Lower Bounds on Random-Self-Reducibility

(Extended Abstract – Structures-1990 Proceedings)

Joan Feigenbaum[*] Sampath Kannan[†] Noam Nisan[‡]

Abstract: Informally speaking, a function f is *random-self-reducible* if, for any x, the computation of $f(x)$ can be reduced to the computation of f on other "randomly chosen" inputs. Such functions are fundamental in many areas of theoretical computer science, including lower bounds, pseudorandom number-generators, interactive proof systems, zero-knowledge, instance-hiding, program-checking, and program-testing. Several examples of random-self-reductions are quite well-known and have been applied in all of these areas.

In this paper we study the limitations of random-self-reducibility and prove several negative results. For example, we show unconditionally that random boolean functions do not have random-self-reductions, even of a quite general nature. For several natural, but less general, classes of random-self-reductions, we show that, unless the polynomial hierarchy collapses, nondeterminstic polynomial-time computable functions are not random-self-reducible.

1 Introduction

Informally speaking, a function f is *random-self-reducible* if, for any x, the computation of $f(x)$ can be reduced to the computation of f on other "randomly chosen" inputs. For concreteness, consider the well-known example of extracting square roots modulo some number N. To compute $\sqrt{x} \bmod N$, one may instead choose a random number r and compute $\sqrt{xr^2} \bmod N$. From this information, $\sqrt{x} \bmod N$ is easily recovered by dividing by r. More general forms of random-self-reducibility may also be defined, allowing, e.g., reductions to several random instances of the function.

Random-self-reducible functions are important in complexity theory and in many applications. For example:

Worst-case hardness implies average-case hardness: Random-self-reducible problems are as hard on average as they are in the worst case. Angluin and Lichtenstein [2] pointed out that this is why they yield good candidates for one-way functions. Blum and Micali [5] used the random-self-reducibility of the "discrete log" function to construct pseudorandom number-generators. Babai [3] used the random-self-reducibility properties of the "parity" problem to obtain a simple proof that a random oracle separates the polynomial hierarchy from PSPACE (an earlier proof by Cai [14] does not use random-self-reducibility).

Program checking, testing, and correcting: If a program computes a random-self-reducible function correctly on *most* inputs, it can be used to compute the function correctly, with high probability, on *all* inputs. Blum, Luby, and Rubinfeld [10] call a program *self-testing* if it is guaranteed to be correct on "random instances"; they use random-self-reducibility to make self-testing programs "correct themselves." The work in [10] builds on *program-checking*, as defined by Blum and Kannan [9], in which

[*] AT&T Bell Laboratories, Room 2C473, 600 Mountain Avenue, Murray Hill, NJ 07974 USA, jf@research.att.com.

[†] Comp. Sci. Division, U. of California, Berkeley, CA 94720 USA, kannan@ernie.berkeley.edu.

[‡] Comp. Sci. Dept., Hebrew University, Jerusalem, ISRAEL, noam@humus.huji.ac.il. Work done at the MIT Laboratory for Comp. Sci.

random-self-reducibility also plays a prominent role. Lipton [23] shows how programs that purport to compute random-self-reducible-functions may be efficiently "tested," where the notion of testability is similar to the notion of correctability in [10].

Interactive proof systems, zero-knowledge, and cryptographic applications: The earliest examples of zero-knowledge proof systems all involve random-self-reductions; see, e.g., Goldwasser, Micali, and Rackoff's original paper on zero-knowledge [20], as well as the subsequent papers of Galil, Haber, and Yung [17], Goldreich, Micali, and Wigderson [19], and Brassard, Chaum, and Crépeau [13]. The practical authentication scheme proposed by Feige, Fiat, and Shamir [15] is also based on random-self-reducibility. Tompa and Woll [28] make explicit the connection between random-self-reducibility and perfect zero-knowledge. Most importantly, the recent characterizations of the power of interactive proof systems (cf. [4, 24, 25, 27]) use random-self-reducibility; in particularly, they use the random-self-reduction of the permanent function exhibited in [6, 23].

Computing with encrypted data: Abadi, Feigenbaum, and Kilian [1] consider the question of whether a weak computer can exploit the resources of a more powerful (but insecure) computer without revealing too much information about its private data. Beaver and Feigenbaum [6] generalize this concept of *instance-hiding schemes* to allow several powerful computers; they show that *all* functions can be computed by multiple powerful computers, none of which obtains the weak computer's input. Beaver, Feigenbaum, Kilian, and Rogaway [7] strengthen these results and give some applications to distributed computation. In each of these works, random-self-reducibility provides natural examples of problems that can be computed with encrypted data.

