ON BEING INCOHERENT WITHOUT BEING VERY HARD

Richard Beigel and Joan Feigenbaum

Abstract. Trakhtenbrot calls a set A *autoreducible* if A is Turing-reducible to A via an oracle Turing machine that never queries the oracle about the input string. Yao considers sets that are autoreducible via probabilistic, polynomial-time oracle Turing machines, and he calls such sets *coherent.* We continue the study of autoreducible sets, including those that are autoreducible via a family of polynomial-sized circuits, which we call *weakly coherent.* Sets that are not weakly coherent are called *strongly incoherent.* We show

∘ If s is superpolynomial and space-constructible, then there is a strongly incoherent set in DSPACE($s(n)$).

∘ If NEEEXP $\not\subseteq$ BPEEEXP, then there is a set in NP that is incoherent.

∘ If A is complete for any of the classes Σ_i^p, Π_i^p, or Δ_i^p, $i \geq 0$, then A is coherent. In particular, all NP-complete sets are coherent.

As corollaries, we obtain new lower bounds on *program checkability* and *random-self-reducibility.* These results answer open questions posed by Yao and by Blum and Kannan.

Key words. Autoreducibility, Incoherence, Program Checkability, Random-Self-Reducibility.

Subject classifications. 68Q15.

1. Introduction

Suppose that A is a set and that x is a string for which we would like to test membership in A. Is there a probabilistic polynomial-time algorithm that, given x and an oracle for A, can determine whether x is in A without asking the oracle "is x in A?"? That is, can membership of x in A be computed efficiently given knowledge of whether certain other strings are in A? If so, then A is said to be a *coherent* set. Almost all natural examples of sets, e.g., paddable sets, are coherent. If membership in A can be computed as above with the additional use of advice functions, then A is said to be *weakly coherent*; sets that are not weakly coherent are said to be *strongly incoherent*.

One goal of this work is to study the properties of coherent and weakly coherent sets, along with their relation to other central notions in complexity, particularly *checkability* [9] and *random-self-reducibility* [14]. Neither the random-self-reducible sets nor the checkable sets have been fully characterized. However checkable and uniformly-random self-reducible sets are coherent, and random-self-reducible sets are weakly coherent; therefore, coherence is a valuable tool in the study of checkability and random-self-reducibility. We obtain the following result about weak coherence.

Theorem: *If s is superpolynomial and space-constructible, then there is a strongly incoherent set in* DSPACE($s(n)$).

Since all sets in BPP are coherent, our space bound cannot be improved without separating BPP from PSPACE. Our technique can also be used to show that there are strongly incoherent sets "just above BPP" in the following sense: If $t(n)$ is superpolynomial and time-constructible, and O is BPP-hard, then there is an incoherent set in $\mathrm{DTIME}(t(n))^O$.

Under a reasonable complexity assumption, we construct incoherent sets in NP, thus answering a question of Yao [29]. Let NEEEXP denote the union, over all polynomials p, of $\mathrm{NTIME}(2^{2^{2^{p(n)}}})$, and define BPEEEXP analogously.

Theorem: *If* NEEEXP $\not\subseteq$ BPEEEXP, *then there is a set in* NP *that is incoherent.*

This is the first condition known to imply the existence of uncheckable sets in NEXP; it thus answers a question of Blum and Kannan [10]. It is also the first condition known to imply the existence of sets in PSPACE that are not uniformly-random self-reducible.

Of course, characterization of the coherent sets requires positive results as well. We provide the following result along those lines.

Theorem: *If A is complete for any of the classes Σ_i^p, Π_i^p, or Δ_i^p, $i \geq 0$, then A is coherent. In particular, all* NP*-complete sets are coherent.*

These results first appeared in our two technical reports [6, 7]. They were presented in preliminary form in [5], together with the results of Bellare and Goldwasser [8].

2. History

Trakhtenbrot [27] calls a set A *autoreducible* if A is Turing-reducible to A via an oracle Turing machine that never queries the oracle about the input string. He considered time- and space-bounded versions of autoreducibility as well. Lynch, Meyer, and Fischer [23] considered resource-bounded versions of autoreducibility in which the resource could be any Blum-complexity measure. Several polynomial-bounded versions of autoreducibility are particularly important in complexity theory. Motivated by previous work on uniformly-random self-reducibility [1, 3, 15, 26] and on efficient program-checking [9, 10], Yao [29] considered sets that are autoreducible via probabilistic, polynomial-time oracle Turing machines, and he called them *coherent*.

