[FFLS94] J. Feigenbaum, L. Fortnow, C. Lund, and D. Spielman. The power of adaptiveness and additional queries in random-self-reductions. *Computational Complexity*, 4:158–174, 1994.

[HNOS93] E. Hemaspaandra, A. Naik, M Ogiwara, and A. Selman. P-selective sets, and reducing search to decision versus self-reducibility. Technical Report 93-21, State University of New York at Buffalo, 1993. To appear in JCSS.

[KL80] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of 12th ACM Symposium on Theory of Computing*, pages 302–309, 1980. An extended version has also appeared as: Turing machines that take advice, *L'Enseignement Mathématique*, 2nd series, 28:191–209, 1982.

[Ko87] K. Ko. On helping by robust oracle machines. *Theoretical Computer Science*, 52:15–36, 1987.

[Lip91] R. Lipton. New Directions in Testing. In *Distributed Computing and Cryptography, DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, 2:191–202. American Mathematical Society, Providence, 1991.

[LV93] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Verlag, 1993.

[Yao90] A. Yao. Coherent functions and program checkers. In *Proceedings of 22nd ACM Symposium on Theory of Computing*, pages 84–94, 1990.

tively coherent) remains open. This seems to be related to the more general question of constructing random-self-reducible functions using techniques other than arithmetization. Also, it would be interesting to know whether, under some reasonable complexity-theoretic assumption, there are sets in NP or even PSPACE that are coherent but not nonuniformly nonadaptively coherent.

Theorem 5 provides some evidence that the notions of self-correctability and random-self-reducibility are different. However, observe that the proof of this theorem does *not* give a self-corrector whose ensemble is P-sampleable; moreover, the fact that the ensemble is not P-sampleable is critical in showing that χ_L is not nonadaptively rsr. Is there a hypothesis that implies the existence of a function that is nonadaptively self-correctable with respect to a P-sampleable ensemble but not nonadaptively rsr? Are there relativized worlds in which these two notions can be separated?

It would also be interesting to know whether there is a result analogous to Theorem 6 in which the self-corrector and the rsr are adaptive.

References

[BBFG91] R. Beigel, M. Bellare, J. Feigenbaum, and S. Goldwasser. Languages that are easier than their proofs. In *Proceedings of 32nd Annual IEEE Symposium on Foundations of Computer Science*, pages 19–28, 1991.

[BF92] R. Beigel and J. Feigenbaum. On being incoherent without being very hard. *Computational Complexity*, 2:1–17, 1992.

[BFKR91] D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway. Security with low communication overhead. In *Advances in Cryptology—CRYPTO '90, Lecture Notes in Computer Science*, 537:62–76. Springer-Verlag, 1991.

[BFT95] H. Buhrman, L. Fortnow, and L. Torenvliet. Using Autoreducibility to Separate Complexity Classes. In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 520–527, 1995.

[BLR93] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting, with applications to numerical problems. *Journal of Computer and System Sciences*, 59:549–595, 1993.

[Boo74] R. Book. Tally languages and complexity classes. *Information and Control*, 26:186–193, 1974.

[Fei93] J. Feigenbaum. Locally random reductions in complexity theory. In *Advances in Computational Complexity Theory, DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, 13:73–98. American Mathematical Society, Providence, 1993.

[FF93] J. Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22:994–1005, 1993.

these additional $k''(n)$ queries is to make its query-distribution dependent only on n, rather than on the specific input x.

More precisely, we seek a polynomially bounded function $k''(n)$ and a P-sampleable ensemble $\{Pr''(x,z)\}_x$ that satisfy the following equation for all z.

$$\frac{k'(n)Pr'(x,z) + k''(n)Pr''(x,z)}{(k'+k'')(n)} = P_{D_n}(z) \tag{4}$$

The nonadaptive rsr (σ'', ϕ'') is then fully specified as follows. On input x, first run σ' to produce z_1, ..., $z_{k'(n)}$. Then do $k''(n)$ independent samplings of Pr'' to produce $z_{k'(n)+1}$, ..., $z_{(k'+k'')(n)}$. Randomly permute the queries z_1, ..., $z_{(k'+k'')(n)}$ and submit them to the oracle. Apply the appropriate inverse permutation to extract the answers $f(z_1)$, ..., $f(z_{k'(n)})$. Use these answers and the function ϕ' to compute $f(x)$.

That (σ'', ϕ'') satisfies requirement 1 (the correctness property) of Definition 1 follows directly from the correctness of (σ', ϕ'). That it satisfies requirement 2 (the local randomness property) follows from Equation (4): Each query z_i, $1 \leq i \leq (k'+k'')(n)$, is distributed according to $\mathcal{D}_n$, and thus its distribution does not depend on the specific input $x \in \{0,1\}^n$.