In this paper, we study the concept of random-self-reducibility in its own right. Preliminary work along these lines can be found in [1, 2, 28], where f is said to be random-self-reducible if computing $f(x)$ can be reduced to computing f on a *single* random input. We consider the more general notion of reducing the computation of $f(x)$ to the computation of f on several random inputs. The importance of this generalization has been demonstrated dramatically by some of the recent works just mentioned (i.e., [4, 6, 7, 10, 23, 24, 25, 27]). Building on results of Beaver and Feigenbaum [6], Lipton [23] shows that some very computationally-complex functions (including the permanent function) have random-self-reductions of this general form. Unless the polynomial hierarchy collapses, such highly complex functions do not have the simpler form of random-self-reductions in which the real instance is mapped to a single random instance (a weak form of this negative result was shown in [1], and a stronger form is shown here). One interpretation of the result in [6, 23] is that every function can be extended to a random-self-reducible function by extending its domain and range. Thus, the natural question to ask is whether every function (with its original domain and range) has a random-self-reduction of this general, multiple-random-instance variety. Similarly, one interpretation of the results of [6, 7] is that every function f has a "random reduction" to *some* function g. Thus, it is natural to ask whether it is possible to take $g = f$. These questions are interesting in their own right and are also motivated by potential applications. Recent works have provided applications of random-self-reducibility to complexity [4, 24, 25, 27] and to self-testing, self-checking, and self-correction of programs [10, 23]; presumably new results on random-self-reducibility would find more applications in complexity theory and in practice.

In this paper, we provide negative answers to these questions. We postpone formal definitions until Section 3, and now state our main results informally.

- Random boolean functions do not have random-self-reductions in which the original instance is mapped to a polynomial number of random instances. Our proof also shows that there is such a non-random-self-reducible boolean function in DSPACE(2^n).

- Every boolean function that has a random-self-reduction in which the original instance is mapped to two random instances is in nonuniform NP $\cap$ coNP. (This question is still open for the case of three random instances or, in general, for constants $k > 2$.)

- Following [1], we consider random-self-reductions of language-membership problems that are one-sided (i.e., they randomize only "yes-instances" of the language). We show that SAT does not have such a reduction unless the polynomial hierarchy collapses; this settles an open question of [1], in which a similar result was proven for $\overline{\text{SAT}}$.

- Following [1], we also consider reductions that randomize only some of the bits of the input instance and leave the rest of the bits fixed. We show that SAT and $\overline{\text{SAT}}$ do not have such reductions unless the polynomial hierarchy collapses. Our result for $\overline{\text{SAT}}$ is best possible and gives a great improvement on the result in [1]: even a self-reduction that randomizes a constant number of the bits of each yes-instance is precluded unless the polynomial hierarchy collapses.

Sections 2 and 3 below contain precise notation, terminology, and definitions for the concepts discussed above. Our main results are given in Section 4. Section 5 contains a brief discussion of very recent related results, and Section 6 contains open problems.

The results given here first appeared in our Technical Memorandum [16]. In what follows, some details of proofs have been omitted because of space limitations; they will appear in the full paper.

2 Preliminaries

We first fix notation for the following concepts, with which we assume familiarity on the part of the reader.

The class of total functions computable in deterministic polynomial time is denoted fP; when restricted to boolean functions, this is just the language class P. The class of total functions computable in nondeterministic polynomial time is denoted fNP; the boolean subclass is NP $\cap$ coNP. Relevant nonuniform versions of these classes are, respectively, fP/*poly*, P/*poly*, fNP/*poly*, and NP/*poly* $\cap$ coNP/*poly*. PH denotes the polynomial hierarchy.

We denote by IP(k) the languages recognizable by k-round interactive-proof systems (cf. Goldwasser, Micali, and Rackoff [20]) and by AM(k) those recognizable by k-round Arthur-Merlin games (cf. Babai and Moran [5]). Relevant nonuniform versions are IP(k)/*poly* and AM(k)/*poly*.

If f is a function on Σ^*, then f_n denotes the restriction of f to inputs of length n; the set of all such inputs is denoted Σ^n. A *random boolean function on* Σ^n is one chosen uniformly from the sample space of size 2^{2^n}. A *random boolean function* f (on Σ^*) is sampled by choosing f_n uniformly, independently for each n.

Throughout this paper, n is the length of the input x, and r is a uniformly chosen random element of $\{0,1\}^m$, where m is bounded by a polynomial in n.

3 Definitions

Here we give formal definitions for the concepts introduced in Section 1.

