An Information-Theoretic Treatment of Random-Self-Reducibility*

(Extended Abstract)**

Joan Feigenbaum and Martin Strauss

AT&T Labs, Murray Hill, NJ 07974 USA
{jf,mstrauss}@research.att.com

Abstract. We initiate the study of random-self-reducibility from an information-theoretic point of view. Specifically, we formally define the notion of a random-self-reduction that, with respect to a given ensemble of distributions, leaks a limited number bits, *i.e.*, produces target instances $y_1, \ldots, y_k$ in such a manner that each y_i has limited mutual information with the input x. We argue that this notion is useful in studying the relationships between random-self-reducibility and other properties of interest, including self-correctability and NP-hardness. In the case of self-correctability, we show that the information-theoretic definition of random-self-reducibility leads to somewhat different conclusions from those drawn by Feigenbaum, Fortnow, Laplante, and Naik [13], who used the standard definition. In the case of NP-hardness, we use the information-theoretic definition to strengthen the result of Feigenbaum and Fortnow [12], who proved, using the standard definition, that the polynomial hierarchy collapses if an NP-hard set is random-self-reducible.

1 Introduction

Informally, a function f is *random-self-reducible* if the evaluation of f at any given instance x can be reduced in polynomial time to the evaluation of f at one or more *random* instances y_i.

Random-self-reducible functions have many applications, including:

Cryptography: The fact that certain number-theoretic functions are random-self-reducible is used extensively in the theory of cryptography — *e.g.*, to achieve *probabilistic encryption* [14] and *cryptographically strong pseudorandom number generation* [9]. Random-self-reductions also provide natural examples of *instance-hiding schemes* [1, 5, 6], in which a weak, private computing device uses the resources of a powerful, shared computing device without revealing its private data.

* Part of this work was done while the second author was at Iowa State University, supported by CCR-9157382.

** A full version of this paper has been submitted for journal publication and is available as AT&T Technical Report 96.13.2.

Reischuk, Morvan (Eds.): STACS'97 Proceedings, LNCS 1200

Interactive proof systems and program checkers, self-testers, and self-correctors: Random-self-reductions are crucial ingredients in many of the original examples of interactive proof systems and program checkers, self-testers, and self-correctors [7, 8, 15]. Intuitively, this is because the verifier, checker, tester, or corrector interrogates the prover or program by comparing its output on the specific input of interest to its outputs on other related random instances. These ideas play a crucial role in the characterization of the language-recognition power of interactive proof systems [4, 18, 19]. A very active theme of current research in this area is the question of whether NP-complete sets have checkers, self-testers, or self-correctors [8, 12, 13]. We explore this theme further in this paper, using an information-theoretic perspective for the first time.

Average-case complexity: A random-self-reduction maps an arbitrary, worst-case instance x in the domain of f to a set of random instances $y_1, \ldots, y_k$ in such a way that $f(x)$ can be computed in polynomial-time, given x, $f(y_1), \ldots, f(y_k)$, and the coin-toss sequence used in the mapping. Thus the average-case complexity of f, where the average is taken with respect to the induced distribution on instances y_i, is the same, up to polynomial factors, as the worst-case randomized complexity of f. One important example of this connection between average-case complexity and worst-case complexity is the result of Lipton [17] that the PERM (permanent of integer matrices) function is random-self-reducible. The PERM function is also #P-complete [21]; thus, if PERM could be computed efficiently *on average* (with respect to the target distribution of the reduction), then *every* function in #P could, with a randomized algorithm, be computed efficiently in the *worst case*. Furthermore, the random-self-reduction for PERM is very simple, whereas average-case hardness proofs are often complicated.

Lower bounds: The random-self-reducibility of the parity function is used in [2] to obtain a simple proof that a random oracle separates the polynomial hierarchy (PH) from PSPACE. (An earlier proof of this result in [10] does not use random-self-reducibility.)

In this work we investigate a new notion, *random-self-reducibility with respect to an ensemble* $D = \{D_n\}_{n \geq 1}$, *leaking* $l(n)$ *bits*. Roughly, a function has this property if $f(x)$ can be recovered with high probability from x, $y_1, \ldots, y_k$ and $f(y_1), \ldots, f(y_k)$, where each y_i, although chosen at random from a distribution that may depend on x, has no more than $l(|x|)$ bits of mutual information with x. In particular, we may require that, for all c, fewer than $1/n^c$ bits of information are leaked. We assume that the random-self-reduction can sample from D — this is justified because, in practice, the reduction can sample from D by requesting more input. This definition is a small modification of the previous definition, but it has implications for the application areas listed above.

