Asymptotic Density and the Theory of Computability: A Partial Survey

Carl G. Jockusch Jr.(✉) and Paul E. Schupp

Department of Mathematics, University of Illinois at Urbana-Champaign,
1409 West Green Street, Urbana, IL 61801, USA
{jockusch,schupp}@math.uiuc.edu
http://www.math.uiuc.edu/~jockusch/

This paper is dedicated to Rod Downey in honor of his important contributions to computability theory.

Keywords: Asymptotic density · Generic computability · Coarse computability · Generic-case complexity

2010 Mathematics Subject Classification: 03D30 · 03D25 · 03D28 · 03D32 · 03D40

1 Introduction

The purpose of this paper is to survey recent work on how classical asymptotic density interacts with the theory of computability. We have tried to make the survey accessible to those who are not specialists in computability theory and we mainly state results without proof, but we include a few easy proofs to illustrate the flavor of the subject.

In complexity theory, classes such as $\mathcal{P}$ and $\mathcal{NP}$ are defined by using worst-case measures. That is, a problem belongs to the class if there is an algorithm solving it which has a suitable bound on its running time over *all* instances of the problem. Similarly, in computability theory, a problem is classified as computable if there is a single algorithm which solves all instances of the given problem.

There is now a general awareness that worst-case measures may not give a good picture of a particular algorithm or problem since hard instances may be very sparse. The paradigm case is Dantzig's Simplex Algorithm (see [6]) for linear programming problems. This algorithm runs many hundreds of times every day for scheduling and transportation problems, almost always very quickly. There are clever examples of Klee and Minty [21] showing that there exist instances for which the Simplex Algorithm must take exponential time, but such examples are not encountered in practice.

Observations of this type led to the development of *average-case complexity* by Gurevich [12] and by Levin [23] independently. There are different approaches

The authors would like to thank the Simons Foundation for its support.

A. Day et al. (Eds.): Downey Festschrift, LNCS 10010, pp. 501–520, 2017.
DOI: 10.1007/978-3-319-50062-1_30

to the average-case complexity, but they all involve computing the expected value of the running time of an algorithm with respect to some measure on the set of inputs. Thus the problem must be decidable and one still needs to know the worst-case complexity.

Another example of hard instances being sparse is the behavior of algorithms for decision problems in group theory used in computer algebra packages. There is often some kind of an easy "fast check" algorithm which quickly produces a solution for "most" inputs of the problem. This is true even if the worst-case complexity of the particular problem is very high or the problem is even unsolvable. Thus many group-theoretic decision problems have a very large set of inputs where the (usually negative) answer can be obtained easily and quickly.

Such examples led Kapovich et al. [20] to introduce generic-case complexity as a complexity measure which is often more useful and easier to work with than either worst-case or average-case complexity. In generic-case complexity, one considers algorithms which answer correctly within a given time bound on a set of inputs of asymptotic density 1. They showed that many classical decision problems in group theory resemble the situation of the Simplex Algorithm in that hard instances are very rare. For example, consider the word problem for one-relator groups. In the 1930's Magnus (see [24]) showed that this problem is decidable but we still have no idea of the possible worst-case complexities over the whole class of one-relator groups. However, for *every* one-relator group with at least three generators, the word problem is generically linear time by Example 4.7 of [20]. Also, in the famous groups of Novikov [31] and Boone (see [33]) with undecidable word problem, the word problem has linear time generic-case complexity by Example 4.6 of [20].

Although it focused on complexity, the paper [20] introduced a general definition of generic computability in Sect. 9.

Let Σ be a nonempty finite alphabet and let Σ^* denote the set of all finite words on Σ. The *length*, $|w|$, of a word w is the number of letters in w. Let S be a subset of Σ^*. For every $n \geq 0$ let $S\rceil n$ denote the set of all words in S of length less than or equal to n. In this situation we can copy the classical definition of asymptotic density from number theory.

Definition 1.1. For every $n \geq 0$, the *density of S up to n* is

$$\rho_n(S) = \frac{|S\rceil n|}{|\Sigma^*\rceil n|}$$

The *density* of S is

$$\rho(S) = \lim_{n\to\infty} \rho_n(S)$$

if this limit exists.

Definition 1.2. Let $S \subseteq \Sigma^*$. We say that S is *generic* if $\rho(S) = 1$ and S is *negligible* if $\rho(S) = 0$.

It is clear that S is generic if and only if its complement $\overline{S} = \Sigma^* \backslash S$ is negligible. Also, the intersection (union) of finitely many generic (negligible) sets is generic (negligible). This notion of genericity should not be confused with notions of genericity from forcing in computability theory and set theory. The latter are related to Baire category rather than density.

Definition 1.3 ([20])**.** Let S be a subset of Σ^* with characteristic function χ_S. A set S is *generically computable* if there exists a *partial computable function* φ such that $\varphi(x) = \chi_S(x)$ whenever $\varphi(x)$ is defined (written $\varphi(x)\downarrow$) and the domain of φ is generic in Σ^*.

We stress that *all* answers given by φ must be correct even though φ need not be everywhere defined, and, indeed, we do not require the domain of φ to be computable. In studying complexity we can clock the partial algorithm and consider it as not answering if it does not answer within the allotted amount of time.

To illustrate that even undecidable problems may be generically easy, we consider the *Post Correspondence Problem* (PCP). Fix a finite alphabet Σ of size $k \geq 2$. A typical instance of the problem consists of a finite sequence of pairs of words $(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)$, where $u_i, v_i \in \Sigma^*$ for $1 \leq i \leq n$. The problem is to determine whether or not there is a finite nonempty sequence of indices $i_1, i_2, \ldots, i_k$ such that

$$u_{i_1} u_{i_2} \ldots u_{i_k} = v_{i_1} v_{i_2} \ldots v_{i_k}$$

holds.

In other words, can finitely many u's be concatenated to give the same word as the corresponding concatenation of v's? Emil Post proved in 1946 [32] that this problem is unsolvable for each alphabet Σ of size at least 2 and this result has been used to show that many other problems are unsolvable. Our exposition of a fast generic algorithm for the PCP follows the book [29] by Myasnikov, Shpilrain, and Ushakov.

The generic algorithm works as follows. Say that two words u and v are *comparable* if either is a prefix of the other. Given an instance $(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)$ of the PCP determine whether or not u_i and v_i are comparable for some i between 1 and n. If not, output "no". Otherwise, give no output.

If the given instance has a solution $u_{i_1} \ldots u_{i_n} = v_{i_1} \ldots v_{i_n}$, then u_{i_1} and v_{i_1} must be comparable. Hence the above algorithm never gives a wrong answer.

We now show that the algorithm gives an answer with density 1 on the natural stratification of instances of the problem. Let I_s be the set of instances $(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)$ where $n \leq s$ and each word u_i, v_i has length at most s. Each I_s is finite, each $I_j \subseteq I_{j+1}$ and every instance of the PCP belongs to some I_s. Let D_s be the set of instances in I_s for which the algorithm gives an output.

