(Asymptotic Properties of Computability)

Paul Schupp

University of Illinois

at Urbana-Champaign

Joint work with Carl Jockusch
and with Rod Downey and Denis Hirschfeldt
and with Rutger Kuyper I and Tim McNicholl

Steklov Institute
May, 2015

The Asymptotic-Generic Point of View

The “asymptotic-generic” point of view is now central in geometric group theory. The purpose of the talk is to show that this point of view interacts with classical computability theory in a very deep way.

Complexity theory

Since the 1970's, motivated by the availability of actual computers, there has been much work on computational complexity. Given that a problem is computable, how long does it take to compute it? In complexity theory, the model of computation still used is that of the Turing machine, exactly because it is so simple.

There is a growing awareness that *worst-case* measures, such as $\mathcal{P}, \mathcal{NP}$ or just being computable often do not give a good overall picture of a particular algorithm or problem.

The Simplex Algorithm

In 1947, Dantzig developed his Simplex Algorithm for linear programming problems. This algorithm now runs many hundreds of times every day for scheduling and transportation problems, almost always very quickly.

There are clever examples of Klee and Minty showing that there exist problems for which the Simplex Algorithm must take exponential time. But one never encounters such examples in practice.

The behavior of the simplex algorithm is the world's longest running computer experiment. In computer experiments what one sees is "Generic behavior".

Density and Generic Sets

Fix a finite alphabet Σ and let Σ^* denote the set of all words over Σ. If $w \in \Sigma$, the *length* $|w|$ of w is the number of letters in w. Since we have a length function we can copy the classical definition of asymptotic density from number theory. For subsets of $\mathbb{N}$ this is classical asymptotic density.

If $A \subseteq \Sigma^*$, then, for $n \geq 1$, the density of A at n is

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

If $\rho(A) = \lim_{n\to\infty} \rho_n(A)$ exists then $\rho(A)$ is the *asymptotic density* of A.

Definition

A set A is generic if $\rho(A) = 1$. A is strongly generic if $\rho_n(A) \to 1$ exponentially fast.

Most sets do not have an asyptotic density but the *upper density*

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

and the *lower density*

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

always exist.

We need the fact that lower and upper densities are always defined.

Generic computability

The original motivation for considering generic computability was as a better complexity measure than either worst-case or average-case complexity. (See Kapovich, Myasnikov, S., Shpilrain, "Generic-case complexity, decision problems in group theory and random walks", J. Algebra, 2003.)

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 $\Phi(x) = \chi_S(x)$ whenever $\Phi(x)$ is defined (written $\Phi(x)\downarrow$) 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.

If we "clock" Φ by requiring that on input w Φ answers in no more than $f(|w|)$ steps or we count Φ as not answering. In this case we say that Φ generically computes S in time f.

An Example from Group Theory

A basic theorem from Kapovich, Miasnikov, S., Shpilrain is the following.

Theorem

Let G be a finitely generated group with a finite index subgroup H which has an infinite quotient group $\overline{H}$ for which the word problem is solvable in the complexity class $\mathcal{C}$. Then the word problem for G has generic-case complexity in the class $\mathcal{C}$. If the group $\overline{H}$ is nonamenable, then the generic-case complexity of the word problem for G is strongly in $\mathcal{C}$.

Magnus solved the word problem for one-relator groups in the 1930's. We have no idea of the possible worst-case complexities of the word problem over the class of all one-relator groups.

Corollary

Let G be a group with a finite presentation involving at least two more generators than relators. In particular, this includes any one-relator group with at least three generators.

By a result of Benjamin Baumslag and Steve Pride, G has a subgroup of finite index that can be mapped homomorphically onto the free group of rank two. Hence the previous theorem shows that the word problem of G is strongly generically linear time.