Trakhtenbrot proved two things that are valuable in the study of coherence. First he noted that $A \oplus A$ is autoreducible via a "very simple" autoreduction, where $B \oplus C = \{0x : x \in B\} \cup \{1x : x \in C\}$. In particular, $A \oplus A$ is coherent. Second he announced a proof that for every space-constructible function s there is a set in DSPACE(s) that is not autoreducible via a $o(s)$-space-bounded oracle Turing machine. Because probabilistic polynomial-time oracle Turing machines can be simulated by polynomial-space oracle Turing machines, it follows that there are incoherent sets in DSPACE($s(n)$) for any superpolynomial, space-constructible s.

Although the construction of non-autoreducible sets is not given explicitly in [27], Traktenbrot used a fairly straightforward diagonalization argument [28]. For standard reasons, any string queried at one stage of the construction cannot be used later in the diagonalization. We point out here that one can do this construction by diagonalizing at points $x_1, x_2, \ldots$ where $|x_{n+1}|$ grows faster than any polynomial in $|x_n|$. In order to perform the diagonalization in a small amount of space, it is important that we construct a set that has low Kolmogorov complexity, e.g., a tally set. The condition on $|x_{n+1}|$ and $|x_n|$ requires that the tally set be, in fact, very sparse. The very-sparseness condition turns out to be useful in and of itself in constructing incoherent sets in NP under reasonable complexity theoretic assumptions; we define it precisely in Section 3, and we exploit it in Section 4.

3. Preliminaries

An *examiner* M is an oracle Turing machine that, on input x, is not permitted to query whether x belongs to the oracle set. A set A is *coherent* if there exists a bounded-error probabilistic polynomial-time (BPP) examiner using oracle A that recognizes A (cf. [29]). A is *deterministically coherent* if there exists a deterministic polynomial-time examiner using oracle A that recognizes A. A is *weakly coherent* if there exists a P/poly examiner (also called a *weak* examiner) using oracle A that recognizes A. The class P/poly consists of all sets recognizable by polynomial-time Turing machines that take polynomial-length advice, or alternatively by polynomial-sized circuits.

If A is not weakly coherent then A is called *strongly incoherent*. Recall that BPP/poly = P/poly, because randomness can be incorporated into nonuniform advice using standard techniques [2]. Hence, there is no need to consider weak probabilistic examiners, because, for a fixed oracle, they are equivalent to weak deterministic examiners. In particular, for a fixed oracle, every BPP examiner is a P/poly examiner, and so every coherent set is weakly coherent.

A *tally* set is a subset of 0^*. A *superpolynomial function* $t(n) : \mathsf{N} \to \mathsf{N}$ is one for which $\lim_{n\to\infty} t(n)/n^c = \infty$, for all positive constants c. A *very fast growing function* $s(n) : \mathsf{N} \to \mathsf{N}$ is one such that for all polynomials p, for all sufficiently large n, $s(n+1) > p(s(n))$. For example, the function

$$s(n) = 2^{2^{n^2}}$$

is very fast growing. A function s is called *well-behaved* if $s(x)$ is computable in time linear in the length of $s(x)$, $Range(s)$ is decidable in linear time, and s^{-1} is computable in linear time on $Range(s)$, where we use the standard binary string representation for integers. A *very sparse tally set* is a subset of $\{0^{s(n)} : n \in \mathsf{N}\}$, where $s(n)$ is very fast growing and well-behaved.

A *length-decreasing, probabilistic (resp. deterministic) self-reduction* for A is a BPP (resp. P) oracle machine N with the following properties: With oracle A, the set accepted by N is A; on input x, N only queries the oracle about strings shorter than x.

FACT 3.1. *If a tally set has a length-decreasing, probabilistic (resp. deterministic) polynomial-time self-reduction, then it is in* BPP *(resp.* P*).*

FACT 3.2. (YAO [29]) *All checkable sets are coherent.*

By Trakhtenbrot's observation, there are arbitrarily complex, coherent sets: For all A, $A \oplus A$ is coherent.

4. Coherent, Incoherent, and Strongly Incoherent Sets

We now construct an incoherent set in NP, assuming that NEEEXP is not contained in BPEEEXP. Building on a result of Yao [29], we construct a strongly incoherent set in DSPACE($s(n)$) for any superpolynomial, space-constructible s — without any assumptions. Finally, we prove that all sets that are complete for Σ_i^p, Π_i^p, or Δ_i^p, $i \geq 0$, are coherent. In particular, all NP-complete sets are coherent.

4.1. Incoherent Sets of Low Complexity. In this section, we exploit sparseness in order to construct incoherent sets in NP, under a plausible complexity assumption. By contrast, in Section 4.3 we will prove that all NP-complete sets are coherent.