Cryptography: One reason that the standard definition of a random-self-reduction is useful in cryptographic constructions is that it naturally provides a notion of privacy of the input x: The recipient of a single target instance y_i receives no information about x except its length. Our new definition provides a robust way to *quantify* the extent to which the privacy of x may be compromised. We assume that x is drawn from an ensemble $D = \{D_n\}_{n \geq 1}$; this ensemble will

be determined by the particular cryptographic scenario in which the reduction is used and is assumed to be known by all of the parties in that scenario. A target instance y_i will also be drawn from a distribution, and this target distribution will depend both on x and the algorithm used in the reduction. The mutual information of the two random variables x and y_i thus measures what a party who receives y_i knows about x that he did not already know from D before receiving y_i. A reduction may be considered sufficiently privacy-preserving if it divulges fully an x that is assigned high probability by D but conceals almost fully an x that is assigned low probability by D.

Self-correction: Blum, Luby, and Rubinfeld [8] defined program self-correction in order to address the following question. Let P be any program that purports to compute f, and suppose that one can determine that, with respect to an ensemble $D = \{D_n\}_{n\geq 1}$, the measure of inputs on which P errs, while not necessarily zero, is limited. Is it possible to write an auxiliary program C that corrects the errors of any such P with high probability? More precisely, on any input $x \in Dom(f)$, C should produce the correct answer $f(x)$ with high probability, and C may call the (potentially faulty) program P several times in the course of its computation. If such a C exists, then the function f is said to be self-correctable with respect to the ensemble D. Blum, Luby, and Rubinfeld observed that every f that has a standard random-self-reduction is also self-correctable, and it is a well-known open question whether the two properties are equivalent, *i.e.*, whether every self-correctable function has a standard random-self-reduction. Recently, Feigenbaum, Fortnow, Laplante, and Naik [13] provided a partial answer to this question by exhibiting, under a plausible complexity theoretic hypothesis, a function that is self-correctable but does not have a standard random-self-reduction.

In Section 3 below, we give evidence that our new definition of random-self-reducibility is better suited than the standard definition to a direct comparison with self-correctability. Specifically, we claim that, because inputs to a self-corrector are drawn from an ensemble D, inputs to a random-self-reduction should also be drawn from such an ensemble if the notions are to be compared meaningfully. To justify this claim, we show that the function f that is proven in [13] to be self-correctable with respect to a particular ensemble D, but not random-self-reducible according to the standard definition, *is* in fact random-self-reducible (leaking 0 bits) according to our definition, in which the random-self-reduction is given the ability to sample from D.

Average-case complexity: If we exhibit a standard random-self-reduction for f, we show that the worst-case complexity of f is no worse than the average-case complexity with respect to an ensemble *of our own construction*, namely the target ensemble of the reduction. In practice, however, we might want to know the average-case complexity of f with respect to an instance ensemble $D = \{D_n\}_{n\geq 1}$ arising in a particular scenario or application. In such a context, we should only consider random-self-reductions that take inputs x from D and produce queries y_i from D (or from a distribution D_x that is close enough to D to allow the average-case complexity with respect to D_x to tell us something

about the average-case complexity with respect to D). Restricting random-self-reductions in such a way may make them considerably harder to construct than they are to construct under the standard definition. Our new definition of a leaky random-self-reduction with respect to D allows us to take advantage of the peculiarities of D and so may make it easier to exhibit reductions that give bounds on the average-case complexity with respect to ensembles that arise naturally.

Some results and all proofs have been omitted from this extended abstract because of space limitations. They can be found in our journal submission, which is available as AT&T Technical Report 96.13.2.

2 Preliminaries

Throughout this paper, f is a function on $\{0,1\}^*$, and x is an arbitrary input for which we would like to determine $f(x)$. Reductions will make a polynomial number of queries, denoted $k(n)$, on inputs of length n. We use r to denote a sequence of fair coin tosses; if $|x| = n$, then $|r| = w(n)$, where w is a polynomially bounded function of n. $D = \{D_n\}_{n \geq 1}$ denotes an ensemble of distributions; there is a constant c such that, for all n, any y to which D_n assigns positive probability satisfies $n^{-c} \leq |y| \leq n^c$.