Turing Machines and the Halting Problem

One of the great accomplishments of twentieth century mathematics was to give a precise definition of what it means to be *computable*. The model usually used is "computable by a Turing machine". One can just think of a Turing machine as an idealized computer: the machine has an infinite memory and we do not care how many operations a computation takes.

Note that each machine is described by a finite, deterministic program. Thus there are only countably many Turing machines and there is an *effective* list of *all* of them.

A machine started on a particular input may or may not eventually halt. The *Halting Problem* for Turing machines is the following decision problem: Is there an algorithm which, given a Turing machine M and a word w in the alphabet of M, decides if M halts when started with input w? Turing showed in 1936 that the Halting Problem is not computable by a Turing machine.

Computably enumerable sets

A basic concept in computability is that of a *computably enumerable* set, written a c.e. set. Intuitively, a set is c.e. if there is an algorithm which enumerates all the elements of A. (Possibly with many repetitions and with no constraints on the order in which elements are enumerated.)

As usual with robust concepts , there are several equivalent definitions. One of the most useful is that A is c.e. if and only if A is the halting set of some Turing machine.

The basic lemma of computability theory is that a set A is computable if and only if both A and its complement $\overline{A}$ are c.e.

Let K be the set of all pairs $< w, M >$ such that the Turing machine M halts on input w. We call K the Halting Problem. It is important that although K is not computable, the set K is computably enumerable: Use "bounded simulation". K is the basic example of a noncomputable c.e. set.

The Halting Problem K is also the original *complete* set. It is complete for the class of c.e. sets, essentially by definition. That A is c.e. means that it is the halting set of a Turing machine M, so to decide if $w \in A$ just ask if $< w, M > \in K$.

Oracle Turing Machines

We often think in terms of relative computability. “I don’t know how to compute A, but if I could compute B *then* I could compute A.” Turing himself formalized relative computability in terms of *oracle Turing machines*.

An oracle Turing machine is just like an ordinary machine except that the machine has a special random access memory where register n contains either a 0 or a 1. The machine has an *oracle for the set A* if the registers contain the characteristic function of A. The machine has a special “branch” instruction which goes to different next instructions depending on whether or not the content of a queried register is 0 or 1.

It is important to stress that oracle Turing machines are still defined by finite, deterministic programs. In particular, there are only countably many oracle Turing machines and it is possible to make an effective list of all such machines. Of course, what a particular machine would actually compute depends on the contents of the oracle registers.

Turing Reducibility

Oracle Turing machines allow us to define one set is “computationally reducible” to another one and when two sets are “computationally equivalent”.

Definition

The set A is Turing reducible to the set B, written $A \leq_T B$ if there is an oracle Turing machine M which, when provided with an oracle for B, computes the set A.

It is clear that $A \leq_T B$ is reflexive and transitive and is thus a partial order on subsets of $\mathbb{N}$.

Definition

Two sets A and B are Turing equivalent, written $A \equiv_T B$, if $A \leq_T B$ and $B \leq_T A$.

It is easy to check that $A \equiv_T B$ is indeed an equivalence relation and its equivalence classes are called *Turing degrees*. Turing degrees are the basic structure of computability theory.

There is a least Turing degree, the degree 0 of computable sets. The degree of the Halting Problem K is denoted by $0'$. A Turing degree is c.e. if it contains a c.e. set. Note that not all sets in a nonzero c.e. degree are c.e. For exmaple, the complement of the Halting Problem is not c.e. since K is not computable. It is not at all obvious that there are any c.e. degrees besides 0 and $0'$. The Friedburg-Muchnik Theorem is a deep result showing that there are infinitely many distinct c.e. degrees. There are also non c.e. degrees between 0 and $0'$. Note that

since a Turing degree contains only countably many sets, there are uncountably many degrees.

Whether or not a set is generically computable depends on how information is distributed in the set. From now on, we only consider subsets of $\mathbb{N}$.

First, *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 has density 0. All the information about A is in a set of density 0.

Note that this example shows that one partial algorithm can generically compute uncountably many sets.

