THE UNIVERSITY OF CHICAGO

ASYMPTOTIC NOTIONS OF COMPUTABILITY

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY
TIAGO ROYER

CHICAGO, ILLINOIS
AUGUST 2025

TABLE OF CONTENTS

ABSTRACT

In computability theory, the classical definition of a Turing machine M solving a problem requires M to *always* halt and *always* give the right answer. If we relax the "always" to "almost always", we give rise to asymptotic notions of computability, where now M halts and gives correct answers only asymptotically always. There are four asymptotic notions of computability: generic, coarse, dense, and effective dense computability.

This dissertation studies these asymptotic notions of computability, specifically the degree structures arising from the associated reducibilities. Two of the main results concerns minimal pairs: pairs of sets which are both noncomputable, and which have no common computational power. We prove that there are only measure-0 many minimal pairs for the generic degrees (Theorem 2.1.4, which is in contrast with the classic Turing degrees, for which there are measure-1 many minimal pairs) and construct a Δ^0_2 minimal pair for the coarse degrees (Corollary 2.3.6).

The third main result concerns attractive degrees. The formalization of the asymptotic notions of computability can be generalized to provide a notion of distance between Turing degrees, called the Hausdorff distance H. It turns out that for every set A, there are either measure-1 many sets which are at a distance 1 from A, or there are measure-1 many such sets which are at a distance 1/2 from A. The former are called dispersive, and the latter are called attractive. We provide a Kolmogorov-complexity flavored sufficient condition for a set to be attractive (Theorem 3.3.1).

CHAPTER 1

INTRODUCTION

The classical setting of computability and complexity theory is of worst-case scenarios; for example, the time complexity of an algorithm is the maximum number of steps, among all inputs of length n, that it takes to return an answer, and in a Turing reduction from a set A to a set B the algorithm must halt on every single input.

There is now general awareness that the strictness of worst-case scenarios may not capture a full picture of a problem or algorithm [12, 13]. Perhaps the best-known example of this phenomenon is the simplex algorithm for linear programming: there are families of instances in which the simplex algorithm takes exponential time to halt; however, on average, the algorithm converges in linear time.

Besides techniques like average-case complexity, we may analyze a problem by allowing the computational device to make mistakes sometimes. Is the problem uniformly difficult, or is the difficulty spread out in a sparse subset of its inputs? The concepts of dense computability, generic computability, coarse computability, and effective dense computability all generalize the notion of computability by requiring the algorithm to get the right answer only for "most of the inputs", rather than for all inputs.

These asymptotic notions of computability give rise to a degree structure analogous to the Turing degrees, but with different properties. This dissertation will mostly be concerned with these degree structures.

This is an expository chapter aggregating results from the literature.

Section 1.1 recapitulates the basic definitions of set density, defines the four asymptotic notions of computability, and defines the corresponding notions of reducibility. Section 1.2 defines minimal pairs for Turing, coarse and dense degrees, and argues about the level of randomness needed to construct those degrees. Section 1.3 shows Igusa's result [11] that there are no minimal pairs for relative generic computability, and Hirschfeldt's result [6] that there are minimal pairs for generic reducibility.

1.1 Background

This section recapitulates the basic definitions and proves some useful theorems.

1.1.1 Notation

We follow the convention from Computability Theory of identifying every natural number $e \in \mathbb{N}$ with the source code of a Turing machine, and vice-versa; for example, we can interpret the binary representation of $e+1$ (except the leading digit 1) as a text file containing the source code of a program in any fixed programming language. This allows us to speak of the eth Turing machine, and gives us an ordering of all Turing machines, which is useful for diagonalization arguments.

The function computed by the eth Turing machine is denoted by $\Phi_e : \mathbb{N} \to \mathbb{N}$. In the interpretation above, if the text file corresponding to e does not represent a valid program, we simply assume that Φ_e is the partial function which is nowhere defined. We assume that all machines are oracle machines, letting $\Phi_e = \Phi_e^{\emptyset}$.

Although the value of $\Phi_e(n)$ is not always defined, we can still run the eth Turing machine on n for a finite number of steps, say s; we denote this by $\Phi_e(n)[s]$, so that if $\Phi_e(n)\downarrow$ then there exists some k for which $\Phi_e(n)[s]\uparrow$ if $s < k$ and $\Phi_e(n)[s]\downarrow = \Phi_e(n)$ for all $s \geq k$.

The indicator function of a set A is denoted by $\mathbb{1}_A$ (so that $\mathbb{1}_A(x) = 1$ if $x \in A$ and $\mathbb{1}_A(x) = 0$ otherwise).

We assume that the set $\mathbb{N}$ of natural numbers contains the number 0. However, in a few cases (e.g. the sets R_k and J_k below), we will construct partitions of $\mathbb{N} \setminus \{0\}$ instead of of $\mathbb{N}$, to simplify some calculations. This means that function definitions which are based on these partitions will leave the value of $f(0)$ undefined. In all cases we can arbitrarily set $f(0) = 0$, so we omit this step of the construction.

Given two sets $A, B \in \mathbb{N}$, we define their join $A \oplus B$ by

$$A \oplus B = \{2n \mid n \in A\} \cup \{2n+1 \mid n \in B\}.$$

We identify a subset A of $\mathbb{N}$ with an infinite string in 2^ω. Therefore, we say that a finite string σ is a prefix of A, denoted $\sigma \prec A$, if the first $|\sigma|$ bits of A, interpreted as a string, agrees with σ. The basic open set $[\![\sigma]\!]$ is defined by

$$[\![\sigma]\!] = \{X \in 2^\omega \mid \sigma \prec X\}.$$

For a subset $A \subseteq \mathbb{N}$ of the natural numbers, we define $A \upharpoonright k = \{n \in A \mid n < k\}$, which is the set A "truncated" to the first k natural numbers. We can naturally identify $A \upharpoonright k$ with a binary string of length k, so in many places where $A \upharpoonright k$ is used as an argument to a computable function, the argument effectively is the corresponding string.

1.1.2 Tools from Measure Theory

The following two theorems from measure theory will be used in several places in this paper.

Theorem 1.1.1 (Lebesgue Density Theorem [5, Theorem 1.2.3]). *Let $\mathcal{A} \subseteq 2^\omega$ be a measurable class, and define $D(\mathcal{A})$ by*

$$D(\mathcal{A}) = \{X \in 2^\omega \mid \lim_{n\to\infty} 2^n \mu([\![X \upharpoonright n]\!] \cap \mathcal{A}) = 1\}.$$

Then $D(A)$ is measurable and $\mu(\mathcal{A} \triangle D(\mathcal{A})) = 0$.

The sets $X \in D(\mathcal{A})$ are said to have *density 1* at $\mathcal{A}$.

Proof. Observe first that $D(\mathcal{A}) \cap D(\overline{\mathcal{A}}) = \emptyset$, so $D(\mathcal{A}) \setminus \mathcal{A} \subseteq \overline{\mathcal{A}} \setminus D(\mathcal{A})$; therefore, it suffices showing that $\mu(\mathcal{A} \setminus D(\mathcal{A})) = 0$.

For $0 < \alpha < 1$ define the set $\mathcal{B}_\alpha$ by

$$\mathcal{B}_\alpha = \{X \in \mathcal{A} \mid \lim_{n\to\infty} 2^n \mu([\![X \upharpoonright n]\!] \cap \mathcal{A}) < \alpha\}.$$

Then $\mathcal{A} \setminus D(\mathcal{A}) = \bigcup_\alpha \mathcal{B}_\alpha$, where the union is over rational $\alpha < 1$. Therefore, it suffices to show that $\mu(\mathcal{B}_\alpha) = 0$ for all $\alpha < 1$.

Suppose by contradiction that $\mu(\mathcal{B}_\alpha) > 0$. Because μ is a regular measure, there exists an open set $V \supset \mathcal{B}_\alpha$ such that $\mu(V) < \mu(\mathcal{B}_\alpha)/\alpha$. Let $U \subseteq V$ be the union of all $[\![\sigma]\!]$ for which $\mu([\![\sigma]\!] \cap \mathcal{A}) < 2^{-|\sigma|}\alpha$, and let $\sigma_0, \sigma_1, \ldots$ be a sequence of strings where the intervals $[\![\sigma_i]\!]$ are pairwise disjoint and $U = \bigcup_i [\![\sigma_i]\!]$.

On one hand, if $X \in \mathcal{B}_\alpha$, then $\mu([\![X \upharpoonright n]\!] \cap A) < \alpha 2^{-n}$ for some n, so $[\![X \upharpoonright n]\!] \subseteq U$ whence $X \in U$. This shows that $\mathcal{B}_\alpha \subseteq U$.

On the other hand,

$$\begin{aligned}
\mu(U \cap \mathcal{B}_\alpha) &= \sum_i \mu([\![\sigma_i]\!] \cap \mathcal{B}_\alpha) \\
&\leq \sum_i \mu([\![\sigma_i]\!] \cap \mathcal{A}) \\
&< \sum_i 2^{-|\sigma_i|}\alpha \\
&= \alpha\mu(U) \\
&\leq \alpha\mu(V) < \mu(\mathcal{B}_\alpha).
\end{aligned}$$

Therefore, U cannot possibly contain $\mathcal{B}_\alpha$, a contradiction. $\square$

A corollary of this theorem is that if $\mu(\mathcal{A}) > 0$, for any $\alpha < 1$ there exists some σ such that $\mu(\mathcal{A} \cap [\![\sigma]\!]) > \alpha 2^{-|\sigma|}$. That is, the relative density of $\mathcal{A}$ in σ can be made arbitrarily close to 1. This is the core of the "majority vote" argument, which will be used in Section 1.2.

A *tailset* is a class $\mathcal{A} \subseteq 2^\omega$ such that if $\sigma X \in \mathcal{A}$ then $\tau X \in \mathcal{A}$ for all τ with $|\tau| = |\sigma|$;

that is, we may flip any finite number of bits of X and still stay inside $\mathcal{A}$. We have the following theorem.

Theorem 1.1.2 (Kolmogorov's 0-1 law). *If $\mathcal{A}$ is a tailset, then either $\mu(\mathcal{A}) = 0$ or $\mu(\mathcal{A}) = 1$.*

Proof. Suppose that $\mu(\mathcal{A}) > 0$, and let $\alpha < 1$ be given. Then by the above corollary to the Lebesgue Density Theorem, there exists some string σ such that $\mu(\mathcal{A} \cap [\![\sigma]\!]) > \alpha 2^{-|\sigma|}$. But because $\mathcal{A}$ is a tailset, we have $\mu(\mathcal{A} \cap [\![\sigma]\!]) = \mu(\mathcal{A} \cap [\![\tau]\!])$ for each τ with $|\tau| = |\sigma|$. This means that $\mu(\mathcal{A}) > \alpha$. Since α was arbitrary, this means $\mu(\mathcal{A}) = 1$. □

1.1.3 Notions of Randomness

In this paper we will deal with three notions of randomness. The most prominent one is Martin-Löf randomness, which we define below.

Definition 1.1.3 ([5, Section 6.4]). Given a set B, a *Martin-Löf test* relative to B is a sequence $\{U_n\}_{n\in\mathbb{N}}$ of B-uniformly $\Sigma^{0,B}_1$ sets[1] such that $\mu(U_n) \leq 2^{-n}$ for all n. We say that a set A *passes* a Martin-Löf test $\{U_n\}_{n\in\mathbb{N}}$ if $A \notin U_n$ for some n, and we say that A is 1-*random* relative to B if A passes every Martin-Löf tests relative to B.

Sets which are 1-random relative to $\emptyset$ are simply called "1-random", or "Martin-Löf random".

Intuitively, a Martin-Löf test $\{U_n\}_{n\in\mathbb{N}}$ corresponds to a procedure for picking out regularities in sets. If $B \in U_n$, it means that U_n picked out some regularity in a prefix of B. If $B \in \bigcap_n U_n$, it means that the test can find regularities in arbitrarily long prefixes of B, so B ought not to be called random.

1. That is, there is a single B-computable sequence of indices $e_1, e_2, \ldots$ such that $U_n = \{X \mid \exists k \Phi^B_{e_n}(X \upharpoonright k) = 1\}$.

The definition is fairly flexible. For example, we can replace U_n with $V_n = \bigcap_{k \leq n} U_k$ to get another test with $\bigcap_n U_n = \bigcap_n V_n$, but with the additional hypothesis that $V_n \supseteq V_{n+1}$. We could require only that $\mu(U_n) \leq f(n)$ for some computable function f with $\lim_{n\to\infty} f(n) = 0$; indeed, for each n, let $V_n = \bigcap_{k \leq m} U_k$ where m is the least integer such that $f(m) \leq 2^{-n}$. Then $\{V_n\}_{n \in \mathbb{N}}$ is a Martin-Löf test and $\bigcap V_n = \bigcap U_n$.

Another possible modification is requiring the sequence $\{U_n\}_{n\in\mathbb{N}}$ to be uniformly $\Sigma_1^{0,B}$, instead of B-uniformly $\Sigma_1^{0,B}$. That is, the definition above requires that there exists a B-computable sequence $e_1, e_2, \ldots$ of indices such that

$$U_n = \{X \mid \exists k \Phi_{e_n}^B (X \upharpoonright k) = 1\}.$$

Because the Turing functional Φ_{e_n} has access to B as an oracle, we may assume that the sequence $e_1, e_2, \ldots$ is computable, rather than B-computable.

One specific class of regularities corresponds to the intuition that we should not be able to predict whether a certain bit of a sequence will be 0 or 1. We can formalize this intuition as follows.

Definition 1.1.4 ([5, Definition 7.4.1]). A *selection function* is a function $F : \mathbb{N} \to \mathbb{N}$ which is strictly increasing. A set A is Church-stochastic if, for every computable selection function f, we have

$$\lim_{n\to\infty} \frac{|\{k < n \mid f(k) \in A\}|}{n} = \frac{1}{2},$$

where $|C|$ denotes the cardinality of the set C.

That is, no matter how we pick the bits $f(0), f(1), \ldots$ to be analyzed, the probability of $A(f(i))$ being 1 tends to $\frac{1}{2}$. We have that 1-randomness is enough to guarantee Church-stochasticity.

Proposition 1.1.5 (see [5, Theorem 7.4.2]). Let A be 1-random. Then A is Church-stochastic.

Proof. Suppose that f is a selection function for which

$$\liminf_{n\to\infty} \frac{|\{k < n \mid f(k) \in A\}|}{n} < \frac{1}{2} - \varepsilon$$

for some $\varepsilon > 0$. (If the lim sup was larger than $\frac{1}{2}+\varepsilon$, then the argument would be analogous.)

Define the set V_n by

$$V_n = \left\{ X \,\middle|\, \frac{|\{k < n \mid f(k) \in X\}|}{n} < \frac{1}{2} - \varepsilon \right\}.$$

Note that $A \in V_n$ for infinitely many n.

By the Chernoff bound [1, Theorem A.1.1],

$$\mu(V_n) \le e^{-2\varepsilon^2 n},$$

so if we set $U_n = \bigcup_{m>n} V_n$ we have

$$\mu(U_n) \le \frac{e^{-2\varepsilon^2 n}}{1 - e^{-2\varepsilon^2}}.$$

Since the V_n are uniformly Σ^0_1 classes, so are the U_n. By the above bound, the sequence $\{U_n\}_{n\in\mathbb{N}}$ is a Martin-Löf test. By construction, we have $A \in V_n$ for infinitely many n, which means that $A \in U_n$ for all n. This contradicts the hypothesis that A is 1-random. □

The converse of the above proposition is false. One striking example is provided by Ville's Theorem, which implies that there is a Church-stochastic set A such that $|\{k < n \mid k \in A\}| \le n/2$ for all n (see e.g. [5, Theorem 6.5.1] for a proof). On the other hand, for several of our applications, Church-stochasticity will be enough.

We define the notion of n-randomness by replacing Σ^0_1 classes with Σ^0_n classes.

Definition 1.1.6 ([5, Section 6.8]). A set A is *n-random* relative to B if, for every B-

uniformly sequence $\{U_n\}_{n\in\mathbb{N}}$ of $\Sigma^{0,B}_n$ classes such that $\mu(U_n) \le 2^{-n}$, we have $A \notin \bigcap_n U_n$.