LEMMA 4.1. *If A is a very sparse tally set and A is coherent, then $A \in$ BPP.*

PROOF. Assume that A is a very sparse tally set and A is coherent. We will prove that A has a length-decreasing, probabilistic polynomial-time self-reduction. The conclusion then follows from Fact 3.1.

Because A is a very sparse tally set, there is, by definition, a very fast growing, well-behaved function $s(n)$ such that $A \subseteq \{0^{s(n)} : n \in \mathsf{N}\}$. Because A is coherent, there is a BPP examiner M using oracle A that recognizes A. We use s and M to construct r, an appropriate self-reduction for A.

On input x, the reduction r first rejects x if $x \notin 0^*$ or $|x| \notin \mathit{Range}(s)$. If $x \in 0^*$ and $|x| \in \mathit{Range}(s)$, then r simulates M on input x. Let $\langle y_1, \ldots, y_m \rangle$ be the sequence of strings for which M, on input x, asks the oracle "is this string in A?" Because M is polynomial-time bounded and s is very fast growing, each query y_i such that $|y_i| > |x|$ is not in A, because its size falls between two elements of $\mathit{Range}(s)$. Because A is a tally set, each query y_i such that $|y_i| = |x|$ is not in A (by definition, the examiner M does not query the oracle about x, and all other strings of length $|x|$ are not in 0^*). Thus r probabilistically reduces membership of x in A to membership in A of a polynomial-length sequence of strings, each of which is shorter than x. □

We use Lemma 4.1 to show, under a plausible assumption, that there is an incoherent set in NP. This answers Yao's question (a) (see [29, Section 6]).

THEOREM 4.2. *If* NEEEXP $\not\subseteq$ BPEEEXP, *then there is an incoherent set in* NP.

PROOF. Assuming that NEEEXP is not contained in BPEEEXP, we will construct a very sparse tally set in NP − BPP and then appeal to Lemma 4.1. We use a standard padding argument in the style of Book [13] and Hartmanis-Sewelson-Immerman [18].

First note that the hypothesis implies that there is a set in $\mathrm{NTIME}\left(2^{2^{2^n}}\right)$ that is not in BPEEEXP. To see this, take a set S in $\mathrm{NTIME}\left(2^{2^{2^{n^c}}}\right) -$ BPEEEXP and let $S' = \{x10^{|x|^c} : x \in S\}$.

Let S' be a set in $\mathrm{NTIME}\left(2^{2^{2^n}}\right) - \mathrm{BPEEEXP}$, and consider the tally set

$$S'' \equiv \{0^{1x} : x \in S'\},$$

where we use the symbol $1x$ to denote the positive integer whose binary representation is $1x$. Because the time needed to decide whether 0^{1x} is in S'' is exactly the time needed to decide whether x is in S', and because $|x| \leq \log_2(|0^{1x}|)$, we have that

$$S'' \in \mathrm{NTIME}\left(2^{2^{2^{\log n}}}\right) = \mathrm{NTIME}\left(2^{2^n}\right).$$

Now let $s(n) = 2^{2^{n^2}}$ and let

$$A = \{0^{s(n)} : 0^n \in S''\}.$$

Note that s is very fast growing and well-behaved, and hence A is a very sparse tally set. Furthermore,

$$A \in \mathrm{NTIME}(n) \subset \mathrm{NP}.$$

So it remains to show that A is not in BPP.

If A were in $\mathrm{BPTIME}(n^c)$, then S'' would be in

$$\mathrm{BPTIME}\left(\left(2^{2^{n^2}}\right)^c\right) = \mathrm{BPTIME}\left(2^{2^{n^2+\log c}}\right).$$

This would imply that S' was in

$$\mathrm{BPTIME}\left(2^{2^{2^{(2n+2)}+\log c}}\right) \subseteq \mathrm{BPEEEXP},$$

thus contradicting the hypothesis. □

Next, we apply Lemma 4.1 to construct an incoherent set without any assumptions. Note that the same result is obtainable as a corollary of Trakhtenbrot's theorem [27]. We will prove a stronger result in Section 4.2.

Theorem 4.3. *If $s(n)$ is superpolynomial and space-constructible, there is an incoherent set in* $\mathrm{DSPACE}(s(n))$.

PROOF. Standard diagonalization techniques suffice to construct a very sparse tally set A such that

$$A \in \mathrm{DSPACE}(s(n)) - \mathrm{PSPACE} \subseteq \mathrm{DSPACE}(s(n)) - \mathrm{BPP}.$$

By Lemma 4.1, A is incoherent. $\square$

Furthermore, we can use Lemma 4.1 to construct an incoherent set "just above BPP." Since every set in BPP is coherent, this complexity bound can hardly be improved.

