Imperfect Computability and asymptotic density

Carl Jockusch

University of Illinois

at Urbana-Champaign

http://www.math.uiuc.edu/∼jockusch/

Dedicated to Rod Downey

January 7, 2017

Introduction

Paul Schupp and I recently wrote an article "Asymptotic Density and the Theory of Computability: A partial survey" which surveys the work on how classical asymptotic density interacts with the theory of computability. It will appear in the Rodfest volume. This talk is largely based on that article.

I thank Paul for bringing me into this area, his help with this talk, and his invaluable collaboration over the years.

Many other people have contributed to this area. In particular, Rod Downey joined our collaboration, and his deep insights have brought our understanding to a new level.

Worst case bounds

Complexity classes such as $\mathcal{P}$ are defined by *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 an 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 the real-life complexity of a problem since hard instances may be very sparse. The paradigm case is Dantzig's Simplex Algorithm for linear programming. This algorithm runs thousands of times every day for purposes ranging from optimizing refinery operations to finding matches on dating sites. It almost always runs very quickly.

Klee and Minty have shown that on certain carefully chosen instances the Simplex Algorithm requires exponential time.

The explanation of the discrepancy between theory and practice is that hard instances are very sparse.

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. Paul Schupp discussed these in his talk here.

Such examples led Kapovich, Myasnikov, Schupp and Shpilrain to introduce generic-case complexity as a complexity measure which is often more useful, realistic, and easier to work with than worst-case complexity. In generic-case complexity, one considers partial 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.

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^*.

Definition

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

$$\rho_n(S) = \frac{|\{w \in S, |w| \leq n\}|}{|w \in \Sigma^*, |w| \leq n\}|}$$

The *density* of S is

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

if this limit exists.

Definition

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

Definition

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 undecidable problems may be generically easy, 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 nonempty finite 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 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.

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.

There is a natural stratification of instances of the problem on which the algorithm gives an answer with density 1. 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.

It is easy to show that the probability that the algorithm diverges on a random element of I_s approaches 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 consider subsets of the the set $\mathbb{N} = \{0, 1, \dots\}$ of natural numbers. In this context, we are using classical asymptotic density, which has been extensively studied in number theory and other areas.

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.

Definition

- A partial function φ is a *generic description* of a set $A \subseteq \omega$ if $\varphi(n) = A(n)$ whenever $\varphi(n)\downarrow$ and the domain of φ has density 1.
- A set $A \subseteq \omega$ is *generically computable* if A has a partial computable generic description.

Note that A is generically computable iff there exist c.e. sets Y, N such that $Y \subseteq A$, $N \subseteq \overline{A}$, and $Y \cup N$ has density 1.

First observe that *every* Turing degree contains a generically computable set and indeed there is a fixed algorithm which witnesses this. 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.

The following sets R_k are frequently useful.

Definition

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

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 $\bigcup_{k=0}^{\infty} R_k = \omega - \{0\}$.

Definition

If $A \subseteq \omega$ then $\mathcal{R}(A) = \bigcup_{n \in A} R_n$.

Proposition

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. $\square$

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.

Densities and C.E. Sets

Theorem (J., S.)

There is a c.e. set A of density 1 which does not have any computable subset of density 1.

To ensure that A has density 1, we meet the following positive requirements:

$$P_k : R_k \subseteq^* A$$

These suffice because of "restricted countable additivity" for ρ.

To ensure that A has no computable subset of density 1 we meet the following negative requirements:

$$N_k : W_k \cup A = \omega \implies \underline{\rho}(W_k \cap R_k) > 0$$

In fact, N_k asserts that $\overline{W_k}$ is not a subset of A of density 1.

What sort of c.e. set is needed as an oracle to construct a c.e. set A of density 1 which has no computable subset of density 1?

A degree $\mathbf{a}$ is *high* if $\mathbf{a}' \geq \mathbf{0}''$ and *low* if $\mathbf{a}' = \mathbf{0}'$.

The positive requirement $R_k \subseteq^* A$ is a typical "thickness" requirement since it requires all but finitely many elements of a given infinite computable set to be put into A. This suggests that a high c.e. oracle is needed.

Instead a new phenomenon arises.

Non-low permitting

Theorem (Downey, J., S.)

Let **a** *be a c.e. degree. Then* **a** *is not low if and only if* **a** *contains a c.e. set A of density 1 with no computable subset of density 1.*

The "only if" direction uses "non-low" permitting, which has turned out to be useful in other contexts. Rod is a master of permitting arguments!

Theorem (Downey, J., S.)

Let **a** *be a nonlow c.e. degree. Then there is a c.e. set A of degree* **a** *such that A has no computable subset of positive density.*

Some positive results

Theorem (Downey, J., S.)

Every c. e. set of upper density 1 *has a computable subset of upper density* 1.

Theorem (Downey, J., S.)

If A *is a c.e. set and* $\epsilon > 0$, *there is a computable set* $C \subseteq A$ *such that*

$$\underline{\rho}(C) > \underline{\rho}(A) - \epsilon$$

Densities and the Arithmetic Hierarchy

There is a tight connection between the positions of sets in the arithmetical hierarchy and the complexity of their densities as real numbers.

