Autoreducibility of Random Sets: a Sharp Bound on the Density of Guessed Bits

Todd Ebert[1] and Wolfgang Merkle[2]

[1] California State University, Long Beach,
Department of Computer Engineering and Computer Science,
1250 Bellflower Blvd., Long Beach, Ca 90840, U.S.A.,
e-mail address: ebert@cecs.csulb.edu

[2] Ruprecht-Karls-Universität Heidelberg, Mathematisches Institut,
Im Neuenheimer Feld 294, D–69120 Heidelberg, Germany,
e-mail address: merkle@math.uni-heidelberg.de

Abstract. A set $A \subseteq \{0,1\}^*$ is called i.o. Turing-autoreducible if A is reducible to itself via an oracle Turing machine that never queries its oracle at the current input, outputs either $A(x)$ or a don't-know symbol on any given input x, and outputs $A(x)$ for infinitely many x. If in addition the oracle Turing machine terminates on all inputs and oracles, A is called i.o. truth-table autoreducible. Ebert and Vollmer obtained the somewhat counterintuitive result that every Martin-Löf random set is i.o. truth-table-autoreducible and investigated the question of how dense the set of guessed bits can be when i.o. autoreducing a random set.
We show that rec-random sets are never i.o. truth-table-autoreducible such that the set of guessed bits has strictly positive constant density in the limit, and that a similar assertion holds for Martin-Löf random sets and i.o. Turing-autoreducibility. On the other hand, our main result asserts that for any computable function r that goes non-ascendingly to zero, any rec-random set is i.o. truth-table-autoreducible such that the set of guessed bits has density bounded from below by $r(m)$.

1 Introduction

For the moment, call a set A *autoreducible* if there is an effective procedure that on input x computes $A(x)$ while having access to the values $A(y)$ for $y \neq x$. Intuitively speaking, for an autoreducible set the information on $A(x)$ can be effectively recovered from the remainder of the set. For example, for any set X the set $\{x0 : x \in X\} \cup \{x1 : x \in X\}$ is autoreducible. Autoreducibility was first studied by Trakhtenbrot [20], see Wagner and Wechsung [22, pp. 483ff]. Since then, the concept of autoreducibility has received some attention in the literature. For example, Buhrman et al. [5] emphasize the role of time-bounded versions of autoreducibility as a tool for proving separation results in complexity theory.

Now consider the question of whether an effective random set, say, a Martin-Löf random set, can be autoreducible. By definition, the bits of an autoreducible set depend on each other in an effective way. This suggests that by exploiting the dependencies, we might come up with an effective betting strategy that succeeds on this set. Indeed, Martin-Löf-random sets are never autoreducible and, similarly, rec-random sets are not autoreducible by reductions that are confined to non-adaptive queries. The two latter assertions can be obtained as corollaries to Theorem 7 below.

Furthermore, we might ask whether for a random set R it is at least possible to recover some of the values $R(x)$ from the values $R(y)$ with $y \neq x$. Trivially, this is possible for finitely many places x, so consider the case of infinitely many x. For the moment, call a set A *i.o. autoreducible* if there is an effective procedure that for infinitely many x computes $A(x)$ while having access to the values $A(y)$ for $y \neq x$, whereas for all other inputs the procedure signals "don't-know". For example, any set that has an infinite computable subset is i.o. autoreducible, and, in particular, any infinite recursively enumerable set is i.o. autoreducible. On the other hand, by a standard diagonalization argument one can easily construct a set that is not i.o. autoreducible.

For an i.o. autoreducible set R there are infinitely many places x where the value of $R(x)$ depends in an effective way on the remainder of the set R. Thus at first glance one may be tempted to assume that effective random sets are not i.o. autoreducible. So the following result of Ebert and Vollmer [7, 8] seems counterintuitive. *Every* Martin-Löf random set is i.o. autoreducible. Moreover, this result has a finite version known as the *hat problem* which is described in section 5 and has received substantial attention outside of academia [16].

