $b_i = 1$ then diverge.

END OF ALGORITHM

For all j, $1 \le j \le k$, if $W_{g(x_1,\ldots,x_a)}$ has the jth element $\vec{v}$, then $\vec{v}$ and $\mathrm{C}^K_a(x_1, \ldots, x_a)$ differ on the bits specified by J_j. Hence they differ on at least $a - b + 1$ places, so $(\forall \vec{v} \in W_{g(x_1,\ldots,x_a)})[\neg(\vec{v} =^{a-b} \mathrm{C}^K_a(x_1, \ldots, x_a))]$. ∎

6 Using Ramsey Theory

Clearly, $\#^A_n \in \mathrm{SEN}(n+1)$, since $\#^A_n(x_1, \ldots, x_n) \in \{0, \ldots, n\}$. We sketch the proof that if $\#^A_n \in \mathrm{EN}(n)$, then A is recursive. The full proof is in [20]. A different proof is in [21].

The following is a high-level description of the proof.

1. If $\#_n^A \in \mathrm{EN}(n)$, then there exists an infinite r.e. tree $\mathcal{T}$ (to be defined in Section 6.1) such that A is one of the infinite branches of $\mathcal{T}$.

2. If $\mathcal{T}$ is an infinite r.e. tree of a certain type, then all the infinite branches of $\mathcal{T}$ are recursive.

3. The infinite r.e. tree $\mathcal{T}$ in 1 is of the type alluded to in 2. To prove this, we will need a Ramsey-type theorem on trees.

6.1 Trees

Definition 6.1 Let $T \subseteq \{0,1\}^*$. T is a *tree* if

$$(\forall \sigma, \tau \in \{0,1\}^*)[(\sigma \in T \wedge \tau \prec \sigma) \Rightarrow \tau \in T].$$

Trees can be finite or infinite. The notions of recursive tree and r.e. tree are defined in the obvious way. We will denote finite trees by italicized letters like T and infinite trees by fancy letters like $\mathcal{T}$.

Example 6.2 Let $n \geq 1$, and let A be a set such that $\mathrm{C}_n^A \in \mathrm{EN}(n^2)$. Choose a recursive function $h \colon \mathsf{N}^n \to \mathsf{N}$ so that C_n^A is n^2-enumerable via h If $n > 1$, there may be sets C *other than* A such that C_n^C is n^2-enumerable via h. The set of *all* sets C (including $C = A$) such that C_n^C is n^2-enumerable via h is just the set of all infinite branches of the following tree:

$$\mathcal{T} = \{\sigma \in \{0,1\}^* : (\forall x_1, \ldots, x_n < |\sigma|)[\sigma(x_1)\cdots\sigma(x_n) \in W_{h(x_1,\ldots,x_n)}]\}.$$

Now $\mathcal{T}$ is r.e., since the definition of $\mathcal{T}$ can be written as

$$\mathcal{T} = \{\sigma \in \{0,1\}^* : (\exists s)(\forall x_1, \ldots, x_n < |\sigma|)[\sigma(x_1)\cdots\sigma(x_n) \in W_{h(x_1,\ldots,x_n),s}]\}.$$

We will want to show that certain trees $\mathcal{T}$ are severely restricted in some sense. We formalize 'severely restricted' by considering the embedding of finite trees in $\mathcal{T}$.

Definition 6.3 Let $k \geq 0$. The *full binary tree of depth k* (denoted by B_k) is the tree $\{\sigma \in \{0,1\}^* : |\sigma| \leq k\}$. Note that if $k > 0$, then the left and right subtrees of B_k are isomorphic to B_{k-1}.

Definition 6.4 Let $k \in \mathsf{N}$, let $\mathcal{T}$ be a nonempty tree (not necessarily infinite), and let $f\colon B_k \to \mathcal{T}$. (Recall that B_k is the full binary tree of depth k.) B_k is *embeddable in* $\mathcal{T}$ *via* f (and f is called an *embedding of* B_k *in* $\mathcal{T}$) if, for every internal node σ of B_k,

- $f(\sigma)$ has two children in $\mathcal{T}$ (hence $f(\sigma)$ is an internal node of $\mathcal{T}$),
- $f(\sigma)0 \preceq f(\sigma 0)$, and
- $f(\sigma)1 \preceq f(\sigma 1)$.

If f is an embedding of B_k in $\mathcal{T}$, then f is one-one, and the set $\{f(\sigma) : \sigma \in B_k\}$ (which is a subset of $\mathcal{T}$, but not necessarily a subtree of $\mathcal{T}$) is the *embedded* B_k; the node $f(\lambda)$ is the *root of the embedded* B_k.