We now show how to obtain the necessary ingredients k'' and Pr''. Let b be a positive integer such that, for all z, $Pr'(x,z)$ and $P_{D_n}(z)$ are integral multiples of 2^{-b}. Such a b can always be found in our model of computation, where the source of randomness is simply an unbiased coin. Rearranging Equation (4) yields

$$Pr''(x,z) = \frac{(k'+k'')(n)P_{D_n}(z) - k'(n)Pr'(x,z)}{k''(n)}. \tag{5}$$

Set $k''(n)$ to be any integer of the form 2^c that is large enough to make $Pr''(x,z)$ positive for all z but small enough so that $2^c = poly(n)$. That such a 2^c exists is guaranteed by the definition of superfluous (see Equation (3)) and the fact that σ' only asks queries that are *not* superfluous.

From Equation (5), we now see that $Pr''(x,z)$ is an integral multiple of $2^{-(b+c)}$. Let

$$g_x(z) = \sum_{w<z} Pr''(x,w)2^{b+c}.$$

Observe that g_x is a #P function and hence by assumption in FP. In order to generate random strings in such a way that each z is produced with probability $Pr''(x,z)$, simply choose a string v by flipping $b+c$ coins and then use binary search to output the largest z for which $g_x(z) < v$.

This completes the proof of the theorem. □

5 Open Questions

The question of whether there exist random-self-reducible functions that are not nonuniformly nonadaptively random-self-reducible (or perhaps even not nonuniformly nonadap-

Recall that the key difference between a nonadaptive rsr and a nonadaptive self-corrector is that the definition of a self-corrector allows its function σ to produce random queries whose distribution depends on the input x. The definition of a nonadaptive rsr, on the other hand, explicitly disallows this, insisting that the distribution depend only on the length n of x. Note, however, that Definition 3 does refer to a distribution, namely $\mathcal{D}_n$, that depends only on n. We use $\mathcal{D}_n$ to define a distribution that can be used in a nonadaptive rsr.

Recall that the ϵ-self-corrector (σ, ϕ) is nonadaptive, which means that we may assume without loss of generality that the random variables $\sigma(1, x, r), \ldots, \sigma(k(n), x, r)$ are identically distributed for each x.

A central notion in our construction is that of a query that is *superfluous* with respect to a given input x. Intuitively, z is superfluous if the answer $P(z)$ that a self-corrector gets by querying the program about z cannot be very useful in computing $f(x)$. Let $Pr(x, z)$ denote the probability that $z = \sigma(1, x, r)$, where r, as usual, is chosen uniformly at random from $\{0,1\}^{w(n)}$. (We use $\sigma(1, x, r)$ for concreteness. Because all the random variables $\sigma(i, x, r)$ are identically distributed, $Pr(x, z)$ could have been defined in terms of any of them and would take on exactly the same values.) Then a query z is superfluous if $Pr(x, z)$ is much bigger than $\mathrm{Prob}_{\mathcal{D}_n}(z)$. Specifically, if

$$Pr(x, z) \geq \frac{1}{\epsilon^2(n)} \cdot k(n) \cdot \mathrm{Prob}_{\mathcal{D}_n}(z), \tag{3}$$

then we say that z is superfluous with respect to x.

The next proposition follows straightforwardly from the definitions.

Proposition 7 *For all n and all $x \in \{0,1\}^n$, the probability that a random z, drawn according to $\mathcal{D}_n$, is superfluous with respect to x is at most $\epsilon^2(n)$.*

Consider a self-corrector (σ', ϕ') that works as follows. On input x, σ' first computes $z_1, \ldots, z_{k(n)}$, a set of random queries produced by σ. For each i, σ' decides whether z_i is superfluous with respect to x; note that assumptions (B) and (C) above make this computable in polynomial time. If z_i is superfluous, then the corrector assumes that $P(z_i) = 0$; otherwise, it queries the program to find out $P(z_i)$. If $k'(n)$ denotes the maximum number of queries to P made by σ', then k' is clearly bounded above by k. The function ϕ' then computes $f(x)$ exactly as ϕ would have. Once again, the following fact is straightforward.

Proposition 8 *(σ', ϕ') is a nonadaptive $(\epsilon - \epsilon^2)$-self-corrector for f.*

Let $Pr'(x, z)$ be the probability that $\sigma'(1, x, r) = z$. Note that $Pr'(x, z) = Pr(x, z)$ if z is not superfluous and is equal to 0 if z is superfluous.

Our nonadaptive rsr (σ'', ϕ'') has the following structure. On input x, it produces $k'(n)$ of its queries by calling σ'. It produces an additional $k''(n)$ queries according to a P-sampleable ensemble described below. It reconstructs $f(x)$ by applying ϕ' to the answers to the queries produced by σ', *i.e.*, it ignores the answers to the other $k''(n)$ queries. The reason for asking

of Definition 1) n times. If u_n belongs to any of these n sets of queries, then accept 1^n, else reject.

If $1^n \in T$, then the above algorithm accepts it with probability at least $\geq 1 - (3/4)^n$, which for $n > 6$ is greater than $3/4$. If $1^n \notin T$, then the witness u_n does not exist and hence is never generated in any of the n simulations of $M_i(1^n)$; this means that the algorithm never accepts an input 1^n that is not in T. Thus, this is an RP algorithm for T. □

Claim 5 *For all i, if R_i is marked fulfilled using the above procedure, then M_i is not a nonadaptive rsr for L.*

Proof of Claim Suppose R_i is marked fulfilled at stage n. The claim is obvious if R_i is marked as fulfilled because the distribution of queries to f is not identical for all $x \in \{0,1\}^n$.