In addition to the issue of simply i.o. autoreducing a random set, Ebert and Vollmer [8] consider the question of how dense the set of guessed bits can be when i.o. autoreducing a random set. A set is i.o. autoreducible with density $r(m)$ if it is i.o. autoreducible such that for all m, at least a fraction of $r(m)$ of the first m bits of the set is guessed. In Theorem 7 it is shown that rec-random sets are never i.o. truth-table-autoreducible with strictly positive constant density (i.e., with density $r(m) = \varepsilon m$ for some $\varepsilon > 0$), and that a similar assertion holds with respect to Martin-Löf random sets and i.o. Turing-autoreducibility. On the other hand, Theorem 11, our main result, asserts that for any computable function r that goes non-ascendingly to 0, any rec-random set is i.o. truth-table-autoreducible with density $r(m)$. So we obtain essentially matching bounds on the density of guessed bits for i.o. truth-table-autoreductions of rec-random sets. These results improve on the work of Ebert and Vollmer and, in particular, our results close the gap between the previously shown bounds, which amount to constant density (for i.o. truth-table autoreductions of Martin-Löf random sets only) and to density worse than $1/\sqrt{m}$. Furthermore, our results on constant bounds are shown by standard probabilistic arguments, which are simpler than the proofs used previously.

1.1 Notation

We use standard notation which is elaborated further in the references [2, 4, 11]. We consider words over the binary alphabet $\{0, 1\}$, which are ordered by the usual length-lexicographical ordering and the $(i+1)$st word in this ordering is denoted by s_i, hence for example s_0 is the empty word λ. Occasionally, we identify words with natural numbers via the mapping $i \mapsto s_i$.

The terms *set* and *class* refer to sets of words and to sets of sets of words, respectively. For any set A, we write $A(x)$ for the value of the characteristic function of A at x and we identify A with the infinite binary sequence $A(0)A(1)\dots$.

An *assignment* is a (total) function from some subset of the natural numbers to $\{0,1\}$. An assignment is *finite* iff its domain is finite. An assignment with domain $\{0,\dots,n-1\}$ is identified in the natural way with a word of length n. The restriction of an assignment β to a set I is denoted by $\beta|I$, thus, in particular, for any set X, the assignment $X|I$ has domain I and agrees there with X.

The class of all sets is referred to as *Cantor space* and is denoted by $\{0,1\}^\infty$. The class of all sets that have a word x as common prefix is called the *cylinder generated by* x and is denoted by $x\{0,1\}^\infty$. For a set W, let $W\{0,1\}^\infty$ be the union of all the cylinders $x\{0,1\}^\infty$ where the word x is in W. We write Prob[.] for probability measures and $\mathbf{E}$[.] for expected values; unless stated otherwise, all probabilities refer to the uniform measure on Cantor space.

2 Random sets

In this section, we briefly review effective random sets and related concepts. For more details please refer to the references [1, 2, 11].

Imagine a player that successively places bets on the individual bits of the characteristic sequence of an unknown set A. The betting proceeds in rounds $i = 1, 2, \dots$. During round i, the player receives as input the length $i-1$ prefix of A and then, first, decides whether to bet on the ith bit being 0 or 1 and, second, determines the stake that shall be bet. The stake might be any fraction between 0 and 1 of the capital accumulated so far, i.e., in particular, the player is not allowed to incur debts. Formally, a player can be identified with a *betting strategy* $b : \{0,1\}^* \to [-1,1]$ where on input w the absolute value of $b(w)$ is the fraction of the current capital that shall be at stake and the the bet is placed on the next bit being 0 or 1 depending on whether $b(w)$ is negative or non-negative.

The player starts with positive, finite capital. At the end of each round, in case of a correct guess, the capital is increased by that round's stake and, otherwise, is decreased by the same amount. So given a betting strategy b, we can inductively compute the corresponding *payoff function* d_b by applying the equations

$$d_b(w0) = d_b(w) - b(w)\cdot d_b(w), \qquad\qquad d_b(w1) = d_b(w) + b(w)\cdot d_b(w)\ .$$

Intuitively speaking, the payoff $d_b(w)$ is the capital the player accumulates till the end of round $|w|$ by betting on a set that has the word w as a prefix.

Definition 1. *A betting strategy b* succeeds *on a set A if the corresponding payoff function d_b is unbounded on the prefixes of A, i.e., if*

$$\limsup_{m\in\omega}\ d_b(A|\{0,\dots,m\})\ =\ \infty.$$

A betting strategy is computable if it is confined to rational values and there is a Turing machine that on input w outputs an appropriate finite representation of $b(w)$. In the context of recursion theory, usually computable betting strategies are considered [1, 17, 19, 23], while in connection with complexity classes one considers betting strategies that in addition are computable within appropriate resource-bounds [2, 10, 11, 13].

Definition 2. *A set is* rec-random *if no computable betting strategy succeeds on it.*

Besides rec-random sets, we consider Martin-Löf-random sets [12]. For the sake of brevity, we only state their definition and in passing note that the class of Martin-Löf-random sets is properly contained in the class of rec-random sets, where both classes have uniform measure 1. Please see Li and Vitányi [9] for more details. Let $\mathrm{W}_0, \mathrm{W}_1, \ldots$ be the standard enumeration of the recursively enumerable sets [18].