Definition

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

Theorem (Downey, J., S.)

Let r be a real number in the unit interval.

- *r is the density of some computable set iff r is a Δ^0_2 real*
- *r is the lower density of some computable set iff r is left-Σ^0_2*
- *r is the density of some c.e. set iff r is left-Π^0_2*
- *r is the upper density of some c.e. set iff r is left-Π^0_2*
- *r is the lower density of some c.e. set iff r is left-Σ^0_3*

This result easily extends via relativization and dualization to characterize the densities, upper densities, and lower densities of sets at all levels of the arithmetic hierarchy.

Asymptotic density and 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 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.

Theorem (Downey, J., McNicholl, S.)

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 which is not the density of any c.e. or co-c.e. set.

Bi-immunity and Absolute Undecidability

I proved that there are nonzero Turing degrees which do not contain any bi-immune sets. This result raises the natural question of how strong a condition on c.e. subsets of A and $\overline{A}$ can be pushed into every non-zero degree. It turns out that one answer involves density.

Miasnikov and Rybalov 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. In other words, a set A is absolutely undecidable if and only if every c. e. subset of A or $\overline{A}$ has density 0. Obviously, every bi-immune set is absolutely undecidable.

The following beautiful and surprising result is due to Bienvenu, Day and Hölzl.

Theorem

Every nonzero Turing degree contains an absolutely undecidable set.

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

Coarse Computability

There is a reasonable concept of imperfect computability which allows incorrect answers.

Definition (J., S.)

Two sets A and B are *coarsely similar*, denoted 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.

Definition (J., S.)

A set A is *coarsely computable* if A is coarsely similar to some 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 set of density 0.

A generic algorithm is always correct when it answers and almost always answers, while a coarse algorithm always answers and is almost always correct.

Proposition (J., S.)

There is a c.e. set which is coarsely computable and absolutely undecidable, and hence not generically computable.

Proof.

It suffices to construct a simple set A of density 0, since any such set is coarsely computable and absolutely undecidable. 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. □

We introduce another useful family of sets.

Definition

Let $I_0 = \{0\}$ 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 (J., S.)

For all A, the set $\mathcal{I}(A)$ is coarsely computable if and only if A is computable.

Proof.

It is clear that if A is computable, then $\mathcal{I}(A)$ is computable. Assume that $\mathcal{I}(A)$ is coarsely computable, and let C be a computable coarse description of $\mathcal{I}(A)$. 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 $A \triangle D$ is finite. □

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 (J., S.)

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 $\mathcal{R}(A)$.

Theorem (J., S.)

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

The proof is an easy application of the limit lemma.

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

Computability at densities less than 1

Generic and coarse computability are computabilites at density 1. But we can just as well consider computability at lower densities.

Definition (D.,J.,S.)

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.

We use lower density rather than upper density because we wish our algorithms to work well on $[0, n]$ for cofinitely many n rather than just infinitely many n.

A natural first question is: Are there sets which are partially 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. Fix $t \geq 0$, and consider the algorithm which, on input n, outputs $A(k)$ if $n \in \mathcal{R}_k$ for some $k \leq t$ and otherwise gives no output. This algorithm gives no incorrect answers, and the density of its domain is $1 - 2^{-(t+1)}$.

Furthermore, $\mathcal{R}(A)$ is generically computable if and only if A is computable.

Definition (D., J., S.)

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

Theorem (D., J., S.)

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

The Coarse Computability Bound

In analogy with partial computability at densities less than 1, Hirschfeldt, Jockusch, McNicholl and Schupp 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 (Hirschfeldt, J., McNicholl, S.)

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

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

Definition

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

For every set A, $\alpha(A) \leq \gamma(A)$.

This is the *only* restriction on the values taken simultaneously by α and γ.

Theorem (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 $A \triangledown C$ is also 1-random and so has density $1/2$.

Let $D(A, B) = \overline{\rho}(A \triangle B)$. It is easily seen that D satisfies the triangle inequality and hence is a pseudometric on $2^{\mathbb{N}}$. Since $D(A, B) = 0$ exactly when A and B are coarsely similar, D is a metric on the space $\mathcal{S}$ of coarse equivalence classes. It is known as the Besicovitch metric.

So we are now working over an actual metric space.

It is easy to see that, for all A,

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

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 $1 - r$.

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 for α shows that $\gamma(\mathcal{R}(A)) = 1$ for every A.

Does every nonzero degree contain a set A such that $\gamma(A) = 1$ and A is not coarsely computable? Every degree not below $\mathbf{0}'$ contains such a set, namely $\mathcal{R}(A)$ for any A of degree $\mathbf{a}$. Also, Downey, J, and S showed that every nonzero c.e. degree contains such a set, namely a c.e. set which is generically computable but not coarsely computable. The following result gives an example of a nonzero degree which contains no such set, The proof uses a crucial lemma due to Joe Miller.

Theorem (H., J., McNicholl, S.)