B_k is *embeddable in* $\mathcal{T}$ if there is an embedding g of B_k in $\mathcal{T}$. For $\tau \in \mathcal{T}$, B_k *is embeddable in* $\mathcal{T}$ *at or below* τ if there is an embedding g of B_k in $\mathcal{T}$ such that $\tau \preceq g(\lambda)$. (We use the term 'below' for consistency with the way we are visualizing trees: The longer the node, the further it lies below the root.)

Note that B_0 (the full binary tree of depth 0) is embeddable in every nonempty tree $\mathcal{T}$, since $|B_0| = 1$.

Definition 6.5 Let $k \geq 0$.

1. A *2-coloring of* B_k (the full binary tree of depth k) is a function that maps the nodes of B_k into some 2-element set (e.g, into the set $\{RED, BLUE\}$ or the set $\{0, 1\}$).

2. Let c be a 2-coloring of B_k. Relative to c: Given $k' \leq k$, and given an embedding of $B_{k'}$ in B_k, the embedded $B_{k'}$ is *monochromatic* if c assigns the same color to all the elements of the embedded $B_{k'}$.

The following Ramsey-type theorem will be useful. It was first proven in [10].

Theorem 6.6 *Let* $k_1, k_2 \in \mathsf{N}$. *Then there exists a number* $r(k_1, k_2)$ *such that, for every 2-coloring of* $B_{r(k_1,k_2)}$ *(the full binary tree of depth* $r(k_1, k_2)$*) with RED and BLUE, there is either a RED embedded* B_{k_1} *or a BLUE embedded* B_{k_2} *in* $B_{r(k_1,k_2)}$. *Moreover,* $r(k_1, k_2) = k_1 + k_2$ *will suffice.*

Corollary 6.7 *Let* $k \in \mathsf{N}$. *Then for every 2-coloring of* B_{2k}*, there is a monochromatic embedded* B_k *in* B_{2k}*.*

6.2 A Lower Bound on the Complexity of $\#_n^A$

Lemma 6.8 *Let $\mathcal{T}$ be an infinite r.e. tree such that, for some $m \in \mathsf{N}$, B_m cannot be embedded in $\mathcal{T}$. Then every infinite branch of $\mathcal{T}$ is recursive.*

Theorem 6.9 *Let $n \geq 1$, and let A be a set such that $\#_n^A \in \mathrm{EN}(n)$. Then A is recursive.*

Proof sketch: Choose a recursive function g so that $\#_n^A$ is n-enumerable via g, i.e.,

$$(\forall x_1, \ldots, x_n)[|W_{g(x_1,\ldots,x_n)}| \leq n \wedge \#_n^A(x_1, \ldots, x_n) \in W_{g(x_1,\ldots,x_n)}].$$

There may be sets C *other than* A such that $\#_n^C$ is n-enumerable via g. The set of *all* sets C (including $C = A$) such that $\#_n^C$ is n-enumerable via g is just the set of all infinite branches of the following tree:

$$\mathcal{T} = \{\sigma : (\forall x_1, \ldots, x_n < |\sigma|)[\sum_{i=1}^{n} \sigma(x_i) \in W_{g(x_1,\ldots,x_n)}]\}.$$

(We are using the fact that, for every set B and all $x_1, \ldots, x_n \in \mathsf{N}$, we have $\#_n^B(x_1, \ldots, x_n) = \sum_{i=1}^n B(x_i)$.) Now $\mathcal{T}$ is r.e., since the definition of $\mathcal{T}$ can be expressed as follows:

$$\mathcal{T} = \{\sigma : (\exists s)(\forall x_1, \ldots, x_n < |\sigma|)[\sum_{i=1}^{n} \sigma(x_i) \in W_{g(x_1,\ldots,x_n),s}]\}.$$

We show that $\mathcal{T}$ satisfies the premise of Lemma 6.8, hence that every infinite branch of $\mathcal{T}$ is recursive. In particular, this will prove that A is recursive.

Suppose, by way of contradiction, that $(\forall m)[B_m$ is embeddable in $\mathcal{T}]$. We use this to show that there exist $z_1, \ldots, z_n \in \mathsf{N}$ and $\tau_0, \ldots, \tau_n \in \mathcal{T}$ such that, for every $j \leq n$, $\sum_{i=1}^n \tau_j(z_i) = j$. Hence $\{0, 1, \ldots, n\} \subseteq W_{g(z_1,\ldots,z_n)}$, in contradiction to the fact that $|W_{g(z_1,\ldots,z_n)}| \leq n$.

∎

7 Using Mindchanges

In this section we study how much convergence matters. We show here that, for sets decided with a bounded number of queries to K, the ability to diverge on incorrect answers does not add power. More succinctly $(\forall n)[\mathrm{Q}(n, K) = \mathrm{QC}(n, K)]$.

The proofs use mindchanges. Here is the intuition: rather than run a computation and ask questions (where wrong answers may lead to divergence) we will instead ask questions about the computation. Having done that, an incorrect response might lead to an incorrect answer, but not to divergence.