Now, assume that the distribution of queries to L by M_i is identical for all inputs $x \in \{0,1\}^n$. Recall that, if R_i is marked fulfilled, then the probability that u_n is queried is less than $1/4$. Suppose it holds that $\|\mathcal{S}_0\| > \|\mathcal{S}_1\|$. This implies that $u_n \in L$. We claim that the probability that M_i accepts u_n is less than $3/4$. This follows from the facts that $\|\mathcal{S}_w\|$ is less than $1/4$ of the total number of random strings and that a majority (that is, $\|\mathcal{S}_0\|$) of the remaining random strings yield $M_i(u_n) = 0$. Thus, $\|\mathcal{S}_w\| + \|\mathcal{S}_1\|$ is less than $3/4$ of the total random strings. The proof for the case $\|\mathcal{S}_0\| \leq \|\mathcal{S}_1\|$ is identical. □

This completes the proof of the theorem. □

4.2 Collapse

In this section, we relate the question of self-correctability versus random-self-reducibility to the question of #P versus FP.

Theorem 6 *Suppose that*

(A) f is nonadaptively self-correctable with respect to $\mathcal{D}$,

(B) $\mathcal{D}$ is P-sampleable, and

(C) $\#P \subseteq$ FP.

Then f is nonadaptively random-self-reducible.

Proof We first note that (B) and (C) together imply that $\mathrm{Prob}_{\mathcal{D}}(x)$ is a polynomial-time computable function. Let (ϕ, σ) be an ϵ-self-corrector for f with respect to $\mathcal{D}$, where $\epsilon(n) = 1/poly(n)$, and assume without loss of generality that this corrector produces a wrong answer with probability at most 2^{-n}. Let the number of random queries produced by σ be $k(n)$. We will construct a nonadaptive rsr for f.

Recall that, because M_i is nonadaptive, we can assume that, for any particular input, all oracle queries are identically distributed. If the distribution of the oracle queries is not identical for all strings in $\{0,1\}^n$ (that is, there are two strings y and z in $\{0,1\}^n$ and a $q \in \{0,1\}^*$ such that the probability of querying q on inputs y and z is different), then mark R_i as fulfilled, and go to the next stage.

Next assume that the queries to the oracle by M_i on all inputs $x \in \{0,1\}^n$ have identical distributions. One of the following equations must hold.

$$Pr_r[M_i(1^n, r) \text{ queries } u_n \,] \geq 1/4, \text{ or} \tag{1}$$
$$Pr_r[M_i(1^n, r) \text{ queries } u_n \,] < 1/4. \tag{2}$$

If Equation (1) holds, then go to the next stage. Otherwise, if Equation (2) holds, then, for all strings r in $\{0,1\}^{w_i(n)}$, simulate $M_i(u_n, r)$.

Because $p(n) = n^c$, for some integer $c > 2$, and $i \leq n$, it follows that, for all n,

$$p_i(2^{2^{p(n)}}) = 2^{i2^{p(n)}} < 2^{2^{p(n+1)}}.$$

Thus, for sufficiently large n, $M_i(1^n)$ cannot query any string of good length m such that $m > n$. This implies that, if a simulation of M_i does not query u_n, its output does not depend on the membership of u_m in L, for any $m \geq n$. We let $\mathcal{Q}(M_i, z, r)$ denote the set of queries made by $M_i(z, r)$, and let $M_i(z, r)$ denote the output of $M_i(z, r)$. We partition the set $\{0,1\}^{w_i(n)}$ into the following three sets.

$$\begin{aligned}
\mathcal{S}_w &= \{r \mid u_n \in \mathcal{Q}(M_i, u_n, r)\} \\
\mathcal{S}_0 &= \{r \mid u_n \notin \mathcal{Q}(M_i, u_n, r) \text{ and } M_i(u_n, r) = 0\} \\
\mathcal{S}_1 &= \{r \mid u_n \notin \mathcal{Q}(M_i, u_n, r) \text{ and } M_i(u_n, r) = 1\}
\end{aligned}$$

If $\|\mathcal{S}_0\| > \|\mathcal{S}_1\|$, then leave u_n in L, else remove u_n from L. Finally, mark requirement R_i fulfilled and go to the next stage. This completes the construction of L.

The proof that χ_L is not nonadaptively rsr is implied by the following two claims.

Claim 4 *For all i, there is a stage ℓ such that R_i is fulfilled at stage ℓ.*

Proof We prove this by contradiction. Suppose that at least one requirement remains unfulfilled at the end of the construction. Note that this means that there must be a unique, smallest i such that R_i remains unfulfilled. Let N be the smallest stage at which we attempted to fulfill this R_i. Recall that R_i can remain unfulfilled only if, for all good numbers m such that $1^m \in T$ and $m > N$, $Pr[M_i(1^m) \text{ queries } u_m \,] \geq 1/4$. We show that the above equation contradicts the hypothesis that T is in $\mathrm{UP} - \mathrm{RP}$ by giving an RP algorithm for T.