Sets which are n-random relative to $\emptyset$ are simply called "n-random".

Recall that the sequence $\{U_n\}_{n\in\mathbb{N}}$ is B-uniformly $\Sigma^{0,B}_n$ if there is a B-computable sequence of indices $e_1, e_2, \dots$ such that

$$U_n = \{X \mid \exists k_1 \forall k_2 \exists k_3 \dots Q k_n \Phi^B_{e_n}(X \restriction k_1, \dots, X \restriction k_n) = 1\},$$

where Q is the quantifier $\exists$ if n is odd and $\forall$ if n is even. Similar to the 1-random case, we can require the sequence to be computable (as opposed to B-computable). Due to the quantifiers, we may also allow the sequence to be $B^{(n)}$-computable.

For a set A, it is true that A is $\Sigma^{0,B}_{n+1}$ if and only if A is $\Sigma^{0,B^{(n)}}_1$, but this is not the case for classes in 2^ω. Therefore, the Martin-Löf tests for n-randomness and the Martin-Löf tests for 1-randomness relative to $\emptyset^{(n-1)}$ are not the same. Surprisingly, these tests still yield the same notions of randomness. Following [5, Section 6.8], we will show this using two lemmas.

Lemma 1.1.7 ([5, Lemma 6.8.1]). Let n be fixed, and denote by μ_e the measure of the $\Sigma^{0,B}_n$ class indexed by e. Then the set

$$\{(e,q) \mid q \in \mathbb{Q} \wedge \mu_e > q\}$$

is a $B^{(n-1)}$-c.e. set. Thus, from an index of a $\Sigma^{0,B}_n$ class S, we can $B^{(n)}$-compute the measure $\mu(S)$.

Proof. Induction on n. For $n = 1$, given e and q, if V is the $\Sigma^{0,B}_1$ class indexed by e, then $V = \bigcup_k [\![\sigma_k]\!]$ for some sequence $\sigma_0, \sigma_1, \dots$ of strings which can be uniformly B-computed from e. We may assume that the strings are mutually incomparable (no string is a prefix of another string), so that

$$\mu(V) = \sum_k 2^{-|\sigma_k|}.$$

Then $\mu(V) > q$ if and only if some finite portion of the sum above is greater than q, which is a B-c.e. property.

For $n > 1$, if V is the $\Sigma_n^{0,B}$ class indexed by e, we can decompose $V = \bigcup_k V_k$ where each V_k is a $\Pi_{n-1}^{0,B}$ class and $V_k \subset V_{k+1}$. Then given k and q, we can $B^{(n-1)}$-compute whether $\mu(V_k) > q$ or not. Then $\mu(V) > q$ if and only if $\mu(V_k) > q$ for some k, which is a $B^{(n-1)}$-c.e. property. □

Lemma 1.1.8 ([5, Theorem 6.8.3]). The following is true for every $n \geq 1$ and every set B.[2]

1. From an index of a $\Sigma_n^{0,B}$ class S and $q \in \mathbb{Q}$, we can B-compute an index of a $\Sigma_1^{0,B^{(n-1)}}$ class U such that $U \supseteq S$ and $\mu(U) < \mu(S) + q$.

2. From an index of a $\Pi_n^{0,B}$ class P and $q \in \mathbb{Q}$, we can B-compute an index of a $\Pi_1^{0,B^{(n-1)}}$ class V such that $V \subseteq S$ and $\mu(V) > \mu(P) - q$.

3. From an index of a $\Sigma_n^{0,B}$ class S and $q \in \mathbb{Q}$, we can $B^{(n)}$-compute an index of a $\Pi_{n-1}^{0,B}$ class V such that $V \subseteq S$ and $\mu(U) > \mu(S) - q$.

4. From an index of a $\Pi_n^{0,B}$ class P and $q \in \mathbb{Q}$, we can $B^{(n)}$-compute an index of a $\Sigma_{n-1}^{0,B}$ class U such that $U \supseteq S$ and $\mu(V) < \mu(P) + q$.

Proof. Note that 2 follows from 1 and that 4 follows from 3 by taking complements. We will prove 4 directly and 1 by induction.

Given a $\Sigma_n^{0,B}$ class S, let $S_0 \subseteq S_1 \subseteq \cdots$ be uniformly $\Pi_{n-1}^{0,B}$ classes such that $S = \bigcup_k S_k$.

For 3, by the lemma above, the set $B^{(n-1)}$ can compute the values of $\mu(S_k)$, so $B^{(n)}$ can compute some K for which $\mu(S_K) > \mu(S_k) - q$ for all $k > K$. Then $\mu(S_K) > \mu(S) - q$ and S_K is a $\Pi_{n-1}^{0,B}$ class contained in S.

2. Compared to [5, Theorem 6.8.3], we simplify the proof a bit by making the claims in parts 3 and 4 a bit weaker.

For 1, if $n = 1$ we may let $U = S$, and if $n > 1$, use part 4 and induction to $B^{(n-1)}$-compute a sequence of indices of $\Sigma^{0,B}_{n-2}$ classes $V_0, V_1, \ldots$ such that $V_k \supseteq S_k$ and $\mu(V_k) < \mu(S_k) + q2^{-k-2}$ for all k. Now, for each k, use part 1 to compute an index for a $\Sigma^{0,B^{(n-2)}}_1$ class U_k such that $U_k \supseteq V_k$ and $\mu(U_k) < \mu(V_k) + q2^{-k-2}$. Finally, let $U = \bigcup_k U_k$. Then U is a $\Sigma^{0,B^{(n-1)}}_1$ class (because it is a union of a sequence $\Sigma^{0,B^{(n-2)}}_1$ classes, whose sequence of indices were $B^{(n-1)}$-computed from an index for S), $U \supseteq S$, and

$$\begin{aligned}
\mu(U - S) &= \mu\left(\bigcup_k U_k - \bigcup_k S_k\right) \\
&\leq \mu\left(\bigcup_k (U_k - S_k)\right) \\
&\leq \sum_k \mu(U_k - S_k) \\
&\leq \sum_k \mu(U_k - V_k) + \mu(V_k - S_k) \\
&\leq \sum_k q2^{-k-2} + q2^{-k-2} = q.
\end{aligned}$$

□

We can now prove the main result.

Theorem 1.1.9 ([5, Corollary 6.8.5]). *A set is n-random relative to B if and only if it is 1-random relative to $B^{(n-1)}$.*

Proof. Suppose that A is not n-random relative to B, and let $U_1, U_2, \ldots$ be a test for which A fails. That is, $U_1, U_2, \ldots$ is a B-uniform sequence of $\Sigma^{0,B}_n$ classes for which $\mu(U_k) < 2^{-k}$, and $A \in \bigcap_k U_k$.

By Lemma 1.1.8, we can B-compute indices for $\Sigma^{0,B^{(n-1)}}_1$ classes $V_1, V_2, \ldots$ with $V_k \supset U_k$ and $\mu(V_k) < \mu(U_k) + 2^{-k}$ for all k. Then $\mu(V_{k+1}) < 2^{-k}$, and $A \in \bigcap_k V_{k+1}$, which shows that A is not 1-random relative to $B^{(n-1)}$.

Conversely, suppose that A is not 1-random relative to $B^{(n-1)}$, and let $U_1, U_2, \ldots$ be a $B^{(n-1)}$-uniformly $\Sigma^{0,B^{(n-1)}}_1$ classes with $\mu(U_k) < 2^{-k}$ and $A \notin \bigcap_k U_k$. Each U_k is also a

$\Sigma_n^{0,B}$ class, and B can uniformly transform an index of U_k into an index of a $\Sigma_n^{0,B}$ class V_k with $V_k = U_k$. Then because $A \notin \bigcap V_k$, we have that A is not n-random relative to B. □

There is a third notion of randomness which we will use in the paper, which arises by replacing the decreasing sequence of $\Sigma_n^{0,B}$ classes with a single $\Pi_n^{0,B}$ class of measure zero.

Definition 1.1.10 ([5, p. 286]). A set A is *weakly n-random* (or Kurtz n-random) relative to B if A is contained in every $\Sigma_n^{0,B}$ class of measure 1, or, equivalently, if A is not contained in any $\Pi_n^{0,B}$ class of measure 0.

If $U_1, U_2, \ldots$ is a sequence of $\Sigma_n^{0,B}$ classes, then $\bigcap_k U_k$ is a $\Pi_{n+1}^{0,B}$ class, so every weakly $(n+1)$-random set is n-random. Similarly, if V is a $\Pi_n^{0,B}$ class with measure zero, by Lemma 1.1.8 item 4 there exists a sequence of $B^{(n)}$-uniformly $\Sigma_{n-1}^{0,B}$ classes $U_1, U_2, \ldots$ (and thus a sequence of $B^{(n)}$-uniformly $\Sigma_n^{0,B}$ classes) with $\mu(U_1) < 2^{-n}$, so every n-random set is also weakly n-random.

We note that being weakly n-random is not the same as being weakly 1-random relative to $\emptyset^{(n-1)}$; see [5, p. 286] for a proof. But we can get a similar result if we allow for an extra quantifier.

Theorem 1.1.11 ([5, Corollary 7.2.5]). *Let $n \geq 2$. Then A is weakly n-random relative to B if and only if A is weakly 2-random relative to $B^{(n-2)}$.*

Proof. If A is not weakly 2-random relative to $B^{(n-2)}$, then A is contained in some $\Sigma_2^{0,B^{(n-2)}}$ class U of measure 1. Because U is also a $\Sigma_n^{0,B}$ class, this means that A is not weakly n-random relative to B.

Conversely, let $A \notin U$ where U is a $\Sigma_n^{0,B}$ class of measure 1. Write $U = \bigcup_k U_k$, where the U_k are uniformly $\Pi_{n-1}^{0,B}$ classes, and use Theorem 1.1.8 item 2 to B-uniformly get $\Pi_1^{0,B^{(n-2)}}$ classes $V_{k,j}$ such that $V_{k,j} \subseteq U_k$ and $\mu(V_{k,j}) > \mu(U_k) - 2^{-j}$. Then $V = \bigcup_{k,j}$ is a $\Sigma_2^{0,B^{(n-2)}}$ class of measure 1, and $A \notin V$, so A is not 2-random relative to $B^{(n-2)}$. □

1.1.4 Density of Sets

Definition 1.1.12 (see [9, Definition 1.1]). For $A \subseteq \mathbb{N}$, we define

$$\rho_n(A) = \frac{|A \restriction n|}{n}.$$

The *upper density* and *lower density* of A, denoted by $\overline{\rho}(A)$ and $\underline{\rho}(A)$, respectively, are the limits

$$\overline{\rho}(A) = \limsup_{n\to\infty} \rho_n(A) \qquad \text{and} \qquad \underline{\rho}(A) = \liminf_{n\to\infty} \rho_n(A)$$

If these two limits coincide, this common number is the *density* of A, denoted by $\rho(A)$. We say that A is *dense* if $\rho(A) = 1$ and *sparse* if $\rho(A) = 0$.

For example, $\rho(\mathbb{N}) = 1$, the density of the set of even numbers is $\frac{1}{2}$, the density of the set of primes is 0, and, for the set

$$\{n \mid \text{there exists a Hadamard matrix of order } n\}$$

it is still an open problem whether it has positive density or not [3].

Of course, $\rho(A)$ does not necessarily exist. If α, β are any two real numbers with $0 \leq \alpha \leq \beta \leq 1$, we can construct a set A with $\underline{\rho}(A) = \alpha$ and $\overline{\rho}(A) = \beta$ by stages. Start defining $A(0) = 0$, so that at the beginning of stage s we already defined $A \restriction s$. Then alternate between defining $A(s) = 1$ until $\rho_s(A) \geq \beta$, and $\rho_s(A) = 0$ until $\rho_s(A) \leq \alpha$.

This notion of density is natural, and most of our theorems will be stated using this definition. However, for proofs, it will be sometimes more convenient to use the following variant.

Definition 1.1.13 ([9, proof of Theorem 5.9]). Let $J_k = [2^k, 2^{k+1}) \cap \mathbb{N}$.[3] If $A \subseteq \mathbb{N}$, we

3. The definition of J_k in [9] is slightly different, namely, $J_k = [2^k - 1, 2^{k+1} - 1) \cap \mathbb{N}$.

define

$$d_k(A) = \frac{|A \cap J_k|}{2^k},$$

and, analogously to the definition of ρ, we define

$$\underline{d}(A) = \liminf_{k\to\infty} d_k(A) \qquad \text{and} \qquad \overline{d}(A) = \limsup_{k\to\infty} d_k(A)$$

and define $d(A)$ to be the common limit if it exists.

The values of $\overline{\rho}$ and $\overline{d}$ may differ, but not by much.

Lemma 1.1.14 ([9, Lemma 5.10]). For all sets $A \subseteq \mathbb{N}$, we have

$$\frac{\underline{d}(A)}{2} \leq \underline{\rho}(A) \qquad \text{and} \qquad \frac{\overline{d}(A)}{2} \leq \overline{\rho}(A) \leq 2\overline{d}(A).$$

Proof. For all k, we have

$$\begin{aligned} d_k(A) &= \frac{|A \cap J_k|}{2^k} \\ &\leq \frac{|A \restriction (2^{k+1})|}{2^k} \\ &= 2\rho_{2^{k+1}}(A). \end{aligned}$$

Since the numbers $\rho_{2^{k+1}}(A)$ are a subsequence of the numbers $\rho_k(A)$, we have

$$\limsup_{k\to\infty} \rho_{2^{k+1}}(A) \leq \overline{\rho}(A),$$

so taking lim sup on both sides of the previous inequality gives

$$\overline{d}(A) \leq 2\overline{\rho}(A).$$

For the other direction, let $\varepsilon > 0$ be given, and let N be large enough that $d_j(A) < \overline{d}(A)+\varepsilon$

for all $j > N$. For $j \leq N$ we can use the obvious bound $d_j(A) < \overline{d}(A) + \varepsilon + 1$, giving

$$\begin{aligned}
\rho_{2^{k+1}-1}(A) &= \frac{|\{0\} \cap A| + |J_0 \cap A| + \cdots + |J_k \cap A|}{2^{k+1}} \\
&\leq \frac{1 + |J_0 \cap A| + \cdots + |J_k \cap A|}{2^{k+1}} \\
&= \frac{1 + 2^0 d_0(A) + 2^1 d_1(A) + \cdots + 2^k d_k(A)}{2^{k+1}} \\
&\leq \frac{1 + 2^0(\overline{d}(A) + \varepsilon + 1) + \cdots + 2^N(\overline{d}(A) + \varepsilon + 1)}{2^{k+1}} \\
&\qquad + \frac{2^{N+1}(\overline{d}(A) + \varepsilon) + \cdots + 2^k(\overline{d}(A) + \varepsilon)}{2^{k+1}} \\
&= \frac{2^{N+1}}{2^{k+1}} + \overline{d}(A) + \varepsilon.
\end{aligned}$$

As N is fixed and ε was arbitrary, this gives

$$\limsup_{k\to\infty} \rho_{2^{k+1}}(A) \leq \overline{d}(A). \tag{1.1}$$

Now, if $2^k \leq n < 2^{k+1}$, then

$$\begin{aligned}
\rho_n(A) &= \frac{|A \restriction n|}{n} \\
&= \frac{2^{k+1}}{n} \cdot \frac{|A \restriction n|}{2^{k+1}} \\
&\leq 2 \cdot \frac{|A \restriction n|}{2^{k+1}} \\
&\leq 2 \cdot \frac{|A \restriction (2^{k+1})|}{2^{k+1}} \\
&= 2\rho_{2^{k+1}}(A).
\end{aligned}$$

Taking lim sup on both sides and combining with inequality 1.1, we get

$$\overline{\rho}(A) \leq 2\overline{d}(A).$$

Let $\varepsilon > 0$ be given. Analogously to the argument that showed inequality 1.1, we can let N be so that $d_j(A) > \underline{d}(A) - \varepsilon$ for all $j > N$, giving

$$\rho_{2^{k+1}}(A) \geq -\frac{2^{N+1}}{2^{k+1}} + \underline{d}(A) - \varepsilon.$$

If $2^k \leq n < 2^{k+1}$, then

$$\begin{aligned}\rho_n(A) &= \frac{|A \restriction n|}{n} \\ &\geq \frac{|A \restriction (2^k)|}{n} \\ &= \frac{2^k}{n}\rho_{2^k}(A) \\ &\geq \frac{1}{2}\rho_{2^k}(A),\end{aligned}$$

which, combining both inequalities and taking lim inf, gives

$$2\underline{\rho}(A) = 2\liminf_{n\to\infty} \rho_n(A) \geq \liminf_{k\to\infty} \rho_{2^k}(A) \geq \liminf_{k\to\infty} d_k(A) = \underline{d}(A). \qquad \square$$

It is false that $\underline{\rho}(A) \leq 2\underline{d}(A)$. If there are infinitely many k such that $J_k \cap A = \emptyset$, then $\underline{d}(A) = 0$, but if the k are sufficiently spaced out and $A(n) = 1$ everywhere else, then we can make $\underline{\rho}(A) = \frac{1}{2}$.