Definition 3.1 *A* k-random-self-reduction *(abbreviated k-rsr) for a function f is a collection of functions ϕ, σ_1, ..., σ_k in fP with the following properties.*

- *For all x and r, $f(x) = \phi(x, r, f(\sigma_1(x,r)), \ldots, f(\sigma_k(x,r)))$.*

- *For all n and all $x \in \Sigma^n$, if r is chosen uniformly from $\{0,1\}^m$, then $\sigma_i(x,r)$ is uniform over Σ^n, for all i such that $1 \le i \le k$.*

Remark: For $i \neq j$, the random variables $\sigma_i(x,r)$ and $\sigma_j(x,r)$ are, in general, dependent.

Remark: The parameter k is, in general, a function of $n = |x|$. The function ϕ is polynomial-time computable, but the total length of its input may be superpolynomial in n if $k(n)$ is superpolynomial.

Remark: A function f that is k-random-self-reducible, in our language, is "randomly-testable of order k over fP" in the language of [23].

Clearly, the notion of nonuniform random-self-reducibility also makes sense. That is, the functions ϕ and σ_i, $1 \leq i \leq k$ can be computed by circuit families instead of TM's.

Definition 3.2
A nonuniform (k,s)-random-self-reduction *for a function f is a collection of functions ϕ, computed by circuit family $\{C_n\}_{n=0}^{\infty}$, and σ_i, $1 \leq i \leq k$, computed by circuit families $\{D_{i,n}\}_{n=0}^{\infty}$, $1 \leq i \leq k$, satisfying the two conditions of Definition 3.1 and the condition that all circuit-sizes $|C_n|$, $|D_{1,n}|$, …, $|D_{k,n}|$ are at most $s(n)$. (The circuits $D_{i,n}$ take as input x and the random bitstring r. The circuit C_n takes x, r, and the values computed by the $D_{i,n}$'s.)*

Both k-rsr's and nonuniform (k,s)-rsr's can also be generalized in the following way: for each x, require only that the probability that $f(x) = \phi(x, r, f(\sigma_1(x,r)), \ldots, f(\sigma_k(x,r)))$ be at least 2/3. In this case we say that f *has a k-rsr (resp. a nonuniform (k,s)-rsr) that makes errors.*

Random-self-reductions are a special case of *locally random reductions* and, even more generally, *instance-hiding schemes.* These notions were defined for $k = 1$ by Abadi, Feigenbaum, and Kilian [1] and for $k > 1$ by Beaver and Feigenbaum [6] and by Beaver, Feigenbaum, Kilian, and Rogaway [7]. For convenience, we include the following, which is a special case of a definition given in [7].

Definition 3.3
A $(1,k)$-locally random reduction *of f to g is a collection of functions ϕ, σ_1, …, σ_k in fP with the following properties.*

- *For all x and r, $f(x) = \phi(x, r, g(\sigma_1(x,r)), \ldots, g(\sigma_k(x,r)))$.*
- *There is a polynomially bounded function $w(n)$ such that, for all n and all $x \in \Sigma^n$, if r is chosen uniformly from $\{0,1\}^m$, then the random variable $\sigma_i(x,r)$ is uniform over $Dom(g) \cap \Sigma^{w(n)}$, for all i such that $1 \leq i \leq k$.*

Clearly a $(1,k)$-locally random reduction is a k-rsr if $g = f$ and $w(n) = n$. Beaver and Feigenbaum [6] show that every function f has a $(1, n - \log n)$-locally random reduction with $w(n) = O(n \log n)$. This general upper bound was improved to $k(n) = n/\log n$, $w(n) = O(n \log n)$ by Beaver, Feigenbaum, Kilian, and Rogaway [7].

Remark: Random-self-reductions, locally random reductions, and instance-hiding schemes can be restricted to f_n in a straightforward manner. Hence, we often use the phrases "f_n is k-random-self-reducible" or "f_n is nonuniformly (k,s)-random-self-reducible" to mean the obvious thing.

Remark: As in random-self-reducibility, the parameter k in locally random reducibility and random-testability is, in general, a function of $n = |x|$.

Notation: Denote by k-RSR the set of boolean functions that have k-rsr's, and denote by poly-RSR the union, over all polynomials $k(n)$, of the sets k-RSR.

We now restrict attention to set-membership problems. We consider two types of self-reductions that are weaker than rsr's. Both types were introduced by Abadi, Feigenbaum, and Kilian [1], but the notation used in [1] is different from that used here.