2.1 Previous Definitions

We start by recalling the standard definitions. For a more in-depth review of these definitions and of their role in complexity theory, see [11].

Definition 1.

- A **basic reduction** of f to g is a pair of polynomial-time computable functions (ϕ, σ) such that, for all n, all $x \in \{0,1\}^n$, and at least $3/4$ of the r's in $\{0,1\}^{w(n)}$, f satisfies

$$f(x) = \phi(x, r, g(\sigma(1, x, r)), \ldots, g(\sigma(k(n), x, r))).$$

  (Note that this is equivalent to the statement that there is a nonadaptive, probabilistic polynomial-time reduction from f to g; we use the term "basic reduction" for brevity.)
- A basic reduction (ϕ, σ) of f to f is a **self-corrector with respect to an ensemble D of distributions** if it is also a basic reduction of f to P for each program P that, for all n, computes f correctly on a set Y_n such that $\Pr_{D_n}(Y_n) \geq 3/4$.
- A basic reduction (σ, ϕ) of f to f is called a **random-self-reduction** of f if, for each i, for all x_1, x_2 such that $|x_1| = |x_2|$, the random variables $\sigma(i, x_1, r)$ and $\sigma(i, x_2, r)$ are identically distributed. (Note that, because r is chosen uniformly at random from $\{0,1\}^{w(n)}$, $\sigma(i, x, r)$ is a random variable, for fixed i and x.)

- Let L be a function on $Dom(f)$. The basic reduction (σ, ϕ) is a **random-self-reduction leaking at most** L if $\sigma(i, x_1, r)$ and $\sigma(i, x_2, r)$ are identically distributed for all x_1, x_2 such that $L(x_1) = L(x_2)$. Thus, "random-self-reduction" is really shorthand for "random-self-reduction leaking at most $|x|$."

Note that we are restricting attention to *nonadaptive* reductions, *i.e.*, those in which the query y_i does not depend on the answers to the queries y_1 through y_{i-1}; see, *e.g.*, [11] for a discussion of the more general, adaptive versions of these reductions. Because of this restriction, we may assume that the random variables $\sigma(i_1, x, r)$ and $\sigma(i_2, x, r)$ are identically distributed, for all i_1, i_2, as pointed out by Szegedy and reported in [12], and thus we use $\sigma(x, r)$ to denote this random variable and σ_x for its distribution. The shorthand "rsr" is used for both "random-self-reduction" and "random-self-reducibility."

As usual, we may reduce the error probability of any of these reductions from $1/4$ to 2^{-n} by running a polynomial number of independent copies of the reduction and returning the plurality answer. One may also parameterize self-correction in such a way that the program P is assumed to err on D_n with probability at most $\epsilon(n)$ (as Blum, Luby, and Rubinfeld did in their original paper [8]).

2.2 Mutual Information

The **mutual information** $I(a; b)$ between two events a, b is defined to be

$$\log \frac{\Pr(ab)}{\Pr(a)\Pr(b)} = \log \frac{\Pr(a \mid b)}{\Pr(a)}.$$

Conventions differ about the mutual information between two events if (at least) one of these events has probability zero. In the context that we use information theory, there is a natural definition, but we defer discussion of that definition to the end of this section.

The **self-information** of the event a is the mutual information between a and a or, equivalently, the maximum, over all random variables B and reals b, of the mutual information between a and the event $B = b$.

2.3 Information-Theoretic Definition of RSR

We introduce three variations on the standard definitions given in Section 2.1. The first concerns the requirement that the distribution on queries be exactly the same for all x's of the same length, the second concerns the sampleability of inputs, and the third concerns a connection between distributions on queries and the natural distributions on inputs (that we introduce).

First, we introduce rsr's that leak at most $l(n)$ bits, as opposed to those that leak at most some function L.

Definition 2. For $n \geq 1$, the **(information-theoretic) leakage** of a basic reduction (ϕ, σ) for f with respect to the ensemble D is

$$\max_{\{x \in \{0,1\}^n, r\}} I(x; \sigma(x, r)).$$

A basic reduction (ϕ, σ) is an **rsr with respect to D, leaking l bits** if, for all $n \geq 1$, its leakage is at most $l(n)$.