The most important application of this lemma is the following equivalence.

Corollary 1.1.15. Let $A \subseteq \mathbb{N}$. The following are equivalent.

1. $\rho(A) = 0$

2. $\overline{\rho}(A) = 0$

3. $d(A) = 0$

4. $\overline{d}(A) = 0$

5. $\rho(\overline{A}) = 1$

6. $\underline{\rho}(\overline{A}) = 1$

7. $d(\overline{A}) = 1$

8. $\underline{d}(\overline{A}) = 1$

Proof. If $\overline{d}(A) = 0$, then $\overline{\rho}(A) \leq 2\overline{d}(A) = 0$. Conversely, if $\overline{\rho}(A) = 0$, then $\overline{d}(A) \leq 2\overline{\rho}(A) = 0$, so 2 and 4 are equivalent.

The equivalence between 1 and 2, and between 3 and 4, is obvious.

Finally, the fact that $d_k(\overline{A}) = 1 - d_k(A)$ and $\rho_n(\overline{A}) = 1 - \rho_n(A)$ directly shows that 1 and 5, 2 and 6, 3 and 7, and 4 and 8 are equivalent. □

It is possible to have $\overline{d}(B)$ defined, but have $\overline{\rho}(B)$ undefined. For example, let B contain all the numbers between 2^n and $2^n + 2^{n-1}$ for even n, and all the numbers between $2^n + 2^{n-1}$ and 2^{n+1} for odd n. Then $d(B) = \frac{1}{2}$, $\underline{\rho}(B) = \frac{1}{3}$, and $\overline{\rho}(B) = \frac{2}{3}$.

Interestingly, the other direction does not occur.

Proposition 1.1.16. If $\rho(B)$ is defined, then so is $d(B)$, and the two values are the same.

Proof. Given $\varepsilon > 0$, let N be large enough that

$$\left| \frac{|\{k \in B : k < n\}|}{n} - \rho(B) \right| < \varepsilon$$

for all $n > N$. If $t > \log_2 N$, then the equation above is true for $n = 2^t$ and $n = 2^{t+1}$, so

$$\begin{aligned}
&\left| \frac{\left|\{k \in B : 2^t \leq k < 2^{t+1}\}\right|}{2^t} - \rho(B) \right| = \\
&\qquad = \left| \frac{\left|\{k \in B : k < 2^{t+1}\}\right|}{2^t} - \frac{\left|\{k \in B : k < 2^t\}\right|}{2^t} - 2\rho(B) + \rho(B) \right| \\
&\qquad \leq \left| \frac{\left|\{k \in B : k < 2^{t+1}\}\right|}{2^t} - 2\rho(B) \right| + \left| \frac{\left|\{k \in B : k < 2^t\}\right|}{2^t} - \rho(B) \right| \\
&\qquad < 2\varepsilon + \varepsilon = 3\varepsilon.
\end{aligned}$$

Computing $\limsup_t$ and $\liminf_t$ then gives

$$\rho(B) - 3\varepsilon \leq \underline{d}(B) \leq \overline{d}(B) \leq \rho(B) + 3\varepsilon,$$

whence $d(B) = \rho(B)$. □

1.1.5 Distances Between Sets and Functions

Using symmetric differences, we can define a notion of distances between sets.

Definition 1.1.17. If A and B are sets, the numbers $d(A, B)$, $\overline{d}(A, B)$, and $\underline{d}(A, B)$ are defined by[4]

$$d(A, B) = d(A \triangle B), \qquad \overline{d}(A, B) = \overline{d}(A \triangle B), \qquad \underline{d}(A, B) = \underline{d}(A \triangle B).$$

(Of course the first is only defined if the second and third are equal.) If n is a number, we also define $d_n(A, B) = d_n(A \triangle B)$.

It will be useful to extend this definition to partial functions as follows.

4. Our definition of $\overline{d}(A, B)$ matches the definition of $D(A, B)$ from [9, Definition 1.8]. We use a slightly different notation because we want to extend it to partial functions in Definition 1.1.18 below.

Definition 1.1.18. If $f, g : \{0.1\}^* \to \{0,1\}^*$ are partial functions, the numbers $\overline{d}(f,g)$ and $\overline{D}(f,g)$ are defined by

$$\overline{d}(f,g) = \overline{d}\big(\{x \in \{0,1\}^* : f(x){\downarrow} \wedge g(x){\downarrow} \wedge f(x) \neq g(x)\}\big),$$
$$\overline{D}(f,g) = \overline{d}\big(\{x \in \{0,1\}^* : f(x){\uparrow} \vee g(x){\uparrow} \vee f(x) \neq g(x)\}\big).$$

$\underline{d}(f,g)$, $d(f,g)$, $d_n(f,g)$, $\underline{D}(f,g)$, $D(f,g)$, and $D_n(f,g)$ are defined analogously.

That is, for $\overline{d}(f,g)$ we ignore the points at which either function is undefined, and for $\overline{D}(f,g)$ we pretend that $f(\sigma) \neq g(\sigma)$ if either function is undefined. In a sense, the first is an "optimistic" view of the distance, as $\overline{d}(f,g)$ is the smallest value of $\overline{d}(\hat{f},\hat{g})$ across all possible total extensions $\hat{f}$ of f and $\hat{g}$ of g; and the second is a "pessimistic" view of the distance, as $\overline{D}(f,g)$ is the largest possible such value.

Note that $\overline{d}(A,B) = \overline{d}(\chi_A, \chi_B) = \overline{D}(\chi_A, \chi_B)$.

1.1.6 *Asymptotic Notions of Computability*

In this section we will explore some relaxations of the notion of computability. Traditionally, to consider that a Turing machine M solves a certain problem, we demand for $M(n)$ to be defined and the correct answer for all inputs n. We will relax this restriction to require $M(n)$ to be defined and correct only for densely many n. The various definitions will vary on how $M(n)$ is required to behave for the other n.

Generic Computability

In this paper we will mostly be concerned with the case where $M(n)$ may not converge, but must be correct where defined.

Definition 1.1.19 ([12, Definition 1.4]). A *generic description* of a function f is a partial function g whose domain has density 1 and $g(n) = f(n)$ wherever g is defined. If g

is partial computable, we say that f is *generically computable*. A set $A \subseteq \mathbb{N}$ is generically computable if its characteristic function is generically computable.

We will frequently identify a set with its indicator function; for example, we will say that "f is a generic description of A" when we mean that f is a generic description of the indicator function of A.

Example 1.1.20. We claim that no 1-random set is generically computable. Let g be any partial computable function with dense domain. In particular, the domain of g is infinite. Let $(a_0, b_0), (a_1, b_1), \ldots$ be a computable enumeration of the graph of g (so that $g(a_i) = b_i$ for all i), and define

$$V_k = \{A \mid A(a_i) = b_i \text{ for all } i \leq k\}.$$

Each V_k is a Σ_1^0 class of measure 2^{-k}, and $\bigcap_{k \in \mathbb{N}} V_k$ is the collection of all sets A whose characteristic function agrees with g where g is defined. Thus, every 1-random set disagrees with g somewhere. Since g was arbitrary, no 1-random set is generically computable. □

Example 1.1.21. For any set $A \subseteq \mathbb{N}$, let $B = \{2^n \mid n \in A\}$. The set B is a "sparsification" of the set A; they are Turing-equivalent. Nevertheless, the function f such that $f(n) = 0$ if n is not a power of 2, and $f(n)\uparrow$ otherwise, is (partial) computable and a generic description of B. Hence the set B will always be generically computable, regardless of A. This shows that every Turing degree contains a generically computable set. □

Example 1.1.22. Let $A \subseteq \mathbb{N}$ be a noncomputable set, and define $B = \bigcup_{k \in A} J_k$, where J_k is as in Definition 1.1.13. Suppose that B were generically computable, and let f be a witness (a generic description of B which is partial computable). By Proposition 1.1.15, for all sufficiently large k, the function f is defined for most $n \in J_k$. So, to compute whether $k \in A$ or not, just try to compute $f(n)$ for all $n \in J_k$, in an interleaved fashion; it must be defined for at least one $n \in J_k$, and as soon as $f(n)$ is calculated for this n, we know that this is the value of $A(k)$.

That would make A computable, a contradiction. Hence every noncomputable set is Turing-equivalent to a set which is not generically computable. □

Coarse Computability

An alternative notion arises if we demand for $M(n)$ to always be defined, but allow the answer to be wrong sometimes.

Definition 1.1.23. A *coarse description* of a total function[5] f is a total function g such that the set $\{n \mid f(n) = g(n)\}$ has density 1. If g is computable, we say that f is *coarsely computable*. A set $A \subseteq \mathbb{N}$ is coarsely computable if its characteristic function is coarsely computable.

Identifying sets with their characteristic functions, we will say that a set B is a coarse description of a set A if $\mathbb{1}_B$ is a coarse description of $\mathbb{1}_A$, or equivalently, if the symmetric difference $A \triangle B$ has density 0.

Example 1.1.24. If A and B are as in Example 1.1.21, then the empty set is a coarse description of B, so all sets are Turing-equivalent to a coarsely computable set.

If A and B are in Example 1.1.22 and B is coarsely computable, let C be computable such that $B \triangle C$ is sparse. Then for all sufficiently large k we have $d_k(B \triangle C) < \frac{1}{3}$, so either $d_k(C) < \frac{1}{3}$ (in which case $k \notin A$) or $d_k(C) > \frac{2}{3}$ (in which case $k \in A$). Hence C computes A. So every noncomputable set is Turing-equivalent to a non-coarsely computable set. □

Example 1.1.25. We can construct a set which is coarsely computable, but not generically computable, through diagonalization, as follows. Let R_e be the set

$$R_e = \{n \in \mathbb{N} : 2^e \mid n \wedge 2^{e+1} \nmid n\}.$$

5. Coarse computability is usually defined only for sets (e.g. [8, Definition 1.1]). We extend the definition for functions for consistency with Definition 1.1.19.

So R_0 is the set of odd numbers, R_1 is the set of even numbers not divisible by four, R_2 is the set of multiples of four which are not multiples of eight, and so on. Note that $\rho(R_i) = 2^{-e-1}$. We will construct a non-generically computable c.e. set A such that $|A \cap R_e| \leq 1$ for all e. So, for each e, all but finitely many elements of A are contained in R_e, so $\rho(A) = 0$, whence the computable set $\emptyset$ is a coarse description of A.

To achieve this, for each e, simulate the eth Turing machine on every number $n \in R_e$, and wait for it to halt in any of those inputs. If it halts on n, set $n \in A$ if the machine rejected it, and leave $n \notin A$ otherwise, and stop simulating the eth Turing machine (we are done with it). This finishes the construction.

Clearly A is c.e., and if the eth machine halts in some $n \in R_e$, by construction the function it is computing cannot be a generic description of A (as it is getting the value of the nth input wrong). But if the machine never halts for any $n \in R_e$, then the density of its domain is at most $1 - 2^{-e-1}$, so it is also not a generic description. Hence A is coarsely computable but not generically computable. □

Example 1.1.26. Conversely, we can construct a c.e. set A which is generically computable, but not coarsely computable. Recall that the sets $J_k = [2^k, 2^{k+1}) \cap \mathbb{N}$ form a partition of $\mathbb{N} \setminus \{0\}$. We will have a countable collection of strategies S_e, one for each Turing machine e. S_e will try to prevent the eth Turing machine from coarsely computing A.

We will construct a partial function f (which will be a generic description of $\mathbb{1}_A$) by stages. Strategy S_e starts acting on stage s, and "claims" the smallest unclaimed J_k for itself. If n is one of the last $2^k - 2^{k-e}$ elements of J_k, we define $f(n) = 0$. Otherwise, S_e simulates the eth Turing machine on n for s steps. If the Turing machine does not halt on all the remaining elements, S_e does nothing, and will try again on the next stage. Otherwise, define $f(n)$ to be the opposite value of what the Turing machine answered, and claim the next smallest unclaimed J_k for the next stage. Finally, let $A = f^{-1}(1)$. This finishes the construction.

If strategy S_e claims only finitely many intervals J_k, it means that the eth Turing machine fails to halt in some element of the last claimed interval. Hence the eth Turing machine cannot coarsely compute A.

If strategy S_e claims infinitely many intervals J_k, let B be the set computed by the eth Turing machine. If n is one of the first 2^{k-e} elements of J_k, by construction we have $A(n) = f(n) \neq B(n)$, so $d_k(A \triangle B) \leq 1 - 2^{-e}$. Since there are infinitely many such k, the set B is not a coarse description of A.

This shows that A is not coarsely computable. To show that A is generically computable, note that if S_e claims infinitely many intervals J_k, then $f(n)\downarrow = A(n)$ for all n in each of these J_k; and if S_e claims only finitely many intervals, it may leave $f(n)$ undefined for up to 2^{k-e} elements in J_k if it is the last claimed interval, but after that S_e stops claiming intervals at all. Hence if D is the domain of f, then $d_k(D) < 1 - 2^{-e}$ for finitely many k. So D is dense, which shows that f is a partial computable generic description of A. □

The following proposition will be useful in Section 1.2.2.

Proposition 1.1.27. If the set X is Church-stochastic, then X is not coarsely computable.

Proof. Let A be any computable set. We will show that A is not a coarse description of X.

Because $\rho(A \cup \overline{A}) = 1$, either $\overline{\rho}(A) > 0$ or $\overline{\rho}(\overline{A}) > 0$; without loss of generality, assume the former. Let $f : \mathbb{N} \to \mathbb{N}$ be the increasing function which satisfies

$$A = \{f(0), f(1), f(2), \ldots\}.$$

Then f is a computable selection function, and because X is Church-stochastic,

$$\lim_{n\to\infty} \frac{|\{k < n \mid f(k) \in X\}|}{n} = \frac{1}{2}.$$

Let $\varepsilon > 0$ be given, and take N large enough that if $n \geq N$ then the limit above is within

ε of $\frac{1}{2}$. Further, take some $n > f(N)$ for which $\rho_n(A) > \overline{\rho}(A) - \varepsilon$, let m be the largest integer for which $f(m) < n$, and note $m \geq N$. Then

$$\begin{aligned}
\rho_n(A \setminus X) &= \frac{|\{k < n \mid k \in A \wedge k \notin X\}|}{n} \\
&= \frac{|\{j \leq m \mid f(j) \in A \wedge f(j) \notin X\}|}{n} \\
&= \frac{|\{j \leq m \mid f(j) \notin X\}|}{n} \\
&\geq \left(\frac{1}{2} - \varepsilon\right) \frac{m+1}{n} \\
&= \left(\frac{1}{2} - \varepsilon\right) \frac{|\{k < n \mid k \in A\}|}{n} \\
&\geq \left(\frac{1}{2} - \varepsilon\right) (\overline{\rho}(A) - \varepsilon).
\end{aligned}$$

Because $\varepsilon > 0$ was arbitrary, this means that $\overline{\rho}(A \setminus X) \geq \overline{\rho}(A)/2$. So A and X disagree on a set with positive upper density, which means that A cannot be a coarse description of X. □

Dense and Effectively Dense Computability

Generic computability arises when the Turing machine M is allowed to not halt sometimes, and coarse computability arises when M is allowed to make a few mistakes. If both of these relaxations are taken together, we get dense computability.

Definition 1.1.28 ([2, Definition 1.3]). A *dense description* of a function f is a function g such that the set

$$\{n \mid g(n)\downarrow \wedge g(n) = f(n)\}$$

has density 1. If g is partial computable, we say that f is *densely computable*. A set is densely computable if its characteristic function is densely computable.