Definition 3.4 *A function $\sigma \in$ fP is a* one-sided 1-rsr *for a set S if is it length-preserving, membership-preserving, and has the property that, for all n and all $x \in \Sigma^n$, if r is chosen uniformly from $\{0,1\}^m$, then $\sigma(x,r)$ is uniformly distributed on $S \cap \Sigma^n$.*

Thus, a one-sided 1-rsr achieves perfect randomization on yes-instances, but may not randomize no-instances at all.

The requirements of Definitions 3.1 and 3.4 can be weakened as follows. For each n, the elements of Σ^n are partitioned into equal-sized *orbits.* Let $O(x)$ denote the orbit of x. Then $\sigma_i(x,r)$ is distributed uniformly on $O(x)$. If there is more than one orbit for each Σ^n, we speak of a *partial k-rsr* or a *one-sided partial 1-rsr*.

Often a partial rsr σ has the property that all elements of an orbit share a suffix.[1] Under these conditions, the action of σ has the following interpretation: let O be a σ-orbit all of whose elements share the suffix v. For any $x \in O$, σ *fixes* v and it *randomizes* the rest of x. The shorter v is in comparison to x, the larger the orbits, the fewer orbits there are, and the closer σ comes to being an rsr. If σ has exactly one orbit, then $|v| = 0$, and σ is an rsr.

Definition 3.5 *A function is a* p-partial rsr *of* S *if it satisfies all of the above conditions and* $p = (n - |v|)/n$.

Definition 3.5 makes sense for both two-sided and one-sided rsr's. Like k, the parameter p is, in general, a function of n. By definition, $0 \leq p(n) \leq 1$. Also, partial random-self-reductions may be computed by circuits as well as TM's (and the notation carries over).

Perhaps the best-known example of a random-self-reducible set S is the set of quadratic residues with Jacobi symbol 1 modulo composites that are the product of two primes (refer to, e.g., [2, 11, 13, 17, 20] for applications). The standard reduction is a (1/2)-partial 1-rsr in which the common suffix is the modulus; that is, the modulus is fixed by the reduction and the residue is randomized. It is also two-sided, i.e., it is a (1/2)-partial 1-rsr of $\overline{S}$. See [1] for an example (based on the graph-isomorphism problem) of a set S with a one-sided (1/2)-partial 1-rsr in which $\overline{S}$ does not seem to have a one-sided p-partial 1-rsr with $p \geq 1/2$.

Finally, note that there *are* random-self-reducible functions at arbitrarily high levels of the time hierarchy. For example, the characteristic function of $\{x : |x|$ encodes a Turing Machine that halts on all inputs$\}$ is 1-rsr, but it is not recursive. See [1, §4] for a longer discussion of this issue.

4 Results

4.1 Random functions are not in poly-RSR

Theorem 4.1 *There is a constant* $c > 1$ *such that, for all polynomials* $k(n)$ *and* $s(n)$*, for all sufficiently large* n*, the probability that a random boolean function* f_n *is nonuniformly* (k, s)*-random-self-reducible is less than* 2^{-c^n}.

Proof (sketch): We use a counting argument to compute the probability that a random f_n satisfies a weaker condition, which we call nonuniform (k, s)-self-reducibility.

Say that the functions ϕ_n and $\sigma_{i,n}$, $1 \leq i \leq k$, constitute a *nonuniform (k,s)-self-reduction* (abbreviated (k, s)-sr) for f_n if they are computable by circuits of size at most $s(n)$ and together satisfy the following two conditions. For all $x \in \Sigma^n$, for all $1 \leq i \leq k$, $\sigma_{i,n}(x) \neq x$. For all $x \in \Sigma^n$, $f_n(x) = \phi_n(f_n(\sigma_{1,n}(x)), ..., f_n(\sigma_{k,n}(x)))$. It is not difficult to show that, if f_n has a nonuniform (k, s)-rsr, then it also has a nonuniform (k, s)-sr. **Claim**: There exist constants $c_1 > 1$ and $c_2 > 1$, such that, for all sufficiently large n, if $k(n) < c_1^n$ and $s(n) < c_1^n$, then the probability that a random f_n on Σ^n has a nonuniform (k, s)-sr is at most $2^{-c_2^n}$.

To see why this claim holds, fix a set of circuits for $\{\phi_n, \sigma_{1,n}, \ldots, \sigma_{k,n}\}$. Note that the number of choices for such a set of circuits is approximately $2^{c_1^{2n}}$. We prove the claim by showing that the probability that $\{\phi_n, \sigma_{1,n}, \ldots, \sigma_{k,n}\}$ is a k-sr for f_n is sufficiently small.