On input 1^n, if $n < N$, then use a finite look-up table to determine whether $1^n \in T$. If $n \geq N$, then simulate the query-generation part of $M_i(1^n)$ (i.e., the calls to the function σ

Proof By standard padding arguments [Boo74], it follows from the hypothesis that there exists a language $T \in \mathrm{UP} - \mathrm{RP}$ such that $T \subseteq \{1^{2^{2^{p(n)}}} \mid n \in N\}$, where p is a monomial n^c, for some integer $c > 2$; this condition on $p(n)$ ensures that there are sufficiently large gaps between consecutive elements of T. We say that a natural number n is *good* if there exists an integer j such that $n = 2^{2^{p(j)}}$. For all n such that $1^n \in T$, let u_n be the unique witness of this fact. Assume, without loss of generality, that $|u_n| = n$, for all n such that $1^n \in T$. Finally, let $\mathcal{W}$ be the set of all witnesses u_n.

For each $n \geq 1$, define a distribution $\mathcal{D}_n$ on $\{0,1\}^n$ as follows. If $1^n \in T$, then for each $x \in \{0,1\}^n$: if $x \in \mathcal{W}$, then $Pr_{\mathcal{D}_n}(x) = 1$, else $Pr_{\mathcal{D}_n}(x) = 0$. If $1^n \notin T$, then for each $x \in \{0,1\}^n$: $Pr_{\mathcal{D}_n}(x) = \frac{1}{2^n}$. Let $\mathcal{D}$ be the ensemble $\{\mathcal{D}_n\}_{n \geq 1}$. All subsets of $\mathcal{W}$ have the following useful property.

Claim 3 *If L is a subset of $\mathcal{W}$, then, for all $\epsilon < 1$, χ_L is ϵ-self-correctable with respect to $\mathcal{D}$.*

Proof It follows directly from the definition of $\mathcal{D}$ that, for any $\epsilon < 1$, any program P that computes χ_L with error probability at most ϵ with respect to $\mathcal{D}$ cannot commit an error on any string in $\mathcal{W}$. Furthermore, membership in $\mathcal{W}$ can be tested in polynomial time. Thus, a self-corrector M for χ_L works as follows on input x: If $x \in \mathcal{W}$, then M outputs $P(x)$; else M outputs 0. □

By Claim 3, it suffices to construct a language L that is a subset of $\mathcal{W}$ but is not nonadaptively rsr. We give a recursive procedure that constructs such an L.

Suppose $\{M_i\}$ is an enumeration of all probabilistic polynomial-time nonadaptive oracle Turing machines in which the running time of M_i is $p_i(n) = n^i$. We assume that each probabilistic Turing machine M_i is implemented as a deterministic machine that takes two arguments as input, namely, an input string x and a random string r such that $|r| = w_i(|x|)$ for some polynomial w_i. The language L is constructed such that, for all $i \in N$, the following set of requirements $\{R_i\}$ is fulfilled:

(*) R_i : M_i is not a nonadaptive rsr for χ_L

We fulfill these requirements in stages. Initially, let L be $\mathcal{W}$. At stage n, we decide whether we should remove u_n from L. If $1^n \in T$, then we attempt to fulfill a requirement at stage n; if we succeed, then we mark the requirement *fulfilled*, else, we mark it *unfulfilled*. All the requirements are initially marked unfulfilled.

The main idea of the construction is as follows. The language L will be a subset of $\mathcal{W}$ and hence nonadaptively self-correctable. If all of the requirements are fulfilled during the construction, then L cannot be nonadaptively rsr. To show that all requirements are fulfilled, it suffices to prove that if there is any unfulfilled requirement, then $T \in \mathrm{RP}$, which is a contradiction.

We now describe the construction of L formally. At stage n, if $1^n \notin T$, then do nothing and go to the next stage. Else, if $1^n \in T$, then let i be the smallest integer such that requirement R_i is as yet unfulfilled.

Claim 2 $\alpha \geq 2^{n/4} - a(n) - c$

Proof To get the claimed lower bound on α, consider the following description of h_n: for the $B_n - \alpha$ values whose last bit is never queried, we will encode all but the last bit of each of these strings, and we will encode the remaining α strings in their entirety. The last bit of each of the $B_n - \alpha$ values would be constructed by simulating the examiner E on all B_n boundary values. Note that, since the last bits of the $B_n - \alpha$ values are never queried by E, with all the above information, E can be simulated unambiguously. Using this description of h_n, we get the following upper bound on $C(h_n \mid n)$.

$$C(h_n|n) \leq B_n \log \frac{2^n}{B_n} - (B_n - \alpha) + a(n) + c.$$

Using the fact that $C(h_n|n) \geq B_n \log \frac{2^n}{B_n}$, simple algebraic manipulation now yields the bound α. □

Clearly, if n is large enough, Claims 1 and 2 cannot simultaneously hold, and so a nonadaptive examiner cannot exist for h, even when allowed polynomial advice.

□ **Lemma 4 and Theorem 2**

4 Random-self-reducibility versus Self-Correctability

In this section, we first show that random-self-reducibility and self-correctability are not equivalent, unless a certain implausible complexity hypothesis holds. Next, we show that, if we restrict attention to self-correctable sets under P-sampleable distributions, then constructing a self-correctable function that is not rsr is as hard as proving that FP $\neq$ #P.