Claim 1.4. $\lim_s \frac{|D_s|}{|I_s|} = 1$

Proof. Put the uniform measure on I_s and let an element $(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)$ of I_s be chosen uniformly at random. To prove the claim, we show that the probability that the algorithm diverges on a random element of I_s approaches 0 as s approaches infinity.

For any fixed values of $v_1, u_2, \ldots, v_n$ the conditional probability that u_1 is a prefix of v_1 is at most $\frac{s+1}{2^s}$ since there at least 2^s words on Σ of length s and at most $s+1$ of these are prefixes of v_1.

Hence, the probability that u_1 is a prefix of v_1 is at most $\frac{s+1}{2^s}$, and the probability that some u_i is comparable with v_i is at most $\frac{2s(s+1)}{2^s}$. So the probability that the algorithm gives no answer on the given instance is at most $\frac{2s(s+1)}{2^s}$, which tends to 0 as s approaches infinity. □

The generic algorithm we described works in quadratic time, so the generic-case complexity of the Post Correspondence Problem is at most quadratic time.

From now on we mainly consider subsets of the set $\mathbb{N} = \{0, 1, \ldots\}$ of natural numbers, which we identify with the set ω of finite ordinals, In terms of the preceding definitions, we are using the 1-element alphabet $\Sigma = \{1\}$ and identifying $n \in \omega$ with its unary representation $1^n \in \{1\}^*$. In this context, we are using classical asymptotic density. If $A \subseteq \mathbb{N}$, then, for $n \geq 1$, the *density of A below n* is

$$\rho_n(A) = \frac{|\{m \in A : m < n\}|}{n}$$

The *(asymptotic) density* $\rho(A)$ of A is $\lim_{n \to \infty} \rho_n(A)$ if this limit exists.

While the limit for density does not exist in general, the *upper density*

$$\overline{\rho}(A) = \limsup_n \{\rho_n(A)\}$$

and the *lower density*

$$\underline{\rho}(A) = \liminf_n \{\rho_n(A)\}$$

always exist.

We use φ_e for the unary partial function computed by the e-th Turing machine. Let W_e be the domain of φ_e. We identify a set $A \subseteq \omega$ with its characteristic function χ_A.

First observe that *every* Turing degree contains a generically computable set. Let $A \subseteq \mathbb{N}$. Let $C(A) = \{2^n : n \in A\}$. Then $C(A)$ is generically computable since the set of powers of 2 is computable and has density 0. All the information about A is in a set of density 0. When given m, the generic algorithm checks if m is a power of 2. If not, the algorithm answers $m \notin C(A)$ and otherwise does not answer. This example shows that one partial algorithm can generically compute uncountably many different sets.

The following sets R_k are extremely useful.

Definition 1.5 ([19], Definition 2.5)**.**

$$R_k = \{m : \ 2^k | m, \ 2^{(k+1)} \nmid m\}.$$

For example, R_0 is the set of odd nonnegative integers. Note that $\rho(R_k) = 2^{-(k+1)}$. The collection of sets $\{R_k\}$ forms a partition of $\omega - \{0\}$ since these sets are pairwise disjoint and $\bigcup_{k=0}^{\infty} R_k = \omega - \{0\}$.

From the definition of asymptotic density it is clear that we have *finite additivity* for densities. Of course we do not have countable additivity for densities in general, since ω is a countable union of singletons. However, we do have countable additivity in the situation where the "tails" of a sequence contribute vanishingly small density to the union of a sequence of sets.

Lemma 1.6 *([19], Lemma 2.6, Restricted countable additivity).* *If $\{S_i\}, i = 0, 1, \ldots$ is a countable collection of pairwise disjoint subsets of ω such that each $\rho(S_i)$ exists and $\overline{\rho}(\bigcup_{i=N}^{\infty} S_i) \to 0$ as $N \to \infty$, then*

$$\rho(\bigcup_{i=0}^{\infty} S_i) = \sum_{i=0}^{\infty} \rho(S_i).$$

Definition 1.7 ([19], Definition 2.7). If $A \subseteq \omega$ then $\mathcal{R}(A) = \bigcup_{n \in A} R_n$.

Our sequence $\{R_n\}$ satisfies the hypotheses of Lemma 1.6, so we have the following corollary.

Corollary 1.8 *([19], Corollary 2.8).* $\rho(\mathcal{R}(A)) = \sum_{n \in A} 2^{-(n+1)}$.

This gives an explicit construction of sets with pre-assigned densities. and shows that every real number $r \in [0, 1]$ is a density.

Proposition 1.9 *([19], Observation 2.11). Every nonzero Turing degree contains a set which is not generically computable since the set $\mathcal{R}(A)$ is generically computable if and only if A is computable.*

Proof. It is clear that $\mathcal{R}(A)$ is Turing equivalent to A. If $\mathcal{R}(A)$ is generically computable by a partial algorithm φ, to compute $A(n)$ search for $k \in R_n$ with $\varphi(k) \downarrow$ and output $\varphi(k)$. Since R_n has positive density, this procedure must eventually answer, and the answer is correct because φ never gives a wrong answer. □

Recall that a set A is *immune* if A is infinite and A does not have any infinite c.e. subset and A is *bi-immune* if both A and its complement $\overline{A}$ are immune. It is clear that no bi-immune set can be generically computable.

Now the class of bi-immune sets is both comeager and of measure 1. This is clear by countable additivity since the family of sets containing a given infinite set is of measure 0 and nowhere dense. Thus the family of generically computable sets is both meager and of measure 0.

There are numerous interactions between the area of this paper and effective randomness. For information on the latter see, for example, [7].

2 Densities and C.E. Sets

Observe that a set A is generically computable if and only if there exist c.e. sets $B \subseteq A$ and $C \subseteq \overline{A}$ such that $B \cup C$ has density 1. In particular, every c.e. set of density 1 is generically computable. This suggests the question of how well c.e. sets can be approximated by computable subsets in general. The following definition gives two ways to measure how good an approximation is.

Definition 2.1 ([8], Definition 3.1). Let $A, B \subseteq \omega$.

(i) Define $d(A, B) = \underline{\rho}(A \triangle B)$, the lower density of the symmetric difference of A and B.
(ii) Define $D(A, B) = \overline{\rho}(A \triangle B)$, the upper density of the symmetric difference of A and B.

To our knowledge the first result on approximating c.e. sets by computable subsets is a result of Barzdin' [3] from 1970 showing that for every c.e. set A and every real number $\epsilon > 0$, there is a computable set $B \subseteq A$ such that $d(A, B) < \epsilon$. We thank Evgeny Gordon for bringing this result to our attention. The following result of Downey, Jockusch, and Schupp improves Barzdin's result from d to D.