Choose at random $2^n/k$ inputs in Σ^n, and fix the value of f_n on all other inputs in an arbitrary manner. By a simple argument, a constant fraction of the chosen $2^n/k$ inputs x have the property that $f_n(\sigma_{i,n}(x))$ is already determined, for all $1 \leq i \leq k$. Say that this constant fraction is bounded below by c_3^n (c_3 depends on c_1). If we flip a coin to determine the value of $f_n(x)$, we will be correct with probability only 1/2 for any chosen x; thus $\{\phi_n, \sigma_{1,n}, \ldots, \sigma_{k,n}\}$ is a k-sr with probability at most $2^{-c_3^n}$. The claim follows, because we can choose c_1 and c_2 so that $2^{c_1^{2n} - c_3^n} < 2^{-c_2^n}$. ∎

[1] Clearly this statement is only meaningful with respect to an agreed-upon encoding of S. Refer to Garey and Johnson [18, Chapter 2] for a discussion of encodings. We assume that all of the sets we consider are encoded "in a standard way," i.e., as in [18, Chapter 2].

Corollary 4.1 *The class* poly-RSR *has measure 0 in the class of all boolean functions.*

Corollary 4.2 *There is a constant $c > 1$ such that, for all polynomials $k(n)$, the probability that a random boolean function f on Σ^* has a k-random-self-reduction that makes errors is less than 2^{-c^n}.*

Proof (sketch): If f has a k-rsr that makes errors, then there is a polynomial s such that, for all sufficiently large n, f_n has an (errorless) nonuniform (k, s)-rsr. ▌

Corollary 4.3 *There is a boolean function f in* DSPACE(2^n) *that is not nonuniformly (k, s)-sr, for any polynomials k and s. A fortiori, there is one that is not nonuniformly (k, s)-rsr. Furthermore, there is a deterministic exponential-space procedure to find such an f.*

Proof (sketch): Essentially, the proof of Theorem 4.1 can be made into an exponential-space, exhaustive-search procedure P that maps n to f_n. The search is over all possibilities for the set of $2^n/k$ inputs, the initial values of f_n on elements not in the set, and the circuits for the self-reduction ϕ_n, $\sigma_{i,n}$, $1 \leq i \leq k$. The resulting f is in DSPACE(2^n), because $f(x)$ can be computed by running P on $n = |x|$ to get f_n and then outputting $f_n(x)$. ▌

Remark: Recall that *every* function on Σ^* has a $(1, n/\log n)$-locally random reduction (cf. Beaver, Feigenbaum, Kilian, and Rogaway [7]). These reductions map length-n instances of f to random elements of $Dom(g) \cap \Sigma^{n \log n}$, where $g \neq f$. Theorem 4.1 and Corollary 4.3 show that the upper bound $k(n) = n/\log n$ could not be achieved by a length-preserving locally random reduction that required g to be equal to f (nor could any polynomial upper bound).

Remark: Theorem 4.1 and Corollary 4.3 also show that random functions are not randomly testable of order k over fP (in the sense of Lipton [23]), for any polynomial function $k(n)$, and that such a non-randomly-testable function can be found in DSPACE(2^n).

Remark: Beaver and Feigenbaum [6, Lemma 2.2] observe that random boolean functions have no 1-oracle instance-hiding schemes (and hence no 1-rsr's). This is, to our knowledge, the only previously published lower bound on the random-self-reducibility of random functions.

4.2 Functions with 2-random-self-reductions are in nonuniform fNP

Theorem 4.2 *If f has a 2-rsr, then it is in* fNP/*poly.*

Proof (sketch): Let ϕ, σ_1, and σ_2 constitute a 2-rsr for f. The goal is to find, for each n, a polynomial-sized set of subset $\{x_1, \ldots, x_m\}$ of Σ^n such that $f(x)$ can be deduced from $\{f(x_1), \ldots, f(x_m)\}$ using the 2-rsr. The pairs x_i, $f(x_i)$ will be given as polynomial advice.

For any subset $\{x_1, \ldots, x_m\}$ of Σ^n, the set $SPAN(x_1, \ldots, x_m)$ is defined inductively as follows. It is the union of $SPAN(x_1, \ldots, x_{m-1})$ and the set of all x such that there exists a coin-toss sequence r for which $\sigma_1(x, r) = z$ and $\sigma_2(x, r) = x_m$, where $z \in SPAN(x_1, \ldots, x_{m-1})$. If such an r exists, we say that x reduces to $\langle z, x_m \rangle$.

It suffices to prove that there exists a sequence $(x_1, \ldots, x_m)$ with $SPAN(x_1, \ldots, x_m) = \Sigma^n$ and m polynomial in n.