THEOREM 4.4. *If $t(n)$ is superpolynomial and time-constructible and O is* BPP*-hard, then there is an incoherent set in* $\mathrm{DTIME}(t(n))^O$.

PROOF. Standard diagonalization techniques suffice to construct a very sparse tally set in $\mathrm{DTIME}(t(n))^O - \mathrm{P}^O$, for any oracle O. If O is BPP-hard, then this tally set must be incoherent. This follows directly from Lemma 4.1 and the fact that $\mathrm{BPP} \subseteq \mathrm{P}^O$. $\square$

COROLLARY 4.5. *If $t(n)$ is superpolynomial and time-constructible, then there is an incoherent set in* $\mathrm{DTIME}(t(n))^{\Sigma_2^p}$.

PROOF. All Σ_2^p-complete sets are BPP-hard. $\square$

4.2. Strongly Incoherent Sets of Low Complexity. In this section we construct strongly incoherent sets in $\mathrm{DSPACE}(s(n))$ for any superpolynomial, space-constructible s. The incoherent sets constructed in the previous section are sparse and belong to NP under a plausible assumption. By contrast, the sets constructed in this section are not polynomial-time reducible to any sparse set; however, we do not know of any plausible condition that permits us to construct strongly incoherent sets in NP.

The proof requires a combinatorial result due to Yao [29]. Consider the following game between two players called P_1 and P_2, with parameters N, m, and t.

1. Player P_1 chooses a sequence of N bits, $b_1, \ldots, b_N$.
2. Player P_2 inspects $b_1, \ldots, b_N$ and stores m bits of information, which we call her *pad* p. Player P_2 may not remember anything else about $b_1, \ldots, b_N$.

3. Player P_1 chooses i between 1 and N and asks P_2 for the value of b_i.

4. Player P_2 refers to her pad p and inspects b_j for t values of j other than i. Player P_2 then answers P_1's question. (Player P_2's strategy may be adaptive but must be deterministic.)

Player P_2 *wins* if she answers correctly. We say that player P_2's strategy is *a winning strategy for* N, m, t if she wins no matter how P_1 plays.

FACT 4.6. (YAO [29]) *Player P_2 has a winning strategy if and only if $(t+1)m$ is greater than or equal to N.*

Player P_2's pad corresponds to advice in the proof below.

THEOREM 4.7. *If $s(n)$ is superpolynomial and space-constructible, then there is a strongly incoherent set in* DSPACE$(s(n))$.

PROOF. Let $m(n) = t(n) = \lfloor\sqrt{s(n)}\rfloor - 1$. Note that $m()$ and $t()$ dominate every polynomial on all but finitely many points. Let M_e denote the e^{th} oracle Turing machine, running in time $t(n)$ with advice of length $m(n)$. If A is weakly coherent, then there exist e and p such that M_e, using oracle A and advice p, decides $x \in A$ correctly for all but finitely many x, without actually querying the oracle about x.

Let $N(n) = s(n)$. Then $(t(n)+1)m(n) < N(n)$. Let $\langle \cdot, \cdot \rangle$ denote a pairing function on the natural numbers. We construct a strongly incoherent set A by the following initial segment argument.

Stage -1**:** Let $A = \emptyset$. Let $n = 3$.

Stage $\langle e, i \rangle$**:** Let $n = \max(n, t(n)) + 1$. Let X be the set containing the first $N(n)$ strings of $\{0,1\}^n$ in lexicographic order. Find a subset A_n of X such that for every advice string p of length $m(n)$, there exists $x \in X$ such that on input x with advice p and oracle $A \cup A_n$

- M_e queries x, or
- M_e accepts x and $x \notin A_n$, or
- M_e rejects x and $x \in A_n$.

If such a set A_n exists at every stage, then clearly the construction guarantees that A differs infinitely often from every weakly coherent language, so A is strongly incoherent. But if A_n did not exist then player P_2 would have a winning strategy in Yao's game above with parameters $N(n), t(n), m(n)$. Finally we note that the set A_n can be found by exhaustive search using space proportional to $N(n) + t(n) + m(n) = O(s(n))$, so $A \in \mathrm{DSPACE}(s(n))$. $\square$

In fact, the language A constructed above is in $(\mathrm{DTIME}(s(n)))^{\Sigma_2^p}$.

THEOREM 4.8. *If $s(n)$ is superpolynomial and time-constructible, then there is a strongly incoherent set in* $(\mathrm{DTIME}(s(n)))^{\Sigma_2^p}$.

4.3. Coherence and Completeness. Coherence enabled us to find plausible examples of uncheckable sets in NP. However, it is unknown whether NP-complete sets are checkable. Coherence will not help us determine the answer, because of the following theorem.