4.1 Separation

Let UEEEXP denote the class of languages accepted by unambiguous nondeterministic Turing machines that run in nondeterministic time $2^{2^{2^{p(n)}}}$ for some polynomial p; similarly, let REEEXP denote the class of languages accepted by probabilistic Turing machines with one-sided error in time $2^{2^{2^{p(n)}}}$ for some polynomial p. For any language L, let χ_L denote the characteristic function of L.

Theorem 5 *If* UEEEXP $\not\subseteq$ REEEXP, *then there exists a language L and an ensemble $\mathcal{D} = \{\mathcal{D}_n\}_{n\geq 1}$ such that, for all $\epsilon < 1$, χ_L is nonadaptively ϵ-self-correctable with respect to $\mathcal{D}$ but* not *nonadaptively random-self-reducible.*

Proof Assume, by way of contradiction, that there exists a deterministic polynomial-time examiner E for h.

Note that, for sufficiently large n, because h is non-zero only on allowed lengths, we can assume that the only non-trivial queries made by E on inputs of length n must also have length n. The next allowed length is exponential in n, and therefore queries of that length cannot be constructed in the time available. We are also able to assume that the values of the function at all the smaller allowed lengths are encoded into the examiner's advice string.

Let E compute h using an advice string of length bounded by a polynomial $a(n)$ and making at most $q(n)$ nonadaptive queries for some polynomial q. To simplify notation, we make the following definition:

$$\tilde{s}_{n,i} = \begin{cases} s_{n,i} & \text{if } s_{n,i} \text{ is even} \\ s_{n,i} - 1 & \text{otherwise} \end{cases}$$

We focus on the behavior of E on all the boundary values, i.e., on inputs $x = BV_{n,i}$ for $1 \leq i \leq B_n$ (where $n = t(m)$ for some m). We show that, if E queries too many inputs of the form $BV_{n,i} + \tilde{s}_{n,i}$, we can shorten the description of the function by encoding $\tilde{s}_{n,i}$ with an index to a query. On the other hand, if the examiner does not query enough inputs of the form $BV_{n,i} + \tilde{s}_{n,i}$, we can shorten the description of the function by running the examiner with partial information. Since the description of f is chosen to be incompressible, we get a contradiction.

Let α be the number of intervals i for which E queries $BV_{n,i} + \tilde{s}_{n,i}$ when its input is one of the boundary values $BV_{n,j}$.

Claim 1 $\alpha \leq \frac{a(n)+c}{\frac{n}{2} - \log q(n) - 1}$

Proof To prove this claim, consider the following description of h_n.

- a complete description of all the $B_n - \alpha$ steps $s_{n,i}$ for which $BV_{n,i} + \tilde{s}_{n,i}$ does not get queried by E on any of the boundary values;
- for the remaining α steps, an index to a block j in which E, on input $BV_{n,j}$, queries $BV_{n,i} + \tilde{s}_{n,i}$, as well as the index to the query itself and the last bit of $s_{n,i}$;
- E's algorithm and advice string.

Note that a program that encodes the above description can be used to construct h_n. Thus,

$$\begin{aligned} C(h_n|n) \quad \leq \quad & \alpha(\log B_n + \log q(n) + 1) \\ & + (B_n - \alpha)(\log \frac{2^n}{B_n}) + a(n) + c \end{aligned}$$

Using the fact that $C(h_n|n) \geq B_n \log \frac{2^n}{B_n}$, simple algebraic manipulation yields the claimed upper bound on α, concluding the proof of Claim 1. □

interval $[2^n + (i-1)\frac{2^n}{B_n}, 2^n + i\frac{2^n}{B_n} - 1]$. The first value in the i^{th} block is called the *boundary value* for block i and is denoted by $BV_{n,i} = 2^n + (i-1)\frac{2^n}{B_n}$. Let $s_{n,1}, \ldots, s_{n,B_n}$ be arbitrary but fixed values between 1 and $\frac{2^n}{B_n}$.

On each block, define $f_n|_{[BV_{n,i}, BV_{n,i+1}-1]} = f_{n,i}$ as follows:

$$f_{n,i}(x) = \begin{cases} s_{n,i} \bmod 2 & \text{if } x = BV_{n,i} \\ 0 & \text{if } x < BV_{n,i} + s_{n,i} \\ 1 & \text{if } x \geq BV_{n,i} + s_{n,i} \end{cases}$$

We call $s_{n,i}$ the "step" of the i^{th} block for length n. Note that each possible set of steps $\mathcal{S} = \{s_{n,i} \mid m \in N, n = t(m), 1 \leq i \leq B_n\}$ completely determines a function f. Let $\mathcal{F}$ denote the set of functions each defined by a set of steps as described above.

We first prove that all functions $f \in \mathcal{F}$ are adaptively coherent; then we prove that there exists a function $h \in \mathcal{F}$ that has no nonadaptive examiner even when provided with polynomial advice.