Claim: Let $S(i)$ denote $|SPAN(x_1, \ldots, x_i)|$. For any $(x_1, \ldots, x_{m-1})$, there exists a choice of x_m for which

$$S(m) - S(m-1) \geq \frac{S(m-1)(2^n - S(m-1))}{2^n}. \tag{1}$$

Thus the size of the span approximately doubles every time we increase m by 1. Once $S(m)$ is greater than $|\Sigma^n|/2$, we are done (this follows from the definition of 2-rsr).

Proof of claim: If $S \subset \Sigma^n$ and $x \notin S$, let $N_S(x)$ be the set of w such that x can be reduced to $\langle z, w \rangle$, with $z \in S$. Observe that the definition of 2-rsr implies that, for any such S and x, $|N_S(x)| \geq |S|$. To see this, note that, if $\sigma_1(x, r) \in S$, then $\sigma_2(x, r) \in N_S(x)$. That is, $\mathrm{Prob}_{r \in \{0,1\}^m}(\sigma_2(x, r) \in N_S(x)) \geq$

$\mathrm{Prob}_{r\in\{0,1\}^m}(\sigma_1(x) \in S)$. Because each $\sigma_i(x,r)$ is uniform over Σ^n, this, in turn, implies that $|N_S(x)| \geq |S|$.

Now set $S = SPAN(x_1, \ldots, x_{m-1})$ and choose x_m at random. For any $x \notin S$, the probability that x_m is in $N_S(x)$ is at least $p = |S|/2^n$. Thus the expected growth in the size of the span is p times the maximum possible growth, namely $|S|(2^n - |S|)/2^n$. ▌

Corollary 4.4 *If* SAT *has a 2-rsr, then the* PH *collapses at the third level.*

Proof: Recall that the boolean subclass of fNP is NP/*poly* ∩ coNP/*poly*. Then apply Theorem 4.2 and a well-known theorem of Yap [31] that, if NP ⊆ coNP/*poly*, the PH collapses at the third level. ▌

4.3 One-sided random-self-reductions

Theorem 4.3 is a special case of the main result of [1]; we include it for completeness.

Theorem 4.3 *If S has a one-sided 1-rsr, then $S \in$* NP/*poly.*

Corollary 4.5 *If* $\overline{\mathrm{SAT}}$ *has a one-sided 1-rsr, then the* PH *collapses at the third level.*

Proof: This follows directly from Theorem 4.3 and Yap's theorem [31]. ▌

Theorem 4.4 *If S has a one-sided 1-rsr, then $S \in$* coNP/*poly.*[2]

Proof (sketch): Suppose that σ is a one-sided 1-rsr for S. We show that $\overline{S} \in \mathrm{IP}(2)/poly$. The theorem then follows from results of Goldwasser and Sipser [21] and Babai and Moran [5] that $\mathrm{IP}(2) \subseteq \mathrm{AM}(4) \subseteq \mathrm{AM}(2) \subseteq \mathrm{NP}/poly$.

For each n, the verifier V is given as advice one element y_n of $S \cap \Sigma^n$ (or the fact that $S \cap \Sigma^n$ is empty).

Let x be a string of length n; prover P wants to convince V that $x \in \overline{S}$. If V's advice string says that $S \cap \Sigma^n$ is empty, then V simply accepts x. Otherwise, V computes $x' = \sigma(x, r_1)$ and $y' = \sigma(y_n, r_2)$, where r_1 and r_2 are chosen uniformly and *independently* from $\{0,1\}^m$, and sends $\{x', y'\}$ to P, challenging P to select the element of $\overline{S}$. If $x \in \overline{S}$, then $x' \in \overline{S}$ and $y' \in S$; so an honest P always succeeds. However, if $x \in S$, then x' and y' are both uniformly-distributed random elements of $S \cap \Sigma^n$. Thus, even a cheating P^* fails to find x' with probability greater than 1/2. ▌

Corollary 4.6 *If* SAT *has a one-sided 1-rsr, then the* PH *collapses at the third level.*

Proof: This follows directly from Theorem 4.4 and Yap's theorem [31]. ▌

4.4 Partial random-self-reductions

All of the results of Subsections 4.1, 4.2, and 4.3 carry over, *mutatis mutandis*, to partial random-self-reductions with polynomially many orbits.

We now show that, if SAT and $\overline{\mathrm{SAT}}$ are encoded in the standard way, then much stronger negative results can be obtained.