On the other hand, every nonzero Turing degree contains a set which is not generically computable. Let $R_k = \{m : 2^k \mid m, 2^{k+1} \nmid m\}$. So R_0 is the set of odd positive integers. Note that $\rho(R_k) = 2^{-(k+1)}$.

For arbitrary A, let $\mathcal{R}(A) = \bigcup_{n \in A} R_n$. Then $\mathcal{R}(A)$ is generically computable if and only if A is computable. The information on whether or not $n \in A$ is spread over a set of positive density.

C.E. sets of density 1

Most sets are not generically computable. If A is an infinite subset of $\mathbb{N}$ we can regard the characteristic function of A as the infinite binary expansion of a number in the unit interval. So we can consider the collection of infinite subsets of $\mathbb{N}$ as the unit interval and use Lebesgue measure. It is easy to show directly that the family of generically computable subsets has measure 0.

Are there any sets which we know must be generically computable from their definition? Yes, a c.e. set A of density 1.

Bounded simulation of a Turing machine whose halting set is A gives a generic algorithm. But can we do better? Is there necessarily a generic algorithm with computable domain?

The answer is “No”. There are c. e. sets of density 1 with no computable subset of density 1. It turns out that this property characterizes an important class of Turing degrees.

The Jump Operator

We can relativize the halting problem. If A is arbitrary, the *jump* of A, written A', is the halting problem for Turing machines with an oracle for A. It is easy to see that if $A \equiv_T B$ then $A' \equiv_T B'$ so we can view the jump as an operator on Turing degrees.

It is a fact that almost all of the results of classical computability theory *relativise*. This means that what oracle, if any, one is using does not matter. For example, take the proof that the Halting Problem for Turing machines is not computable by a Turing machine. In the proof, replace the words "Turing machine" by "Turing machine with an oracle for A". Not only is the statement correct but the proof as given is correct.

The Baker-Gill-Solovay Theorem says that the question $\mathcal{P} = \mathcal{NP}$ does Not relativise. There are oracles A for which $\mathcal{P}_\mathcal{A} = \mathcal{NP}_\mathcal{A}$ and oracles B for which $\mathcal{P}_\mathcal{B} \neq \mathcal{NP}_\mathcal{B}$.

Non-low c.e. degrees

If $\underline{\mathbf{a}}$ is a c.e. degree then $0' \leq_T \underline{\mathbf{a}}' \leq_T 0''$. A nonzero c.e. degree is *low* if $\underline{\mathbf{a}}' = 0'$, that is, $\underline{\mathbf{a}}'$ is as low as possible.

Theorem

(Downey, Jockusch, S.) A c.e. degree $\underline{\mathbf{a}}$ *is Not low if and only if* $\underline{\mathbf{a}}$ *contains a c.e. set A of density* 1 *with no computable subset of density* 1*.*

The theorem shows that there is indeed a strong connection between density and classical concepts of computability theory.

The Arithmetic Hierarchy

The original complexity hierarchy is, of course, the Borel Hierarchy. (Borel, “Leçons sur la theorie des fonctions”, 1898.) Other complexity hierarchies imitate the definition in the appropriate context. The Arithmetic Hierarchy in computability theory was defined by Kleene.

Let Δ_1 denote the class of computable sets, which are our “basic” sets.

Define Σ_1 to be the class of all subsets of $\mathbb{N}$ of the form $\{x : (\exists \overrightarrow{y})R(x, \overrightarrow{y})\}$ where $R(x, \overrightarrow{y})$ is a computable relation. So Σ_1 is the class of c.e. sets.

Define Π_1 to be the class of all subsets of $\mathbb{N}$ of the form $\{x : (\forall \overrightarrow{y})R(x, \overrightarrow{y})\}$ where $R(x, \overrightarrow{y})$ is a computable relation. These are the complements of Σ_1 sets, the co-c.e. sets.