Clearly all generically computable sets and all coarsely computable sets are densely computable. It is interesting to note, however, that there are sets which are neither generic nor coarsely computable, but which are densely computable. In a sense, this whole is larger than the sum of its two parts.

Example 1.1.29. Let A be the set constructed in Example 1.1.26 and B be the set constructed in Example 1.1.25. Then the join $A \oplus B$ of A and B is neither generically nor coarsely computable, but it is densely computable. □

If we disallow both relaxations (we still demand the Turing machine to always halt, and we forbid it from giving the wrong answer), we can still construct an asymptotic notion of computability which is weaker than just being computable. We allow the machine to return the special symbol □, which means that the machine does not know the answer.

Definition 1.1.30 ([2, Definition 1.4]). An *effective dense description* of a function $f : \mathbb{N} \to \mathbb{N}$ is a total function $g : \mathbb{N} \to \mathbb{N} \cup \{\Box\}$ such that, for all n, we have either $f(n) = g(n)$ or $f(n) = \Box$, and the set

$$\{n \mid g(n) \neq \Box\}$$

has density 1. If g is computable, then f is *effectively densely computable.* A set is effectively densely computable if its characteristic function is effectively densely computable.

Of course, all effectively densely computable sets are also generically computable and coarsely computable, and a modification of Example 1.1.21 shows that there are effectively densely computable sets which are not computable.

Example 1.1.31. We will modify Example 1.1.26 to construct a set which both generically and coarsely computable, but not effectively densely computable.

Strategy S_e will again simulate the eth Turing machine on the beginning of its claimed intervals. Suppose S_e claimed interval J_k, and that the eth Turing machine computes the function f_e. First we again define $A(n) = 0$ for all of the last $2^k - 2^{k-e}$ elements n of J_k.

Once the machine halts on all of the 2^{k-e} starting elements of J_k, if $f_e(n) \neq \Box$ for any of those n, we can define $A(n) = 1 - f_e(n)$ for the smallest one, $A(n) = 0$ for the rest, and S_e can stop claiming intervals. Otherwise, we have $f_e(n) = \Box$ for all of the 2^{k-e} starting elements of J_k; define $A(n) = 0$ for all $n \in J_k$ and claim another interval. This finishes the construction.

If strategy S_e claims infinitely many intervals, it means that $f_e(n) = \Box$ in all the beginning 2^{k-e} elements of J_k for infinitely many k, so f_e is not an effective dense description of A. If strategy S_e claims only finitely many intervals, it means that f_e is undefined in some of the claimed intervals, which thus mean that the eth Turing machine does not effectively densely compute A. Therefore, A is not effectively densely computable.

Finally, the same reasoning as in Example 1.1.26 shows that A is indeed generically computable; and $A(n) = 1$ for at most one $n \in J_k$, so $\emptyset$ is a coarse description of A, which shows that A is also coarsely computable. $\Box$

The following proposition will be useful in Section 1.2.3.

Proposition 1.1.32. If the set X is Church-stochastic, then X is not densely computable.

Proof. Let $g : D \subseteq \mathbb{N} \to \{0, 1\}$ be a partial computable function with dense domain. We will show that g is not a dense description of X.

Because $\rho(D) = 1$, at least one of the sets

$$\{n \mid g(n)\downarrow = 1\} \quad \text{and} \quad \{n \mid g(n)\downarrow = 0\}$$

has positive upper density. Assume it is the former without loss of generality.

Let $q \in \mathbb{Q}$ satisfy

$$\frac{q}{2} \leq \overline{\rho}(\{n \mid g(n)\downarrow = 1\}) \leq q.$$

Define the computable set B as follows.

Start with $k = 0$. On stage s, evaluate $g(n)$ for $n \in J_k, J_{k+1}, \ldots$ and wait until, for some $i \geq k$, we have

$$|\{n \in J_i \mid g(n)\downarrow = 1\}| > q2^{i-2}.$$

Because $\overline{\rho}(\{n \mid g(n)\downarrow = 1\}) > q/2$, by Lemma 1.1.14, we have $\overline{d}(\{n \mid g(n)\downarrow = 1\}) > q/4$, so for each fixed k there is some $i \geq k$ for which the condition above is satisfied. Then define B on J_i to be the first $q2^{i-2}$ elements of $\{n \in J_i \mid g(n)\downarrow = 1\}$, and $B(n) = 0$ if $n \in J_k \cup J_{k+1} \cup \cdots \cup J_{i-1}$. Set $k = i$ and go to the next stage.

Again by Lemma 1.1.14, this means that $\overline{\rho}(B) > q/8$, so B has a positive upper density. The argument now follows the proof of Theorem 1.1.27 to show that B and X disagree in a set of positive upper density. Therefore g is not a dense description of X. □

1.1.7 Enumeration Operators and Reducibilities

In order to talk about generic degrees, we have to define "generic reduction" between two sets, but simply using Turing reductions will not work.

Generic descriptions are functions. To use functions as oracles, we will consider their graphs; formally, we let $\langle \cdot, \cdot \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be any fixed pairing function (say, Cantor's pairing function), and define

$$\operatorname{graph}(f) = \{\langle x, f(x) \rangle \mid x \in \operatorname{Dom}(f)\}.$$

The problem of using graph(f) directly as an oracle for a Turing machine is that the machine can query whether a certain number is in the domain of f or not. For example, for any set A, we can define $f(2^k) = 0$ if $k \in A$, leave $f(2^k)$ undefined if $k \notin A$, and set $f(n) = 0$ if n is not a power of 2. Then f is a generic description of the empty set, but it is easy to see that $\operatorname{graph}(f) \geq_{\mathrm{T}} A$. Hence for every set A, there is a generic description of $\emptyset$ which computes it.

The issue is that the function directly queries whether a pair is in graph(f) or not. To prevent this, we will use enumeration operators instead.

Definition 1.1.33 ([18, Section 9.7]). Let $W \subseteq \mathbb{N}^* \times \mathbb{N}$ be a set of pairs (F, k), where $k \in \mathbb{N}$ and F is a finite subset of $\mathbb{N}$. Then W is an *enumeration operator* if the set of codes for the elements in W is a c.e. set. (We will identify W with the set of codes.) For any set A, we define W^A by

$$W^A = \{k \mid (F, k) \subseteq W \text{ for some } F \subseteq A\},$$

and we say that W^A is enumeration reducible to A.

The name comes from the idea of a program that can transform any enumeration of A into an enumeration of a set B. Since the elements of A may be presented out of order, all the algorithm ever sees are finite subsets $F \subseteq A$. If the algorithm decides to enumerate k into B, we place the pair (F, k) into W. The result is that $B = W^A$.

As a direct consequence of this definition, if A is a c.e. set, then any W^A is also c.e. for any enumeration operator W.

Proposition 1.1.34. For infinite sets, the set W may be taken to be computable; that is, for every enumeration operator W there exists a computable enumeration operator V such that $W^A = V^A$ for all infinite sets A.

Proof. To decide whether $(F, k) \in V$ or not, let M be a Turing machine which recognizes W and simulate M on all pairs (F', k) with $F' \subseteq F$ for $|F|$ steps. Place $(F, k) \in V$ if M accepts any pair, and define $(F, k) \notin V$ otherwise. This makes V computable.

For any k, if $k \in V^A$, it means that $(F, k) \in V$ for some $F \subseteq A$, which means that $(F', k) \in W$ for some $F' \subseteq F$. Hence $(F', k) \in W$ for some $F' \subseteq A$, so $k \in W^A$.

Conversely, if A is infinite and $k \in W^A$, this means that $(F, k) \in W$ for some $F \subseteq A$. Suppose it takes s steps for M to accept (F, k). Let $S \subseteq A$ be any subset of A with s elements. Then, by construction, $(F \cup S, k) \in V$, and $F \cup S \subseteq A$. Therefore, $k \in V^A$. □

Proposition 1.1.35. The set W may be assumed to be monotonic; that is, for every enumeration operator W there exists an enumeration operator V such that $W^A = V^A$ for all sets A, and if $(F, k) \in V$ then $(F', k) \in V$ for all $F' \supset F$.

Proof. Define

$$V = \{(F, k) \mid \exists F' \subseteq F[(F', k) \in W]\}.$$

As $V \supseteq W$, we have $W^A \subseteq V^A$ for all A. Furthermore, if $k \in V^A$, then $(F, k) \in V$ for some $F \subseteq A$, so $(F', k) \in W$ for some $F' \subseteq F$, so $(F', k) \in W$ for some $F' \subseteq A$, which means $k \in W^A$. Hence $V^A = W^A$. □

An observation which will be important below is that we may compose enumeration operators.

Proposition 1.1.36 ([18, Theorem XX]). For any enumeration operators V and W there exists an enumeration operator U such that $U^A = V^{W^A}$ for all sets A.

Proof. Replacing V and W if needed, we may assume that they are monotonic. Define U via

$$U = \left\{(F, k) \;\middle|\; \exists F'\big[(F', k) \in V \wedge \forall l \in F'[(F, l) \in W]\big]\right\}$$

U is an enumeration operator because all of the candidate F' must be finite.

Let $k \in V^{W^A}$. Then $(F, k) \in W$ for some set $F \subseteq W^A$. For each $l \in F$ we have $l \in W^A$, so there exists some $F'_l \subseteq A$ such that $(F'_l, l) \in W$. Define $F' = \bigcup_{l \in F} F'_l$. Because W is monotonic, we also have $(F', l) \in W$ for all $l \in F$. Then $(F, k) \in U$, by construction, so $k \in U^A$.

Conversely, let $k \in U^A$, so $(F, k) \in U$ for some $F \subseteq A$. By definition of U, this means that $(F', k) \in V$ for some F' such that $(F, l) \in W$ for all $l \in F'$. Because $F \subseteq A$, this means that $l \in W^A$ for all $l \in F'$, so $F' \subseteq W^A$. Hence $k \in V^{W^A}$. □

Dense and Generic Reducibility

Definition 1.1.37 (see [2, Definition 3.2]). A set A is *nonuniformly generically reducible* to a set B, denoted by $A \leq_{\mathrm{ng}} B$, if for every generic description f of A, there exists a generic description g of B such that f is enumeration reducible to g.

Definition 1.1.38 (see [2, Definition 3.2]). A is *uniformly generically reducible* to B, denoted by $A \leq_{\mathrm{ug}} B$, if there is a single enumeration operator W such that W^g is a generic description of A for all generic descriptions g of B.

A direct consequence of Proposition 1.1.36 is that both $\leq_{\mathrm{ng}}$ and $\leq_{\mathrm{ug}}$ are transitive, and therefore the definitions below make sense.

Definition 1.1.39. The *uniform* (resp. *nonuniform*) *generic degree* of a set A is the collection of sets B such that $A \leq_{\mathrm{ug}} B$ and $B \leq_{\mathrm{ug}} A$ (resp. $A \leq_{\mathrm{ng}} B$ and $B \leq_{\mathrm{ng}} A$).

For example, the class of generically computable sets is the generic degree of $\emptyset$. In this case, the uniform and the nonuniform generic degrees coincide, but this is not the case in general.

Example 1.1.40. Let A be any set, and define the sets $\mathcal{R}(A)$ and $\tilde{\mathcal{R}}(A)$ by [2, Section 5]

$$\mathcal{R}(A) = \bigcup_{k \in A} R_k$$

and

$$\tilde{\mathcal{R}}(A) = \bigcup_{k \in A} J_k.$$

Clearly, $A \geq_{\mathrm{T}} \mathcal{R}(A)$ and $A \geq_{\mathrm{T}} \tilde{\mathcal{R}}(A)$.

Because each R_k has positive density, if f is any generic description of $\mathcal{R}(A)$, then $f(n)$ is defined for some $n \in R_k$. Since generic descriptions must be correct wherever defined, we can uniformly reconstruct A from any generic description of $\mathcal{R}(A)$ by simply waiting for some $n \in R_k$ to be in the domain of f. Because $A \geq_{\mathrm{T}} \tilde{\mathcal{R}}(A)$, this means that we can compute $\tilde{\mathcal{R}}(A)$ from any generic description of $\mathcal{R}(A)$, whence

$$\tilde{\mathcal{R}}(A) \leq_{\mathrm{ug}} \mathcal{R}(A).$$

If f is any generic description of $\tilde{\mathcal{R}}(A)$, then for all sufficiently large k, we have that $f(n)$ is defined for at least one $n \in J_k$. Hence we can nonuniformly compute $\mathcal{R}(A)$ from f by "memorizing" the values of $A(k)$ for the finitely many non-sufficiently large k, and just waiting for $f(n)$ to be defined for some $n \in J_k$ if k is sufficiently large. Thus

$$\mathcal{R}(A) \leq_{\mathrm{ng}} \tilde{\mathcal{R}}(A).$$

Finally, choose A to be a non-autoreducible set [19, Exercise 7.3.8] (that is, there is no Turing reduction Φ such that $A(k) = \Phi^{A \setminus \{k\}}(k)$ for all k). We claim that

$$\mathcal{R}(A) \not\leq_{\mathrm{ug}} \tilde{\mathcal{R}}(A)$$

in this case. Indeed, suppose that there exists an enumeration operator W which witnesses $\mathcal{R}(A) \leq_{\mathrm{ug}} \tilde{\mathcal{R}}(A)$. For any k, define $f(n) = A(j)$ if $n \in J_j$ and $j \neq k$, and let $f(n)$ undefined otherwise. Because f is a generic description of $\tilde{\mathcal{R}}(A)$, we must have $W^f(n)\downarrow = A(k)$ for some $n \in R_k$. As f can be uniformly computed from $A \setminus \{k\}$, this shows that $A(k)$ can be uniformly computed from $A \setminus \{k\}$, which shows that A is autoreducible, a contradiction.

Hence the sets $\mathcal{R}(A)$ and $\tilde{\mathcal{R}}(A)$ are nonuniformly generically equivalent, but not uniformly generically equivalent. □

We will use a similar definition for dense reducibility.

Definition 1.1.41 ([2, Definition 3.2]). A set A is *nonuniformly densely reducible* to a set B, denoted by $A \leq_{\mathrm{nd}} B$, if for every dense description f of A, there exists a dense description g of B such that f is enumeration reducible to g. A is *uniformly generically reducible* to B, denoted by $A \leq_{\text{ud}} B$, if there is a single enumeration operator W such that W^g is a dense description of A for all dense descriptions g of B.

Again, Proposition 1.1.36 guarantees that the definition below makes sense.

Definition 1.1.42. The *uniform* (resp. *nonuniform*) *dense degree* of a set A is the collection of sets B such that $A \leq_{\text{ud}} B$ and $B \leq_{\text{ud}} A$ (resp. $A \leq_{\text{nd}} B$ and $B \leq_{\text{nd}} A$).

The distinction between uniform and nonuniform is again necessary; see [2, Corollary 5.8] for an example of a pair of sets which are nonuniformly densely equivalent, but not uniformly densely equivalent.

Coarse and Effectively Dense Reducibilities

Coarse and effectively dense descriptions of sets must be defined everywhere, so there is no need to use enumeration operators.

Definition 1.1.43 (see [2, Definition 3.1]). A set A is *nonuniformly coarsely reducible* to a set B, denoted by $A \leq_{\text{nc}} B$, if every coarse description of B computes a coarse description of A. The set A is *uniformly coarsely reducible* to B, denoted by $A \leq_{\text{uc}} B$, if there is a single Turing functional Φ such that Φ^C is a coarse description of A for all coarse descriptions C of B.

The definition of uniform and nonuniform coarse degrees is analogous to the previous definitions. See [8, Theorem 2.6] for an example of two nonuniformly coarsely equivalent sets which are not uniformly coarsely equivalent.

The definition for effectively dense degrees is analogous.

Definition 1.1.44 ([2, Definition 3.1]). A set A is *nonuniformly effectively densely reducible* to a set B, denoted by $A \leq_{\text{ned}} B$, if every effectively dense description of B computes an effectively dense description of A. The set A is *uniformly effectively densely reducible* to B, denoted by $A \leq_{\text{ued}} B$, if there is a single Turing functional Φ such that Φ^C is an effectively dense description of A for all effectively dense descriptions C of B.