Let $\sigma \circ D$ be the ensemble of distributions on y produced by the experiment of sampling x from D then sampling y from σ_x. Thus, if we are given x from D and we query y, the mutual information between x and y is

$$\log \frac{\Pr_{\sigma_x}(y)}{\Pr_{\sigma \circ D}(y)} = \log \frac{\Pr(y \mid x)}{\Pr(y)},$$

and the leakage of a basic reduction is bounded by l if and only if, for all x and y, $\Pr_{\sigma_x}(y) \leq 2^l \Pr_{\sigma \circ D}(y)$.

Both self-correctors and leaky rsr's have associated ensembles of instances. We are motivated by the case in which these ensembles arise in nature, and thus we assume that additional data points are abundant:

Convention 3 *We assume the functions ϕ and σ of a self-corrector can sample from D_n (by requesting more input). We assume samples from D_n are independently and identically distributed.*

Thus, in our model, sampling from $\sigma \circ D$ is feasible.

The following definition is useful in the context of average-case complexity.

Definition 4. An rsr with respect to D is **reciprocating** if there is a polynomial $p(n)$ such that, for all x, y, $\Pr_{\sigma_x}(y) \leq p(n) \Pr_D(y)$.

That is, the rsr is reciprocating if the distribution on queries conditioned on x is approximated from above by the distribution on inputs. If f has a reciprocating rsr with respect to D leaking (the function) $|x|$, then the distributions σ_x are actually the one common distribution σ, and the worst-case running time of f is at most polynomially worse than the average-case running time with respect to the natural ensemble D.

An important special case of a reciprocating rsr is that in which D_n and σ_x are uniform distributions. Traditional rsr theory allows two arbitrary distributions, D_n and a common distribution σ for all x, but requires no connection between D_n and σ; we allow different distributions for each x but require σ_x to be loosely bounded from above by both D and $\sigma \circ D$.

It should now be clear why our definition of leakage captures the notion of the (worst-case) amount of information about the private instance x that is given away by a basic reduction. Some further comments on the definition are in order:

- The phrase "... leaking l *bits*" (as opposed to "... leaking l") is used to distinguish information-theoretic leakage from the standard notion of functional leakage (which is referred to as "... leaking L").

- It is $I(x;\sigma(x,r))$ that is bounded and not the absolute value $|I(x;\sigma(x,r))|$: There is no limit on the amount of negative (*i.e.*, misleading) information about x that may be "divulged" by the queries.
- As in the literature on standard rsr's, we assume that we make our queries of powerful players that are not allowed to collude. Thus the mutual information between an input x and a single query y_i is limited in the worst case, but that between x and a pair of queries y_i, y_j is not.

We now argue that leaking $O(\log n)$ bits is reasonable. Standard rsr theory focuses on functional leakage of $n = |x|$, and we wish to develop an analogous notion of a "reasonable amount" of information-theoretic leakage. First we argue that divulging $n = |x|$ is equivalent to leaking at least $\log n + \log\log n$ bits, information-theoretically. Suppose that x is chosen from a distribution on $\{0,1\}^*$ rather than a distribution on $\{0,1\}^{|x|}$; so $|x|$ is initially unknown. Because $\sum_n \Pr(|x| = n) = 1 < \infty$, for infinitely many x's, $\Pr(|x| = n) < 1/(n \log n)$; therefore, for arbitrarily large n, the self-information, $-\log \Pr(|x| = n)$ bits, contained in $n = |x|$ is at least $\log n + \log\log n$ bits. (If only finitely many instances x are possible, then no complexity-theoretic treatment is meaningful.) Furthermore, the next lemma shows that leaking $O(\log n)$ bits is a robust notion for rsr's, in the same way that erring with probability at most $1/2 - \epsilon$ is a robust notion for probabilistic algorithms.

Lemma 5. *If f is rsr with respect to D, leaking $O(\log n)$ bits, then, for any polynomial $p(n)$, f is rsr with respect to D, leaking $1/p(n)$ bits.*

This argument that $O(\log n)$ bits is the "right" amount of leaking justifies the following definition.

Definition 6. A function has a **leaky rsr with respect to** D if it is rsr with respect to D leaking at most $O(\log n)$ bits.

Another useful way to view the information-theoretic leakage of a basic reduction is in relationship to the entropy of D. For example, suppose that D_n is the uniform distribution on $\{0,1\}^n$ and that a basic reduction leaks $n/2$ bits. Each x drawn from D_n contains n bits of self-information, and therefore the basic reduction preserves $n/2$ bits of privacy. This should be considered more privacy-providing, for example, than any basic reduction (ϕ, σ) with respect to an ensemble D in which D_n is uniform on $2^{n/3}$ of the strings in $\{0,1\}^n$, even if (ϕ, σ) is a standard rsr, because the latter provides only $n/3$ bits of privacy even *before* queries are made.