Note that an existential quantifier corresponds to union and a universal quantifier corresponds to intersection. Also note that $\Delta_1 = \Sigma_1 \cap \Pi_1$ by the fundamental lemma.

We continue by alternating quantifiers to obtain the Arithmetic Hierarchy

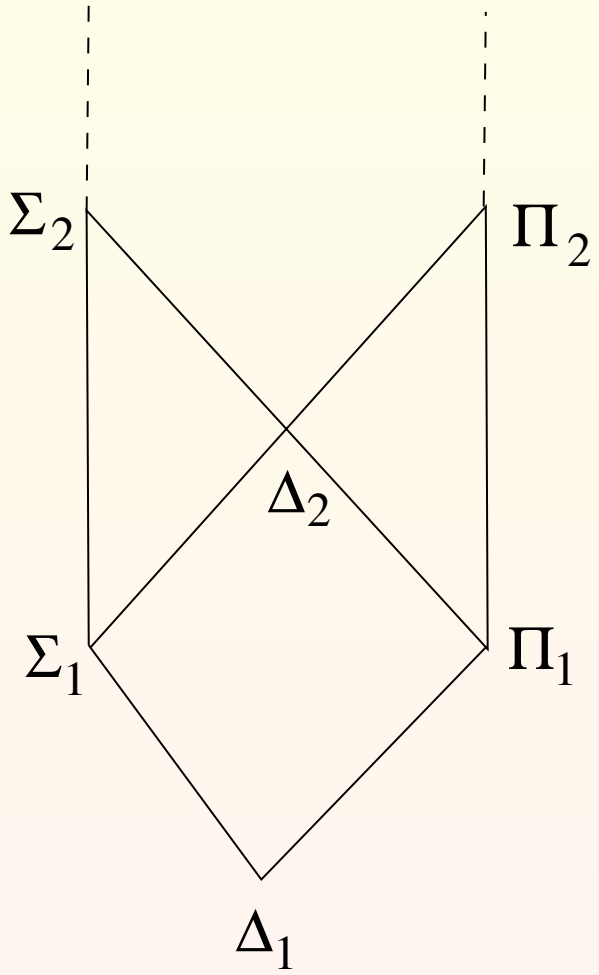

Figure: The Arithmetic Hierarchy

Post's Theorem

The levels of the Arithmetic Hierarchy are exactly correlated with the number of iterations of the jump operator.

Theorem (Post's Theorem)

1. $A \in \Sigma_{n+1} \iff A$ *is c.e. in* 0^n.
2. $A \in \Delta_{n+1} \iff A \leq_T 0^n$.

The Complexity of Densities

One of striking things to emerge from looking at density is that there is an extremely tight connection between the complexity of a set in the arithmetic hierarchy and the complexity of its densities as real numbers.

To discuss the complexity of real numbers, *fix* a computable bijection between the rationals and $\mathbb{N}$. Then we can talk about the position of a set of rationals in the Arithmetic Hierarchy.

Definition

Define a real number r to be left-Σ_n if its corresponding lower Dedekind cut in the rationals,

$$\{q \in \mathbb{Q} : q < r\},$$

is Σ_n.
Define "left-Π_n"and "left-Δ_n" analogously.

Characterization of Densities

Theorem

(Downey, Jockusch, S.) Let r be a real number in the interval $[0,1]$. Then the following hold:

(i) *r is the lower density of some set in Δ_n if and only if r is left-Σ_{n+1}.*

(ii) *r is the upper density of some set in Δ_n if and only if r is left-Π_{n+1}.*

(iii) *r is the density of some set in Δ_n if and only if and only if r is left-Δ_{n+1}.*

(iv) *r is the lower density of some set in Σ_n if and only if r is left-Σ_{n+2}.*

(v) *r is the upper density of some set in Σ_n if and only if r is left-Π_{n+1}.*

(vi) *r is the density of some set in Σ_n if and only if r is left-Π_{n+1}.*

In short, the position of a set in Arithmetic Hierarchy strictly limits the complexities of its densities as real numbers and vice versa.