Theorem 2.2 *([8], Corollary 3.10). For every c.e. set A and real number $\epsilon > 0$, there is a computable set $B \subseteq A$ such that $D(A, B) < \epsilon$.*

Jockusch and Schupp ([19], Theorem 2.22) showed that there is a c.e. set of density 1 which does not have any computable subset of density 1. It turns out that this property characterizes an important class of c.e. degrees, where a c.e. degree is one which contains a c.e. set. Recall that if $\mathbf{a}$ is a Turing degree with $A \in \mathbf{a}$, then the *jump* of $\mathbf{a}$, denoted $\mathbf{a}'$, is the Turing degree of the halting problem for machines with an oracle for A. If $\mathbf{a}$ is a c.e. degree then $\mathbf{0}' \leq \mathbf{a}' \leq \mathbf{0}''$. A degree $\mathbf{a}$ is *low* if $\mathbf{a}' = \mathbf{0}'$, that is, $\mathbf{a}'$ is as low as possible. A degree $\mathbf{a}$ is *high* if $\mathbf{a}' \geq \mathbf{0}''$.

Downey et al. [8] proved the following characterization of non-low c.e. degrees.

Theorem 2.3 *([8], Corollary 4.4). Let $\mathbf{a}$ be a c.e. degree. Then $\mathbf{a}$ is not low if and only if $\mathbf{a}$ contains a c.e. set A of density 1 with no computable subset of density 1.*

With Eric Astor they also proved the following result.

Theorem 2.4 *([8], Corollary 4.2). There is a c.e. set A of density 1 such that the degrees of subsets of A of density 1 are exactly the high degrees.*

One of the striking things to emerge from considering density and computability is that there is a very tight connection between the positions of sets in the arithmetical hierarchy and the complexity of their densities as real numbers.

Fix a computable bijection between the rationals and $\mathbb{N}$, so we can classify sets of rationals in the arithmetical hierarchy.

Definition 2.5. Define a real number r to be *left-*Σ^0_n if its corresponding lower cut in the rationals, $\{q \in \mathbb{Q} : q < r\}$, is Σ^0_n. Define "left-Π^0_n" analogously.

Jockusch and Schupp [19] proved that a real number $r \in [0,1]$ is the density of a computable set if and only if r is a Δ^0_2 real. Downey et al. [8] carried this much further and proved the following results.

Theorem 2.6 *([19], Theorem 2.21, [8] Corollary 5.4, Theorems 5.6, 5.7, and 5.13). Let r be a real number in the interval $[0,1]$ and suppose that $n \geq 1$. Then the following hold:*

(i) r is the density of some set in Δ^0_n if and only if r is left-Δ^0_{n+1}.
(ii) r is the lower density of some set in Δ^0_n if and only if r is left-Σ^0_{n+1}.
(iii) r is the upper density of some set in Δ^0_n if and only if r is left-Π^0_{n+1}.
(iv) r is the lower density of some set in Σ^0_n if and only if r is left-Σ^0_{n+2}.
(v) r is the upper density of some set in Σ^0_n if and only if r is left-Π^0_{n+1}.
(vi) r is the density of some set in Σ^0_n if and only if r is left-Π^0_{n+1}.

This result follows by relativization from characterizing the densities, upper densities, and lower densities of the computable and c.e. sets.

2.1 Asymptotic Density and the Ershov Hierarchy

The correlation of densities and position in the arithmetical hierarchy is further clarified by considering densities of sets in the Ershov Hierarchy. The Shoenfield Limit Lemma shows that a set A is Δ^0_2 exactly if there is a computable function g such that for all x, $A(x) = lim_s g(x,s)$. Roughly speaking, the Ershov Hierarchy classifies Δ^0_2 sets by the number of s with $g(x,s) \neq g(x,s+1)$. A set A is n-c.e. if there exists a computable function g as above such that, for all x, $g(x,0) = 0$ and there are at most n values of s such that $g(x,s) \neq g(x,s+1)$.

The 1-c.e. sets are just the c.e. sets. The 2-c.e. sets, also called the d.c.e. sets, are sets which are the differences of two c.e. sets. Since the densities of c.e. sets are precisely the left-Π^0_2 reals in the unit interval, one is led to suspect that the densities of the 2-c.e. sets should be exactly the differences of two left-Π^0_2 reals which are in the unit interval. This is true but there is something to prove since the difference of A and B may have a density even though A and B do not have densities. Let $\mathcal{D}_2$ denote the set of reals which are the difference of two left Π^0_2 reals. Downey et al. [9] proved the following results.

Theorem 2.7 *([9], Corollary 4.3). For every $n \geq 2$, the densities of the n-c.e. sets coincide with the reals in $\mathcal{D}_2 \cap [0,1]$.*

It follows that there is a real r which is the density of a 2-c.e. set but not of any c.e. or co-c.e. set.

Say that a Δ^0_2 set A is f-c.e. if there is a computable function g such that, for all x, $g(x,0) = 0$, $A(x) = \lim_s g(x,s)$, and $|\{s : g(x,s) \neq g(x,s+1)\}| \leq f(x)$.

Theorem 2.8 *([9], Corollary 5.2). Let f be any computable, nondecreasing, unbounded function. If A is a Δ^0_2 set that has a density, then the density of A is the same as the density of a set B such that B is f-c.e.*

2.2 Bi-immunity and Absolute Undecidability

If A is bi-immune then any c.e. set contained in either A or $\overline{A}$ is finite so being bi-immune is an extreme non-computability condition. Jockusch [18] proved that there are nonzero Turing degrees which do not contain any bi-immune sets. This raises the natural question of how strong a non-computability condition can be pushed into every non-zero degree. Miasnikov and Rybalov [28] defined a set A to be *absolutely undecidable* if every partial computable function which agrees with A on its domain has a domain of density 0. We might suggest the term *densely undecidable* as a synonym for "absolutely undecidable", since being absolutely undecidable is a weaker condition than being bi-immune. The following beautiful and surprising result is due to Bienvenu et al. [4].

Theorem 2.9 *([4]). Every nonzero Turing degree contains an absolutely undecidable set.*

The theorem was proved using the Hadamard error-correcting code, which the authors of [4] rediscovered to prove the result.

3 Coarse Computability

The following definitions suggest another quite reasonable concept of "imperfect computability".

Definition 3.1 ([19], Definition 2.12). Two sets A and B are *coarsely similar*, which we denote by $A \sim_c B$, if their symmetric difference $A \triangle B = (A \setminus B) \cup (B \setminus A)$ has density 0. If B is any set coarsely similar to A then B is called a *coarse description* of A.

It is easy to check that $\sim_c$ is an equivalence relation. Any set of density 1 is coarsely similar to ω, and any set of density 0 is coarsely similar to $\emptyset$.

Definition 3.2 ([19], Definition 2.13). A set A is *coarsely computable* if A is coarsely similar to a computable set. That is, A has a computable coarse description.

We can think of coarse computability in the following way: The set A is coarsely computable if there exists a *total* algorithm φ which may make mistakes on membership in A but the mistakes occur only on a negligible set. A generic algorithm is always correct when it answers and almost always answers, while a coarse algorithm always answers and is almost always correct. Note that all sets of density 1 or of density 0 are coarsely computable.

Using the Golod-Shafarevich inequality, Miasnikov and Osin [27] constructed finitely generated, computably presented groups whose word problems are not generically computable. Whether or not there exist finitely presented groups whose word problem is not generically computable is a difficult open question. The situation for coarse computability is very different.

Observation 3.3 ([19], Observation 2.14). The word problem of any finitely generated group $G = \langle X : R \rangle$ is coarsely computable.