2.4 Zero Probability

One final issue remains. Suppose that D_n is a distribution such that $\Pr_D(x_0) = 0$. How should this be handled? Ideally, our analysis should be continuous, *i.e.*, the case in which $\Pr_D(x_0)$ is equal to zero should be treated similarly to the case in which $\Pr_D(x_0)$ is small.

First, should an rsr with respect to D be required to recover $f(x_0)$? Because we regard an rsr as a transformation from worst case to average case, we answer "yes."

A related question is whether there should be a limit to the leakage when $\Pr_D(x_0) = 0$. The leakage is defined in terms of the mutual information $I(x_0; y)$, which in turn is defined in terms of a conditional probability $\Pr(y|x_0)$. In many contexts, there is no sensible way to define the conditional probability when $\Pr(x_0) = 0$. We, however, use mutual information in just one context, in which there is a natural and meaningful definition. The mutual information between x and y is taken to be

$$\log \frac{\Pr_{\sigma_x}(y)}{\Pr_{\sigma \circ D}(y)}.$$

Even if $\Pr_D(x_0) = 0$, it still makes sense to talk about σ_{x_0} and about recovering $f(x_0)$ from $f(y)$ with y chosen from σ_{x_0}. Also, if the denominator is zero but the numerator is non-zero, which can only happen if $\Pr_D(x_0) = 0$, then we define the mutual information to be $+\infty$ — if we query such a y, then we have (completely) leaked the fact that the input is one to which D assigns probability zero. Similarly, if the denominator is non-zero but the numerator is zero then the mutual information is defined to be $-\infty$ (not a constraint). Finally, if both numerator and denominator are zero (whether $\Pr_D(x_0) = 0$ or not), then the mutual information is defined to be zero (*i.e.*, not a constraint), because we could easily define a new rsr that, with small probability, queries from the uniform ensemble and disregards the answers; in this new rsr, the numerator and denominator would be equal, non-zero quantities. These conventions make our analysis continuous.

3 RSR versus Self-correction

Feigenbaum, Fortnow, Laplante, and Naik [13] consider the question of whether all self-correctible functions are rsr. They show that the question cannot be settled with current techniques, because there are currently unrefutable hypotheses that support both answers. Nonetheless, one can interpret the results of [13] as evidence that there is a function f and an ensemble D such that f does not have a standard rsr but is self-correctable with respect to D — [13] gives a plausible hypothesis that implies the existence of such an f and D and an implausible hypothesis that implies the nonexistence of such an f and D.

Here we revisit the question, using our notion of rsr's that can sample from D, which we have argued in Section 1 is more suitable for a fair comparison of rsr with self-correction. While we cannot show definitively that all self-correctible functions have such rsr's, our results lend themselves to the opposite interpretation of those in [13].

3.1 Inequivalence

Feigenbaum, Fortnow, Laplante, and Naik [13] observe that certain functions f are self-correctible with respect to a singleton ensemble D, *i.e.*, one that, for each

n, puts all the weight on one string in $\{0,1\}^n$: The crux is that any program that computes f correctly with probability $3/4$ with respect to D already computes f correctly on the one positive-probability string. While the peculiarities of D can make a function self-correctible, they do not necessarily endow that function with a standard rsr, as shown in [13]. We show here that, for such peculiar ensembles D, all self-correctible functions have rsr's that sample D. In fact, the rsr that we exhibit leaks zero bits beyond $n = |x|$; the proof relies on Convention 3 only.

Theorem 7. *Suppose $p(n)$ is a polynomial and $T \subseteq 1^* \cap \mathrm{UP}$. For each n such that $1^n \in T$, let u_n be the unique witness of this, and without loss of generality assume that $|u_n| = n$. Let W be the set of all such u_n's. Let δ_{u_n} denote the distribution on $\{0,1\}^n$ that puts all the weight on u_n, and define D_n by*

$$D_n = \begin{cases} \delta_{u_n} & 1^n \in T \\ \textit{uniform} & \textit{otherwise} \end{cases}$$

If L is a subset of W, then L has a leaky rsr with respect to D.