Computability at densities less than 1

Definition

. 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 $\phi(n)\downarrow$ and with the lower density of Domain(ϕ) greater than or equal to r.

Note that A is computable at density 1 means that A is generically computable. A first natural question is: Are there sets which are computable at all densities $r < 1$ but which are not generically computable?

Actually, we have already seen that every nonzero Turing degree contains such sets. Recall that $\mathcal{R}(A) = \bigcup_{n \in A} R_n$. Any set of the form $\mathcal{R}(A)$ is computable at all densities less than 1. For any $t \geq 0$ the finite list of which numbers in $[0, t]$ are in A, allows one to calculate $\bigcup_{k \leq t} R_k$. But $\mathcal{R}(A)$ is generically computable if and only if A is computable.

The partial computability bound

For any set $A \subseteq \mathbb{N}$ we can define the *partial computability bound*, $\alpha(A)$ as $\alpha(A) = \sup\{r : A \text{ is computable at density } r\}$.

It is not difficult to prove that for any $r \in [0, 1]$. there is a set A of density r with $\alpha(A) = r$.

Coarse computability

There is another natural model of less than perfect computability. Two sets A and B are *coarsely similar* if their symmetric difference $A \triangle B$ has density 0. It is not difficult to check that this is indeed an equivalence relation. Any set coarsely similar to A is called a *coarse description* of A.

Say that a set A is *coarsely computable* if A is coarsely similar to a computable set C.

We can think of this in the following way. We now have a *total* "algorithm" for A which may make mistakes, (the algorithm for C) , but is correct on a set of density 1.

In particular, all sets of density 0 and all sets of density 1 are coarsely computable.

Miasnikov and Osin have given a beautiful example of a finitely generated, computably presented group whose word problem is not generically computable. In essence, the Golod-Shafarevch inequality allows one to do Post's construction of a simple set. alIt is a difficult question whether or not there exists a *finitely presented* group whose word problem is not generically computable.

Observation. The word problem of *any* finitely generated group G is coarsely computable.

There are two cases. If G is finite then the word problem is computable.

If G is infinite, then, since G is finitely generated, the set of words not equal to the identity has density 1. So always answer "No".

The coarse computability bound

In analogy with the partial computabilty bound, for any set $A \subseteq \mathbb{N}$ we can define the *coarse computability bound*, $\gamma(A)$ as

$$\gamma(A) = \sup\{r : \exists \text{ computable } C)\ [\underline{\rho}(\ \{n : A(n) = C(n)\}\ ) \geq r]\}.$$

Lemma

For every $A \subseteq \mathbb{N}$, $\alpha(A) \leq \gamma(A)$. In particular, if $\alpha(A) = 1$ then $\gamma(A) = 1$.

The lemma is not obvious but follows from results about the densities of computable subsets of c.e. sets.

The following is a recent theorem of Greg Igusa.

Theorem

If $0 \leq r < s \leq 1$ then there is a set A with $\alpha(A) = r$ and $\gamma(A) = s$.

For the moment think of finite strings of $0's$ and $1's$ as initial segments of characteristic functions. If σ, τ are two such strings then $\sigma \subseteq \tau$ means that τ extends σ. If σ is such a string and A is a set then $A \supset \sigma$ means the characteristic function of A extends σ.

Definition

A set $A \subset \mathbb{N}$ is 1-generic if for every c.e. set X of finite strings of 0's and 1's either

1. $(\exists \tau \subset A)[\tau \in X]$ *or,*
2. $(\exists \tau \subset A)(\forall \sigma \supset \tau)[\sigma \notin X]$.

A Turing degree $\boldsymbol{d}$ is 1-generic if $\boldsymbol{d}$ contains a 1-generic set.

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

Proof.