Definition 3. *A class $\mathcal{N}$ is called a* Martin-Löf null class *iff there exists a computable function $g\colon \mathbb{N} \to \mathbb{N}$ such that for all i*

$$\mathcal{N} \subseteq \mathrm{W}_{g(i)}\{0,1\}^\infty \quad \textit{and} \quad \mathrm{Prob}[\mathrm{W}_{g(i)}\{0,1\}^\infty] < \frac{1}{2^i}\,.$$

A set is Martin-Löf-random *is it is not contained in any Martin-Löf null class.*

We conclude this section by stating a well-known result for the construction of betting strategies, and leave its proof as an exercise (hint: for each round, distribute the capital uniformly with respect to the elements of Θ that are consistent with the information currently known).

Remark 4 *Let I be a finite set and let Θ be a non-empty subset of all assignments on I. Then there is a betting strategy b that, by betting on places in I, increases its capital by a factor of $\alpha = 2^{|I|}/|\Theta|$ for all sets B where the restriction of B to I is in Θ.*

3 Autoreducibility

In this section, we give formal definitions of the concepts of autoreducibility of Turing and truth-table type. Recall the concept of an *oracle Turing machine* [4], which is a Turing machine that during its computation has access to a set X, the oracle. In case an oracle Turing machine M with oracle X terminates on input x, let $M(X, x)$ denote the computed value; otherwise, $M(X, x)$ is undefined. But rather than use standard oracle machines M whose defined outputs $M(X, x)$ belong to $\{0, 1\}$, we also allow the machines to output a special "don't-know-symbol" $\perp$, which has the intended meaning of signaling that the correct value is not known. We emphasize however that the results of this paper still hold when using the standard model, in that rather than output "don't-know", the Turing machine would simply query the oracle about the value of x, and output that value. This of course would require a slightly different definition of i.o. autoreducibility, which was originally used by Ebert [7].

Definition 5. *Let M be an oracle Turing machine (with output in $\{0, 1, \perp\}$) and let $\mathrm{Q}(M, A, x)$ be the set of query words occurring during the computation of M on input x and with oracle A. Then M* autoreduces *a set A* on a set *E if and only if, for all inputs x, $x \notin \mathrm{Q}(M, A, x)$ and either $A(x) = M(A, x)$ or $M(A, x) = \perp$, with the case $A(x) = M(A, x)$ occurring for all x in E.*

A set A is i.o. Turing autoreducible *iff it is autoreduced by some oracle Turing machine on an infinite set. A set A is* Turing autoreducible *iff it is autoreduced by some oracle Turing machine on the set of all words.*

The definitions of i.o. truth-table autoreducibility *and* truth-table autoreducibility *are the same, except that M is now required to be total, i.e., for all oracles X and for all inputs x, the computation of $M(X, x)$ eventually terminates. Such machines can be represented by a pair of computable functions $h(x)$ and $g(x)$ where, for input x, $h(x)$ computes the query words and $g(x)$ computes the machine output as a function of how the queries about those words are answered. From here on* T-autoreducibility *and* tt-autoreducibility *stand for Turing and truth-table autoreducibility respectively.*

In addition to i.o. autoreducing a random set, Ebert and Vollmer [8] considered the density of the set of guessed bits when i.o. autoreducing a random set.

Definition 6. *For all $m > 0$, the* density *$\rho(E, m)$ of a set E up to m is defined by*

$$\rho(E, m) \;=\; \frac{|E \cap \{\mathrm{s}_0, \dots, \mathrm{s}_{m-1}\}|}{m} \,.$$

An oracle Turing machine M i.o. T-autoreduces a set X with density *$r(m)$ if M i.o. T-autoreduces X on a set E such that $\rho(E, m) \geq r(m)$ for all $m > 0$ (i.e., the density of the set of words x such that M guesses $X(x)$ is always at least $r(m)$).*

A set A is called i.o. T-autoreducible with density *$r(m)$ if there is some oracle Turing machine that i.o. T-autoreduces A with density $r(m)$. Concepts like* i.o. tt-autoreducible with density *$r(m)$ are defined in the same manner.*

Relative to the above definition, Ebert and Vollmer [8] proved that no Martin-Löf random set is i.o. tt-autoreducible with strictly positive constant density (i.e., with density $r(m) = \varepsilon m$ for some constant $\varepsilon > 0$), and that every Martin-Löf random set can be i.o. tt-autoreduced with density bounded above by $1/\sqrt{m}$.