Proof. If G is finite then the word problem is computable. If G is an infinite group, the set of words on $X \cup X^{-1}$ which are not equal to the identity in G has density 1 and hence is coarsely computable. (See, for example, [35].) □

It is easy to check that the family of coarsely computable sets is meager and of measure 0. In fact, if A is coarsely computable, then A is neither 1-generic nor 1-random. This is a consequence of the fact that if A is 1-random and C is computable, then the symmetric difference $A \triangle C$ is also 1-random, and the analogous fact also holds for 1-genericity. The result now follows because 1-random sets have density 1/2 [30], and 1-generic sets have upper density 1.

Proposition 3.4 *([19], Proposition 2.15). There is a c.e. set which is coarsely computable but not generically computable.*

Proof. Recall that a c.e. set A is *simple* if $\overline{A}$ is immune. It suffices to construct a simple set A of density 0, since any such set is coarsely computable but not generically computable. This is done by a slight modification of Post's simple set construction. Namely, for each e, enumerate W_e until, if ever, a number $>e^2$ appears, and put the first such number into A. Then A is simple, and A has density 0 because for each e, it has at most e elements less than e^2. □

The following construction shows that c.e. sets may be neither generically nor coarsely computable.

Theorem 3.5 *([19], Theorem 2.16). There exists a c.e. set which is not coarsely similar to any co-c.e. set and hence is neither coarsely computable nor generically computable.*

Proof. Let $\{W_e\}$ be a standard enumeration of all c.e. sets. Let

$$A = \bigcup_{e \in \omega} (W_e \cap R_e)$$

Clearly, A is c.e. We first claim that A is not coarsely similar to any co-c.e. set and hence is not coarsely computable. Note that

$$R_e \subseteq A \triangle \overline{W_e}$$

since if $n \in R_e$ and $n \in A$, then $n \in (A \setminus \overline{W_e})$, while if $n \in R_e$ and $n \notin A$, then $n \in (\overline{W_e} \setminus A)$. So, for all e, $(A \triangle \overline{W_e})$ has positive lower density, and hence A is not coarsely similar to $\overline{W_e}$. It follows that A is not coarsely computable. Of course, this construction is simply a diagonal argument, but instead of using a single witness for each requirement, we use a set of witnesses of positive density.

Suppose now for a contradiction that A were generically computable. Let W_a, W_b be c.e. sets such that $W_a \subseteq A$, $W_b \subseteq \overline{A}$, and $W_a \cup W_b$ has density 1. Then A would be generically similar to $\overline{W_b}$ since

$$A \triangle \overline{W_b} \subseteq \overline{W_a \cup W_b}$$

and $\overline{W_a \cup W_b}$ has density 0. This shows that A is not generically computable. □

We introduce the following construction which will be used repeatedly.

Definition 3.6 ([19]). Let $\mathcal{I}_0 = \{0\}$ and for $n > 0$ let I_n be the interval $[n!, (n+1)!)$. For $A \subseteq \omega$, let $\mathcal{I}(A) = \cup_{n \in A} I_n$.

Theorem 3.7 *([19], proof of Theorem 2.20). For all A, the set $\mathcal{I}(A)$ is coarsely computable if and only if A is computable.*

Proof It is clear that $\mathcal{I}(A) \equiv_T A$, so it suffices to show that if A is not computable then $\mathcal{I}(A)$ is not coarsely computable. If $\mathcal{I}(A)$ is coarsely computable, we can choose a computable set C such that $\rho(C \triangle \mathcal{I}(A)) = 0$. The idea is now that we can show that A is computable by using "majority vote" to read off from C a set D which differs only finitely from A. Specifically, define

$$D = \{n : |I_n \cap C| > (1/2)|I_n|\}.$$

Then D is a computable set and we claim that $A \triangle D$ is finite. To prove the claim, assume for a contradiction that $A \triangle D$ is infinite. If $n \in A \triangle D$, then more than half of the elements of I_n are in $C \triangle \mathcal{I}(A)$. It follows that, for $n \in A \triangle D$,

$$\rho_{(n+1)!}(C \triangle \mathcal{I}(A)) \geq \frac{1}{2}\frac{|I_n|}{(n+1)!} = \frac{1}{2}\frac{(n+1)! - n!}{(n+1)!} = \frac{1}{2}(1 - \frac{1}{n+1}).$$

As the above inequality holds for infinitely many n, it follows that $\overline{\rho}(C \triangle \mathcal{I}(A)) \geq 1/2$, in contradiction to our assumption that $\rho(C \triangle \mathcal{I}(A)) = 0$. It follows that $A \triangle D$ is finite and hence A is computable. □

A similar argument shows that if A is not computable then $\mathcal{I}(A)$ is also not generically computable. We thus have the following result.

Theorem 3.8 *([19], Theorem 2.20). Every nonzero Turing degree contains a set which is neither coarsely computable nor generically computable.*

Since $\mathcal{R}(A)$ is generically computable if and only if A is computable, it seems natural to ask about the coarse computability of $\mathcal{R}(A)$. Post's Theorem shows that the sets Turing reducible to $0'$ are precisely the sets which are Δ^0_2 in the arithmetical hierarchy. Using the limit lemma one can prove the following result.

Theorem 3.9 *([19], Theorem 2.19). For all A, the set $\mathcal{R}(A) = \bigcup_{n \in A} R_n$ is coarsely computable if and only if $A \leq_T 0'$.*

In particular, if A is any noncomputable set Turing reducible to $0'$ then $\mathcal{R}(A)$ is coarsely computable but not generically computable.

4 Computability at Densities Less Than 1

Generic and coarse computability are computabilites at density 1. Downey et al. [8] took the natural step of considering computability at densities less than 1.

Definition 4.1 ([8], Definition 5.9). If $r \in [0,1]$, a set A is *partially computable at density* r if there exists a partial computable function φ agreeing with $A(n)$ whenever $\varphi(n)\downarrow$ and with the lower density of domain(φ) greater than or equal to r.

A natural first question is: Are there sets which are computable at all densities $r < 1$ but are not generically computable? Actually, we have already seen that every nonzero Turing degree contains such sets. Any set of the form $\mathcal{R}(A)$ is partially computable at all densities less than 1, as Asher Kach observed. Note that for any $t \geq 0$, the set $\bigcup R_k$ where $k \leq t$ and $k \in A$ is a computable set whose symmetric difference with $\mathcal{R}(A)$ is contained in $\bigcup\{R_k : k > t\}$, and the latter set has density 2^{-t-1}. Furthermore, $\mathcal{R}(A)$ is generically computable if and only if A is computable.

This "approachability" phenomenon holds very generally.

Definition 4.2 ([8], Definition 6.9). If $A \subseteq \omega$, the *partial computability bound* of A is
$\alpha(A) := sup\{r : A \text{ is computable at density } r\}$.