THEOREM 4.9. *Let $\mathcal{C}$ be a complexity class that has a polynomial-time Turing complete set that is length-decreasing, probabilistically (resp. deterministically) self-reducible. Let A be polynomial-time Turing complete for $\mathcal{C}$. Then A is coherent (resp. deterministically coherent).*

PROOF. Let K be a polynomial-time Turing complete set for $\mathcal{C}$, and let r_{KK} be a length-decreasing self-reduction for K. Let r_{AK} (resp. r_{KA}) be a polynomial-time Turing reduction from A to K (resp. from K to A). An examiner for A proceeds as follows.

M starts by simulating r_{AK} on input x. Whenever it reaches a point at which it must know whether, say, y is in K, it starts to simulate r_{KA} on input y. Recall that M has access to an A-oracle. Thus, if r_{KA} can decide whether y is in K without asking whether x is in A, the examiner M has made progress and can proceed with the simulation of r_{AK} on input x. Otherwise, M can try both possible values for $\chi_A(x)$. If the value obtained for $\chi_K(y)$ is the same in both cases, then M knows that this answer for $\chi_K(y)$ is correct and once again proceeds with the simulation of r_{AK} on input x.

The only other possibilities are $x \in A \leftrightarrow y \in K$ or $x \in A \leftrightarrow y \notin K$, and thus it suffices for M to determine $\chi_K(y)$. In this case, M aborts the simulation of r_{AK}. Instead, it applies r_{KK} to y. Once again, M has made progress, because all subsequent values $\chi_K(y')$ that it tries to determine satisfy $|y'| < |y|$. $\square$

Corollary 4.10. *If A is polynomial-time Turing complete for any class Σ_i^p, Π_i^p, Δ_i^p, $i \geq 0$, then A is deterministically coherent. In particular, all* NP*-complete sets are deterministically coherent.*

Note that Theorem 4.9 also implies the coherence of PSPACE-complete sets. However, this follows from the fact that those sets are checkable [25].

5. Checkability

From a complexity-theoretic point of view, checkability provides an interesting new perspective on the (presumed) difference between *computing* a value and *verifying* the result of a computation. Traditionally, this issue is represented by the (presumed) gap between P and NP $\cap$ coNP. If one also allows randomness and interaction to be used in the computation or verification, then the issue is represented by the (presumed) gap between BPP and what Blum [9] calls "checkable sets" — the latter gap is more dramatic, because EXP-complete sets are checkable [4].

The following definition of checkability is equivalent to the one given by Blum [9]. A set A is *checkable* if there is a probabilistic, polynomial-time oracle machine C (called the *checker*) with the following properties. On input x, the first thing C does is query the oracle about membership of x. Thereafter, C's goal is to "check" whether the oracle answered this first question correctly. Let $C(O, x)$ denote the random variable computed by C with oracle O on input x. For all x, $C(A, x) = \mathit{correct}$, with probability at least 3/4. For all B and all x such that $\chi_A(x) \neq \chi_B(x)$, $C(B, x) = \mathit{faulty}$ with probability at least 3/4. (Note that $C(B, x)$ can be either *correct* or *faulty* if $B \neq A$ but $\chi_A(x) = \chi_B(x)$.) The checker's error probability can be made exponentially small using standard amplification techniques. If the oracle machine C is deterministic, checking is equivalent to *self-helping* (cf. Schöning [24] and Ko [20]).

Fortnow, Rompel, and Sipser showed that A has a multi-prover interactive proof system (MIP) if and only if it has a single-prover proof system in which the prover's answers to a question q from the verifier depend only on q and not on previous questions [16, Theorem 1]. Thus, when the provers get together before the protocol to "agree on a strategy," their agreement amounts to choosing a function f that they will use to answer the verifier's questions; in response to question q, they will always send $f(q)$. Given this result, it is easy to see (and is remarked in [16]) that, if A has an MIP system, A must be in NEXP.

In the terms we have used to describe Fortnow, Rompel, and Sipser's results, A has a program checker if and only if both A and its complement have MIP systems in which the function f computed by the provers is χ_A. Thus it follows that all checkable sets are in NEXP $\cap$ coNEXP.

It is known that PSPACE-complete sets and EXP-complete sets are checkable and thus coherent [4, 22, 25]. However, it remains open whether NP-complete sets (and, more generally, the complete sets at each level of the polynomial hierarchy) have program checkers.

Yao [29] observed that all checkable sets are coherent. Furthermore, if A has a multi-prover interactive proof system in which the provers simply answer questions about A then it is easy to see that A is coherent. In this section, we apply some results from Section 4 to obtain negative results about checkability.

We obtain the following corollary to Theorem 4.2.