We improve on these results in two respects. First, in Theorem 7, we show that no rec-random set is i.o. tt-autoreducible with positive constant density and that Martin-Löf random sets are not i.o. T-autoreducible with such density. Second, in Theorem 11, we close the gap between constant density and density $1/\sqrt{m}$ by showing that any rec-random set can be i.o. tt-autoreduced with density $1/g(m)$ for any unbounded and non-decreasing computable function g. Equivalently, we can achieve density $r(m)$ for any computable rational-valued function r that goes non-ascendingly to 0. Hence, when tt-autoreducing a rec-random set, the fraction of guessed bits cannot be constant in the limit but can be chosen to decrease as slowly as any computable function that goes non-ascendingly to 0.

4 Autoreducing random sets: negative results on density

Ebert and Vollmer [8] show that no Martin-Löf-random set is i.o. tt-autoreducible with positive constant density. Theorem 7 extends this statement to the larger class of rec-random sets, while for Martin-Löf-random sets the result is strengthened to i.o. T-autoreducibility. Theorem 7 is demonstrated by probabilistic arguments, which are more simple than previous methods.

Theorem 7. (a) *No rec-random set is i.o. tt-autoreducible with strictly positive constant density.*

(b) *No Martin-Löf-random set is i.o. T-autoreducible with strictly positive constant density.*

Proof. The proofs of (a) and (b) rely on Claims 1 and 2 below. For the sake of brevity, we only demonstrate (a). Assertion (b) then follows by a similar but slightly more involved argument. Moreover, the proof of Claim 1 is left as an easy exercise.

Fix an oracle Turing machine M and a rational $\varepsilon_0 > 0$. We want to show that if M i.o. autoreduces a set with density ε_0, then this set cannot be random, i.e., an appropriate betting strategy succeeds on this set. We argue that for any w, the fraction of sets in the cylinder $w\{0,1\}^\infty$ that are i.o. autoreduced by M with density ε_0 is bounded by a constant $\delta < 1$. Let $\varepsilon = \varepsilon_0/2$ and for any m let $i(m) = \lceil \frac{m}{\varepsilon} \rceil$ and $I(m) = \{s_m, \ldots, s_{m+i(m)-1}\}$; that is, in the case where $m = |w|$, the interval $I(m)$ contains the first $i(m)$ words not in the domain of w. For any set X and any finite set I, let

$$\begin{aligned} \text{correct}(X, I) &= |\{x \in I : M(X,x) = X(x)\}|, \\ \text{incorrect}(X, I) &= |\{x \in I : M(X,x) = 1 - X(x)\}| . \end{aligned}$$

That is, the set of all inputs x in I such that $M(X,x)$ is defined and differs from $\perp$ is partitioned into the inputs of correct and incorrect guesses. In the remainder of this proof and with M and ε understood from the context, we say a set X is consistent with a word w if

(i) X is an extension of w,
(ii) for all x in $I(m)$, $M(X,x)$ exists and is computed without querying the oracle at place x,
(iii) $\text{incorrect}(X, I(m)) = 0$,
(iv) $\text{correct}(X, I(m)) \geq \varepsilon |I(m)|$.

Claim 1 *If M i.o. T-autoreduces a set with density ε_0, then this set is consistent with any prefix of itself.*

For any word w, let the random set X_w be an extension of w where the bits of X_w that are not already determined by w are obtained by independent tosses of a fair coin. Let $\delta = 1/(1+\varepsilon)$ and observe that $\delta < 1$ due to $\varepsilon > 0$.

Claim 2 *For any word w, the probability that X_w is consistent with w is at most δ.*

Proof. Fix w of length m. We claim that

$$\mathbf{E}[\text{correct}(X_w, I(m))] \quad = \quad \mathbf{E}[\text{incorrect}(X_w, I(m))] \ . \tag{1}$$

For proof, observe that the values $X_w(y)$ with y in $I(m)$ are stochastically independent, and hence by (ii) the same holds for $X_w(x)$ and $M(X_w, x)$. Furthermore, since $X_w(x)$ is chosen uniformly from $\{0, 1\}$, it follows that the expected number of correct and incorrect answers at x are equal. Thus, Equation 1 follows by linearity of expectation.

We conclude the proof of Claim 2 by distinguishing two cases.

Case I: $\mathbf{E}[\text{correct}(X_w, I(m))] \leq \delta\varepsilon|I(m)|$.

The random variable correct$(X_w, I(m))$ is non-negative, hence the case assumption implies that the probability that correct$(X_w, I(m))$ is at least $\varepsilon|I(m)|$ is at most δ. Also, the probability that X_w is consistent with w is at most δ by condition (iv) in the definition of consistency.