In this section we give a proof that uses the notion of a mindchange. In Section 8 we will combine mindchanges with $0''$-priority arguments.

Definition 7.1 Let A be an r.e. set, let $e, x, m \in \mathsf{N}$, and let $\{A_s\}_{s\in\mathsf{N}}$ be a recursive enumeration of A. $M_e^A(x)$ *changes its mind at least* m *times with respect to* $\{A_s\}_{s\in\mathsf{N}}$ if there exist $s_0, \ldots, s_m \in \mathsf{N}$ and $b \in \{0,1\}$ such that $s_0 < \cdots < s_m$ (if $m > 0$) and, for every $i \le m$,

$$\begin{aligned} i \text{ even} \quad &\Rightarrow \quad M_{e,s_i}^{A_{s_i}}(x)\downarrow = b, \\ i \text{ odd} \quad &\Rightarrow \quad M_{e,s_i}^{A_{s_i}}(x)\downarrow = 1-b. \end{aligned}$$

Lemma 7.2 *Let A be an r.e. set, let $n \ge 1$, let $X \in \mathrm{Q}(n, A)$, and let $e \in \mathsf{N}$ such that $X \in \mathrm{Q}(n, A)$ via M_e^A. Let*

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

Then the following hold.

1. *For every x, $M_e^A(x)$ changes its mind at most $2^n - 1$ times.*
2. *$B \in \mathrm{Q}(n, A)$ (hence $B \le_{\mathrm{T}} A$) and B is r.e.*

Theorem 7.3 *Let A be an r.e. set. Then $(\forall n \ge 1)[\mathrm{Q}(n, A) \subseteq \mathrm{QC}(n, K)]$. In particular, $(\forall n \ge 1)[\mathrm{Q}(n, K) = \mathrm{QC}(n, K)]$.*

Proof: Let $n \geq 1$. We show that $\mathrm{Q}(n, A) \subseteq \mathrm{QC}(n, K)$.

Let $X \in \mathrm{Q}(n, A)$. We show that $X \in \mathrm{QC}(n, K)$. Choose e so that $X \in \mathrm{Q}(n, A)$ via M_e^A. Let

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

Note that B is r.e. (by Lemma 7.2.2), hence $B \leq_{\mathrm{m}} K$.
ALGORITHM

1. Input x.
2. Find the least s such that $M_{e,s}^{A_s}(x) \downarrow\in \{0, 1\}$. (Such s exists, since $M_e^A(x) \downarrow\in \{0, 1\}$.) Let $b = M_{e,s}^{A_s}(x)$.
3. Note that $\langle x, 0\rangle \in B$, since $M_e^A(x) \downarrow\in \{0, 1\}$, and that (by Lemma 7.2.1) there exists $m' \leq 2^n - 1$ such that $\{\langle x, 0\rangle, \ldots, \langle x, m'\rangle\} \subseteq B$ and $\{\langle x, m' + 1\rangle, \langle x, m' + 2\rangle, \langle x, m' + 3\rangle, \ldots\} \subseteq \overline{B}$. Thus $M_e^A(x)$ changes its mind exactly m' times, and

   $$X(x) = \begin{cases} b, & \text{if } m' \text{ is even;} \\ 1 - b, & \text{otherwise.} \end{cases}$$

   So we wish to determine the parity of m'. We will actually compute m'.

   Using the fact that $B \leq_{\mathrm{m}} K$, obtain numbers $z_1, \ldots, z_{2^n-1}$ such that, for every m with $1 \leq m \leq 2^n - 1$, $\langle x, m\rangle \in B$ iff $z_m \in K$. Note that $m' = \#_{2^n-1}^K(z_1, \ldots, z_{2^n-1})$.
4. Compute $m' = \#_K^{2^n-1}(z_1, \ldots, z_{2^n-1})$ by a binary search that uses n queries to K. Note that even if incorrect answers are supplied, this computation converges.
5. If m' is even, output b; else output $1 - b$.

END OF ALGORITHM

∎

8 Using $0''$ Priority Arguments

In this section we sketch a theorem that uses a $0''$ priority argument. We present the motivation for using $0''$ and the construction; however, we leave the justification for the reader. The main point of our writeup is to show why a 0" seems to be needed.

In Theorem 7.3 we showed that $(\forall n \geq 1)[\mathrm{Q}(n,K) = \mathrm{QC}(n,K)]$. The question arises as to whether this property is special for K. We show that there exist incomplete r.e. sets that have the same property. The proof codes mind change information into a set A while trying to make A incomplete. The mindchange information is infinitary in nature, and hence leads to infinite injury.