Proposition 4.1 *If* SAT *has a one-sided* (1/2)*-partial 1-rsr, then* $\overline{\mathrm{SAT}} \in$ IP(2)*.*

Proof (sketch): Let σ be a one-sided (1/2)-partial 1-rsr for SAT. By Definition 3.5, σ fixes the last $n/2$ bits of every satisfiable formula. Let x be a $\overline{\mathrm{SAT}}$ instance of length n and s be the length-$(n/2)$ suffix of x. The verifier constructs the satisfiable formula $y = s \vee \overline{s}$ and applies σ to both x and y. If the prover can distinguish the results, then $x \in \overline{\mathrm{SAT}}$. ▌

Corollary 4.7 *If* SAT *has a one-sided* (1/2)*-partial 1-rsr, then the* PH *collapses at the second level.*

Proof: Use Proposition 4.1 and the result of Bopanna, Hastad and Zachos [12] (see also the related work of Klapper [22]). ▌

Proposition 4.2 *There is a constant c_0 such that, for all $c \geq c_0$, if* $\overline{\mathrm{SAT}}$ *has a one-sided* (c/n)*-partial 1-rsr, then* NP = coNP*.*

[2] This theorem was first proven by one of the authors in 1987 and has already been referred to in the literature (e.g., [6]); it is published here for the first time.

Proof (sketch): Let σ be a one-sided (c/n)-partial 1-rsr for $\overline{\mathrm{SAT}}$. By Definition 3.5, σ fixes the last $n-c$ bits of every unsatisfiable formula. We show that this implies that $\overline{\mathrm{SAT}}$ can be recognized in NP. Let x be a formula of length n and s be the length-$(n-c)$ suffix of x. Take the conjunction of s with $u \wedge \overline{u}$, where u is any variable; pad this conjunction out so that it is a formula y of length n. Then y is unsatisfiable and shares the suffix s with x. Guess a random coin-toss sequence r and accept x if $\sigma(x,r) = y$. The constant c_0 just has to be big enough to accommodate the encoding of $u \wedge \overline{u}$. ▌

Proposition 4.2 shows that $\overline{\mathrm{SAT}}$ fails in the strongest possible way to be random-self-reducible in the style of quadratic residues, isomorphic graphs, etc. Even if $\overline{\mathrm{SAT}}$ had such an rsr with *constant-sized orbits*, the PH would collapse to NP. Proposition 4.1 shows that SAT fails (in a less extreme way) to be random-self-reducible in the usual style.

5 Recent Related Work

Independently of the results presented here, Yao [29] defined the notion of an *examiner* for a function; examiners are generalizations of uniform $poly(n)$-sr's, which can be defined analogously to the nonuniform $(poly(n), poly(n))$-sr's used in Theorem 4.1. Yao calls functions that have examiners *coherent* functions. The same argument that shows that if a function has a nonuniform (k,s)-rsr, then it has a nonuniform (k,s)-sr can be used to show that a function in (uniform) poly-RSR is coherent.

Yao [29] shows that there is a boolean function in DSPACE($2^{n^{\log\log n}}$) that is incoherent. This result should be contrasted with our Corollary 4.3, in which a stronger negative result is obtained for a weaker class of examiners.

Beigel and Feigenbaum [8] continued Yao's work on coherence by showing unconditionally that there is an incoherent set in DSPACE($n^{\log^* n}$) and that, if NEXPTIME $\not\subseteq$ BPEXPTIME, then there is an incoherent set in NP.[3] The first of these theorems represents an improvement of the theorem of Yao and of the uniform version of Corollary 4.3.

Yao [30] also provides progress on one of the open questions we raise here (see Section 6). He shows that there is a boolean function f in DSPACE($2^{n^{\log\log n}}$) that is not $(1,2)$-locally random reducible to any boolean function g. Note that, in the locally random reductions given in [6, 7], the target functions g are not boolean — there, $|g(x)| = \Theta(\log|x|)$.

6 Open Problems

Random-self-reducibility is an interesting, fundamental concept, and there is a lot of work to be done before it is fully understood. We state several questions about k-rsr's and their generalizations, $(1,k)$-locally random reductions and k-oracle instance-hiding schemes.

Question 1: Does Theorem 4.2 hold if the parameter 2 is replaced by any constant? That is, are functions with k-rsr's, where k is constant, in fNP/*poly*? (This is true if the random variables $\sigma_i(x,r)$ are $(k-1)$-wise independent.)

Question 2: Does Theorem 4.2 hold for locally random reductions and instance-hiding schemes as well as rsr's? That is, if f has a $(1,2)$-locally random reduction (or, more generally, a 2-oracle instance-hiding scheme), is it in fNP/*poly*?