Lemma 3 *All functions $f \in \mathcal{F}$ are adaptively coherent.*

Proof We provide an adaptive examiner for f. Given an input x whose length is not an allowed length, the examiner always returns zero. Consider $n = t(m)$ for some m. There are three possibilities:

1. If $x = BV_{n,i}$ for some i, then the examiner uses binary search to find the value of $s_{n,i}$ and outputs the parity of this value.

2. If x lies in the interval $[BV_{n,i}+1, BV_{n,i+1}-1]$ and oracle queries using its two immediate neighbors as inputs produce the same value, then the examiner outputs this value.

3. If the values of f on the neighbors of x differ, then x is either $BV_{n,i} + s_{n,i} - 1$ or $BV_{n,i} + s_{n,i}$. A query to $BV_{n,i}$ distinguishes these two cases, and the examiner returns zero in the first case and one in the second. □

Next, we define the function $h \in \mathcal{F}$ as follows. For each allowed length n, let s_n denote an incompressible string of length $B_n \log \frac{2^n}{B_n} = \frac{3n}{4} B_n$. Suppose that $s_n = s_{n,1} \cdot s_{n,2} \cdots s_{n,B_n}$ where, for all i, $s_{n,i} \in \{0,1\}^{\frac{3n}{4}}$. Let h_n be the function in $\mathcal{F}$ obtained by taking the strings $s_{n,1} \ldots, s_{n,B_n}$ as the steps of length n. Since h_n can be fully constructed from s_n and vice-versa, we write that $C(h_n|n) \geq C(h_n| \, |s_n|) \geq |s_n| = B_n \log \frac{2^n}{B_n}$.

Lemma 4 *The function h is not nonadaptively coherent, even with an examiner that takes polynomial advice.*

probabilistic polynomial-time algorithm that, on input 1^n, outputs each string z with probability exactly $Prob_{\mathcal{D}_n}(z)$. If P is a program that purports to compute f, then $error(P, f, \mathcal{D}_n)$ is the probability that $P(z) \neq f(z)$, when z is chosen randomly according to $\mathcal{D}_n$. If $error(P, f, \mathcal{D}_n) \leq \epsilon(n)$, for all $n \geq 1$, we say that $error(P, f, \mathcal{D}) \leq \epsilon$.

Definition 3 *A* **nonadaptive $\epsilon(n)$-self-corrector** *for the function f with respect to $\mathcal{D}$ consists of a pair of polynomial-time computable functions σ and ϕ with the following property: If $error(P, f, \mathcal{D}) \leq \epsilon$, then, for all x, $f(x) = \phi(x, r, P(\sigma(1, x, r)), \ldots, P(\sigma(k(n), x, r)))$ for at least $3/4$ of all r's in $\{0,1\}^{w(n)}$.*

Adaptive versions of self-correctors can be defined; see [BLR93] for details. We say that "f is nonadaptively self-correctable with respect to $\mathcal{D}$" if it has a nonadaptive $\epsilon(n)$-self-corrector with respect to $\mathcal{D}$ for some function $\epsilon(n) = 1/poly(n)$.

If an ϵ-self-corrector (σ, ϕ) is nonadaptive, we may assume without loss of generality that the random variables $\sigma(1, x, r)$, ..., $\sigma(k(n), x, r)$ are identically distributed for all x. If this were not the case, we could construct a new ϵ-self-corrector (σ', ϕ') for which this is the case as follows. On input x, r', first choose a random permutation π of $\{1, \ldots, k(n)\}$. Let r be the unused portion of r'. For $1 \leq i \leq k(n)$, let $\sigma'(i, x, r') = \sigma(\pi(i), x, r)$. Let $\phi'(x, r', a_1, \ldots, a_{k(n)}) = \phi(x, r, a_{\pi^{-1}(1)}, \ldots, a_{\pi^{-1}(k(n))})$. The same assumption can be made without loss of generality about a nonadaptive random-self-reduction.

3 Adaptiveness versus Advice

In this section, we show that there exist functions that are adaptively coherent but not nonadaptively weakly coherent. The following proposition, which can be proved using a standard Chernoff bound argument, states that it suffices to consider only deterministic examiners with polynomial advice.

Proposition 1 *If E is a nonadaptive examiner that uses an advice string of length $s(n)$, then E can be replaced by a deterministic nonadaptive examiner that uses an advice string of length $s(n) + p(n)$, where $p(n)$ is a polynomial.*

Theorem 2 *There exists a function $f : \{0,1\}^* \longrightarrow \{0,1\}$ that is adaptively coherent but not nonadapively coherent, even with polynomial advice.*

Proof Define $t(m)$ to be a tower of m 2's, i.e., $t(0) = 1$ and, for all $m \geq 1$, $t(m) = 2^{t(m-1)}$. Consider a function f that is zero on all inputs except on those inputs whose length $n = t(m)$ for some natural number m. Such integers $n = t(m)$ are called *allowed lengths*. The restriction of f to strings of an allowed length n, denoted by f_n, is defined using the following scheme.