If A is computable from a Δ^0_2 1-generic set and $\gamma(A) = 1$, then A is coarsely computable.

This theorem shows that a limit of computable sets under the hypothesis is actually coarsely computable.

Theorem (H., J., McNicholl, S.)

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 $\mathcal{I}(A)$ given before 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. Stuart Kurtz defined a set A to be *weakly 1-generic* if A meets every dense c.e. set S of finite binary strings.

Theorem (H.,J.,McNicholl,S.)

If A is a weakly 1-generic set, then $\gamma(A) = 0$.

Corollary

Every hyperimmune degree contains a set A with $\gamma(A) = 0$.

The corollary follows since Kurtz proved that every hyperimmune degree contains a weakly 1-generic set.

Theorem (Andrews, Cai, Diamondstone, J, and Lempp)

Every PA degree contains a set A such that $\gamma(A) = 0$.

It follows that there are hyperimmune-free degrees which contain a set A with $\gamma(A) = 0$.

The Γ Question

It is natural to ask whether *every* nonzero degree contains a set A such that $\gamma(A) = 0$.

This question is investigated and answered in the negative in Andrews, Cai, Diamondstone, Jockusch and Lempp, where the following definition was introduced.

Definition (ACDJL)

If $\mathbf{d}$ 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

(Terwijn, Zambella) 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.)

The next two theorems are due to Andrews, Cai, Diamondstone, Jockusch and Lempp.

Theorem

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

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

This raises the natural question:

Question

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

Benoit Monin proved the remarkable result that $\Gamma(\mathbf{d})$ is actually 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.

Relative generic computability

One might 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.

Igusa proved the remarkable and surprising result that there are no minimal pairs for the non-transitive notion of relative generic computability.

Theorem (Igusa)

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

Jockusch and Schupp introduced *uniform generic reducibility* (denoted $\leq_g$). Recall that 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, given any generic description of A, outputs a generic description of B.

Since computing a partial function amounts to enumerating its graph, this is made precise using enumeration operators. Define uniform generic degrees in the obvious way.

For example, for any A, $\mathcal{R}(A)$ and $\mathcal{I}(A)$ have the same uniform generic degree.

The Turing degrees are embeddable in the uniform generic degrees by the map which sends the Turing degree of A to the uniform generic degree of $\mathcal{R}(A)$. This embedding is not a surjection.

It is not known if there exist minimal pairs, or minimal degrees, in the uniform generic degrees.

A *hyperarithmetical* set is a set computable from any set that can be obtained by iterating the jump operator through the computable ordinals.

Igusa proved the following striking characterization.

Theorem

A set A is hyperarithmetical if and only if there is a set B of density 1 such that $\mathcal{R}(A) \leq_g B$.

Coarse reducibility and coarse degrees

Recall that a *coarse description* of a set A is a set C which agrees with A on a set of density 1.

Definition (H., J. Kuyper, S.)

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

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 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.

Proposition

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 require that Φ not ask whether its input belongs to its oracle. Figueira, Miller and Nies 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 studied 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

If $\mathcal{E}(X) \leq_{uc} \mathcal{I}(X)$ then X is autoreducible. Therefore if X is 1-random or 1-generic then $\mathcal{E}(X) \leq_{nc} \mathcal{I}(X)$ but $\mathcal{E}(X) \not\leq_{uc} \mathcal{I}(X)$.

Coarse degrees and algorithmic randomness

The coarse degrees of sufficiently random sets are not in the range of the embedding induced by $\mathcal{E}$ and in fact are quasiminimal in the following sense.

A uniform coarse degree $\mathbf{x}$ is *quasiminimal* if $\mathbf{x} > \mathbf{0}$ and $\mathbf{0}$ is the only uniform coarse degree less than or equal to $\mathbf{x}$ containing a set of the form $\mathcal{E}(A)$. The same definition applies to nonuniform coarse degrees.

Theorem (H., J.,Kuyper, S.)

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 quasiminimal in both the uniform and nonuniform coarse degrees.

Theorem (Cholak, Igusa)

If X is either 1-random or 1-generic, then the degree of X is quasiminimal in the uniform coarse degrees.

A minimal pair in the coarse degrees

Theorem (H., J., Kuyper, S.)

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.

The same authors showed that this result fails if weak 3-randomness is replaced by 2-randomness, so a high level of randomness is needed!

Effective Hausdorff dimension and 1-randomness

The *effective Hausdorff dimension* of a set A is $\liminf_n \frac{K(A\upharpoonright n)}{n}$, where K is prefix-free Kolmogorov complexity.

The following remarkable result is due to Noam Greenberg, Joe Miller, Alexander Shen, and Linda Brown Westrick.

Theorem (MGSW)

Let $A \subseteq \omega$. Then A has effective Hausdorff dimension 1 if and only if A is coarsely similar to some 1-random set.

Thus, the sets of effective Hausdorff dimension 1 are precisely the sets which can be obtained by perturbing a 1-random set on a set of density 0. The same authors obtain corresponding results for sets of effective Hausdorff dimension less than 1.