Corollary 5.1. *If* NEEEXP $\not\subseteq$ BPEEEXP, *then there exists a set A in NP such that*

- *A is uncheckable, and furthermore*
- *in any MIP system for A, the correct provers must have more power than simply enough to decide membership in A.*

Corollary 5.2. *If* NEEEXP $\not\subseteq$ BPEEEXP, *then there is an uncheckable set in* IP.

Corollary 5.2 provides half of the answer to the question of Blum and Kannan [10]. The other half is provided by Babai, Fortnow, and Lund [4], who show that there is a checkable set that is not in IP, unless EXP = PSPACE.

We were the first to use sparse sets in the study of checkability [6, 7]. Subsequently, Bellare and Goldwasser [8] used sparse sets in the direct construction of an uncheckable set B in NP, under the weaker assumption that NEE $\not\subseteq$ BPEE. (EE denotes the union, over all positive constants c, of DTIME$(2^{2^{cn}})$. NEE and BPEE denote the analogous nondeterministic and bounded-probabilistic time classes.) Note that Bellare and Goldwasser's set B is not necessarily incoherent; so it might be necessary to assume that NEEEXP $\not\subseteq$ BPEEEXP in order to construct an incoherent set in NP.

Sparse sets were also used by Impagliazzo and Sudan [19] to derive negative results on reductions from NP-search problems to NP-decision problems.

By Theorem 4.7 we have

COROLLARY 5.3. *If $s(n)$ is superpolynomial and space-constructible, there is an uncheckable set in* DSPACE$(s(n))$.

6. Random-Self-Reducibility

Random-self-reducibility and uniformly-random self-reducibility are important concepts in complexity theory [11, 12, 14, 15], especially in connection with program checking and secure protocols. If a set is *uniformly-random self-reducible* in the sense of [15], then it is coherent. If a set is *random-self-reducible* in the sense of [14], then it is weakly coherent. Thus, all of our negative results about coherence yield negative results about random-self-reducibility as well.

In this section, we examine the connection between coherence and random-self-reducibility. To simplify our discussion, we use slightly different terminology from that of [14, 15], where the definitions given in this section first appeared. Close connections between random-self-reducibility and program checking have been demonstrated by Blum, Luby, and Rubinfeld [11, 12].

Intuitively, a set A is random-self-reducible if an arbitrary membership query "is x in A?" can be reduced in polynomial time to the evaluation of one or more *randomly distributed* membership queries "is y_i in A?". This notion is of central importance in many areas of theoretical computer science, including average-case complexity, lower bounds, interactive proof systems, zero-knowledge, instance-hiding, and pseudorandom number generation; refer to [14] for a thorough discussion of these connections.

DEFINITION 6.1. (FEIGENBAUM-FORTNOW [14]) *A set A is random-self-reducible (abbreviated rsr) if there is a probabilistic, polynomial-time oracle machine ϕ with the properties given below. Let $q_i^B(x,r)$ denote the i^{th} query asked by ϕ when the oracle is B, the input is x, and the random tape holds r. Let $p(n)$ be an upper bound on the number of queries asked by ϕ when $|x| = n$, and let r be chosen uniformly from $\{0,1\}^{w(n)}$, where w is a polynomially bounded function of n.*

- *For all x, for at least $3/4$ of all r, $\chi_A(x) = \phi^A(x,r)$, the result of running ϕ with oracle A, input x, and random tape r.*
- *For all x and y such that $|x| = |y|$, for all i such that $1 \le i \le p(|x|)$, the random variables $q_i^A(x,r)$ and $q_i^A(y,r)$ are identically distributed.*

(Note that we do not require that $q_i^B(x,r)$ and $q_i^B(y,r)$ be identically distributed if $B \neq A$.)

For example, complete sets for PSPACE and EXP are rsr [14].

DEFINITION 6.2. (FEIGENBAUM-KANNAN-NISAN [15]) *A set A is uniformly-random self-reducible (abbreviated ursr) if it has a random-self-reduction in which each random variable $q_i^A(x,r)$ is uniformly distributed over $\{0,1\}^{|x|}$.*

Uniformly-random self-reducible sets are treated at length in [15].

Definitions 6.1 and 6.2 are phrased in terms of sets but can be extended *mutatis mutandis* to functions. One celebrated example of a uniformly-random self-reducible function is the permanent of matrices over finite fields [21].