We conclude that the self-correctible function exhibited in [13] fails to be rsr simply because the standard definition of rsr does not allow the reduction to sample from the ensemble D. The fact that a standard rsr leaks at most n is not really used.

3.2 Equivalence

In [13], it is shown that, if PF = #P, then any function that is self-correctible with respect to a P-sampleable ensemble also has a standard rsr. In this section, we proceed along similar lines, drawing a weaker conclusion from a weaker hypothesis: If P = NP, then any function that is self-correctible with respect to a P-sampleable D has a leaky rsr with respect to D. The proof in [13] uses the assumption PF = #P to conclude that a P-sampleable ensemble is simply a function in P. Instead, we use the weaker hypothesis P = NP and the technique of universal hashing [20] merely to approximate the ensemble. This is sufficient to produce a leaky rsr.

Theorem 8. *Let $D = \{D_n\}_{n \geq 1}$ be a P-sampleable ensemble. If* P = NP *and f has a self-corrector with respect to D, then f has a leaky, reciprocating rsr with respect to D.*

Proof. (sketch) Our proof builds on [13]. Suppose the self-corrector (ϕ, σ) makes $k(n) = n^{O(1)}$ queries and fails with probability at most 2^{-n} to correct any program that correctly computes f on $\{0,1\}^n$ with probability at least $\epsilon(n) = 1/n^{O(1)}$, with respect to D_n.

Fix c. A query z is called *superfluous* with respect to x if

$$\Pr_{\sigma_x}(z) \geq \frac{k}{\epsilon^2} \Pr_{D_n}(z),$$

and *very superfluous* if

$$\Pr_{\sigma_x}(z) \geq \frac{k}{\epsilon^2}\left(1+\frac{1}{n^c}\right)\Pr_{D_n}(z).$$

The term superfluous originated in [13], where it is used to describe z's with the property that no self-corrector can rely heavily on $f(z)$ to recover $f(x)$. The term has additional meaning here: If we promise that our rsr queries from D (a goal that originated in [13] and is related to reciprocity), then a query z is superfluous precisely when it has more than $\log(k/\epsilon^2)$ bits of mutual information with x.

In [13], it is shown how to construct a standard rsr with target ensemble D, by preparing samples z from σ_x and another distribution $\Pr''(x,z)$, then querying the non-superfluous samples from σ_x mixed with the $\Pr''$ samples. The non-superfluous queries z do not reveal much about x so burying them among a modest number queries from $\Pr''$ that are not needed to reconstruct $f(x)$ results in a target ensemble that reveals sufficiently little about x.

The construction of $\Pr''$ depends on identifying superfluous queries. We can do this approximately, using universal hashing. The construction is given in full detail in our journal submission and is omitted here because of space limitations. For each x, the reduction we construct queries an instance z with probability $\Pr_{\tilde{D}_x}(z) = \Pr_{D_n}(z)(1 \pm O(1/n^c))$.

What is leaked? For a fixed x, the mutual information between x and a query z from $\tilde{D}_x$ is

$$\log \frac{\Pr_{\tilde{D}_x}(z)}{\Pr_{\tilde{D}\circ D_n}(z)}.$$

$\tilde{D}_x$ is close to D uniformly in x, and so

$$\Pr_{\tilde{D}\circ D_n}(z) = (1 \pm O(1/n^c)) \Pr_{D_n}(z).$$

Also $\Pr_{\tilde{D}_x}(z) = (1 \pm O(1/n^c))\Pr_{D_n}(z)$, so the leakage is $\log(1 \pm O(1/n^c)) = O(1/n^c)$.

We conclude that f has a leaky reciprocating rsr.

4 Random-self-reducibility of SAT

Feigenbaum and Fortnow [12] show that SAT is not random-self-reducible unless the PH collapses at the third level. In this section, we show an analogous result for our notion of leaky rsr.

The language class $\mathrm{AM}^{\mathrm{poly}}$ is defined in [12]; it consists of languages with Arthur-Merlin protocols [3] in which Arthur gets advice and the subsequent AM protocol need only be valid on the correct advice. In [12], it is shown that, if S is in NP and has a standard rsr, then the complement of S is in $\mathrm{AM}^{\mathrm{poly}}$. Since it is known [3, 16] that $\mathrm{AM}^{\mathrm{poly}} = \mathrm{NP/poly}$, membership of $\overline{\mathrm{SAT}}$ in $\mathrm{AM}^{\mathrm{poly}}$ implies that co-NP $\subseteq$ NP/poly, whence the PH collapses [22].