To simplify notation, we interpret n-bit strings as integers in the range $[2^n, 2^{n+1} - 1]$. The domain $[2^n, 2^{n+1} - 1]$ is divided into $B_n = 2^{n/4}$ equal "blocks," so that the ith block spans the

shortest program that outputs x given y as input. We use the notation $C(x|y)$ to denote the Kolmogorov complexity of x relative to y. More formally,

$$C(x|y) = \min_p\{|p| \mid p(y) = x\}.$$

If x is a string of length n, then we say that x is *incompressible* if $C(x|n) \geq n$. We will use the following well-known fact [LV93]: For each n, there exists at least one incompressible string of length n.

The following definition of random-self-reducibility was first stated formally in [FF93]. It is a special case of "locally random reducibility," which was first formalized in [BFKR91].

Definition 1 *Let $f : \{0,1\}^* \mapsto \{0,1\}^*$, $k(n)$ be a polynomial, and M be a probabilistic polynomial-time oracle Turing machine. M is a* **$k(n)$-random-self-reduction** *for f if it has the following properties.*

1. *For all $x \in \{0,1\}^*$, $f(x) = M^f(x)$ with probability at least 2/3.*
2. *For all $x \in \{0,1\}^*$, $M^f(x)$ makes at most $k(|x|)$ oracle queries $q_1^f(x)$, $q_2^f(x)$, ..., $q_{k(|x|)}^f(x)$.*
3. *For all n, all $x_1, x_2 \in \{0,1\}^n$, and each $i \in \{1, 2, \ldots, k(n)\}$, the random variables $q_i^f(x_1)$ and $q_i^f(x_2)$ are identically distributed.*

If M queries the oracle f nonadaptively, then it is said to be a **nonadaptive $k(n)$-random-self-reduction** *for f. In the nonadaptive case, we denote by σ and ϕ the functions that, respectively, map x and r to the oracle queries and compute $f(x)$ from x, r, and the oracle answers.*

Property 3 is sometimes called the *instance-hiding property* or the *local randomness property*. Note that it is not required to hold if the oracle queried is $f' \neq f$. Furthermore, there is no requirement that the i^{th} and j^{th} random variables, $i \neq j$, be jointly instance-hiding – in fact, $q_i(x)$ and $q_j(x)$ together may determine x.

We say simply that "f is adaptively (nonadaptively) rsr" if it is adaptively (nonadaptively) $poly(n)$-random-self-reducible, and the precise polynomially bounded number of queries does not matter.

We use the following definition of coherent functions.

Definition 2 *A function f is said to be* **coherent** *if, for every $\varepsilon > 0$, there is a probabilistic polynomial-time Turing machine E called the* **examiner** *that, on input x, can query an f-oracle multiple times to obtain the value of f at any point except x. E computes $f(x)$ with probability at least $1 - \varepsilon$. We say that f is* **nonadaptively weakly coherent** *if E takes polynomial advice and makes only nonadaptive queries to its oracle.*

In the following definition of self-correction, based on the one in [BLR93], $\mathcal{D}$ denotes an ensemble $\{\mathcal{D}_n\}_{n\geq 1}$ of probability distributions. The ensemble is *P-sampleable* if there is a

Blum, Luby, and Rubinfeld observed that every random-self-reducible function is self-correctable. Moreover, all known self-correcting schemes use some form of random-self-reducibility. This immediately raises the question of whether the two notions are equivalent, *i.e.*, whether every self-correctable function is random-self-reducible. Our first result gives a conditional negative answer to this question.

> **Theorem**: If UEEEXP $\not\subseteq$ REEEXP, then there exists a function f that is non-adaptively self-correctable but *not* nonadaptively random-self-reducible.

Although the complexity classes UEEEXP and REEEXP may appear obscure, we note that the hypothesis UEEEXP $\not\subseteq$ REEEXP is only used to construct a sufficiently sparse tally language in UP $-$ RP. The existence of arbitrarily sparse intractable languages is often put forth as a reasonable complexity hypothesis; see, *e.g.*, [BBFG91, BF92, FFLS94, HNOS93, Ko87].

Next, we consider functions that are self-correctable with respect to polynomial-time sampleable ensembles. We show that proving that such functions are not random-self-reducible is as hard as proving that FP $\neq$ #P (and hence as hard as proving that P $\neq$ PP).

> **Theorem**: If $\#P \subseteq$ FP, then every function that is nonadaptively self-correctable with respect to a P-sampleable ensemble is also nonadaptively random-self-reducible.

Although the assumption that $\#P \subseteq$ FP is generally believed to be false, it cannot be disproven with current techniques. Therefore, our result says something about the tractability of proving or disproving the equivalence of random-self-reducibility and self-correctability.

We give preliminary definitions in Section 2. Our result on adaptive versus nonuniform coherence is proven in Section 3, and our results on random-self-reducibility versus self-correctability are proven in Section 4. We conclude with open questions in Section 5.

2 Preliminaries

Throughout this paper, f is a function from $\{0,1\}^*$ to $\{0,1\}$ or $\{0,1\}^*$, and x is an arbitrary input for which we would like to determine $f(x)$. 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.