Theorem 4.3 *([8], Theorem 6.10). If $r \in [0,1]$, then there is a set A of density r with $\alpha(A) = r$.*

Proof. Let $.b_0b_1...$ be the binary expansion of r. By Corollary 1.8 the set $D = \bigcup_{b_i=1} R_i$ has density r. We let $A = D \cup S$ where S is a simple set of density 0 (Proposition 3.4). If $s < r$ we can take enough digits of the expansion of r so that if $t = .b_1 \dots b_n$ then $s < t < r$. The set C which is the union of the R_j where $j \leq n, b_j = 1$ is a computable subset of A of density t so A is computable at density t. Since we can take t arbitrarily close to r, it follows that $\alpha(A) \geq r$. To show that $\alpha(A) \leq r$, assume that φ is a computable partial function which agrees with A on its domain W. We must show that $\underline{\rho}(W) \leq r$. For $i \in \{0,1\}$, let $T_i = \{n : \varphi(n) = i\}$, so $W = T_0 \cup T_1$. Then T_0 is c.e. and $T_0 \subseteq \overline{A} \subseteq \overline{S}$, so T_0 is finite because S is simple. Also $T_1 \subseteq A$, so $\underline{\rho}(T_1) \leq \underline{\rho}(A) = r$, so $\underline{\rho}(W) \leq r$, as needed to complete the proof. □

In analogy with partial computability at densities less than 1, Hirschfeldt et al. [15] introduced the analogous concepts for coarse computability. We define

$$A \triangledown C = \{n : A(n) = C(n)\}$$

and call $A \triangledown C$ the *symmetric agreement* of A and C. Of course, the symmetric agreement of A and C is the complement of the symmetric difference of A and C.

Definition 4.4 ([15], Definition 1.5). A set A is *coarsely computable at density* r if there is a computable set C such that the lower density of the symmetric agreement of A and C is at least r, that is

$$\underline{\rho}(A \nabla C) \geq r$$

Definition 4.5 ([15], Definition 1.6). If $A \subseteq \mathbb{N}$, the *coarse computability bound* of A is

$$\gamma(A) := \sup\{r : A \text{ is coarsely computable at density } r\}$$

Proposition 4.6 *([15], Lemma 1.7). For every set A, $\alpha(A) \leq \gamma(A)$.*

This result follows easily from Theorem 2.2.

The next result is due to Greg Igusa and shows that this is the *only* restriction on the values taken simultaneously by α and γ.

Theorem 4.7 *(Igusa, personal communication). If r and s are real numbers with $0 \leq r \leq s \leq 1$, there is a set A such that $\alpha(A) = r$ and $\gamma(A) = s$.*

The coarse computability bound of every 1-random set A is $1/2$. This is because for every computable set C, the set $A \nabla C$ is also 1-random and so has density $1/2$.

Recall that we defined the distance function $D(A, B) = \overline{\rho}(A \bigtriangleup B)$. It is easily seen that D satisfies the triangle inequality and hence is a pseudometric on Cantor space 2^ω. Since $D(A, B) = 0$ exactly when A and B are coarsely similar, D is actually a metric on the space $\mathcal{S}$ of coarse similarity classes.

Note that A is coarsely computable at density 1 if and only if A is coarsely computable. To exhibit many sets with $\gamma = 1$ which are not coarsely computable, again consider sets of the form $\mathcal{R}(A) = \bigcup_{n \in A} R_n$. Essentially the same argument as before shows that $\gamma(\mathcal{R}(A)) = 1$ for every A. For each k, use the finite list of which $i \leq k$ are in A, to answer correctly on $\bigcup_{i=0}^{k} R_i$ and answer "yes" on all R_l with $l > k$. This algorithm is correct with density at least $1 - \frac{1}{2^{k+1}}$.

Lemma 4.8 *([15]). For $A \subseteq \omega, \underline{\rho}(A) = 1 - \overline{\rho}(\overline{A})$*

For each $n, \rho_n(A) = 1 - \rho_n(\overline{A})$, so the lemma follows by taking the least upper bound of both sides. As a corollary we have

$$\underline{\rho}(A \nabla C) = 1 - D(A, C).$$

So, $\gamma(A) = 1$ if and only if A is a limit of computable sets in the pseudo-metric D. In general, $\gamma(A) = r$ means that the distance from A to the family $\mathcal{C}$ of computable sets is $1 - r$.

Theorem 4.9 *([15], Theorems 3.1 and 3.4). For every $r \in (0, 1]$ there is a set A with $\gamma(A) = r$ such that A is* not *coarsely computable at density r, and a set B such that $\gamma(B) = r$ and B is coarsely computable at density r.*

We have seen that if A is not Δ_2^0 then $\mathcal{R}(A)$ is Turing equivalent to A, and $\gamma(\mathcal{R}(A)) = 1$, but $\mathcal{R}(A)$ is not coarsely computable. Also, every non-zero c.e. degree contains a c.e. set A which is generically computable but not coarsely computable ([8], Theorem 4.5). So the question is whether or not *every* nonzero Turing degree contains a set A such that $\gamma(A) = 1$ but A is not coarsely computable. The following result gives a negative answer. The proof includes a crucial lemma due to Joe Miller.

Theorem 4.10 *([15], Theorem 5.12). If A is computable from a Δ_2^0 1-generic set and $\gamma(A) = 1$, then A is coarsely computable.*

Theorem 4.11 *([15], Theorem 2.1). Every nonzero (c.e.) degree contains a (c.e.) set B such that $\alpha(B) = 0$ and $\gamma(B) = \frac{1}{2}$.*

Proof. Given A, let $B = \mathcal{I}(A)$. The majority vote argument about $\mathcal{I}(A)$ in the proof of Theorem 3.7 actually shows that if A is not computable then $\gamma(\mathcal{I}(A)) \leq \frac{1}{2}$. If E is the set of even numbers, then $E \nabla \mathcal{I}(A)$ has density $1/2$, so $\gamma(\mathcal{I}(A)) \geq \frac{1}{2}$. Also, it is easily seen $\alpha(\mathcal{I}(A)) = 0$ if A is noncomputable. □

We observe that large classes of degrees contain sets A with $\gamma(A) = 0$.

A set $S \subseteq 2^{<\omega}$ of finite binary strings is *dense* if every string has some extension in S. Kurtz [22] defined a set A to be *weakly* 1*-generic* if A meets every dense c.e. set S of finite binary strings.

Theorem 4.12 *([15], proof of Theorem 2.1). If A is a weakly 1-generic set, then $\gamma(A) = 0$.*

Proof. If f is a computable function then, for each $n, j > 0$, define

$$S_{n,j} = \left\{\sigma \in 2^{<\omega} : |\sigma| \geq j \ \&\ \rho_{|\sigma|}(\{k < |\sigma| : \sigma(k) = f(k)\}) < \frac{1}{n}\right\}.$$

Each set $S_{n,j}$ is computable and dense. A meets each $S_{n,j}$ since A is weakly 1-generic. Thus $\{k : f(k) = A(k)\}$ has lower density 0. □

Let D_n be the finite set with canonical index n, so $n = \sum\{2^i : i \in D\}$.