Case II: $\mathbf{E}[\text{correct}(X_w, I(m))] > \delta\varepsilon|I(m)|$.

By (1), we also have $\mathbf{E}[\text{incorrect}(X_w, I(m))] > \delta\varepsilon|I(m)|$. The latter implies that the probability that incorrect$(X_w, I(m))$ is strictly larger than 0 is at least $\delta\varepsilon$ since by definition, the random variable incorrect$(X_w, I(m))$ is bounded by $|I(m)|$. By condition (iii), the probability that X_w is consistent is then at most

$$1 - \delta\varepsilon = 1 - \frac{\varepsilon}{1+\varepsilon} = \frac{1}{1+\varepsilon} = \delta \ . \qquad \square$$

Proof of (a). We can assume that M is in fact a tt-reduction, say, $M = (g, h)$. Fix A such that M i.o. tt-autoreduces A with density $r(m) = \varepsilon_0$. It suffices to show that A is not rec-random, and this is done by constructing a computable betting strategy that succeeds on A.

Partition the set of words into consecutive intervals $J_0, J_1, \ldots$ where the cardinality of J_i is denoted by l_i. The J_i are defined via specifying the l_i inductively. For all i, let $s_i = l_0 + l_1 + \ldots + l_i$. Let $l_0 = 1$ and for all $i \geq 0$ choose l_{i+1} so large that J_{i+1} contains $I(s_i)$ as well as the query sets $h(x)$ for all x in $I(s_i)$. This way for any word w of length s_i, the consistency with w of any set X that extends w depends only on the restriction of X to the interval J_{i+1}. Call an assignment on J_{i+1} consistent with such a word w if the assignment is the restriction of a set that is consistent with w. Then for any given word w of length s_i, the following holds with respect to the assignments on J_{i+1}: it can be decided effectively whether a given assignment is consistent with w; if w is a prefix of A, then the assignment obtained by restricting A is consistent; the consistent assignments comprise a fraction of at most δ of all assignments.

These assertions hold by definition of consistency and by Claims 1 and 2, respectively. Now consider the betting strategy that for each interval J_i, uses the capital accumulated up to the first element of the interval in order to bet according to Remark 4 against all assignments on this interval that are not consistent with the already seen length s_i prefix of the unknown set. If the unknown set is indeed A, then by the preceding discussion on each interval,

b increases its capital at least by the constant factor $1/\delta > 1$. Furthermore, the betting strategy b is computable since consistency can be decided effectively. $\square$

As a corollary to Theorem 7, we have that no rec-random set is truth-table autoreducible, and that no Martin-Löf-random set is Turing autoreducible. These corollaries are complemented by the following results of Merkle and Mihailović [14]. There are rec-random sets that are weak truth-table autoreducibility. There are Martin-Löf random sets that are selfreducible, i.e., autoreducible while asking only queries that are strictly smaller than the current input, with respect to the reducibility *being recursively enumerable in.*

A set is p-random if it cannot be succeeded upon by a betting strategy that is computable in polynomial time [2, 11]. Apparently, the arguments used in this section do not carry over to show that p-random sets are not T- or tt-autoreducible via oracle machines that run in polynomial time. The latter statements have been considered by Buhrman et al. [6]. They argue that these questions are closely related to major open questions in complexity theory.

5 Autoreducing random sets: positive results on density

Ebert and Vollmer [8] prove that any Martin-Löf random sequence is i.o. tt-autoreducible and that in fact a certain density can be achieved. These results rely on the use of appropriate error-correcting codes, and our main result Theorem 11 is shown by similar methods. First we review perfect one-error-correcting codes and their relation to i.o. autoreducibility.

Definition 8. *Any subset of* $\{0,1\}^n$ *is called a* code *of (*codeword) length n. *The* unit sphere *around a word* w *is the set of all words in* $\{0,1\}^n$ *that differ from* w *by at most one position. A code* C *of length* n *is called* perfect one-error-correcting *if the set* $\{0,1\}^n$ *is the disjoint union of the unit spheres around the words in* C.

For any perfect one-error-correcting code of length n, it is necessary that the unit-sphere volume of $(n+1)$ divides 2^n. This condition is also sufficient [21].

Fact 9 *Let* $n = 2^k - 1$ *for integer* $k > 0$. *Then there is a perfect one-error-correcting code* C_n *of length* n. *Furthermore, the codes* C_n *can be chosen so that* $\bigcup_{\{n:n=2^k-1\}} \mathrm{C}_n$ *is decidable in polynomial time. For example, such codes are given by the well-known family of binary Hamming codes [21].*

The hat problem and its solution [7, 8, 16] (see also Aspnes et al. [3] for related problems) described in Remark 10 show neatly how error-correcting codes can be used when autoreducing random sets.