The technique is inspired by Downey-Jockush [11]. In fact, the theorem could be derived from their work. However, our construction is more flexible in that we can add additional requirements. In particular, we can obtain a high set A, which their construction could not do (they proved that the sets they get must be low_2).

Definition 8.1 If $e \in \mathsf{N}$ then $M_e^{A(=n)}$ is oracle Turing machine M^A modified so that if it tries to ask more than n questions then it diverges.

Theorem 8.2 *Let $n \in \mathsf{N}$ and let C be any nonrecursive r.e. set. There exists a nonrecursive r.e. sets A such that $\mathrm{Q}(n,A) = \mathrm{QC}(n,A)$, and $C \not\leq_{\mathrm{T}} A$.*

Proof sketch:

We construct an r.e. set A to satisfy the following requirements.

$$\begin{array}{rl} N_e: & \{e\}^A \text{ total } \Rightarrow \{e\}^A \neq C \\ Q_e: & \{e\} \text{ total } \Rightarrow \{e\} \neq A \\ P_e: & M_e^{A(=n)} \text{ total } \Rightarrow M_e^{A(=n)} \in \mathrm{QC}(n,A) \end{array}$$

We discuss the requirements and how to satisfy them as if this were going to be a finite injury argument with priority ordering

$$N_0, Q_0, P_0, N_1, Q_1, P_1, \ldots.$$

In reality we are going to use a priority tree; however, discussing it as if it were finite injury will *motivate* the use of the priority tree.

We will meet the N_e requirement by using the standard method of preserving agreement (see [25]). We will meet the Q_e requirement by using the standard method of holding onto a witness x: if $\{e\}(x)\downarrow$ then diagonalize, and if $\{e\}(x)\uparrow$ then Q_e is satisfied with no action needed. Note that Q_e only enumerates a finite number of elements and hence causes only finite injury.

The alert reader may be asking herself "how come the P_e requirements cannot be broken down into an infinite number of finitary requirements called $P_{e,x}$, where $P_{e,x}$ codes the status of $M_e^{A(=n)}(x)$ into a $\mathrm{QC}(n,A)$ comptutation?" This would not work. In the end we have to have a $\mathrm{QC}(n,A)$ algorithm for $M_e^{A(=n)}$. This algorithm can of course use some finite information. If the requirement P_e gets injured finitely often then the $\mathrm{QC}(n,A)$ algorithm will indeed need some finite information. If we split it up, and each $P_{e,x}$ gets injured finitely often, then the algorithm would have to code infinite information.

We informally describe how to satisfy the P_e requirements. Let $x \in \mathsf{N}$. We take action on coding $M_e^{A(=n)}(x)$ into A the first time we spot $s \in \mathsf{N}$ such that, for $0 \le y \le x$, $M_{e,s}^{A_s(=n)}(y)\downarrow$. If this never occurs then P_e is satisfied since $M_e^{A(=n)}$ is not total. If this does occur then we declare 2^n-1 *traces* $tr(e,x,1),\ldots,tr(e,x,2^n-1)$. These traces are picked larger than all numbers seen so far. In later stages if we spot $M_e^{A(=n)}(x)$ changing its mind for the kth time then we enumerate $tr(e,x,k)$ into A. Note the following.

1. Since $M_e^{A(=n)}(x)$ asks n questions the number of mindchanges is $\le 2^n-1$. Hence we have declared enough traces.

2. If $M_e^{A(=n)}(x)$ is total and the traces are declared and enumerated as above then $M_e^{A(=n)} \in \mathrm{QC}(n,A)$ as follows. On input x find the least $s \in \mathsf{N}$ and $b \in \{0,1\}$ such that $M_{e,s}^{A_s(=n)}(x)\downarrow= b$ and 2^n-1 traces are declared. (Such an s exists since $M_e^{A(=n)}$ is total.) Using a binary search and queries to A find the least k such that $tr(e,x,k) \in A$ but $tr(e,x,k+1) \notin A$. Note that k is the number of mind changes $M_e^A(x)$ makes. If k is even then output b, else output $1-b$. Note that if incorrect query answers are supplied then this computation still converges (though it may be wrong).

3. P_e *does not* try to preserve $M_{e,s}^{A_s(=n)}(x)\downarrow$. P_e merely codes mindchange information into A.

4. If at stage $t > s$ we observe $M_{e,t}^{A_t(=n)}(x) \uparrow$, or $M_{e,t}^{A_t(=n)}(x) = M_{e,s}^{A_s(=n)}(x)$ but with different query answers then, *no trace is enumerated*. Traces are enumerated only when A changes *and* the new computation converges *and* the new result is different.