The definition of uniform and nonuniform effectively degrees is analogous to the previous definitions. See [2, Corollary 5.3] for a pair of sets which are nonuniformly effectively densely equivalent, but which are not uniformly effectively densely equivalent.

1.1.8 Function Trees

This section defines function trees (using the same definition as in [16, Section V.5]), and states a theorem from [9, Corollary 5.11] (Lemma 1.1.47) in terms of function trees.

As noted below, function trees correspond to "traditional" trees with no isolated paths. We will use them when proving Theorem 2.3.5 essentially for notational convenience. If T is a tree without isolated paths and we are working on a string $\tau \in T$, we will frequently want to take extensions $\tau_0, \tau_1 \succcurlyeq \tau$ immediately after the first "bifurcation" after τ. That is, we first replace τ with the shortest extension $\hat{\tau}$ of τ in the tree such that both $\tau * 0$ and $\tau * 1$ are also strings in the tree (i.e. we navigated until the first "bifurcation" after τ in the tree, and every extension of τ is comparable with $\hat{\tau}$); then we let τ_0 be the extension of $\hat{\tau} * 0$ until the nearest bifurcation, and likewise for τ_1. If T were a function tree, then $\tau = T(u)$ for some string u, and τ_0, τ_1 are just called $T(u * 0)$ and $T(u * 1)$.

Definition 1.1.45 ([16, Definition V.5.1]). A *partial function tree* is a partial function $T : \{0,1\}^* \to \{0,1\}^*$ satisfying the following properties:

1. For all $v \preccurlyeq u$, if $T(u)$ is defined then $T(v)$ is also defined.

2. For all u such that $T(u)$ is defined, either $T(u * 0)$ and $T(u * 1)$ are both undefined, or both are defined to incomparable extensions of $T(u)$.

A function tree T can be made into a "traditional" tree by taking prefixes of strings in the image of T; that is, the set

$$\{\tau \mid \exists u[T(u)\downarrow \succcurlyeq \tau]\}$$

is a tree. Condition 2 implies that this tree has no isolated paths, and conversely, every tree without isolated paths can be converted into a partial function tree. We will sometimes identify the two notions.

Definition 1.1.46. We say that a partial function tree T *refines* a partial function tree $\hat{T}$ if for all u, if $T(u)$ is defined, then it equals $\hat{T}(v)$ for some v.

As functions, the definition above says that $\operatorname{Im} T \subseteq \operatorname{Im} \hat{T}$; as trees, T is a subtree of $\hat{T}$.

As an example, a function tree T always refines itself, and if we define $\hat{T}$ by setting $\hat{T}(\emptyset) = T(\emptyset)$ and $\hat{T}(0 * u) = T(1 * u), \hat{T}(1 * u) = T(0 * u)$ for all σ, then T and $\hat{T}$ refine each other. Intuitively, these two function trees represent the same "traditional tree", so ultimately it does not really matter which of those is our representative.

The following lemma will be used in the proof of Claim 2.3.5.8.

Lemma 1.1.47 ([10, Theorem 7.13]). There exists a total computable function tree T_0 such that if A, B are distinct paths on T_0 then $d(A, B) = 1/2$.

Remark 1.1.48. The original theorem states the result for ρ instead of d. However, due to Lemma 1.1.14, convergence under ρ and convergence under d are equivalent.

1.1.9 Effective Completeness Theorems

A metric space is complete if every converging Cauchy sequence has a limit. If we use d the metric from Definition 1.1.17, we can talk about convergence in the coarse equivalence classes. However it is fairly easy to construct a sequence of coarsely computable sets which converge to a set which is not coarsely computable. In a sense, the completeness of $(2^\omega, d)$ is not effective.

However, there are additional hypothesis that do ensure convergence to a coarsely computable set. We mention two such theorems.

Lemma 1.1.49 ([9, Corollary 5.11]). Let B be any set and suppose there exists a $\emptyset'$-computable function f such that, for each k, the number $f(k)$ is an index for a computable set $\Phi_{f(k)}$ such that $\overline{d}(B, \Phi_{f(k)}) < 2^{-k}$. Then B is coarsely computable.

Each $f(k)$ is, of course, a computable index for the computable set $\Phi_{f(k)}$; it is just the sequence $f(0), f(1), \ldots$ which may be only $\emptyset'$-computable. We cannot replace $\emptyset'$ with any non-Δ^0_2 set, though; see Proposition 2.4.1. In Section 2.4 we discuss this and versions of this theorem for the other asymptotic notions of computability. See also Open Problem 5.4.

Lemma 1.1.50 ([9, Theorem 5.12]). Let G be a Δ^0_2 1-generic set, let $A \leq_{\mathrm{T}} G$, and suppose that there exists a sequence of computable sets $B_0, B_1, \ldots$ such that $\overline{d}(A, B_k) < 2^{-k}$. Then A is coarsely computable.

Note that the sequence does not need to be uniform (unlike Lemma 1.1.49), but the set A itself must be computable by some Δ^0_2 1-generic set.

1.2 Minimal Pairs and Randomness

Informally speaking, if A and B are any two sets, we can ask whether they have some common "computational power". Clearly both A and B can compute any computable set, but is there anything more? Perhaps surprisingly, there are many pairs without any common computational power (besides the computable sets), which we call "minimal pairs".

We start with the Turing degrees.

1.2.1 Minimal Pairs in the Turing Degrees

Definition 1.2.1 ([19, p. 135]). The *upper cone* above a set A is the collection

$$\{B \mid B \geq_{\mathrm{T}} A\},$$

and the *lower cone* below A is the collection

$$\{B \mid B \leq_{\mathrm{T}} A\}.$$

The intersection of these two cones is the *Turing degree* of A.

For any two sets A and B, it is easy to see that their upper cones intersect; in fact, the intersection of the upper cones above A and B is exactly the upper cone above $A \oplus B$.

Trivially, the lower cones also intersect, because all lower cones contain the computable sets. And, in some cases, their intersection contain *only* the computable sets.

Definition 1.2.2 ([19, Definition 6.2.1]). A *minimal pair* for the Turing degrees is a pair of sets (A, B) such that neither A nor B are computable, but if $C \leq_{\mathrm{T}} A$ and $C \leq_{\mathrm{T}} B$, then C is computable.

That is, the intersection of the lower cones below A and B is the class of computable sets.

We will provide a proof of existence of minimal pairs based on the Lebesgue Density Theorem, using a "majority vote" argument.

Lemma 1.2.3 ([5, Theorem 8.12.6]). If A is noncomputable, then the upper cone above A has measure zero.

Proof. Let V be the upper cone above A, and assume it has positive measure.

For each e, define $V_e = \{B \mid \Phi_e^B = A\}$; that is, the class V_e contains all the oracles that e can use to compute A. Note that V is the union of all V_e, so if $\mu(V) > 0$ then $\mu(V_e) > 0$ for some e.

By the Lebesgue Density Theorem, there is some $\sigma \in 2^{<\omega}$ such that $\mu(V_e \cap [\![\sigma]\!]) > \frac{2}{3}\mu([\![\sigma]\!])$, so we can do a majority vote inside $[\![\sigma]\!]$: given n, compute $\Phi_e^\tau(n)[|\tau|]$ for each $\tau \succcurlyeq \sigma$, until either

- the measure of the τ such that $\Phi_e^\tau(n)[|\tau|]\downarrow = 0$ surpasses $\frac{1}{2}\mu([\![\sigma]\!])$, in which case we know that $A(n) = 0$; or
- the measure of the τ such that $\Phi_e^\tau(n)[|\tau|]\downarrow = 1$ surpasses $\frac{1}{2}\mu([\![\sigma]\!])$, in which case we know that $A(n) = 1$.

One of these two must happen, and when it does the majority vote is the correct answer.

Therefore, if the measure of V is nonzero, then A is computable. □

We have the following as an immediate consequence of this lemma.

Proposition 1.2.4. Each noncomputable set forms a minimal pair with measure-1 many sets.

Proof. Let A be noncomputable. There are countably many noncomputable sets $D \leq_T A$, and for each of those sets, the upper cone $V_D = \{C \mid C \geq_T D\}$ has measure zero. We claim that if B is noncomputable and not in any of these cones, then (A, B) forms a minimal pair for the Turing degrees.

Indeed, if $C \leq_T A, B$, then clearly $B \in V_C$; by construction, this is only possible if C is computable. So (A, B) is a minimal pair for the Turing degrees.

Since there are countably many $D \leq_T A$, the family of valid such B has measure 1. □

Hence, a simple application of Fubini's Theorem shows that the collection of pairs (A, B) which form a minimal pair for the Turing degrees has measure 1. Intuitively, this means that any two sets picked "at random" will be a minimal pair for the Turing degrees. In fact, we can show that a fairly low degree of randomness suffices.

Proposition 1.2.5 ([5, Corollary 8.12.4]). If A and B are relatively weakly 2-random, then A and B form a minimal pair for the Turing degrees.

Proof. Let C be a noncomputable set such that $C \leq_{\mathrm{T}} B$, and suppose for the sake of contradiction that $C \leq_{\mathrm{T}} A$. Let e satisfy $\Phi_e^A = C$, and define

$$\mathcal{S} = \{X \mid \forall n \exists s [\Phi_e^X(n)[s] = C(n)]\}.$$

Then $\mathcal{S}$ is a $\Pi_2^{0,B}$ class which is contained in the upper cone above C. By Lemma 1.2.3, the class $\mathcal{S}$ has measure 0. Since A is weak 2-random relative to B, this means that $A \notin \mathcal{S}$, a contradiction. □

1.2.2 Minimal Pairs in the Coarse Degrees

For coarse reducibility, we have a similar definition, though we have to pay attention to uniformity.

Definition 1.2.6. Two sets A and B form a *minimal pair for the uniform coarse degrees* if both are non-coarsely computable and if C is any set such that $C \leq_{\mathrm{uc}} A, B$, then C is coarsely computable.

Being a minimal pair for the nonuniform coarse degrees is defined similarly.

Because the relation $\leq_{\mathrm{uc}}$ has stricter requirements than $\leq_{\mathrm{nc}}$, if two sets form a minimal pair for the nonuniform coarse degrees then they also form a minimal pair for the uniform coarse degrees, so it suffices to show the former.

Lemma 1.2.7 ([8, Theorem 5.2]). If A is non-coarsely computable and X is weakly 3-random relative to A, then X does not compute any coarse description of A.

Proof. Suppose for the sake of contradiction that $A \leq_{\mathrm{nc}} X$, and let Φ be a Turing functional such that Φ^X is a coarse description of A. Define the class $\mathcal{P}$ by

$$\mathcal{P} = \{Y \mid \Phi^Y \text{ is a coarse description of } A\}.$$

For Y to be in $\mathcal{P}$, it must be the case that Φ^Y is total (which is a Π^0_2 property) and that $\lim_k \rho_k(\Phi^Y \triangle A) = 0$, which we can express as

$$\forall \varepsilon \exists K \forall s, k > K \Big[\big(\forall n < k (\Phi^Y(n)[s]\downarrow)\big) \implies \rho_k(\Phi^Y[s] \triangle A) < \varepsilon \Big],$$

which is a $\Pi^{0,A}_3$ property. Hence $\mathcal{P}$ is a $\Pi^{0,A}_3$ class.

Because X is weakly 3-random relative to A, it is not contained in any $\Pi^{0,A}_3$ class of measure 0; since $X \in \mathcal{P}$, we must have $\mu(\mathcal{P}) > 0$. Using the Lebesgue Density Theorem, there exists some σ such that $\mu(\mathcal{P} \cap [\![\sigma]\!]) > \frac{5}{6}2^{-|\sigma|}$. By replacing $\Phi^{\tau Y}$ with $\Phi^{\sigma Y}$ for all $|\tau| = |\sigma|$ (except if $\tau = X \restriction |\sigma|$), we may assume that $\mu(\mathcal{P}) > \frac{4}{5}$; that is, for more than $\frac{4}{5}$ of all Y, the set Φ^Y is a coarse description of A. (The exception of not replacing if $\tau = X \restriction |\sigma|$ exists solely to keep $X \in \mathcal{P}$, and it is not really needed in the rest of the argument.)

Define the set D as follows. Recall that $J_k = [2^k, 2^{k+1}) \cap \mathbb{N}$. For each k, find some integer s_k and a finite set S_k of s_k-sized strings such that $\Phi^\sigma(n)\downarrow$ for all $n \in J_k$ and all $\sigma \in S_k$, and $|S_k| > \frac{4}{5}2^{n_k}$. We know such set exists because $\mu(\mathcal{P}) > \frac{4}{5}$. Then pick a set $R_k \subseteq S_k$ such that $|R_k| > \frac{1}{2}2^{n_k}$ which minimizes

$$\max_{\sigma, \tau \in R_k} d_k(\Phi^\sigma \triangle \Phi^\tau).$$

Finally, let $D(n) = \Phi^\tau(n)$ for all $n \in J_k$, where τ is some fixed element of R_k (say, the lexicographically least element). We claim that D is a coarse description of A, which implies that A is coarsely computable.

Let $\varepsilon > 0$ be given, and define $\mathcal{B}_k$ by

$$\mathcal{B}_k = \{Y \mid \Phi^Y(n){\downarrow} \text{ for all } n \in J_k \text{ and } d_k(\Phi^Y \triangle A) < \varepsilon\}.$$

Note that in the definition of S_k, all s_k-sized prefixes of elements in $\mathcal{B}_k$ are "valid choices".

Suppose that $\mu(\mathcal{B}_k) > \frac{4}{5}$ for some k. This means that at least $\frac{4}{5}$ of all the strings σ of length s_k satisfy $d_k(\Phi^\sigma \triangle A) < \varepsilon$, thus at least $\frac{3}{5}2^{s_k}$ strings in S_k satisfy $d_k(\Phi^\sigma \triangle A) < \varepsilon$, which means that there is a set $R \subseteq S_k$ with $|R| > \frac{3}{5}2^{s_k}$ for which

$$\max_{\sigma,\tau \in R} d_k(\Phi^\sigma \triangle \Phi^\tau) < 2\varepsilon.$$

Since R_k minimizes the value above, we know that $d_k(\Phi^\sigma \triangle \Phi^\tau) < 2\varepsilon$ for all $\sigma, \tau \in R_k$.

Since $\mu(\mathcal{B}_k) > \frac{4}{5}$, at least one $\sigma \in R_k$ is a prefix of an element in $\mathcal{B}_k$, so if $\tau \in R_k$ is the string used to define D we have

$$\begin{aligned} d_k(D \triangle A) &= d_k(\Phi^\tau \triangle A) \\ &\leq d_k(\Phi^\tau \triangle \Phi^\sigma) + d_k(\Phi^\sigma \triangle A) \\ &\leq 2\varepsilon + \varepsilon = 3\varepsilon. \end{aligned}$$

If $Y \in \mathcal{P}$, by Corollary 1.1.15 we have $Y \in \mathcal{B}_k$ for all sufficiently large k. Hence the union of the sets $\mathcal{C}_k = \bigcap_{j>k} \mathcal{B}_k$ contains all elements in $\mathcal{P}$. This means that $\mu(\mathcal{C}_k) > \frac{4}{5}$ for all large enough k, which implies that $\mu(\mathcal{B}_k) > \frac{4}{5}$ for all large enough k.

Therefore, $d_k(D \triangle A) < 3\varepsilon$ for all large enough k. Since ε was arbitrary, this shows that D is a coarse description of A, contradicting the hypothesis that A is not coarsely computable. $\square$

Corollary 1.2.8 ([8, Corollary 5.3]). If A and B are relatively weakly 3-random, then they form a minimal pair for relative coarse computability.

Proof. By Propositions 1.1.5 and 1.1.27, neither A nor B is coarsely computable, so we have to show that if C is coarsely computable relative to A and B, then C is coarsely computable.

If not, let Y be a coarse description of C which is computable in B. Then A is weakly 3-random relative to Y, so by the lemma above we have Y is not coarsely computable relative to A, which means that X is not coarsely computable relative to A, a contradiction. □

1.2.3 Minimal Pairs in the Dense Degrees

For the dense degrees, the definitions are similar.

Definition 1.2.9. Two sets A and B form a *minimal pair for the uniform dense degrees* if both are non-densely computable and if C is any set such that $C \leq_{\mathrm{uc}} A, B$, then C is densely computable.