Remark 10 *In the hat problem with* n *players, a binary sequence of* n *bits, the true word, is chosen by independent tosses of a fair coin. Player* i *is assigned bit* i *(or, equivalently, is assigned one of two possible hat colors according to bit* i*). Afterwards, each player may cast a guess on its own bit under the following* *conditions. The players know the bits of all other player but not their own bit. Once the game begins, the players are neither allowed to communicate nor do they know whether the other players have already guessed. The team wins if and only if there is no incorrect guess and at least one correct guess.*

We now provide a strategy for the hat problem with n players where n has the form $2^k - 1$. Fix a perfect one-error-correcting code C_n of length n and call the unit spheres with center in C_n the designated spheres. Each player behaves according to the following rule, where the consistent words of player i are the true word and the unique word that differs from the true word at position i. In case both consistent words are in the same designated sphere, guess according to the assumption that the true word is the (unique) consistent word that differs from the center of this sphere. Otherwise, do not guess.

Under this strategy, all players guess incorrectly and the team looses in case the true word is the center of a designated sphere. Otherwise, i.e., if the true word differs from the center of some (uniquely determined) sphere at position i, the team wins since player i guesses correctly while no other players guess. Hence the team wins with probability $1 - 1/(n+1)$, the fraction of non-center words.

The solution of the hat problem described in Remark 10 is used in the proof of Theorem 11.

Theorem 11. *Let $g : \mathbb{N} \to \mathbb{N}$ be any computable function that is unbounded and non-decreasing. Then every rec-random set is i.o. tt-autoreducible with density $r(m) = 1/g(m)$.*

An equivalent formulation of Theorem 11 is the following. For any computable, rational-valued function r that goes non-ascendingly to 0, every rec-random set is i.o. tt-autoreducible with density $r(m)$. This formulation shows that the bound given by Theorem 11 essentially matches the negative result on constant bounds in Theorem 7.

Proof. Fix any rec-random set R. We have to show that R is i.o. tt-autoreducible with density $r(m)$ by some oracle Turing machine M. For every $k > 0$, let $l_k = 2^k - 1$. For further use, fix perfect one-error-correcting codes C_k of length l_k such that given x and k, we can decide effectively whether x is in $\bigcup \mathrm{C}_k$. The function g is unbounded, thus we can define a computable sequence $t_0 < t_1 < \dots$ where $g(t_k) \geq l_{k+1}$ and hence $r(t_k) \leq \frac{1}{l_{k+1}}$. For every $k > 0$, partition the set of words into consecutive intervals $J_k^1, J_k^2, \dots$ of length l_k. Let c_k be minimum such that the t_kth word s_{t_k} is in $J_k^{c_k}$ and let $H_k = J_k^2 \cup \dots \cup J_k^{c_k}$.

We construct M from oracle machines $M_1, M_2, \dots$, to which we refer as modules. Intuitively, module k is meant to ensure density $1/l_k$ in the interval between s_0 and s_{t_k}. Before defining the modules, we describe their properties and how they are combined to form M.

Module k never queries any place outside the set $J_k^1 \cup H_k$ and outputs $\perp$ on all inputs that are not in H_k. Furthermore, on oracle R, module k never makes a wrong guess and guesses exactly one bit in each of the intervals J_k^2 through $J_k^{c_k}$. In addition, we will ensure that the M_k are uniformly effective in the sense that there is an oracle machine M_0 such that the values $M_k(X, x)$ and $M_0(X, \langle x, k\rangle)$ either are both undefined or are both defined and equal (where $\langle ., .\rangle$ denotes the usual pairing function [18]).

Then M is obtained by running the modules in parallel as follows. For input x, if x is equal to s_0 or s_1, then M outputs $R(x)$. Otherwise, M determines the finitely many k such that $x \in H_k$ and simulates the corresponding modules with input x and the current oracle. If any of these modules outputs a value different from $\perp$, then M outputs the value output by the least such module; otherwise, M outputs $\perp$. From the properties stated so far, we can already prove that M works as required.

Claim 1 *The oracle Turing machine M i.o tt-autoreduces the set R with density $r(m)$.*

Proof. From the module assumptions, it is immediate that M is computable, queries its oracle non-adaptively, and never queries the oracle at the current input. Moreover, on oracle R, the modules always guess correctly, hence so does M.