If f is any computable function let $S_{f,n,j}$ be the set of all finite binary strings σ such that $|\sigma| \geq j$ and

$$\rho_{|\sigma|}(\{k < |\sigma| : \sigma(k) = f(k)\}) < \frac{1}{n}$$

Any finite binary string can be extended to a string in $S_{f,n,j}$ so the first condition of 1-genericity must hold, so $\{k : f(k) = A(k)\}$ has lower density 0. Thus $\gamma(A) = 0$. □

If A is a 1-random set then $\gamma(A) = 1/2$. (1-randomness is the infinitary version of Kolmogorov complexity.)

It is a theorem that the density of a 1-random set is $1/2$. So either of the constant algorithms works at density $1/2$. If there were an algorithm which worked at density greater than $1/2$ this would define a test showing that the set was not random.

Theorem

(Jockusch,S.) Every non-zero Turing degree contains a set A with $\gamma(A) \leq 1/2$.

Proof.

Let $I_n = [n!, (n+1)!)$. If A is not computable let

$$\mathcal{I}(A) = \cup_{n \in A} I_n$$

Then $\mathcal{I}(A)$ is Turing equivalent to A. If there is a computable C which agrees with A with lower density greater than $1/2$ we can compute A by majority vote. Our proposed algorithm for $n \in A$ is to say that $n \in A$ if more than half of the elements of I_n are in C.

Since the intervals I_n grow so quickly, our proposed algorithm can only make finitely many mistakes. Correcting these finitely many mistakes, we have a correct total algorithm for A. □

Note that this procedure is not unform. It depends on which set C is

The distance D

If $A, B \subseteq \mathbb{N}$, define $D(A,B) = \overline{\rho}(A \triangle B)$ A Venn diagram argument shows that D satisfies the triangle inequality and so D is a pseudo-metric on subsets of $\mathbb{N}$.

Lemma

If $A \subseteq \mathbb{N}$ then $\underline{\rho}(A) = 1 - \overline{\rho}(\overline{A})$.

Definition

If $A \subseteq \mathbb{N}$ and $\mathcal{S} \subseteq \mathcal{P}(\mathbb{N})$ let

$$\delta(A, \mathcal{S}) = \inf\{D(A,S) : S \in \mathcal{S}\}.$$

So $\gamma(A) = 1 - \delta(A, \mathcal{C})$ where $\mathcal{C}$ is the class of computable sets. Thus $\gamma(A) = 1$ if and only if A is a limit of computable sets in the pseudometric.

As we have seen, for every set A the set $\mathcal{R}(A)$ is coarsely computable at every density $r < 1$. So when is $\mathcal{R}(A)$ coarsely computable?

Theorem

(Jockusch, S.) The set $\mathcal{R}(A)$ is coarsely computable if and only if $A \in \Delta_2$.

So if $\underline{\mathbf{d}}$ is a Turing degree with $\underline{\mathbf{d}} \not\leq_T 0'$ then $\underline{\mathbf{d}}$ contains a set A such that $\gamma(A) = 1$ but A is not coarsely computable.

Coarse computability and generic computability are actually very different in many ways.

Theorem

(Downey, Jockusch, S.) Every nonzero c.e. degree contains a set A which is generically computability but not coarsely computable.

So, if a nonzero Turing degree $\boldsymbol{d}$ is either not below $0'$ or is c.e. then $\boldsymbol{d}$ contains a set A such that $\gamma(A) = 1$ but A is not coarsely computable. This raises the

Question

Does every nonzero Turing degree contain a set A with $\gamma(A) = 1$ such that A is not coarsely computable?

Theorem

(Hirschfeldt, Jockusch, McNicholl, S.) If $\boldsymbol{d} \leq_T 0'$ *is a* 1*-generic degree then: If* $A \leq_T \boldsymbol{d}$ *and* $\gamma(A) = 1$ *then A is coarsely computable.*

So there is at least some condition implying that if A is a limit of computable sets in the pseudo-metric D then A is coarsely computable.