Recall that a set A is *hyperimmune* if A is infinite and there is no computable function f such that the sets $D_{f(0)}, D_{f(1)}, \ldots$ are pairwise disjoint and all intersect A, where D_n is the finite set with canonical index n. A degree $\mathbf{a}$ is called *hyperimmune* if it contains a hyperimmune set and otherwise *hyperimmune-free.* Kurtz [22] proved that the weakly 1-generic degrees coincide with the hyperimmune degrees. We thus have the following corollary.

Corollary 4.13 *([15], Theorem 2.2). Every hyperimmune degree contains a set A with $\gamma(A) = 0$.*

A degree $\mathbf{a}$ is called *PA* if every infinite computable tree of binary strings has an infinite $\mathbf{a}$-computable path.

Proposition 4.14 *([1], Proposition 1.8). If* **a** *is PA, then* **a** *contains a set* A *with* $\gamma(A) = 0$.

Proof. It is straightforward to construct an infinite computable tree T of binary strings such that the paths through T are exactly the sets X which, on every interval I_n, disagree with the partial computable function φ_n on all arguments where the latter is defined. Then an easy argument shows that $\gamma(X) = 0$ for every path X through T, and T has an **a**-computable path since **a** is PA. □

It is easily seen that $\alpha(\mathcal{I}(A)) = 0$ whenever A is noncomputable, and hence every nonzero degree contains a set B such that $\alpha(B) = 0$. In view of the preceding results on hyperimmune and PA degrees it is natural to ask whether *every* nonzero degree contains a set B such that $\gamma(B) = 0$.

This question is investigated and answered in the negative in Andrews et al. [1], where the following definition was introduced.

Definition 4.15 ([1]). If **d** is a Turing degree,

$$\Gamma(\mathbf{d}) = \inf\{\gamma(A) : A \leq_T \mathbf{d}\}$$

Recall that the majority vote argument shows that if A is any noncomputable set then $\gamma(\mathcal{I}(A)) \leq 1/2$. Therefore if a Turing degree has a Γ-value greater than $1/2$ then it is computable and so has Γ-value 1.

We call a function g a *trace* of a function f if $f(n) \in D_{g(n)}$ for every n.

Definition 4.16 (Terwijn and Zambella [34]). A set A is *computably traceable* if there is a computable function p with the property that every A-computable function f has a computable trace g such that $(\forall n)[|D_{g(n)}| \leq p(n)]$. (Note that p is independent of f.)

Theorem 4.17 *([1], Theorem 1.10). If* A *is computably traceable, then* A *is coarsely computable at density* $\frac{1}{2}$.

The proof is a probabilistic argument. Since the computably traceable sets are closed downwards under Turing reducibility, it follows easily that $\Gamma(\mathbf{a}) = \frac{1}{2}$ for every degree $\mathbf{a} > \mathbf{0}$ which contains a computably traceable set.

Theorem 4.18 *([1], Theorem 1.12). If* A *is a* 1*-random set of hyperimmune-free Turing degree and* $B \leq_T A$*, then* B *is coarsely computable at density* $\frac{1}{2}$.

In summary, we know the following.

- $\Gamma(\mathbf{0}) = 1$.
- If $\mathbf{a} > \mathbf{0}$, then $\Gamma(\mathbf{a}) \leq \frac{1}{2}$.
- If **a** is hyperimmune or PA, then $\Gamma(\mathbf{a}) = 0$.
- If **a** is computably traceable and nonzero, then $\Gamma(\mathbf{a}) = \frac{1}{2}$.
- If **a** is both 1-random and hyperimmune-free, then $\Gamma(\mathbf{a}) = \frac{1}{2}$.

The following question was raised in [1].

Question 4.19. What is the range of Γ? Does it equal $\{0, \frac{1}{2}, 1\}$?

Monin [25] has recently announced the remarkable result that $\Gamma(\mathbf{d})$ is equal to 0, 1/2 or 1 for every degree $\mathbf{d}$. Together with the results just above, this gives a positive answer to the second half of the above question, and thus a natural trichotomy of the Turing degrees according to their Γ-values. In contrast, Matthew Harrison-Trainor [13] has just announced that the range of the analogue for Γ for many-one degrees is $[0, 1/2] \cup \{1\}$.

Monin and Nies [26] have also recently extended and unified some of the above results on Γ using Schnorr randomness. In particular they showed the existence of degrees $\mathbf{a}$ with $\Gamma(\mathbf{a}) = \frac{1}{2}$ which are neither computably traceable nor 1-random. They also gave a new proof of Liang Yu's unpublished result that there are degree $\mathbf{a}$ with $\Gamma(\mathbf{a}) = 0$ such that $\mathbf{a}$ is neither hyperimmune nor PA.

5 Generic and Coarse Reducibility and Their Corresponding Degrees

One might first consider relative generic computability: That is, what sets are generically computable by Turing machines with a full oracle for a set A? Say that a set B is *generically A-computable* if there is a generic computation of B using a *full* oracle for A. It is easy to see that this notion is not transitive because we start with full information but compute only partial information. For example, let $A = \emptyset$ and let $B = \{2^n : n \in C\}$ where C is any set which is not generically computable. Then B is generically A-computable and C is generically B-computable, but C is not generically A-computable. The following is a remarkable and surprising result of Igusa [16] showing there are no minimal pairs for this non-transitive notion of relative generic computability.

Theorem 5.1 *([16], Theorem 2.1). For any noncomputable sets A and B there is a set C which is not generically computable but which is both generically A-computable and generically B-computable.*

Generic reducibility (denoted $\leq_g$) was introduced by Jockusch and Schupp [19] (Sect. 4), and we review the definition here. A *generic description* of a set A is a partial function θ which agrees with A on its domain and has a domain of density 1. Note that A is generically computable if and only if A has a partial computable generic description. The basic idea is then that $B \leq_g A$ if and only if there is an effective procedure which, from any generic description of A, computes a generic description of B. Since computing a partial function is tantamount to enumerating its graph, this is made precise using enumeration operators. These are similar to Turing reductions but use only *positive* oracle information and also output only positive information. An *enumeration operator* is a c.e. set W of pairs $\langle n, D\rangle$ where $n \in \omega$ and D is a finite subset of ω. (Here we identify finite sets with their canonical indices and pairs with their codes in saying that W is c.e. The membership of $\langle n, D\rangle$ in W means intuitively that from the positive information that D is a subset of the oracle, W computes that n belongs to the output.) Hence if W is an enumeration operator and $X \subseteq \omega$, define

$$W^X := \{n : (\exists D)[\langle n, D\rangle \in W \ \& \ D \subseteq X]\}$$

Note that from any enumeration of X one may effectively obtain an enumeration of W^X. If θ is a partial function, let $\gamma(\theta) = \{\langle a, b\rangle : \theta(a) = b\}$, so $\gamma(\theta)$ is a set of natural numbers coding the graph of θ. We can now state our formal definition of generic reducibility.

Definition 5.2. The set B is *generically reducible* to the set A (written $B \leq_g A$) if there is a fixed enumeration operator W such that, for every generic description θ of A, $W^{\gamma(\theta)} = \gamma(\delta)$ for some generic description δ of B.

This reducibility is also called "uniform generic reducibility" and denoted $\leq_{ug}$. (There is also a nonuniform version, $\leq_{ng}$, of generic reducibility which we do not consider in this survey.)