FACT 6.3. *All ursr sets coherent.*

PROOF. Let A be ursr. Using standard amplification techniques, we can reduce to 2^{-n} the probability that the function ϕ outputs a wrong answer. An examiner for A can choose r, query the A-oracle to obtain the answers $a_1 = \chi_A(q_1^A(x,r)), \ldots, a_{p(n)} = \chi_A(q_{p(n)}^A(x,r))$, and output $\phi(x,r,a_1,\ldots,a_{p(n)})$. This examiner only fails if ϕ fails or at least one of the $q_i^A(x,r)$'s is equal to x. By definition of ursr, this probability is at most $(p(n)+1)/2^n$, which suffices to show that A is coherent. □

FACT 6.4. *All rsr sets are weakly coherent.*

PROOF. Let A be rsr. For each n, we will construct an advice string that can be used by an examiner for A. As in Fact 6.3, start by amplifying so that the error probability of ϕ is at most 2^{-n}.

For each i, there are at most $5p(n)$ instances y such that the probability that $q_i^A(x,r) = y$ is at least $1/5p(n)$, where x is an arbitrary element of $\{0,1\}^n$ and r is chosen uniformly at random. The advice for length n consists of y and $\chi_A(y)$, for all such y, for $1 \le i \le p(n)$.

Using this advice, the examiner proceeds as follows on input x. It chooses r and starts running the reduction on input x and random string r. When it reaches a query $y_i = q_i^A(x,r)$, it looks up $a_i = \chi_A(y_i)$ in the advice if $y_i = x$; otherwise, it obtains a_i by querying the oracle. It outputs $\phi(x,r,a_1,\ldots,a_{p(n)})$.

This examiner fails if some $y_i = x$ but is not in the advice, or if ϕ errs. Thus the probability of failure is less than $1/5 + 2^{-n}$. □

Corollary 6.5. *The results in Section 4 have the following implications for random-self-reducibility.*

- *If $s(n)$ is superpolynomial and space-constructible, then there is a set in* DSPACE($s(n)$) *that is not rsr.*
- *If $t(n)$ is superpolynomial and time-constructible, then there is a set in* $\mathrm{DTIME}(t(n))^{\Sigma_2^p}$ *that is not rsr.*

The first of these results is a direct improvement of Feigenbaum, Kannan, and Nisan's construction [15] of a set in DSPACE(2^n) that is not ursr.

Finally, we remark that these notions, although related, are provably different. It is easy to show that there are uniformly-random self-reducible sets that are not checkable, that there are coherent sets that are not checkable, and that there are coherent sets that are not uniformly-random self-reducible. It is unknown, however, whether there is a checkable set that is not random-self-reducible; there are specific checkers that do not use random-self-reductions as defined here (e.g., the graph-isomorphism checker in [10]), but the sets in question may be random-self-reducible.

For NP-complete sets, we have the following behavior:

- All NP-complete sets are coherent.
- If any NP-complete set is nonadaptively random-self-reducible, then the polynomial hierarchy collapses at the third level [14].
- It is unknown whether NP-complete sets are checkable.

7. Open Questions

Open questions include:

- Are NP-complete sets checkable?
- Is there a plausible assumption that implies the existence of strongly incoherent sets in the polynomial hierarchy?

Many interesting questions about random-self-reducibility remain open as well. Refer to Feigenbaum and Fortnow [14] for details.

Acknowledgements

The first author was supported in part by NSF grants CCR-8808949 and CCR-8958528.

We thank Andy Yao for his suggestion that we look at coherence and the polynomial hierarchy. We thank Lance Fortnow, Jeremy Kahn, and Peter Shor for suggesting improvements and corrections to an earlier version of the paper. Finally, we thank Manuel and Lenore Blum for their support and encouragement.

References

[1] M. Abadi, J. Feigenbaum, and J. Kilian, On Hiding Information from an Oracle, *J. Comput. System Sci.* **39** (1989), 21–50.

[2] L. Adleman, Two Theorems on Random Polynomial Time, in *Proc. of the 19th Symposium on Foundations of Computer Science* (1978), IEEE, 75–83.

[3] D. Angluin and D. Lichtenstein, *Provable Security of Cryptosystems: A Survey*, YALEU/DCS/TR-288, 1983.

[4] L. Babai, L. Fortnow, and C. Lund, Nondeterministic Exponential Time has Two-Prover Interactive Protocols, *Computational Complexity* **1** (1991), 3–40.

[5] R. Beigel, M. Bellare, J. Feigenbaum, and S. Goldwasser, Languages that are Easier than their Proofs, in *Proc. of the 32nd Symposium on Foundations of Computer Science* (1991), IEEE, 19–28.

[6] R. Beigel and J. Feigenbaum, *On the Complexity of Coherent Sets*, AT&T Bell Laboratories Technical Memorandum, February, 1990.

[7] R. Beigel and J. Feigenbaum, *Improved Bounds on Coherence and Checkability*, Yale University Technical Report, YALEU/DCS/TR-819, September, 1990.