5. If $M_e^{A(=n)}$ is total then P_e may enumerate an infinite number of traces.

6. Assume $M_e^{A(=n)}$ is not total. Then there exists x_0 such that $M_e^{A(=n)}(x_0) \uparrow$. Since the $M_e^{A(=n)}(x_0)$ computation asks at most n queries there exists s_0 such that for all $s \geq s_0$ the computation of $M_{e,s}^{A_s(=n)}(x_0)$ uses correct information about A. Recall that in order for elements to be enumerated for the sake of $M_e^{A(=n)}(y)$ during stage s one must have $M_{e,s}^{A_s(=n)}(x) \downarrow$ for all $x < y$. Hence, during all stages $s \geq s_0$, no trace $tr(e, y, k)$ for $y \geq x_0$ will ever be enumerated. In brief, if $M_e^{A(=n)}$ is not total then P_e enumerates a finite number of traces.

At first glance it looks like the P-requirements of higher priority can injure N_e infinitely often. This is true; however, the reasons for it are subtle. Let $e' < e$. The following sequence of events shows how $P_{e'}$ might injure N_e. Assume $s_0 < s_1 < s_2 < s_3$.

1. During stage s_0 the computation $M_{e',s_0}^{A_{s_0}(=n)}(x) \downarrow = b$ is spotted and traces $tr(e', x, 1), \ldots, tr(e', x, 2^n - 1)$ are declared.

2. During stage s_1 q is enumerated into A and causes $M_{e,s_1}^{A_{s_1}(=n)}(x) \uparrow$. Note that this *does not* cause any trace to be enumerated.

3. During stage s_2 N_e acts and sets a restraint $r(e, s) > \max\{tr(e', x, k) : 1 \leq k \leq 2^n - 1\}$.

4. During stage s_3 $M_{e,s_3}^{A_{s_3}(=n)}(x) \downarrow = 1 - b$. $P_{e'}$ enumerates $tr(e', x, 1)$ which injures N_e.

The key problem is that some traces are associated to computations that *currently* are diverging but may soon converge; hence the traces *may* enter A.

Definition 8.3 Assume N_e wants to act at stage s. Let $tr(e', x, k)$ be a trace associated to $P_{e'}$, $e' \leq e$, and $M_{e',s}^{A_s(=n)}(x) \uparrow$. Then $tr(e', x, k)$ is called a *trace* *threatening* N_e or just a *threatening trace*. It *stops threatening at stage* t if $M_{e',t}^{A_t(=n)}(x)\downarrow$ and $tr(e',x,k)$ either goes in (if there was a mindchange) or not.

The problem N_e has with threatening traces is *uncertainty*. Consider the following optimistic scenarios.

1. N_e *knows* that $M_{e'}^{A(=n)}$ is not total. N_e can ignore the threatening traces declared by $P_{e'}$ since $P_{e'}$ will only enumerate a finite number of traces. Hence $P_{e'}$ may injure N_e finitely often which is tolerable.
2. N_e *knows* that $M_{e'}^{A(=n)}$ is total. If N_e wants to preserve the $\{e\}^A(x)$ then it will first note if any element queried in that computation is a threatening trace declared by $P_{e'}$. If so then N_e will not act now— N_e will wait until the trace stops threatening. This must happen since $M_{e'}^{A(=n)}$ is total.

The problem of course is that N_e *does not know* if $M_{e'}^{A(=n)}$ is total. The key idea is that we will have 2^e different *strategies* working on N_e; one for each combination of guesses as to which $M_{e'}^{A(=n)}$, $0 \le e' \le e$, are total. We have motivated using a priority tree.

Level 0 is the root. Level $3e$ ($3e+1, 3e+2$) will be concerned with requirement N_e (Q_e, P_e). At the levels associated to the P_e requirement the tree will branch both ways. At all other levels the nodes have only one outcome (i.e., one branch). The construction could be done without putting the Q_e nodes on the tree; however, this makes the construction easier to extend later.

At a node associated to P_e we think of going to the left as guessing "$M_e^{A(=n)}$ is total" and going to the right as guessing "$M_e^{A(=n)}$ is non-total." We think of the nodes associated to the N_e requirements as having *opinions* about the behavior of the $P_{e'}$ requirements with $e' \le e$. These opinions will be embodied in the definition of $\{\eta\}^A$ (Definition 8.7.4).

Convention 8.4 If η is a node and $|\eta| = 3e$ ($3e+1, 3e+2$) then N_η (Q_η, P_η) refers to both the requirement N_e (Q_e, P_e) and the attempt to satisfy N_e (Q_e, P_e) using the information and assumptions at node η. For example the statement 'N_η is satisfied' will mean that N_e is satisfied by the actions that happen at node η.

Convention 8.5 When we use the expression N_η (Q_η, P_η) we will be assuming that $|\eta| \equiv 0 \pmod 3$ ($|\eta| \equiv 1 \pmod 3$, $|\eta| \equiv 2 \pmod 3$).