We say that a computation *takes polynomial advice* if $h : N \mapsto \{0,1\}^*$ is such that, for some polynomial $a(n)$, $|h(n)| \leq a(n)$, and, for each x, the computation is provided with $h(|x|)$ as auxiliary input. The function h is called the *advice function*, and $h(n)$ is called the *advice string* for length n. Our definition of computations that take advice is based on the original definition by Karp and Lipton [KL80], but it is slightly more general.

We will use the Kolmogorov complexity notion of *incompressibility* of strings. (See [LV93] for an excellent treatment of the general theory of Kolmogorov complexity.) We say that c is the *Kolmogorov complexity of string x relative to string y* if c is the length of the

1 Introduction

Consider a function f that we wish to compute by a probabilistic, polynomial-time oracle Turing machine M as follows. M is allowed to consult the function f as an oracle $q(n)$ times, for some polynomial q, under the restriction that, for all input strings x and y of length n, for all strings z, and for all i, $1 \leq i \leq q(n)$, the probability that z is the i^{th} query that M makes on input x is identical to the probability that z is the i^{th} query that M makes on input y. If f can be computed in this manner, then it is said to be *random-self-reducible*. Random-self-reducible functions have the useful property that, even though they may be hard to compute directly, they can be computed efficiently using $q(n)$ f-oracles *without revealing their input to the oracles*. In addition to their application to cryptography, these functions have been used extensively in areas such as average-case complexity, lower bounds, program checking, testing, and correcting, and probabilistically checkable proofs. For references and explanations of these applications, see [Fei93].

Yao [Yao90] defined the notion of *coherence*, which is the weakest form of randomized self-reducibility. A function f is *coherent* if there exists a probabilistic polynomial-time oracle Turing machine called the *examiner* that computes f using f as an oracle *without querying the input*. Buhrman, Fortnow, and Torenvliet [BFT95] used the property of deterministic coherence (called *autoreducibility*) as a tool to separate complexity classes. It is known that all nonadaptively random-self-reducible functions are nonadaptively coherent with polynomial-sized advice [BF92].

Despite notable progress in our understanding of these topics [BF92, FF93, FFLS94, BFT95], many complexity-theoretic questions about random-self-reducibility and coherence remain open. This paper examines two of them. We first address the power of adaptiveness and advice in coherence. Feigenbaum *et al.* [FFLS94] showed that there is a random-self-reducible function f that is not *nonadaptively* random-self-reducible. However, the function f they exhibited can be computed in polynomial time if the Turing machine is provided with polynomial-sized advice as an auxiliary input [KL80]. Indeed, all known results that separate adaptive from nonadaptive self-reductions [HNOS93, FFLS94] do so using functions computable in P/poly.

We show that adaptive examiners are more powerful than nonadaptive examiners, even if the nonadaptive ones use polynomial advice.

> **Theorem**: There exists a coherent function that is *not* nonadaptively coherent, even with polynomial advice.

Next, we study the relationship between random-self-reducibility and *self-correctability* [BLR93]. Blum, Luby, and Rubinfeld [BLR93] defined program self-correction in order to address the following question. Let P be a program for which one can determine that the number of inputs on which P errs, while not necessarily zero, is limited. Is it possible to write an auxiliary program C that corrects P's errors with high probability? More precisely, on any input, C should produce the correct answer with high probability, and C may call the (potentially faulty) program P several times in the course of its computation.

ABSTRACT

We address two questions about self-reducibility – the power of adaptiveness in examiners that take advice and the relationship between random-self-reducibility and self-correctability. We first show that adaptive examiners are more powerful than nonadaptive examiners, even if the nonadaptive ones are nonuniform. Blum *et al.* [Blum, Luby and Rubinfeld, Journal of Computer and System Sciences, 59:549–595, 1993] showed that every random-self-reducible function is self-correctable. However, whether self-correctability implies random-self-reducibility is unknown. We show that, under a reasonable complexity hypothesis, there exists a self-correctable function that is not random-self-reducible. For P-sampleable distributions, however, we show that constructing a self-correctable function that is not random-self-reducible is as hard as proving that $\mathrm{P} \neq \mathrm{PP}$.

These results were presented in preliminary form at the IEEE Conference on Computational Complexity Theory, Philadephia PA, May 1996.

DIMACS Technical Report 96-16
June, 1996

On Coherence, Random-Self-Reducibility, and Self-Correction

by

Joan Feigenbaum[1]
AT&T Research
600 Mountain Avenue
Murray Hill, NJ 07974
jf@research.att.com

Lance Fortnow[2]
Computer Science Department
University of Chicago
Chicago, IL 60637
fortnow@cs.uchicago.edu

Sophie Laplante[3]
Computer Science Department
University of Chicago
Chicago, IL 60637
sophie@cs.uchicago.edu

Ashish V. Naik[4]
Computer Science Department
University of Chicago
Chicago, IL 60637
naik@cs.uchicago.edu

[1]Permanent Member of DIMACS
[2]Research supported in part by NSF grant CCR 92-53582.
[3]Partially supported by an NSERC doctoral fellowship and NSF grant CCR 92-53582.
[4]Research conducted in part at State University of New York at Buffalo and supported in part by NSF grant CCR 92-53582.

DIMACS is a cooperative project of Rutgers University, Princeton University, AT&T Bell Laboratories and Bellcore.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.