In Section 2, we argued that $O(\log n)$ bits of leakage is the "right" amount to allow. That is, one should require that $\Pr_{\sigma_x}(z) \leq n^{O(1)} \Pr_{\sigma \circ D_n}(z)$. For the results of this section, we need tighter control on the leakage. Relatively tight one-sided bounds will allow us to deduce the two-sided bounds necessary for the results of this section.

Theorem 9. *Let S be a set in* NP. *If χ_S has an rsr making $k(n) \geq 4$ queries and leaking $1/k^5$ bits with respect to an ensemble D, then $\overline{S}$ is in* $\mathrm{AM}^{\mathrm{poly}}$.

Corollary 10. *If SAT has an rsr making $k(n) \geq 4$ queries and leaking $1/k^5$ bits, then the PH collapses.*

5 Conclusions and Open Problems

We have modified the standard notion of random-self-reducibility. We have argued that our modification is natural in many of the contexts in which random-self-reducibility arises, including cryptography, average-case complexity, and program self-correction.

The results in Section 3.1 raise the following question. Can one exhibit, perhaps under a plausible complexity theoretic hypothesis, a self-correctable function that has no leaky rsr, not even an rsr that samples from the ensemble D?

In Section 4, we show that SAT has no rsr making k queries and leaking $1/k^5$ bits, unless the PH collapses. Can this result be improved to show that SAT has no leaky rsr (leakage independent of k), unless the PH collapses? A more useful and possibly easier task would be to show that SAT has no leaky, reciprocrating rsr.

Can our information-theoretic formulation be used to analyze adaptive rsr's and to resolve some of the questions about adaptiveness that have gone unresolved for standard rsr's?

References

1. M. Abadi, J. Feigenbaum, and J. Kilian. On Hiding Information from an Oracle. *Journal of Computing and System Sciences*, 39:21–50, 1989.
2. L. Babai. Random oracles separate PSPACE from the polynomial-time hierarchy. *Information Processing Letters*, 26:51–53, 1987.
3. L. Babai and S. Moran. Arthur-Merlin games: A randomized proof system and a hierarchy of complexity classes. *Journal of Computing and System Sciences*, 36:254–276, 1988.
4. L. Babai, L. Fortnow, and C. Lund. Nondeterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3-40, 1991.
5. D. Beaver and J. Feigenbaum. Hiding instance in multioracle queries. In *Proc. 7th Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, vol. 415, pages 37–48. Springer, Berlin, 1990.

6. D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway. Locally random reductions: Improvements and applications. *Journal of Cryptology*, to appear. Preliminary version in *Proc. Crypto '90*, pages 62–76, under the title "Security with low communication overhead."
7. M. Blum and S. Kannan. Designing programs that check their work. *Journal of the ACM*, 42:269–291, 1995.
8. M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting, with applications to numerical problems. *Journal of Computing and System Sciences*, 59:549–595, 1993.
9. M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13:850–864, 1984.
10. J. Cai. With probability one, a random oracle separates PSPACE from the polynomial hierarchy. *Journal of Computing and System Sciences*, 38:68–85, 1989.
11. J. Feigenbaum. Locally random reductions in interactive complexity theory. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 13, pages 73–98. American Mathematical Society, Providence, 1993.
12. J. Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22:994–1005, 1993.
13. J. Feigenbaum, L. Fortnow, S. Laplante, and A. Naik. On coherence, random-self-reducibility, and self-correction. In *Proc. 11th Conference on Computational Complexity*, pages 59–67. IEEE Computer Society Press, Los Alamitos, 1996.
14. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computing and System Sciences*, 28:270–299, 1984.
15. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18:186–208, 1989.
16. S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. *Advances in Computing Research*, vol. 5, pages 73–90. JAI Press, Greenwich, 1989.
17. R. Lipton. New directions in testing. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 2, pages 191–202. American Mathematical Society, Providence, 1991.
18. C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39:859–868, 1992.
19. A. Shamir. IP = PSPACE. *Journal of the ACM*, 39:869–877, 1992.
20. M. Sipser. A complexity theoretic approach to randomness. In *Proc. 15th Symposium on Theory of Computation*, pages 330–335. ACM Press, New York, 1983.
21. L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.
22. C. Yap. Some consequences of nonuniform conditions on uniform classes. *Theoretical Computer Science*, 26:287–300, 1983.