Let E be the set of inputs x such that $M(R, x)$ differs from $\perp$. Fix k and assume that m is in J_k^1 through $J_k^{c_k}$. Then we have $\rho(E, m) \geq 1/l_k$. For $k = 1$, this follows because the intervals have length $l_1 = 1$ and thus, by assumption, module 1 and hence M guess all bits in these intervals. For $k > 1$, M guesses the first two bits of the first interval, while module k and hence M guess at least one bit in the remaining intervals, where the intervals have length l_k. So if m is in interval J_k^j, up to m there are at most jl_k words and at least j guesses, so the density up to m is at least $1/l_k$. Hence $\rho(E, m) \geq 1/l_k$.

In order to prove $\rho(E, m) \geq r(m)$, fix any m and choose k such that $t_{k-1} \leq m < t_k$. Then we have

$$\rho(E, m) \geq \frac{1}{l_k} \geq r(t_{k-1}) \geq r(m) \ , \tag{2}$$

where the inequalities follow because m is in J_k^1 through $J_k^{c_k}$, by choice of t_{k-1}, and since r is non-ascending. $\square$

In order to construct modules that have the required properties, let w_k^j be the restriction of R to J_k^j, i.e., $R = w_k^1 w_k^2 \ldots$, where $|w_k^j| = l_k$, and let $w_k = w_k^1 \oplus \ldots \oplus w_k^{c_k}$, where $\oplus$ is bit-wise exclusive-or. The idea underlying the construction is as follows. The code words in C_k comprise such a small fraction of all words of length l_k that in case infinitely many words w_k were in C_k, there would be a computable betting strategy that succeeds on R. But R is assumed to be rec-random, hence w_k is not in C_k for almost all k. So in order to guess bits of w_k and then also of w_k^1 through $w_k^{c_k}$, we can apply the solution to the hat problem described in Remark 10.

Claim 2 *For almost all k, w_k is not in C_k.*

Proof. We bet on the bits of an unknown set X. Similar to the definition of w_k, let v_k^j be the restriction of X to the interval J_k^j and let $v_k = v_k^1 \oplus \ldots \oplus v_k^{c_k}$. Recall that C_k contains a fraction of $a_k = 1/2^k$ of all length l_k words and observe that the mapping $o_k : u \mapsto v_k^1 \oplus \ldots \oplus v_k^{c_k-1} \oplus u$ is a bijection. Thus o_k maps a fraction of a_k of all length l_k words to C_k. When betting on the places in $J_k^{c_k}$, we have already seen v_k^1 through $v_k^{c_k-1}$. Under the assumption that v_k is in C_k, we can exclude all but a fraction of a_k of the possible assignments to $J_k^{c_k}$ and hence, by betting in favor of these assignments we can increase our capital by a factor of $1/a_k$ in case the assumption is true.

Now consider the following betting strategy. For every k, a portion of a_k of the initial capital 1 is exclusively used for bets on the interval $J_k^{c_k}$. On each such interval, the bets are in favor of the a_k-fraction of assignments that o_k maps to C_k. Then the capital a_k increases to 1 for all k such that v_k is in C_k and consequently the strategy succeeds on any set such that the latter occurs for infinitely many k. But no computable betting strategy can succeed on the rec-random sequence R, hence Claim 2 follows. We leave it to the reader to show that this strategy is indeed computable. Observe in this connection that each word is contained in at most finitely many intervals of the form $J_k^{c_k}$ and consequently at most finitely many of the strategies related to these intervals might act in parallel. □

Now construct the modules. By Claim 1, fix k_0 such that w_k is not in C_k for all $k > k_0$. First assume $k \leq k_0$. Consider the least elements of the intervals J_k^2 through $J_k^{c_k}$ and for all these x, hard-wire $R(x)$ into module k. On all these x, module k outputs $R(x)$, while the module outputs $\perp$ for all other inputs.

Next assume $k > k_0$. On inputs that are not in H_k, module k simply outputs $\perp$. Now consider an input x in H_k and suppose that x is the i_0th element of interval $J_k^{j_0}$, where then $2 \leq j_0 \leq c_k$. For the given oracle X, define v_k and the v_k^j as in the proof of Claim 2, i.e., v_k^j is the restriction of X to J_k^j and v_k is the bit-wise exclusive-or of the v_k^j. Then on input x, module k queries its oracle at the remaining words in $J_k^1 \cup H_k$, i.e., the module obtains all bits of the words $v_k^j = v_k^j(1)v_k^j(2)\ldots v_k^j(l_k)$, $j = 1, \ldots, c_k$, except for the bit $X(x) = v_k^{j_0}(i_0)$. In order to guess this bit, the module tries to guess bit $v_k(i_0)$. From the latter and from the already known bits $v_k^j(i_0)$ for $j \neq j_0$, the bit $v_k^{j_0}(i_0)$ can be computed easily since $v_k(i_0)$ is the exclusive-or of the $v_k^j(i_0)$.