Definition 8.6 If η and η' are nodes then η' *has higher priority than* η if either η is an ancestor of η' or η is on a branch to the right of the branch that η' is on. We denote that η' has higher priority than η by $\eta \leq_{pri} \eta'$. Note that if $\eta' \prec \eta$ then $\eta \leq_{pri} \eta'$.

We now describe the concepts needed for N_η's actions.

Definition 8.7 Let $|\eta| = 3e$.

1. If $\eta = \lambda$ then $\hat{\eta} = \lambda$. If $\eta = \eta_1 000$ then $\hat{\eta} = \hat{\eta}_1 0$. If $\eta = \eta_1 001$ then $\hat{\eta} = \hat{\eta}_1 1$. Let $\hat{\eta} = b_0 \cdots b_{e-1}$. Our intention is that η assumes that, for $0 \leq i \leq e$, $b_i = 1$ iff $M_i^{A(=n)}$ is total.

2. If for all $0 \leq i \leq e$, $b_i = 1$ iff $M_i^{A(=n)}$ is total, then η is *correct*.

3. Let i be such that $b_i = 1$. Let η_1 be such that $|\eta_1| = 3i + 2$ and $\eta_1 \prec \eta$ (i.e., N_η assumes $M_i^{A(=n)}$ is total and has to defer to P_{η_1}'s actions). Let $s \in \mathsf{N}$. Let $tr(\eta_1, x, k)$ be a trace declared for P_{η_1} during some stage $t \leq s$. If η is visited during stage s and $M_{i,s}^{A_s(=n)}(x) \uparrow$ then $tr(\eta_1, x, k)$ is a *trace threatening* N_η or just a *threatening trace*. (We do not care about traces associate to requirements of higher priority that are not prefixes of η since N_η believes they will be visited only finitely often. If N_η is wrong then a different node will be used to satisfy N_e.)

4. We define $\{\eta\}^A$ such that, if η is correct then $\{e\}^A = \{\eta\}^A$. We actually define $\{\eta\}_s^{A_s}(y)$ and take $\{\eta\}^A(y)$ to be the limit as $s \to \infty$. To calculate $\{\eta\}_s^{A_s}(y)$ simulate $\{e\}_s^{A_s}(y)$. If one of the queries made is a threatening trace threatening N_η then $\{\eta\}_s^{A_s}(y) \uparrow$; otherwise $\{\eta\}_s^{A_s}(y) = \{e\}_s^{A_s}(y)$.

5. We define the standard length of agreement and restraint functions. If η is visited during stage s then

$$\begin{aligned} l(\eta, s) = &\ \max\{x : (\forall y < x)[\{e\}_s^{A_s}(x) = C_s(x)];\\ r(\eta, s) = &\ \max\{use(\eta, A_s, y, s) : y < l(\eta, s)\}; \end{aligned}$$

   If η is not visited during stage s then $l(\eta, s) = l(\eta, s-1)$ and $r(\eta, s) = r(\eta, s-1)$ are undefined. The function $l(\eta, s)$ is called *the length of agreement*. The function $r(\eta, s)$ is called *the restraint function*.

The P_η and Q_η requirements will have to respect the restraints imposed by higher priority N-type requirements. Hence we have the following definition.

Definition 8.8 Let η be any node on the tree. Then

$$R(\eta, s) = \max\{r(\eta', s) : \eta \leq_{pri} \eta', |\eta'| \equiv 0 \pmod 3\}.$$

Note that this is the restraint imposed *on* node η, not the restraint imposed *by* node η. We could do the entire construction and proof only using $R(\eta, s)$ when $|\eta| \equiv 1, 2 \pmod 3$; however, we will use the $|\eta| \equiv 0 \pmod 3$ case for convenience.

The following facts will be useful. Their proofs are based soly on the shape of the priority tree and the definition of $R(\eta, s)$; they are left to the reader.

Fact 8.9 *Let* $\eta_1 \prec \eta \prec \eta' \prec \eta''$, $|\eta_1| = 3e - 1$, $|\eta| = 3e$, $|\eta'| = 3e + 1$, $|\eta''| = 3e + 2$. *Then*

1. $R(\eta, s)$ *is the maximum of* $\{r(\eta^*, s) : \eta^*$ *is to the left of* η $\}$ *and* $\{R(\eta_1, s)\}$.
2. $R(\eta, s) = R(\eta', s) = R(\eta'', s)$.

We now describe the concepts needed for P_η's actions. If the function $M_e^{A(=n)}$ looks like it is converging for longer and longer initial segments then it is worth taking action on. Hence we have the following definition.