Question 3: Do arbitrary boolean functions have 2-oracle instance-hiding schemes (or even k-oracle schemes, where k is constant)? Note that a stronger negative result, like Theorem 4.1 on rsr, does *not* hold for instance-hiding, by the results of [6, 7]. A partial negative result on $(1,2)$-locally random reductions for arbitrary functions was given recently by Yao [30]; see Section 5.

References

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

[3] Indeed, they showed unconditionally that there is an incoherent set in any class "just above PSPACE" – see [8] for details.

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

[3] L. Babai. Random Oracles Separate PSPACE from the Polynomial-time Hierarchy, *Inform. Proc. Letters* **26** (1987), 51–53.

[4] L. Babai, L. Fortnow, and C. Lund. Nondeterministic Exponential Time has Two-Prover Interactive Protocols, Technical Report 90-03, University of Chicago Computer Science Department, January, 1990.

[5] L. Babai and S. Moran. Arthur-Merlin Games: A Randomized Proof System, and a Hierarchy of Complexity Classes, *J. Comput. System Sci.* **36** (1988), 254–276.

[6] D. Beaver and J. Feigenbaum. Hiding Instances in Multioracle Queries, *Proceedings of the 7th STACS* (1990), Springer Verlag LNCS 415, 37–48.

[7] D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway. Cryptographic Applications of Locally Random Reductions, AT&T Bell Laboratories Technical Memorandum, November 15, 1989.

[8] R. Beigel and J. Feigenbaum. On the Complexity of Coherent Sets, AT&T Bell Laboratories Technical Memorandum, February 19, 1990.

[9] M. Blum and S. Kannan. Designing Programs that Check Their Work, *Proceedings of the 21st STOC* (1989), ACM, 86–97.

[10] M. Blum, M. Luby, and R. Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems, *Proc. of the 22nd STOC* (1990), ACM.

[11] M. Blum and S. Micali. How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits, *SIAM J. Comput.* **13** (1984), 850–864.

[12] R. Boppana, J. Hastad, and S. Zachos. Does co-NP Have Short Interactive Proofs?, *Inform. Proc. Letters* **25** (1987), 127–132.

[13] G. Brassard, D. Chaum, and C. Crépeau. Minimum Disclosure Proofs of Knowledge, *J. Comput. System Sci.* **37** (1988), 156–189.

[14] J. Cai. With Probability One, a Random Oracle Separates PSPACE From the Polynomial-Time Hierarchy, *J. Comput. System Sci.* **38** (1989), 68–85.

[15] U. Feige, A. Fiat, and A. Shamir. Zero-Knowledge Proofs of Identity, *J. Cryptology* **1** (1988), 77–94.

[16] J. Feigenbaum, S. Kannan, and N. Nisan. Lower Bounds on Random-Self-Reducibility, AT&T Bell Laboratories Technical Memorandum, December 4, 1989.

[17] Z. Galil, S. Haber, and M. Yung. Minimum-Knowledge Interactive Proofs for Decision Problems, *SIAM J. Comput.* **18** (1989), 711-739.

[18] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

[19] O. Goldreich, S. Micali, and A. Wigderson. Proofs that Yield Nothing but Their Validity, and a Methodology of Cryptographic Protocol Design, *Proceedings of the 27th FOCS* (1986), IEEE, 174–187.

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

[21] S. Goldwasser and M. Sipser. Public Coins vs. Private Coins in Interactive Proof Systems, *Advances in Computing Research — Vol. 5: Randomness and Computation*, S. Micali (ed.), JAI Press, Greenwich, 1989, 73–90.

[22] A. Klapper. Generalized Lowness and Highness and Probabilistic Complexity Classes, *Math. Sys. Theory* **22** (1989), 37–45.

[23] R. Lipton. New Directions in Testing, manuscript, October, 1989.

[24] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. The Polynomial-Time Hierarchy has Interactive Proofs, manuscript, December, 1989.

[25] N. Nisan. Co-SAT has Multiprover Interactive Proofs, manuscript, November, 1989.

[26] R. Rivest. Workshop on Communication and Computing, MIT, October, 1986.

[27] A. Shamir. IP = PSPACE, manuscript, December, 1990.

[28] M. Tompa and H. Woll. Random Self-Reducibility and Zero-Knowledge Interactive Proofs of Possession of Information, *Proceedings of the* 28*th FOCS* (1987), IEEE, 472–482.

[29] A. Yao. Coherent Functions and Program Checking, *Proceedings of the* 22*nd STOC* (1990), ACM.

[30] A. Yao. Private commuication, February, 1990.

[31] C. Yap. Some Consequences of Nonuniform Conditions on Uniform Classes, *Theor. Comput. Sci.* **26** (1983), 287–300.