In order to guess $v_k(i_0)$, module k mimics the solution of the hat problem with l_k players that is given in Remark 10. More precisely, the module computes the remaining bits of v_k from the $v_k^j(i)$ and obtains two consistent words u_0 and u_1. In case for some r, the word u_r is in C_k, the module guesses $u_{1-r}(i_0)$ while the module abstains from guessing, otherwise. By assumption on k, on oracle R the word v_k is not in C_k, hence the discussion in Remark 10 shows that module k never guesses incorrectly and guesses exactly one bit in each of the intervals J_k^2 through $J_k^{c_k}$ (in fact, for some i, in each interval bit i is guessed). □

Acknowledgments. We acknowledge helpful hints from Klaus Ambos-Spies and Heribert Vollmer.

References

1. K. Ambos-Spies and A. Kučera. Randomness in computability theory. In P. A. Cholak et al. (eds.), *Computability theory and its applications. Current trends and open problems.* Proceedings of a 1999 AMS-IMS-SIAM joint summer research conference, Boulder, USA, AMS Contemporary Mathematics 257:1–14, 2000.
2. K. Ambos-Spies and E. Mayordomo. Resource-bounded measure and randomness. In A. Sorbi (ed.), *Complexity, logic, and recursion theory*, p. 1-47. Dekker, New York, 1997.
3. J. Aspnes, R. Beigel, M. Furst, and S. Rudich. *On the expressive power of voting polynomials*, Combinatorica 14(2), p. 135-148, 1994.
4. J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity*, vol. I and II. Springer, 1995.
5. H. Buhrman, L. Fortnow, D. van Melkebeek, and L. Torenvliet, *Separating complexity classes using autoreducibility.* SIAM Journal on Computing 29:1497-1520, 2000.
6. H. Buhrman, D. van Melkebeek, K. W. Regan, D. Sivakumar, M. Strauss, *A generalization of resource-bounded measure, with an application to the BPP vs. EXP problem.* SIAM Journal on Computing 30(2):576-601, 2000.
7. T. Ebert. *Applications of Recursive Operators to Randomness and Complexity.* Ph.D. Thesis, University of California at Santa Barbara, 1998.
8. T. Ebert, H. Vollmer. *On the Autoreducibility of Random Sequences.* In: M. Nielsen and B. Rovan (eds.), Mathematical Foundations of Computer Science 2000, Lecture Notes in Computer Science 1893: 333-342, Springer, 2000.
9. M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*, 2nd ed.. Springer, 1997.
10. J. H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, 44:220–258, 1992.
11. J. H. Lutz. The quantitative structure of exponential time. In L. A. Hemaspaandra and A. L. Selman (eds.), *Complexity Theory Retrospective* II, p. 225–260, Springer, 1997.
12. P. Martin-Löf. The definition of random sequences. *Inform. and Control* 9(6):602–619, 1966.
13. E. Mayordomo. *Contributions to the Study of Resource-Bounded Measure.* Doctoral dissertation, Universitat Politècnica de Catalunya, Barcelona, Spain, 1994.
14. W. Merkle and N. Mihailović. *On the construction of effective random sets.* In: Mathematical Foundations of Computer Science 2002, this volume.
15. P. Odifreddi. *Classical Recursion Theory.* North-Holland, Amsterdam, 1989.
16. S. Robinson. Why mathematicians now care about their hat color. *NY Times*, April 10, 2001.
17. C.-P. Schnorr. A unified approach to the definition of random sequences. *Mathematical Systems Theory*, 5:246–258, 1971.
18. R. I. Soare. *Recursively Enumerable Sets and Degrees.* Springer, 1987.
19. S. A. Terwijn. *Computability and Measure.* Doctoral dissertation, Universiteit van Amsterdam, Amsterdam, Netherlands, 1998.
20. B. A. Trakhtenbrot, *On Autoreducibility.* Soviet Math. doklady, 11:814-817, 1970.
21. J.H. van Lint. *Introduction to coding theory*, 3rd edition. Springer, 1999.
22. K. Wagner and G. Wechsung. *Computational Complexity.* Deutscher Verlag der Wissenschaften, Berlin, 1986.
23. Y. Wang. *Randomness and Complexity.* Doctoral dissertation, Universität Heidelberg, Mathematische Fakultät, INF 288, Heidelberg, Germany, 1996.