It is easily seen that $\leq_g$ is transitive since the maps induced by enumeration operators are closed under composition.

Definition 5.3. The *generic degree* of A is $\{B : B \leq_g A \ \& \ A \leq_g B\}$.

We have seen that the map $\widehat{\mathcal{R}}$ which sends the Turing degree of A to the generic degree of $\mathcal{R}(A)$ embeds the Turing degrees into the generic degrees, since any generic algorithm for $\mathcal{R}(A)$ will compute A, and the proof of this is uniform. The generic degrees have a least degree under the ordering induced by $\leq_g$, and this least degree consists of the generically computable sets.

Define B to be *enumeration reducible* to A (written $B \leq_e A$) if there is an enumeration operator W such that $W^A = B$.

Enumeration reducibility leads analogously to the *enumeration degrees*, i.e. equivalence classes under the equivalence relation $A \leq_e B$ and $B \leq_e A$. The Turing degrees can be embedded in the enumeration degrees by the map which takes the Turing degree of A to the enumeration degree of $A \oplus \overline{A}$. An enumeration degree $\mathbf{a}$ is called *quasi-minimal* if it is nonzero and no nonzero enumeration degree $\mathbf{b} \leq \mathbf{a}$ is in the range of this embedding. The following definition is analogous:

Definition 5.4 ([17]). A generic degree $\boldsymbol{a}$ is *quasi-minimal* if it is nonzero and no nonzero generic degree $\mathbf{b} \leq \mathbf{a}$ is in the range of the embedding $\widehat{\mathcal{R}}$ of the Turing degrees into the generic degrees defined above.

The following result gives a connection between quasi-minimality for enumeration degrees and generic degrees.

Lemma 5.5 *([19], Lemma 4.9). If A is a set of density 1 which is not generically computable and the enumeration degree of A is quasi-minimal, then the generic degree of A is also quasi-minimal.*

It is shown in the proof of Theorem 4.8 of [19] that there is a set A which meets the hypotheses of the lemma. It follows that there exist quasi-minimal generic degrees which contain sets of density 1.

It is therefore natural to consider generic degrees which are *density*-1, that is, generic degrees which contain a set of density 1 [17].

A *hyperarithmetical* set is a set computable from any set that can be obtained by iterating the jump operator through the computable ordinals. The class of such sets coincides with the class of Δ^1_1 sets, which are those sets which can be defined by a prenex formula of second-order arithmetic with all set quantifiers universal and also by a prenex formula with all set quantifiers existential. Igusa [17] proves the following striking characterization.

Theorem 5.6 *([17], Theorem 2.15). A set A is hyperarithmetical if and only if there is a density-1 set B such that $\mathcal{R}(A) \leq_g B$.*

Cholak and Igusa [5] consider the question of whether or not every non-zero generic degree bounds a non-zero density-1 generic degree. By the results of [17] a positive answer would show that there are no minimal generic degrees and a negative answer would show that there are minimal pairs in the generic degrees. However, it is not yet known whether or not there are minimal degrees or minimal pairs in the generic degrees.

Recall that a *coarse description* of a set A is a set C which agrees with A on a set of density 1. Hirschfeldt et al. [14] introduced both uniform and nonuniform versions of coarse reducibility and their corresponding degrees.

Definition 5.7 ([14], Definition 2.1). A set A is *uniformly coarsely reducible* to a set B, written $A \leq_{uc} B$, if there is a fixed oracle Turing machine M which, given any coarse description of B as an oracle, computes a coarse description of A. A set A is *nonuniformly coarsely reducible* to a set B, written $A \leq_{nc} B$ if every coarse description of B computes a coarse description of A.

These coarse reducibilities induce respective equivalence relations $\equiv_{uc}$ and $\equiv_{nc}$.

Definition 5.8 ([14]). The *uniform coarse degree* of A is $\{B : B \equiv_{uc} A\}$ and the *nonuniform coarse degree* of A is $\{B : B \equiv_{nc} A\}$.

We can embed the Turing degrees into both the nonuniform and the uniform coarse degrees. We have already seen that the function $\mathcal{I}$ induces an embedding of the Turing degrees into the nonuniform coarse degrees since $\mathcal{I}(A) \leq_T A$ and each coarse description of $\mathcal{I}(A)$ computes A, but the adjustments needed to compute A depend on the coarse description used.

To construct an embedding of the Turing degrees into the uniform coarse degrees we need more redundancy. The following map is slightly different from but equivalent to the map used in [14], Proposition 2.3.

Proposition 5.9 *([14]). Define $\mathcal{E}(A) = \mathcal{I}(\mathcal{R}(A))$. The function $\mathcal{E}$ induces an embedding of the Turing degrees into the uniform coarse degrees.*

Recall that a set X is *autoreducible* if there exists a Turing functional Φ such that for every $n \in \omega$ we have $\Phi^{X \setminus \{n\}}(n) = X(n)$. Equivalently, we could require that Φ not ask whether its input belongs to its oracle. Figueira et al. [11] showed that no 1-random set is autoreducible and it is not difficult to show that no 1-generic set is autoreducible.

Dzhafarov and Igusa [10] study various notions of "robust information coding" and introduced uniform "mod-finite", "co-finite" and "use-bounded from below" reducibilities. Using the relationships between these reducibilities and generic and coarse reducibility, Igusa proved the following result.

Theorem 5.10 *(see [14], Theorem 2.7). If $\mathcal{E}(X) \leq_{uc} \mathcal{I}(X)$ then X is autoreducible. Therefore if A is 1-random or 1-generic then $\mathcal{E}(X) \leq_{nc} \mathcal{I}(X)$ but $\mathcal{E}(X) \not\leq_{uc} \mathcal{I}(X)$.*

There are striking connections between coarse degrees and algorithmic randomness. The paper [14] shows the following.

Theorem 5.11 *([14], Corollary 3.3). If X is weakly 2-random then $\mathcal{E}(A) \not\leq_{nc} X$ for every noncomputable set A, so the degree of X is quasi-minimal (in the obvious sense) in both the uniform and nonuniform coarse degrees.*

For the uniform coarse degrees, this result was strengthened by independently motivated work by Cholak and Igusa [5].

Theorem 5.12 *([5]). If A is either 1-random or 1-generic, then the degree of A is quasiminimal in the uniform coarse degrees.*

Theorem 5.13 *([14], Corollary 5.3). If Y is not coarsely computable and X is weakly 3-random relative to Y, then their nonuniform coarse degrees form a minimal pair for both uniform and nonuniform coarse reducibility.*

Astor et al. [2] introduced "dense computability" as a weakening of both generic and coarse computability.

Definition 5.14 ([2]). A set A is *densely computable* (or *weakly partially computable*) if there is a partial computable function φ such that $\underline{\rho}(\{n : \varphi(n) = A(n)\}) = 1$.