A minimal pair for the nonuniform dense degrees is defined similarly.

Because dense computability is more flexible than coarse computability, showing a result similar to Theorem 1.2.8 will require more work. We start with a lemma.

Definition 1.2.10 ([2, p. 169]). If $\mathcal{S} = \{\mathcal{S}_n \subseteq 2^\omega\}_{n\in\mathbb{N}}$ is a sequence of measurable subsets of 2^ω and $A \in 2^\omega$, define $\mathcal{S}(A) = \{n \mid A \in \mathcal{S}_n\}$.

Lemma 1.2.11 ([2, Lemma 6.1]). Let $a, b, q \in [0, 1]$ and $\mathcal{S} = \{\mathcal{S}_n\}_{n\in\mathbb{N}}$. Suppose that

$$\overline{\rho}(\{n \mid \mu(\mathcal{S}_n) < q\} > a$$

and

$$\mu(\{A \mid \rho(\mathcal{S}(A)) = 1\}) > b.$$

Then $(1 - q)a + b \leq 1$.

Proof. Fix $r < 1$ and define the class $\mathcal{X}_n$ by

$$\mathcal{X}_n = \{A \mid \forall k > n[\rho_k(\mathcal{S}(A)) > r]\}.$$

The union of all $\mathcal{X}_n$ contains the set $\{A \mid \rho(\mathcal{S}(A)) = 1\}$, and $\mathcal{X}_n \subseteq \mathcal{X}_{n+1}$. Therefore, there exists some N such that $\mu(\mathcal{X}_n) > b$ for all $n > N$. Now fix $n > N$ to satisfy

$$\rho_n(\{k \mid \mu(\mathcal{S}_k) < q\}) > a$$

(such n exist by hypothesis), and consider the equality

$$\frac{1}{n}\sum_{j<n}\int_{2^\omega} \mathbb{1}_{\mathcal{S}_j}\mathrm{d}\mu = \frac{1}{n}\int_{2^\omega}\sum_{j<n}\mathbb{1}_{\mathcal{S}_j}\mathrm{d}\mu. \tag{1.2}$$

This equality is true because integrals commute with finite sums. We will provide an upper bound for the left side of the equality and a lower bound for the right side of the equality which, together, will give us the result.

For the right side, we will compare it with $\rho_n(\{n \mid \mu(\mathcal{S}_n) < q\})$; we have

$$\frac{1}{n}\sum_{j<n}\int_{2^\omega} \mathbb{1}_{\mathcal{S}_j}\mathrm{d}\mu = \frac{1}{n}\sum_{j<n}\mu(\mathcal{S}_j).$$

Due to the choice of n, at least an of the $\mathcal{S}_j$ satisfy $\mu(\mathcal{S}_j) < q$. For the remaining $n - an$ of them, we can simply bound $\mu(\mathcal{S}_j)$ by 1. This gives

$$\begin{aligned}\frac{1}{n}\sum_{j<n}\int_{2^\omega} \mathbb{1}_{\mathcal{S}_j}\mathrm{d}\mu &\leq \frac{anq + (n - an)}{n} \\ &= 1 - (1-q)a.\end{aligned}$$

For the left side, we can restrict the integration domain to get

$$\frac{1}{n}\int_{2^\omega}\sum_{j<n}\mathbb{1}_{\mathcal{S}_j}\mathrm{d}\mu \geq \frac{1}{n}\int_{\mathcal{X}_n}\sum_{j<n}\mathbb{1}_{\mathcal{S}_j}\mathrm{d}\mu.$$

Because $n > N$, if $A \in \mathcal{X}_n$ we have $\rho_n(\mathcal{S}(A)) > r$, which expands to

$$\sum_{j<n}\mathbb{1}_{S_j}(A) > rn.$$

Substituting into the integral gives

$$\begin{aligned}\frac{1}{n}\int_{2^\omega}\sum_{j<n}\mathbb{1}_{\mathcal{S}_j}\mathrm{d}\mu &\geq \frac{1}{n}\int_{\mathcal{X}_n} rn\mathrm{d}\mu \\ &= r\mu(\mathcal{X}_n) > rb.\end{aligned}$$

Combining both inequalities gives $rb < 1-(1-q)a$, which rearranges to $(1-q)a+rb < 1$. Because $r < 1$ was arbitrary, we finish the proof. □

The lemma above will be used to prove the following theorem, which will enable us to implement the "majority vote" argument for dense degrees.

Theorem 1.2.12 ([2, Proposition 6.2]). *Let $\mathcal{S} = \{\mathcal{S}_n \subseteq 2^\omega\}_{n\in\mathbb{N}}$ and q satisfy*

$$\mu(\{A \mid \rho(\mathcal{S}(A)) = 1\}) > q.$$

Then

$$\rho(\{n \mid \mu(\mathcal{S}_n) \geq q\}) = 1.$$

Proof. The result is trivial if $q = 0$, so assume $q > 0$. Define p by

$$p = \overline{\rho}(\{n \mid \mu(\mathcal{S}_n) < q\}),$$

and assume for the sake of contradiction that $p > 0$. Let $\varepsilon > 0$ be such that $\frac{1}{p\varepsilon} \in \mathbb{N}$ and

$$\mu(\{A \mid \rho(\mathcal{S}(A)) = 1\} > q + \varepsilon.$$

We will construct a subsequence $\mathcal{T}$ of $\mathcal{S}$ such that setting $a = 1-\varepsilon$ and $b = q+\varepsilon$ then $a, b, q, \mathcal{T}$ satisfy the hypothesis of the lemma above, which is a contradiction because $a(1-q)+b = 1 + q\varepsilon$ in this case. Thus $p = 0$.

Define the set $I = \{n_0 < n_1 < \cdots\}$ to be the collection of all n for which either $n \equiv 0 \pmod{\frac{1}{p\varepsilon}}$ or $\mu(\mathcal{S}_n) < q$, and define

$$\mathcal{T}_j = \mathcal{S}_{n_j}.$$

Observe that $\underline{\rho}(I) \geq p\varepsilon$, because I contains all multiples of $\frac{1}{p\varepsilon}$, and by definition of p we have $\overline{\rho}(I) \leq p\varepsilon + p$. Intuitively, $\mathcal{T}$ has a positive fraction of all entries of $\mathcal{S}$.

Let $\delta > 0$ be given. There is some N such that $\rho_n(I) < p(1+\varepsilon) + \delta$ for all $n > N$. On the other hand, by definition of p, there are infinitely many n such that

$$\rho_n(\{i \mid \mu(\mathcal{S}_i) < q\}) > p - \delta.$$

If $n > N$ satisfies this condition, then

$$\begin{aligned}
\rho_k(\{n \mid \mu(\mathcal{T}_n) < q\}) &= \rho_k(\{j \mid \mu(\mathcal{S}_{n_j}) < q\}) \\
&= \frac{\rho_{n_k}(I \cap \{n \mid \mu(\mathcal{S}_n) < q\})}{\rho_{n_k}(I)} \\
&\geq \frac{\rho_{n_k}(\{n \mid \mu(\mathcal{S}_n) < q\})}{\rho_{n_k}(I)} \\
&> \frac{p-\delta}{p(1+\varepsilon)+\delta}.
\end{aligned}$$

This means that

$$\overline{\rho}(\{n \mid \mu(\mathcal{T}_n) < q\}) \geq \frac{p-\delta}{p(1+\varepsilon)+\delta}.$$

Since $\delta > 0$ was arbitrary, this means that

$$\overline{\rho}(\{n \mid \mu(\mathcal{T}_n) < q\}) \geq \frac{1}{1+\varepsilon} > 1 - \varepsilon.$$

Therefore, a, b and $\mathcal{T}$ satisfy the first condition of Lemma 1.2.11.

Now, since $\mathcal{T}$ is a subsequence of $\mathcal{S}$, if $\rho(\mathcal{S}(A)) = 1$ then

$$\begin{aligned}
\rho(\mathcal{T}(A)) &= \lim_k \rho_k(\{j \mid A \in \mathcal{S}_{n_j}\}) \\
&= \lim_k \frac{n_k}{k} \rho_{n_k}(\{i \mid a \in \mathcal{S}_i\} \cap I) \\
&= \lim_k \frac{n_k}{k} \rho_{n_k}(\mathcal{S}(A) \cap I) \\
&= \lim_k \frac{\rho_{n_k}(\mathcal{S}(A) \cap I)}{\rho_{n_k}(I)} \\
&= \lim_k \frac{\rho_{n_k}(I) - \rho_{n_k}(I \cap \overline{\mathcal{S}(A)})}{\rho_{n_k}(I)} \\
&= 1 - \lim_k \frac{\rho_{n_k}(I \cap \overline{\mathcal{S}(A)})}{\rho_{n_k}(I)} \\
&\geq 1 - \lim_k \frac{\rho_{n_k}(\overline{\mathcal{S}(A)})}{\rho_{n_k}(I)} \\
&= 1,
\end{aligned}$$

because the numerator goes to zero whereas the denominator stays (asymptotically) between $p\varepsilon$ and $p(1+\varepsilon)$. This means that

$$\mu(\{A \mid \rho(\mathcal{T}(A)) = 1\}) \geq \mu(\{A \mid \rho(\mathcal{S}(A)) = 1\}) > q + \varepsilon.$$

Hence b and $\mathcal{T}$ satisfy the second condition of Lemma 1.2.11. As noticed before, this means that $1 + q\varepsilon \leq 1$, a contradiction. □

Now we can show the analogue to Proposition 1.2.7 for dense reducibility.

Proposition 1.2.13 ([2, Theorem 6.4]). If A is non-densely computable and X is weakly 4-random relative to A, then there are no X-computable dense descriptions of A. (In particular, $A \not\leq_{\mathrm{nd}} X$.)

Proof. Suppose not, and let Φ be a Turing functional for which Φ^X is a dense description of A. Define the class

$$\mathcal{F} = \{Y \mid \Phi^Y \text{ is a dense description of } A\}.$$

Note that $\mathcal{F}$ is a $\Pi_4^{0,A}$ class; indeed,

$$Y \in \mathcal{F} \quad \Leftrightarrow \quad \forall k \exists N \forall n > N \exists s \Big[\big| \{x < n \mid \Phi^Y(x)[s]\downarrow = A(x)\} \big| > n(1 - 2^{-k}) \Big]$$

Since $X \in \mathcal{F}$ and X is weakly 4-random relative to A, it follows that $\mu(\mathcal{F}) > 0$. By replacing Φ if necessary, we may assume that $\mu(\mathcal{F}) > \frac{4}{5}$; indeed, let σ satisfy $\mu(\mathcal{F} \cap [\![\sigma]\!]) > \frac{5}{6} 2^{-|\sigma|}$ (which exists by the Lebesgue Density Theorem) and if $|\tau| = |\sigma|$ replace $\Phi^{\tau Z}$ with $\Phi^{\sigma Z}$, except if $\tau = X \upharpoonright |\sigma|$ (this exception guarantees $X \in \mathcal{F}$).

As before, our goal is to show that A is densely computable, resulting in a contradiction. We will use Theorem 1.2.12. For each n, define $\mathcal{S}_n = \{Y \mid \Phi^Y(n)\downarrow = A(n)\}$, and $\mathcal{S} = \{\mathcal{S}_n\}_{n \in \mathbb{N}}$. Each $\mathcal{S}_n$ is a $\Sigma_1^{0,A}$ class, and thus measurable. Furthermore, $\mathcal{S}(Y) = \{n \mid \Phi^Y(n)\downarrow = A(n)\}$, so, by definition, $\rho(\mathcal{S}(Y)) = 1$ if and only if Φ^Y is a generic description of A. Hence

$$\mu(\{Y \mid \rho(\mathcal{S}(Y)) = 1\}) > \frac{4}{5}.$$

Therefore, by Theorem 1.2.12,

$$\rho(\{n \mid \mu(\mathcal{S}_n) \geq \tfrac{4}{5}\}) = 1.$$

We define the partial function g as follows. For $g(n)$, find an i, an s, and a set R of s-sized strings for which $\Phi^\sigma(n)\downarrow = i$ and $|R| > 2^{s-1}$ (that is, more than half of all possible

s-sized strings σ agree that $\Phi^\sigma(n)\downarrow = i$) and define $g(n) = i$. Since such set exists for at most one i, this partial function is computable. If $\mu(\mathcal{S}_n) > \frac{4}{5}$ then the set R above exists and $g(n) = A(n)$. Since this happens for densely many n, it follows that g is a dense description of A. Thus A is densely computable. □

Proposition 1.2.14 ([2, Corollary 6.7]). If X and Y are relatively weakly 4-random, then they form a minimal pair for relative dense computability.

Proof. By Propositions 1.1.5 and 1.1.32, neither X nor Y are densely computable. Let C be a set which is densely computable relative to both X and Y, and let Φ be a Turing functional for which Φ^Y is a dense description of C.

By the Low Basis Theorem [19, Theorem 3.7.2], there exists a completion D of Φ^Y which is low relative to Y. By Theorem 1.1.11, the set X is weakly 2-random relative to Y'', so it is weakly 2-random relative to D'', so it is weakly 4-random relative to D. This means there are no X-computable dense descriptions of D, which means there are no X-computable dense descriptions of C, which contradicts the hypothesis that C is densely computable relative to C. □

1.3 Minimal Pairs for Generic Computability

As we've seen in the previous section, the Turing, coarse, and dense degrees have measure-1 many minimal pairs. The situation for generic computability is more complicated, and is the object of study of this section and Section 2.1.

1.3.1 There are no minimal pairs for relative generic computability

A result from Igusa [11], which predates the results on coarse and dense degrees, states that, for the more restricted notion of relative generic computability, there are actually *no* minimal pairs.

Theorem 1.3.1 ([11, Theorem 2.1]). *If A and B are noncomputable sets, then there is a set C which is generically computable relative to both A and B, but which is not generically computable.*

To obtain the set C, we will construct two total Turing functionals Φ and Ψ such that, for almost all sets X and Y, the sets Φ^X and Ψ^Y are dense, but the union $\Phi^X \cup \Psi^Y$ does not contain any dense c.e. subset.

Under these conditions, we can let $C = \Phi^X \cup \Psi^Y$. The set X can generically compute C by letting $f(n) = 1$ if $\Phi^X(n) = 1$ and leaving $f(n)$ undefined otherwise. Then f is an X-computable function with dense domain which agrees with $\mathbb{1}_C$ wherever it is defined. The set Y similarly generically computes C. And C is not generically computable, because if f agrees with $\mathbb{1}_X$ wherever it is defined, then $f^{-1}(1)$ is a dense subset of X, which thus cannot be c.e.

Definition 1.3.2 ([11, p. 514]). A set X has a *gap of size* 2^{-e} *at* J_i if the last 2^{i-e} elements of J_i are not in X; algebraically, if $X \cap [2^{i+1} - 2^{i-e}, 2^{i+1}) = \emptyset$.

Lemma 1.3.3 ([11, Lemma 2.3]). Let X be a set such that all elements missing from it are due to entire gaps being missing. Then X is dense if and only if for each e it only has finitely many gaps of size e.

Proof. Under these constraints, the set X has a gap of size 2^{-e} at J_i if and only if $d_i(X) \leq 1 - 2^{-e}$. The lemma then follows from Corollary 1.1.15. $\square$

We first prove the following special case of 1.3.1.

Theorem 1.3.4 ([11, Proposition 2.5]). *There exist total Turing functionals Φ and Ψ such that, if A and B are not Δ^0_2 sets, then $\Phi^A \cup \Psi^B$ is dense but contains no dense c.e. set.*

Proof. We will construct the total Turing functionals Φ and Ψ in stages. At the beginning of stage s, we will have $\Phi^X(n)$ and $\Psi^Y(n)$ defined for all $n < 2^s$, using only the first s bits of X and Y. (In other words, we will have defined $\Phi^\sigma(n)$ for all $n < 2^s$ and all strings σ with $|\sigma| = s$, and similarly for Ψ.) For definiteness, define $\Phi^X(0) = \Psi^Y(0) = 1$ for all X, Y.