Definition 8.10 Let $|\eta| = 3e + 1$, and $s \in \mathsf{N}$. If η is visited during stage s then

$$\begin{aligned} lc(\eta, s) = &\ \max\{x : (\forall y \leq x)[M_{e,s}^{A_s(=n)}(x) \downarrow]. \\ mlc(\eta, s) = &\ \max\{lc(\eta, s), mlc(\eta, s - 1)\}. \end{aligned}$$

If η is not visited during stage s then then $lc(\eta, s) = lc(\eta, s - 1)$ and $mlc(\eta, s) = mlc(\eta, s - 1)$. The function $lc(\eta, s)$ is called *length of convergence.* The function $mlc(\eta, s)$ is called *the maximum length of convergence.*

The numbers we use as witnesses for the Q_η and traces for the P_η are picked ahead of time. The following definition formalizes this.

Definition 8.11 Let $\{\widehat{Q_e} : e \in \mathsf{N}\}$ be a recursive partition of the evens into an infinite number of infinite recursive sets. Let $\{\widehat{P_\eta} : \eta \in \{0,1\}^*\}$ be a recursive partition of the odds into an infinite number of infinite recursive sets. The elements of $\widehat{Q_e}$ will be potential witnesses for any Q_η with $|\eta| = 3e + 1$. The elements of $\widehat{P_\eta}$ will be potential traces for P_η.

We now describe what information is stored at each node.

1. If $|\eta| = 3e$ then the only information stored at η is $r(\eta, s)$.
2. If $|\eta| = 3e + 1$ then the only information stored at η is an index for $\widehat{Q_e}$ and a Boolean variable SAT_e which is TRUE if *any* of the Q_η with $|\eta| = 3e + 1$ is satisfied.
3. If $|\eta| = 3e + 2$ then the only information stored at η is an index for $\widehat{P_\eta}$, $mlc(\eta, s)$, and the traces that have been declared.

Convention 8.12 When we use the expressions $r(\eta, s)$ we are implicitly assuming that $|\eta| \equiv 0 \pmod 3$. When we use the expressions $lc(\eta, s)$ and $mlc(\eta, s)$ we are implicitly assuming that $|\eta| \equiv 1 \pmod 3$.

CONSTRUCTION

Stage 0: $A_0 = \emptyset$. For all η set $r(\eta, 0) = 0$ and $mlc(\eta, 0) = 0$. For all e set $SAT_e = FALSE$.

Stage s: Visit nodes along a path of length s starting at the root.

Case 0: $|\eta| = 3e$. We work on N_η. Compute $r(\eta, s)$. Note that for all η' to the left of η we have $r(\eta', s) = r(\eta', s - 1)$ by definition of restraint. Hence $R(\eta, s)$ can be determined. By Fact 8.9, for all nodes η' such that $|\eta'| \equiv 1, 2 \pmod 3$, if η' is visited in this stage, the value of $R(\eta', s)$ will be known when that node is entered.

Case 1: $|\eta| = 3e + 1$. We work on Q_η. If $SAT_e = TRUE$ then do nothing and exit the node. Otherwise let

$$x = \mu y[y \in \widehat{Q_e} \wedge y \geq R(\eta, s)].$$

If $\{e\}_s(x) \downarrow= b$ then do the following. If $b = 0$ then put x into A. If $b = 1$ then do nothing but note that $x \notin A$. Set SAT_e to TRUE. If an element was enumerated which is less than $r(\eta', s)$ for some $\eta' \leq_{pri} \eta$ then $N_{\eta'}$ is said to be *injured.* If SAT_e is set to TRUE then we will say that the requirement Q_η *acts.*

Case 2: $|\eta| = 3e + 2$. We work on P_η.

I) (Take care of old traces.) For all $x \leq mlc(\eta, s)$ and $k \leq 2^n - 1$ if $M_{e,s}^{A_s(=n)}(x)$ has changed its mind k times (total) then enumerate all elements of $\{tr(\eta, x, 1), tr(\eta, x, 2), \ldots, tr(\eta, x, k)\}$ that are larger than $R(\eta, s)$ into A. (Some of these traces may already be in A from prior visits to η. For those traces no action is needed.) If an element was enumerated which is less than $r(\eta', s)$ for some $\eta' \leq_{pri} \eta$ then $N_{\eta'}$ is said to be *injured.*

II) (Plant new traces and exit.) Compute $lc(\eta, s)$ and $mlc(\eta, s)$. If $lc(\eta, s) \leq mlc(\eta, s-1)$ then exit η on the right. If $lc(\eta, s) > mlc(\eta, s-1)$ then do the following: (1) For every $x \leq mlc(\eta, s)$ such that no traces for x have been declared, declare traces $tr(\eta, x, 1), tr(\eta, x, 2), \ldots, tr(\eta, x, 2^n - 1)$ for x. These traces are the least elements of $\widehat{P_\eta}$ that are bigger than any number seen so far in the construction. (2) Exit on the left.