In other words, the partial computable function may diverge on some arguments and give wrong answers on others but agrees with the characteristic function of A on a set of density 1. It is obvious that every generically computable set and every coarsely computable set is densely computable. Note that if A is generically computable but not coarsely computable and B is coarsely computable but not generically computable then $A \oplus B$ is neither generically computable nor coarsely computable, where, as usual, $A \oplus B = \{2n : n \in A\} \cup \{2n+1 : n \in B\}$. But $A \oplus B$ is densely computable by using the generic algorithm on even numbers and the coarse algorithm on odd numbers. Thus dense computability is strictly weaker than the disjunction of coarse computability and generic computability.

We can consider weak partial computability at densities less than 1.

Definition 5.15 ([2]). Let $r \in [0,1]$. A set A is *weakly partially computable* at density r if there exists a partial computable function such that $\underline{\rho}(\{n : \varphi(n) = A(n)\}) \geq r$. Let

$$\delta(A) = sup\{r : A \text{ is weakly partially computable at density } r\}.$$

It is easy to show the following.

Lemma 5.16 *([2]). For all $A, \delta(A) = \gamma(A)$.*

Proof. If A is weakly partially computable at density r by a partial computable function φ, then by Theorem 2.2 $\text{dom}(\varphi)$ has a computable subset C such that $\underline{\rho}(C) > \underline{\rho}(\text{dom}(\varphi)) - \epsilon$. Let h be the total computable function defined by $h(n) = \varphi(n)$ if $n \in C$ and $h(n) = 0$ otherwise. Since $A \cap C \subseteq \{n : A(n) = \varphi(n)\}$ it follows that A is coarsely computable at density $r - \epsilon$. So $\gamma(A) \geq \delta(A)$. Since $\delta(A) \geq \gamma(A)$ by definition, the two are equal. □

Definition 5.17. A partial function Θ is a *dense description of* A if $\{n : \Theta(n) = A(n)\}$ has density 1.

Using dense descriptions one can define dense reducibility and dense degrees as in [2].

References

1. Andrews, U., Cai, M., Diamondstone, D., Jockusch, C., Lempp, S.: Asymptotic density, computable traceability, and 1-randomness. Fundam. Math. **234**, 41–53 (2016)
2. Astor, E., Hirschfeldt, D., Jockusch, C.: Dense computability, upper cones and minimal pairs. In preparation
3. Barzdin, J.: On a frequency solution to the problem of occurrence in a recursively enumerable set. Proc. Steklov Inst. Math. **133**, 49–56 (1973)
4. Bienvenu, L., Day, A., Hölzl, R.: From bi-immunity to absolute undecidability. J. Symb. Log. **78**(4), 1218–1228 (2013)
5. Cholak, P., Igusa, G.: Bounding a density-1 and quasiminimality in the generic degrees. Preprint
6. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press and McGraw-Hill (2001). Section 29.3: The simplex algorithm, pp. 790–804
7. Downey, R.G., Hirschfeldt, D.R.: Algorithmic Randomness and Complexity. Theory and Applications of Computability. Springer, Springer, New York (2010)
8. Downey, R.G., Jockusch Jr., C.G., Schupp, P.E.: Asymptotic density and computably enumerable sets. J. Math. Log. **13** (2013). 1350005, 43 pp.
9. Downey, R.G., Jockusch Jr., C.G., McNicholl, T.H., Schupp, P.E.: Asymptotic density and the Ershov Hierarchy. Math. Log. Q. **61**, 189–195 (2015)
10. Dzhafarov, D.D., Igusa, G.: Notions of robust information coding, computability. To appear
11. Figueira, S., Miller, J.S., Nies, A.: Indifferent sets. J. Log. Comput. **19**, 425–443 (2009)

12. Gurevich, Y.: Average case completeness. J. Comput. Syst. Sci. **42**, 346–398 (1991)
13. Harrison-Trainor, M.: The Gamma questions for many-one degrees. Preprint. arXiv: 1606.05701
14. Hirschfeldt, D., Jockusch, C., Kuyper, R., Schupp, P.: Coarse reducibility and algorithmic randomness. J. Symb. Log. **81**, 1028–1046 (2016)
15. Hirschfeldt, D.R., Jockusch, C.G., McNicholl, T., Schupp, P.E.: Asymptotic density and the coarse computability bound. Computability **5**, 13–27 (2016)
16. Igusa, G.: Nonexistence of minimal pairs for generic computability. J. Symb. Log. **78**, 51–522 (2012)
17. Igusa, G.: The generic degrees of density-1 sets and a characterization of the hyperarithmetic reals. J. Symb. Log. **80**, 1290–1314 (2015)
18. Jockusch, C.: The degrees of bi-immune sets. Zeitschr. f. math. Logik und Grundlagen d. Math. **15**, 135–140 (1969)
19. Jockusch, C.G., Schupp, P.E.: Generic computability, Turing degrees, and asymptotic density. J. Lond. Math. Soc. **85**, 472–490 (2012)
20. Kapovich, I., Myasnikov, A., Schupp, P., Shpilrain, V.: Generic-case complexity, decision problems in group theory and random walks. J. Algebra **264**, 665–694 (2003)
21. Klee, V., Minty, G.: How good is the simplex algorithm? Inequalities, III (Proceedings of Third Symposium, University of California, Los Angeles, 1969; dedicated to the memory of Theodore S. Motzkin), pp. 159–175. Academic Press, New York (1972)
22. Kurtz, S.A.: Notions of weak genericity. J. Symb. Log. **48**, 764–770 (1983)
23. Levin, L.: Average case complete problems. SIAM J. Comput. **15**, 285–286 (1986)
24. Lyndon, R.C., Schupp, P.E.: Combinatorial Group Theory. Classics in Mathematics. Springer, Heidelberg (2000)
25. Monin, B.: Asymptotic density, error-correcting codes. https://www.lacl.fr/benoit.monin/ressources/papers/resolution_gamma.pdf
26. Monin, B., Nies, A.: A unifying approach to the Gamma question. In: 30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, pp. 585–596, 6–10 July 2015
27. Myasnikov, A., Osin, D.: Algorithmically finite groups. J. Pure Appl. Algebra **215**, 2789–2796 (2011)
28. Myasnikov, A., Rybalov, A.: Generic complexity of undecidable problems. J. Symb. Log. **73**, 656–673 (2008)
29. Myasnikov, A., Shpilrain, V., Ushakov, A.: Non-commutative Cryptography and Complexity of Group-Theoretic Problems. Mathematical Surveys and Monographs, vol. 177. American Mathematical Society (2011)
30. Nies, A.: Computability and Randomness. Oxford University Press, Oxford (2009)
31. Novikov, P.S.: On the algorithmic unsolvability of the word problem in group theory. Trudy Math. Inst. Steklov **44**, 3–143 (1955)
32. Post, E.: A variant of a recursively unsolvable problem. Bull. Am. Math. Soc. **52**, 264–268 (1946)
33. Rotman, J.: An Introduction to the Theory of Groups. Graduate Texts in Mathematics, 4th edn. Springer, New York (1995)
34. Terwijn, S.A., Zambella, D.: Algorithmic randomness and lowness. J. Symb. Log. **66**, 1199–1205 (2001)
35. Woess, W.: Cogrowth of groups and simple random walks. Arch. Math. **41**, 363–370 (1983)