Strategy e starts acting on stage e. Its goal is to either diagonalize against W_e being a subset of any $\Phi^X \cup \Psi^Y$, or, failing that, at least make sure W_e is not a dense set. To do so, on stage s, enumerate W_e for s steps, and define $T_{e,s}$ to be the 4-ary tree of all pairs $\langle\sigma, \tau\rangle$ such that the first $2^{|\sigma|}$ bits of $W_e[s]$ are contained in $\Phi^X \cup \Psi^Y$. Formally,

$$T_{e,s} = \left\{ \langle\sigma, \tau\rangle \;\middle|\; k := |\sigma| = |\tau| \leq s \wedge W_e[s] \restriction 2^k \subseteq \Phi^\sigma \cup \Psi^\tau \right\}$$

Note that, since the first 2^s bits of Φ^X and Ψ^Y were already defined, the tree $T_{e,s}$ is finite and uniformly computable.

If $T_{e,s}$ has no members $\langle\sigma, \tau\rangle$ with $|\sigma| = |\tau| = s$, then Strategy e succeeded, as $W_e[s] \subseteq W_e$ and for any X, Y, the set $\Phi^X \cup \Psi^Y$ is missing some element of $W_e[s]$.

Otherwise, let $\langle\sigma, \tau\rangle$ be the leftmost pair of $T_{e,s}$ with $|\sigma| = |\tau| = s$. Let $\hat{\sigma}$ and $\hat{\tau}$ be the longest prefixes of σ and τ such that the pair $\langle\hat{\sigma}, \hat{\tau}\rangle$ is unmarked, mark it, and place a gap of size 2^{-e} at P_s in both $\Phi^{\hat{\sigma}}$ and $\Psi^{\hat{\tau}}$. That is, Strategy e demands for both $\Phi^{\hat{\sigma}}$ and $\Psi^{\hat{\tau}}$ to have gaps of size 2^{-e} at P_s.

(Intuitively, if T_e is the union of all $T_{e,s}$, we will "sacrifice" the leftmost path of T_e, in the sense that the diagonalization will fail here; but since we are placing gaps of size 2^{-e}, for W_e to be contained in $\Phi^X \cup \Psi^Y$ it must not be dense.)

We essentially let the strategies act independently. On stage s, only the first s strategies acted, placing gaps at P_s for various values of X and Y. Define the remaining values of $\Phi^X(n)$ and $\Psi^Y(n)$ to be 1 (for $n < 2^{s+1}$).

This defines $\Phi^X(n)$ and $\Psi^Y(n)$ for all X and Y and all $n < 2^{s+1}$, so the construction is finished. We now prove correctness.

For each e, let T_e be the union of all $T_{e,s}$ for all s. If, for some s, the tree $T_{e,s}$ has no members of length s, then so will T_e. Therefore, T_e is finite, and Strategy e stopped acting at stage s. This means that W_e is not contained in $\Phi^X \cup \Psi^Y$ for any X, Y.

So suppose that T_e is infinite. Note that, for a pair $\langle\sigma, \tau\rangle$ with $|\sigma| = |\tau| < s$, if $\langle\sigma, \tau\rangle \notin T_{e,s}$, then $\langle\sigma, \tau\rangle \notin T_{e,t}$ for any $t > s$. This implies that the tree T_e is computable. Let X_e, Y_e be the two sets corresponding to the leftmost path in T_e; note that these sets are Δ^0_2.

Since all prefixes $\langle\sigma, \tau\rangle$ of (X_e, Y_e) of length s are contained in $T_{e,s}$, they were chosen infinitely often by Strategy e (because infinitely often they were the leftmost pair of $T_{e,s}$), so all prefixes of X_e and Y_e are marked. Therefore, $\Phi^{X_e} \cup \Psi^{Y_e}$ has infinitely many gaps of size 2^{-e}. But because $W_e \subseteq \Phi^{X_e} \cup \Psi^{Y_e}$, by the definition of $T_{e,s}$, the set W_e itself has infinitely many gaps of size 2^{-e}. Hence W_e is not dense. This means that, for all X and Y, the set $\Phi^X \cup \Psi^Y$ contains no dense c.e. subset.

Finally, we will show that if, for all e, we have $X \neq X_e$ and $Y \neq Y_e$, then $\Phi^X \cup \Psi^Y$ is dense. Let $\langle\sigma, \tau\rangle$ be a prefix of (X_e, Y_e) which is not a prefix of (X, Y). At some stage, the pair $\langle\sigma, \tau\rangle$ will be marked by Strategy e, and because the pair (X_e, Y_e) is the leftmost path of T_e, at a further stage t, all marked pairs will be extensions of $\langle\sigma, \tau\rangle$. Then, for all $s > t$, Strategy e will never place a gap of size 2^{-e} at P_s in $\Phi^X \cup \Psi^Y$.

Since A and B are not Δ^0_2 and all X_e and Y_e are, the set $\Phi^X \cup \Psi^Y$ is dense (as it has only finitely many gaps of size 2^{-e}), but it contains no dense c.e. set. □

In the proof above, we constructed two total Turing functionals Φ and Ψ which almost always yield a dense set. The sets X_e and Y_e are the only exceptions. The theorem works for any non-Δ^0_2 set because X_e and Y_e themselves are Δ^0_2.

The sets X_e and Y_e are the leftmost path of a certain computable tree. To prove the theorem for noncomputable Δ^0_2 sets, we will choose different paths.

Proof of 1.3.1. The case where the noncomputable sets A and B are not Δ^0_2 is covered by Theorem 1.3.4, so assume A is Δ^0_2 but B is not.

Modify the proof of the theorem above as follows. By the recursion theorem, we may assume we have an index for the tree T_e. Since T_e is a computable 4-ary tree, by the cone-avoidant basis theorem [5, Theorem 2.19.10], there is a Δ^0_2 path Z_e through T_e which does not compute A, and it is possible to uniformly compute a Δ^0_2 index for Z_e. Then, on stage s, Strategy e computes $Z_e[s]$ (an approximation to Z_e) and chooses $\langle \sigma, \tau \rangle$ to be the longest prefix of $Z_e[s]$ which is contained in $T_{e,s}$, instead of taking the leftmost path. The rest of the construction (marking prefixes of $\langle \sigma, \tau \rangle$ and placing gaps) is the same.

By the same arguments as before, if $X \neq X_e$ and $Y \neq Y_e$, then $\Phi^X \cup \Psi^Y$ is dense, and for all X and Y the set $\Phi^X \cup \Psi^Y$ contains no dense c.e. subset.

But here, if T_e is infinite, we will have $Z_e = (X_e, Y_e)$, so neither X_e nor Y_e compute A. Therefore, we have $A \neq X_e$ for all e. Since $B \neq Y_e$ for all e, because B is not Δ^0_2, the set $\Phi^A \cup \Psi^Y$ will be dense.

Finally, if A and B are both Δ^0_2, we can let $Z_e \not\geq_{\mathrm{T}} A \oplus B$. The argument is the same.□

1.3.2 There are minimal pairs for generic reducibility

We saw in the last section that the notion of relative generic computability is "broken enough" that it has no minimal pairs. It turns out that, for generic reducibility, we do have minimal pairs.

Theorem 1.3.5 ([6, Theorem 7], see also [7]). *There exists a minimal pair for nonuniform generic reducibility. More explicitly, there exists two sets A_0 and A_1 which are not generically computable, but if B is nonuniformly generically reducible to both A_0 and A_1, then B is generically computable.*

We fix a computable enumeration $\{W_e\}_{e\in\mathbb{N}}$ of the enumeration operators.

Proof. We will construct four Δ^0_2 objects: the sets A_0 and A_1, and the generic descriptions f_0 and f_1 of A_0 and A_1, respectively. If W_{e_1} and W_{e_2} are two enumeration operators, the functions f_0 and f_1 will try to diagonalize against $W_{e_0}^{f_0}$ and $W_{e_1}^{f_1}$ being generic descriptions of the same set; or, failing that, making sure that this common set is generically computable in the first place.

For an enumeration operator W_e, we denote by $W_e^{f_i}[s]$ the set of all k such that (F, k) is enumerated at the sth stage of computation of W_e, for some F contained in the sth stage of f_i.

The functions f_i will be 1 wherever defined; this will simplify the argument. First, we define $f_i(i) = 1$ and leave $f_i(1-i)$ undefined for $i = 0, 1$.

Claim 1.3.5.1. Under these conditions, if there are indices e_0, e_1 such that $W_{e_0}^{f_0}$ and $W_{e_1}^{f_1}$ are generic descriptions of the same set B, then there exists an index e such that $W_e^{f_0}$ and $W_e^{f_1}$ are also generic descriptions of the set B.

This allows us to consider only one enumeration operator each time, simplifying the notation.

Proof. Essentially, let W^f behave like W_{e_0} if $f(0) = 1$, behave like W_{e_1} if $f(1) = 1$, and not do anything otherwise.

Formally, let

$$W = \Big\{(F \cup \{(0,1)\}, k) \mid (0,1) \notin F \wedge (F, k) \in W_{e_0}\Big\}$$
$$\cup \Big\{(F \cup \{(1,1)\}, k) \mid (1,1) \notin F \wedge (F, k) \in W_{e_1}\Big\}.$$

(Recall that, for enumeration reductions, the sets $W_{e_i}^{f_i}$ must be graphs of characteristic functions, so the elements x are pairs of numbers.) Then $W^{f_0} = W_{e_0}^{f_0}$ and $W^{f_1} = W_{e_1}^{f_1}$. $\square$

Continuing the proof, let R_e be as in Example 1.1.25. To make sure that A_0 and A_1 are not generically computable, we will satisfy the following set of requirements:

$$\mathcal{P}_{e,i}: \text{ if } \mathrm{Dom}\,\Phi_e \cap R_e \text{ is infinite, then } \Phi_e(n) \neq A_i(n) \text{ for some } n \in R_e$$

and

$$\mathcal{N}_e : \forall x \forall s \Big[\text{ if } x \in W_e^{f_0}[s] \cap W_e^{f_1}[s], \text{ then either } x \in W_e^{f_0} \text{ or } x \in W_e^{f_1} \Big]$$

The requirements $\mathcal{P}_{e,i}$ make sure that A_i is not generically computable; if these requirements are satisfied, then either $\Phi_i(n) \neq A_i(n)$ for some n, so that Φ_e is not a generic description of A_i, or $\operatorname{Dom}\Phi_e \cap R_e$ is finite, so that Φ_e is not a generic description of anything because $\operatorname{Dom}\Phi_e$ is not dense.

The requirements $\mathcal{N}_e$ are the fallback if we fail to diagonalize against $W_e^{f_i}$. If $W_e^{f_0}$ and $W_e^{f_1}$ both describe the same set B, define $h(x) = 1$ if $x \in W_e^{f_0}[s] \cap W_e^{f_1}[s]$ for some s, and leave h undefined otherwise. The function h is partial computable, and by requirement $\mathcal{N}_e$, if $h(x) = 1$ then either $x \in W_e^{f_0}$ or $x \in W_e^{f_1}$. So $h = W_e^{f_0} \cup W_e^{f_1}$, which shows that h itself is a generic description of B; hence, B is generically computable.

In isolation, satisfying the requirement $\mathcal{P}_{e,i}$ is easy: we just have to compute successive approximations $\Phi_e[s]$ to Φ_e, and if it converges for some $n \in R_e$, we mark $f_i(n)$ as undefined and define $A_i(n) = 1 - \Phi_e(n)$. (Recall that f_i is 1 where it is defined.) As long as at least one such n is not restricted by higher priority requirements, the requirement $\mathcal{P}_{e,i}$ will be satisfied.

The difficult part is not conflicting with the requirements $\mathcal{N}_e$. Intuitively, $\mathcal{N}_e$ says that, if at some stage s we realize that $W_e^{f_0}(x)[s]$ and $W_e^{f_1}(x)[s]$ agree for some x, then we must commit to preserving this computation in at least one of the two sides.

The fact that the functions f_i are 1 wherever defined makes things easier. In order to preserve the computation $W_e^{f_i}(x)[s]$, let u be the use of this computation; that is, $x \in W_e^{f_i}[s]$ if and only if $(F, x) \in W_e[s]$ for some $F \subseteq f_i$; let $u = \max F$. Then restrict the values of $f_i(n)$ for $n \leq u$ from changing. (Note that we only need to preserve the computation in one of the sides.)

But this also allows us to "restore" computation states: if this computation is violated at a further step t, we can restore the value $W_e^{f_i}(x)[s]$ by simply making F a subset of $f_i[t]$

again. There will never be a conflict of values because f_i is 1 wherever defined. The only issue is that this might re-define the value of $f_i(n)$ for some n, injuring some $\mathcal{P}_{e,i}$.

So, in order to satisfy all requirements, we let the $\mathcal{P}_{e,i}$ issue restraints, rather than the $\mathcal{N}_e$. Specifically, each $\mathcal{P}_{e,i}$ will try to choose some $n \in R_e$ to serve as a witness to $A_i \neq \Phi_e$, and it will make $f_i(n)$ undefined in the process. If $\mathcal{P}_{e,i}$ is allowed to make $f_i(n)$ undefined (that is, it does not violate any restraints), it issues the restraint that no lower-priority requirement may further undefine any $f_{1-i}(k)$ for $k < s$. This guarantees that all computations $W_e^{f_{1-i}}(k)[s]$ for $k < s$ will be preserved, even if the corresponding computations $W_e^{f_i}(k)[s]$ are not.

If $\mathcal{P}_{e,i}$ is injured by a higher-priority requirement, then we define $f_i(n) = 1$ again, which restores the computations as outlined above.

Now, if $\mathcal{P}_{e,i}$ is impeded to act, then we must try to satisfy it by marking $f_i(n)$ as undefined for some larger n. But this eventually will be the case, as if u is the largest use ever preserved by any higher-priority requirement, then $\mathcal{P}_{e,i}$ just needs to undefine some $n > u$.

Finally, neither A_0 nor A_1 is generically computable, and if B is nonuniformly generically reducible to both A_0 and A_1, then there are indices e_0 and e_1 such that $W_{e_0}^{f_0}$ and $W_{e_1}^{f_1}$ are generic descriptions of B, so by the claim there is a single index e such that $W_e^{f_0}$ and $W_e^{f_1}$ are generic descriptions of B, and thus by the requirement $\mathcal{N}_e$, the set B is generically computable. $\square$

CHAPTER 2

MINIMAL PAIRS

Theorem 1.2.8 shows that coarse reducibility admits minimal pairs, and Theorem 1.2.14 shows the same for dense reducibility. In both cases, "sufficiently random" pairs of sets automatically form minimal pairs, and these two proofs have a somewhat measure-theoretical flavor to it. As a result, there are measure-1 many minimal pairs for both coarse and dense reducibilities.

In contrast, the proof of Theorem 1.3.5, which constructs minimal pairs for the generic degrees, is significantly more delicate. In Section 2.1, we will show in fact that there are only measure-0 many such minimal pairs. This means that this "measure-theoretic-flavored" proof style cannot possibly work for generic degrees, so the more delicate construction of Theorem 1.3.5 is in fact a necessity.

The existence of minimal pairs for effective dense reducibility is still an open question.

The minimal pair construction for generic degrees gives us two Δ^0_2 sets. In contrast, the proofs for coarse and dense reducibilities need that at least one half of the minimal pair to be weakly 3-random and weakly 4-random, respectively. This motivates Question 8 from [6]: are there any Δ^0_2 minimal pairs for coarse degrees?

In Section 2.3 we give a positive answer to this question. We will do so by modifying Yates's proof that there exists minimal Turing degrees below any given nonzero c.e. degree ([20]; see also [17, Theorem XI.4.9]). Yates's construction is one of the most elaborate constructions of minimal Turing degrees; it is a "full approximation argument". In full approximation constructions, pretty much everything that we build is computable, and then we typically define the object of interest as a limit of these computable things.

Our setting has the additional wrinkle that coarse objects are just approximations. When proving that a certain set A has minimal degree, for any set B below A we have to either use B to compute A, or compute B ourselves. In Theorem 2.3.5 we will achieve the first case, but in the second case (when B would be coarsely computable) we will have to settle for constructing a sequence of computable of sets approximating B. This compromise is the reason why we do not get a minimal degree from Theorem 2.3.5.