[8] M. Bellare and S. Goldwasser, *The Complexity of Decision versus Search*, MIT-LCS Technical Memorandum, TM-444, April, 1991.

[9] M. Blum, *Designing Programs to Check Their Work*, Intl. Comput. Sci. Inst. Technical Report, 88-009, 1988.

[10] M. Blum and S. Kannan, Designing Programs that Check Their Work, in *Proc. of the 21st Symposium on Theory of Computing* (1989), ACM, 86–97.

[11] M. Blum, M. Luby, and R. Rubinfeld, Self-Testing/Correcting with Applications to Numerical Problems, in *Proc. of the 22nd Symposium on Theory of Computing* (1990), ACM, 73–83.

[12] M. Blum, M. Luby, and R. Rubinfeld, Program Result Checking against Adaptive Programs and in Cryptographic Settings, in *Distributed Computing and Cryptography*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science **2** (1991), AMS, 107–118.

[13] R. Book, Tally Languages and Complexity Classes, *Inf. and Control* **26** (1974), 186–193.

[14] J. Feigenbaum and L. Fortnow, On the Random-Self-Reducibility of Complete Sets, in *Proc. of the 6th Structure in Complexity Theory Conference* (1991), IEEE, 124–132.

[15] J. Feigenbaum, S. Kannan, and N. Nisan, Lower Bounds on Random-Self-Reducibility, in *Proc. of the 5th Structure in Complexity Theory Conference* (1990), IEEE, 100–109.

[16] L. Fortnow, J. Rompel, and M. Sipser, On the Power of Multiprover Interactive Protocols, in *Proc. of the 3rd Structure in Complexity Theory Conference* (1988), IEEE, 156–161.

[17] S. Goldwasser, S. Micali, and C. Rackoff, The Knowledge Complexity of Interactive Proof Systems, *SIAM J. Comput.* **18** (1989), 186–208.

[18] J. Hartmanis, V. Sewelson, and N. Immerman, Sparse Sets in NP$-$P: EXPTIME versus NEXPTIME, in *Proc. of the 15th Symposium on Theory of Computing* (1983), ACM, 382–391.

[19] R. Impagliazzo and M. Sudan, Private communication, March, 1991.

[20] K. Ko, On Helping by Robust Oracle Machines, *Theor. Comput. Sci.* **52** (1987), 15–36.

[21] R. LIPTON, New Directions in Testing, in *Distributed Computing and Cryptography*, DIMACS Series on Discrete Mathematics and Theoretical Computer Science **2** (1991), American Mathematical Society, 191–202.

[22] C. LUND, L. FORTNOW, H. KARLOFF, AND N. NISAN, Algebraic Methods for Interactive Proof Systems, in *Proc. of the 31st Symposium on Foundations of Computer Science* (1990), IEEE, 2–10.

[23] N. A. LYNCH, A. R. MEYER, M. J. FISCHER, Relativization of the Theory of Computational Complexity, *Trans. Amer. Math. Soc.* **220** (1976), 243-287.

[24] U. SCHÖNING, Robust Algorithms: A Different Approach to Oracles, *Theor. Comput. Sci.* **40** (1985), 57–66.

[25] A. SHAMIR, IP = PSPACE, in *Proc. of the 31st Symposium on Foundations of Computer Science* (1990), IEEE, 11–15.

[26] M. TOMPA AND H. WOLL, Random Self-Reducibility and Zero-Knowledge Interactive Proofs of Possession of Information, in *Proc. of the 28th Symposium on Foundations of Computer Science* (1987), IEEE, 472–482.

[27] B. TRAKHTENBROT, On Autoreducibility, *Doklady Akad. Nauk. SSSR* **192** (1970), 1224–1227 — in Russian. Translation in *Soviet Math. Dokl.* **11** (1970), 814–817.

[28] B. TRAKHTENBROT, Autoreducible and not autoreducible predicates and sets, in *Studies in Theory of Algorithms and Mathematical Logic*, Academy of Sciences of the USSR — Computing Center, A. Markov and N. Petri (eds.), 1973, 211–234 — in Russian.

[29] A. C. YAO, Coherent Functions and Program Checkers, in *Proc. of the 22nd Symposium on Theory of Computing* (1990), ACM, 84–94.

Manuscript received 20 September 1990

RICHARD BEIGEL
Yale University
Department of Computer Science
P. O. Box 2158, Yale Station
New Haven, CT 06520-2158 USA
beigel-richard@cs.yale.edu

JOAN FEIGENBAUM
AT&T Bell Laboratories
Room 2C-473
600 Mountain Avenue
Murray Hill, NJ 07974-0636 USA
jf@research.att.com