END OF CONSTRUCTION

Let η_s be the path traversed in stage s. Let

$$b_e = \begin{cases} 1 & \text{if } (\exists^\infty s)[b_0 b_1 \cdots b_{e-1} 1 \prec \hat{\eta}_s] \\ 0 & \text{otherwise.} \end{cases}$$

Let $\hat{\eta}_\infty = b_0 b_1 b_2 b_3 \cdots$. Let $\eta_\infty = b_0 00 b_1 00 b_2 00 b_3 00 \cdots$. We call η_∞ *the true path.*

The following lemmas can easily be established.

Lemma 8.13 *Every Q_η acts at most once and enumerates at most one element.*

Lemma 8.14 *Let $e \in \mathsf{N}$. Let η, η', and η'' be such that $\eta \prec \eta' \prec \eta'' \prec \eta_\infty$, $|\eta| = 3e$, $|\eta'| = 3e + 1$, and $|\eta''| = 3e + 2$. Then*

1. *η is correct. (See Definition 8.7.)*
2. *N_η is satisfied.*

3. *(a)* $\lim_{s\to\infty} R(\eta, s) < \infty$.
   (b) $\lim_{s\to\infty} R(\eta', s) < \infty$.
   (c) $\lim_{s\to\infty} R(\eta'', s) < \infty$.

4. $Q_{\eta'}$ *is satisfied.*

5. $P_{\eta''}$ *is satisfied.*

6. *If* $b_e = 0$ *then* $P_{\eta''}$ *enumerates only finitely many traces and* $M_e^{A(=n)}$ *is not total. If* $b_e = 1$ *then* $M_e^{A(=n)}$ *is total.*

∎

One can easily adjust this construction to get A high by replacing Q_e with the usual thickness requirements to obtain highness, and putting the two guesses (cofinite vs. finite) on the tree.

9 Acknowledgment

We would like to thank Andrew Lee and Maciek Smuga-Otto for proofreading and helpful discussion.

References

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

[2] R. Beigel, W. Gasarch, J. T. Gill, and J. C. Owings. Terse, superterse, and verbose sets. *Information and Computation*, 103:68–85, 1993.

[3] R. Beigel, W. Gasarch, and E. Kinber. Frequency computation and bounded queries. *Theoretical Computer Science*, pages 177–192, 1996.

[4] R. Beigel, W. Gasarch, M. Kummer, G. Martin, T. McNicholl, and F. Stephan. On the query complexity of sets. In *21st International Symposium on Mathematical Foundations of Computer Science (MFCS '96), Cracow, Poland*, 1996.

[5] R. Beigel, W. Gasarch, M. Kummer, G. Martin, T. McNicholl, and F. Stephan. On the complexity of $\mathrm{odd}_{\mathrm{n}}^{\mathrm{a}}$, 1997.

[6] J. Cai and L. A. Hemachandra. Enumerative counting is hard. *Information and Computation*, 82(1):34–44, July 1989.

[7] G. Cohen and P. Frankl. Good coverings of Hamming spaces with spheres. *Discrete Mathematics*, 56:125–131, 1989.

[8] G. Cohen, M. Karpovsky, and H. Mattson. Covering radius — survey and recent results. *IEEE Trans. Inform. Theory*, IT-31:338–343, 1985.

[9] G. Cohen, A. Lobstein, and N. Sloane. Further results on the covering radius of codes. *IEEE Trans. Inform. Theory*, IT-32:680–694, 1986.

[10] W. Deuber. A generalizationg of ramsey's theorem for regular trees. *Journal of Combinatorial Theory (Series B)*, 18:18–23, 1975.

[11] R. Downey and C. Jockusch. T-degrees, jump classes, and strong reducibilities. *Transactions of the AMS*, 301:103–120, 1987.

[12] W. 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.

[13] W. Gasarch and G. Martin. Bounded queries in recursion theory. Unpublished manuscript.

[14] W. I. Gasarch. A hierarchy of functions with applications to recursive graph theory. Technical Report 1651, University of Maryland, Dept. of Computer Science, 1985.

[15] L. Hay. Letters to bill gasarch, 1985.

[16] I. Honkala. Modified bounds for covering codes. *IEEE Trans. Inform. Theory*, IT-37:351–365, 1991.

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

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

[19] C. G. Jockusch and R. I. Soare. Π^0_1 classes and degrees of theories. *Transactions of the AMS*, 173:33–56, 1972.

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

[21] M. Kummer and F. Stephan. Effective search problems. *Mathematical Logic Quarterly*, 40, 1994.

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

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

[24] R. Smullyan. *Theory of Formal Systems*. Princeton University Press, Princeton, New Jersey, 1961. Annals of Mathematical Studies Vol 47.

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

[26] G. V. Wee. Improved sphere bounds on the covering radius of codes. *IEEE Trans. Inform. Theory*, IT-34:237–245, 1988.