Bi-immunity and Dense undecidability

A basic concept of computability theory is that of a bi-immune set. A set A is *bi-immune* if neither A nor its complement contain an infinite c.e. set.

This is a very strong condition. Note that the condition says that any partial algorithm for a bi-immune set can only have a finite domain.

A theorem of Jockusch shows that there are non-zero Turing degrees which do not contain any bi-immune set.

Dense Undecidability

Miasnikov and Rybalov defined a set A to be *absolutely undecidable* if every partial computable function which agrees with the characteristic function of A on its domain has a domain of density 0. A better terminology would be *densely undecidable*. If A is absolutely undecidable then $\alpha(A) = 0$ but there are c.e. sets with $\alpha(A) = 0$ which are not absolutely undecidable.

Theorem

(Bienvenu, Day and Hözl) Every nonzero Turing degree contains a set which is absolutely undecidable.

So the question of how strong an undeciability condition one can force into every non-zero Turing degree is answered in terms of density.

The previous result raises the analogous question for γ. Does every nonzero Turing degree contain a set A with $\gamma(A) = 0$?

We have seen that every nonzero Turing degree contains a set A with $\gamma(A) \leq 1/2$.

Theorem

(Andrews, Cao, Diamondstone, Jockusch,Lempp) If $\mathbf{d}$ *contains a computably traceable set then* $\gamma(A) \geq 1/2$ *for all* $A \in \mathbf{d}$.

We can cosider how γ globablly interacts with Turing degrees.

Definition

If $\boldsymbol{d}$ *is a Turing degree, let*

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

We have just seen that if $\boldsymbol{d}$ is a non-zero computably traceable degree then $\Gamma(\boldsymbol{d}) = 1/2$. There is only one degree with $\Gamma = 1$, the zero degree of computable sets. We have also seen that if a degree contains a 1-generic set then $\Gamma = 0$. So we now have the "Γ-question".

Question

Can Γ *have any values other than* $0, 1/2, 1$ *?*

Thank you!

Minimal pairs

Definition

Two noncomputable sets A, B are a minimal pair for Turing reducibility if $S \leq_T A$ and $S \leq_T B$ imply that S is computable.

As an example, take the even and odd halves of a random set. They have nothing to do with each other and so form a minimal pair.

Actually, the set of minimal pairs has measure 1, so almost all pairs are minimal.

Relative generic computability

One can ask, “Given an oracle for a set A, what sets can one then generically compute?”

A beautiful theorem of Greg Igusa shows that there are no minimal pairs for relative generic computability.

Theorem

If A and B are not generically computable then there is a set C which is generically computable from both A and B but such that C is not generically computable.

So the situation is the extreme opposite from the situation for Turing computability.

Coarse Reducibility and Coarse Degrees

One can define *coarse degrees* in a natural way. Recall that a *coarse description* of a set A is any set coarsely similar to A.

Definition

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 descriptiion of A.

It is clear that $\leq_C$ is reflexive and transitive.

Definition

The nonuniform coarse degree of A is $\{B : B \leq_{nc} A, A \leq_{nc} B\}$.

There is a fairly natural embedding of Turing degrees into coarse degrees. The image of the Turing degrees is very sparse.

A set A is *weakly n-random* if A is not contained in a Π^0_n-class of measure 0. The class of all weakly n-random sets has measure 1.

The embedding of Turing degrees misses the coarse degree of every weakly 2-random set.

So what has happened to the situation for minimal pairs?

Theorem

(Hirschfeldt, Jockusch, S.) If A and B are weakly 3-random, then A and B are a minimal pair for coarse computability. Any set S which is coarsely computable from both A and B must be coarsely computable.

We see that there is a deep interaction between algorithmic randomness and coarse computability, the features of which are still to be developed.

Thank you!