So we will need a bit more work to get Corollary 2.3.6 (that there exists a Δ^0_2 minimal pair for the coarse degrees); specifically we will use an approximation theorem, Lemma 1.1.50, which states that, under certain circumstances, simply having a collection of sets approximating B is enough to guarantee that B is coarsely computable.

The proof of Lemma 1.1.50 [9, Theorem 5.12] uses Lemma 1.1.49. Hence if we were to translate Corollary 2.3.6 to dense degrees, we would also need to translate Lemma 1.1.49, but the proof does not quite work. In Section 2.4 we talk about this limitation, and analyze the limits of variants for the other asymptotic notions of computability.

2.1 There are only a few minimal pairs for generic reducibility

In the proof of Theorem 1.3.4, we constructed two total Turing functionals Φ and Ψ, and two collection of Δ^0_2 sets X_e and Y_e such that, if $X \neq X_e$ for all e and $Y \neq Y_e$ for all e, then $\Phi^X \cup \Psi^Y$ is a dense set without dense c.e. subsets. In this section we will be plugging in generic descriptions in Φ and Ψ, to provide a measure-theoretic quantification of Hirschfeldt's Theorem 1.3.5.

Given a partial function $f : \mathbb{N} \to \{0, 1\}$, write $X \succcurlyeq f$ if f can be extended to the characteristic function of X (that is, $f(x) = 0$ implies $x \notin X$ and $f(x) = 1$ implies $x \in X$). For any Turing functional Φ define $\mathcal{W}_\Phi$ by $\mathcal{W}^f_\Phi(n) = 1$ if $\Phi^X(n) = 1$ for all $X \succcurlyeq f$, and leave $\mathcal{W}^f_\Phi(n)$ undefined otherwise. For example, if f is the characteristic function of X, then $\mathcal{W}^f_\Phi$ is just Φ^X but with the zeros replaced with "undefined".

We want to use $\mathcal{W}_\Phi$ and $\mathcal{W}_\Psi$ as enumeration operators, where Φ and Ψ are the Turing functionals defined in the proofs of Theorems 1.3.1 and 1.3.4.

Proposition 2.1.1. If Φ is a Turing functional, then $\mathcal{W}_\Phi$ is an enumeration operator.

Proof. This follows by compactness. Intuitively, to compute $\mathcal{W}_\Phi^f(n)$, we verify all strings σ which agree with the partial function f (that is, if $f(s)\downarrow$ and $s < |\sigma|$ then $f(s) = \sigma(s)$) whether $\Phi^\sigma(n) = 1$. If $\Phi^X(n) = 1$ for all $X \succcurlyeq f$, then for some length t, all strings σ agreeing with f with length t will satisfy $\Phi^\sigma(n)\downarrow = 1$, so we enumerate $\mathcal{W}_\Phi^f(n) = 1$ at this moment. (If no such t exists, then for each t there exists a string σ of length t agreeing with f for which $\neg(\Phi^\sigma(n)\downarrow = 1)$. Hence by the Weak König's Lemma [19, Theorem 8.3.1] there exists some $X \succcurlyeq f$ for which $\neg(\Phi^X(n)\downarrow = 1)$, so we are correct in not enumerating anything.)

Formally, to enumerate a pair (F, k) into $\mathcal{W}_\Phi$, we first must have $k = 1$, and the finite set F be the graph of a partial $\{0, 1\}$-valued function h. (Note that h has finite domain.) Then compute $\Phi^\sigma(n)[|\sigma|]$ for all strings σ which agree with h. If for some length t, all such σ of length t satisfy $\Phi^\sigma(n)[|\sigma|]\downarrow = 1$ then enumerate (F, k). This finishes the construction.

By construction, if $\mathcal{W}_\Phi^f(n)\downarrow = 1$, then $(\operatorname{graph} h, 1) \in \mathcal{W}_\Phi^f$ for some finite $\operatorname{graph} h \subseteq \operatorname{graph} f$, so we have $\Phi^X(n)\downarrow = 1$ for all $X \succcurlyeq h$, and hence $\Phi^X(n)\downarrow = 1$ for all $X \succcurlyeq f$. Conversely, if $\Phi^X(n)\downarrow = 1$ for all $X \succcurlyeq$, by compactness [19, Theorem 8.3.1] there exists a t for which these computations of $\Phi^X(n)$ use only the first t bits of X, and thus the construction above enumerates $(F, 1)$ into $\mathcal{W}_\Phi$ where F is the graph of $f \upharpoonright t$. $\square$

In the proof of Theorem 1.3.4, in order to diagonalize against dense c.e. sets, the sets X_e and Y_e were "sacrificed" in the sense that Φ^{X_e} is not dense, but Φ^X is if $X \neq X_e$ for all e. We have a similar result here.

Proposition 2.1.2. If $f \neq X_e$ for all e (i.e. for all e there exists some n where $f(n) \downarrow\neq X_e(n)$), then $\mathcal{W}_\Phi^f$ has density 1.

This implies that $\mathcal{W}_\Phi^f$ is a generic description of all sets containing $\mathcal{W}_\Phi^f$.

Proof. Consider strategy e. If T_e is finite, then this strategy places only finitely many gaps, so assume that T_e is infinite and let n satisfy $f(n) \neq X_e(n)$.

At some stage t, the first $n+1$ bits of X_e will converge (thinking of $(X_{e,s}, Y_{e,s})$ as being the string pair marked by the eth strategy on stage s). Hence beyond stage t, strategy e will only place gaps in Φ^X for $X \not\succcurlyeq f$. Therefore, assuming by induction on e that, on stage t, all strategies $e' < e$ also stopped placing gaps on Φ^X for $X \not\succcurlyeq f$, this means that for any $s > t$, the density of Φ^f in P_s is at least $1 - 2^{-e}$.

By induction, $\mathcal{W}_\Phi^f$ has density 1. □

Call two sets A and X coarsely similar if the symmetric difference $A \triangle X$ has density zero. If A and X are not coarsely similar, then no generic description of A is also a generic description of X.

Proposition 2.1.3. Let A be a set which is not coarsely similar to any X_e and B a set which is not coarsely similar to any Y_e. Then (A, B) do not form a minimal pair for the uniform generic degrees.

Proof. Let $C = \Phi^A \cup \Psi^B$. The set C has density 1, and by the proof of Theorem 1.3.4 it contains no density-1 c.e. subset. Thus, C is not generically computable.

However, for any partial $f \preccurlyeq A$ with dense domain, we have $\mathcal{W}_\Phi^f \subseteq C$, and since A is not coarsely similar to any X_e we know that $f \neq X_e$ for all e. By the previous proposition $\mathcal{W}_\Phi^f$ is dense, being thus a generic description of C.

This means that C is generically reducible to A, and analogously C is generically reducible to B. □

Two sets which are not generically equivalent may still have a common generic description; for example, if C is any density-1 non-generically computable set then C and $\mathbb{N}$ can both be generically described by the function which is 1 in C and undefined otherwise. Hence not being generically equivalent does not imply not being coarsely similar, and thus the above result cannot be used to show that in any minimal pair for the generic degrees at least one side contains a Δ_2^0 set, for example.

But using randomness we can get a measure-theoretic sense of how rare minimal pairs are.

Theorem 2.1.4. *If A and B are both 2-random, then (A, B) does not form a minimal pair for the generic degrees.*

This means that the collection of pairs (A, B) which form a minimal pair for the generic degrees has measure zero. This contrasts with the situation for Turing degrees (Theorem 1.2.4).

Proof. If A is 2-random, then it is 1-random relative to $\emptyset'$, so it is 1-random relative to X_e. By the relativized form of Proposition 1.1.5, the set A is Church-stochastic with respect to X_e. This means that $A \triangle X_e$ has positive upper density, otherwise A would be coarsely computable relative to X_e, contradicting Proposition 1.1.27. Similarly, B is not coarsely similar to any Y_e.

Therefore, by Proposition 2.1.3, the pair (A, B) is not a minimal pair for the generic degrees. □

2.2 Notation

For the remainder of this chapter we will generally think of sets as collections of strings. As such, strings will play three different roles. For clarity, strings in different roles will be represented by different variable names.

- x, y: "General" strings (e.g. in the definition of $\overline{d}$ below) and arguments to the partial functions Φ_e^A;
- σ, τ: Prefixes of infinite oracle tapes (e.g. Φ_e^σ); and
- u, v, w: Arguments to function trees (defined in Section 1.1.8).

We denote the empty string by $\emptyset$, and reserve ε for small positive real numbers.

Throughout this text, let $\langle\cdot,\cdot\rangle : \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ be a fixed computable bijection between $\mathbb{N}\times\mathbb{N}$ and $\mathbb{N}$.

2.3 A Δ^0_2 Minimal Pair for the Coarse Degrees

Definition 2.3.1. A pair of strings (σ,τ) *has an* (e,ε)*-split*, or (e,ε)*-splits*, if there exists an n such that $\Phi_e^\sigma(x)$ and $\Phi_e^\tau(x)$ are defined for all strings x of length n; and the proportion of the disagreement for strings of length n is at least ε, i.e.

$$\frac{\left|\{x\in\{0,1\}^* : |x| = n \wedge \Phi_e^\sigma(x)\neq\Phi_e^\tau(x)\}\right|}{2^n} > \varepsilon.$$

In terms of Definition 1.1.18, this is equivalent to $d_n(\Phi_e^\sigma,\Phi_e^\tau) > \varepsilon$, with the added restriction that Φ_e^σ and Φ_e^τ must be defined for all strings of length n.

Recall the definition of function trees from Section 1.1.8

Definition 2.3.2. A partial function tree T is (e,ε)-splitting if for every σ such that $T(\sigma*0)$ and $T(\sigma*1)$ are both defined, this pair (e,ε)-split. Conversely,[1] T is without (e,ε)-splits if no pair of strings in the image of the tree (e,ε)-split.

The intuition behind the notion of (e,ε)-splitting is the following.

Proposition 2.3.3. Let T be a partial computable function tree, let A be a path on T, and assume that Φ_e^A is total. If the tree is (e,ε)-splitting, then any coarse approximation of Φ_e^A computes A. If the tree is without (e,ε)-splits, then there exists a computable set $\hat{B}$ such that $\overline{d}(\Phi_e^A,\hat{B})\leq\varepsilon$, and an index for $\hat{B}$ can be found uniformly given an index for T.

1. Finite trees can be both (e,ε)-splitting and without (e,ε)-splits, but all our trees will be infinite.

Proof. First suppose that T is (e, ε)-splitting, and let B be a coarse approximation to Φ_e^A. By definition, $\lim_{n\to\infty} d_n(B, \Phi_e^A) = 0$. By changing only finitely many strings in B, we may assume that $d_n(B, \Phi_e^A) < \varepsilon/2$ for all n.

We compute A from B as follows. Start setting $u = \emptyset$; because A is a path on T we know that $T(u) \preccurlyeq A$, and that either $T(u * 0) \preccurlyeq A$ or $T(u * 1) \preccurlyeq A$. Simulate $\Phi_e^{T(u*i)}(x)$ for $i = 0, 1$ and all values of x until we find a length n which witnesses the fact that the pair $T(u * 0), T(u * 1)$ has an (e, ε)-split. For this n, the proportion of the disagreement between $\Phi_e^{T(u*0)}$ and $\Phi_e^{T(u*1)}$ is at least 2^{-k}, so for exactly one of those the distance to B is at most 2^{-k-1}. Suppose this happens with $\Phi_e^{T(u*0)}$; then we know that $T(u * 0) \preccurlyeq A$, so we may replace u with $u * 0$ and restart this procedure. The case $\Phi_e^{T(u*1)}$ is identical. This produces progressively larger prefixes of A, which means that A is computable from B.

Now assume that T has no (e, ε) splits. Define the set $\hat{B}$ as follows. For each n, compute $\Phi_e^{T(u)}(x)$ for all u and all strings x of length n until we find some u such that $T(u)$ is defined and $\Phi_e^{T(u)}(x)$ is defined for all x of length n. Set $\hat{B}(x) = \Phi_e^{T(u)}(x)$ for all such x.

Because A is a path on T, infinitely many prefixes of A have the form $T(v)$ for some v, so once the search above finds v we know that $\Phi_e^{T(v)}(x)$ will be defined. This makes $\hat{B}$ computable (the search may stop beforehand but this does not affect convergence). And if $\hat{B}$ is defined according to $\Phi_e^{T(u)}$ for strings of length n, we know its distance to the corresponding $\Phi_e^{T(v)}$ is at most ε (otherwise the pair $T(u)$ and $T(v)$ would (e, ε)-split). Hence $\overline{d}(\hat{B}, \Phi_e^A) \le \varepsilon$. □

We will construct a set A with the following property: every set B coarsely computable by A is either in the same coarse degree as A, or can be "approximated well" by computable sets. If we have a tree T which is (e, ε)-splitting for some $\varepsilon > 0$, the proposition above says we are in the first case. Otherwise we get an approximation to Φ_e^A, which on its own is not enough to conclude that Φ_e^A is coarsely computable; we start by picking a Δ_2^0 1-generic G as the other half of the minimal pair, then we construct a sequence of progressively better approximations, and feed it to Lemma 1.1.50.

Recall from Section 2.2 that $\langle \cdot, \cdot \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ is some fixed computable bijection between $\mathbb{N} \times \mathbb{N}$ and $\mathbb{N}$.

Definition 2.3.4. The $\langle e, k\rangle$-state of a pair of strings (σ, τ) is the number

$$\sum \{2^{-\langle i,j\rangle} \mid \langle i, j\rangle \leq \langle e, k\rangle \wedge \text{the pair } (\sigma, \tau) \text{ has an } (i, 2^{-j})\text{-split}\}.$$

The $\langle e, k\rangle$-state at stage s is defined similarly, but the witness n in the definition of $(i, 2^{-j})$-split must be smaller than s, and all computations $\Phi_i^\sigma(x)$ and $\Phi_i^\tau(x)$ must finish within s steps.

Observe that the $\langle e, k\rangle$-state at stage s can be uniformly computed from e, k, and s. Furthermore, if σ is longer than s, extending it will not change the $\langle e, k\rangle$-state at stage s (as no computation can query $\sigma(x)$ for $x \geq n$), and similarly for τ. So when searching for pairs with higher $\langle e, k\rangle$-states we just have to analyze pairs (σ, τ) where both strings are shorter than s. That is, given e, k, s, a pair with largest $\langle e, k\rangle$-state at stage s can be found computably.

The $\langle e, k\rangle$-state is an integer multiple of $2^{-\langle e,k\rangle}$, and it is always a number between 0 and 1.[2] Furthermore, the $(\langle e, k\rangle + 1)$-state of a pair (σ, τ) is always equal to either its $\langle e, k\rangle$-state, or its $\langle e, k\rangle$-state plus $2^{-\langle e,k\rangle-1}$. Hence if we replace (σ, τ) with a pair $(\hat{\sigma}, \hat{\tau})$ with higher $\langle e, k\rangle$-state, we also increase its $(\langle e, k\rangle + 1)$-state.

We can now proceed with the proof of the main technical result of this section.

Theorem 2.3.5. *Let C be a noncomputable c.e. set. Then there exists a non-coarsely computable set $A \leq_T C$ such that, for every set B which is coarsely reducible to A, either B has the same coarse degree of A, or there exists a sequence of computable sets B_k such that $\overline{d}(B, B_k) < 2^{-k}$.*

2. This deviates slightly from the definition in Odifreddi, where the e-state is an integer between 0 and 2^e.

Proof sketch. Our proof will be a "full approximation argument" for the most part. We will construct a uniformly computable sequence of trees $T^s_{\langle e,k\rangle}$, then define another sequence of trees $T^*_{\langle e,k\rangle}$ by taking the limit of these trees, and pick A to be a path in the intersection. (We will guarantee that the intersection is non empty simply by making the trees nested.)

We have to ensure that, for each e, if Φ^A_e is total, then either $A \leq_{\text{nc}} \Phi^A_e$ or Φ^A_e is coarsely computable. The trees $T^*_{\langle e,k\rangle}$ will attempt to do the former by trying to be $(e, 2^{-k})$-splitting.

Intuitively, if $T^*_{\langle e,k\rangle}$ succeeds and is an $(e, 2^{-k})$-splitting tree, then Proposition 2.3.3 says any coarse approximation to Φ^A_e coarsely computes A, i.e. $A \leq_{\text{nc}} \Phi^A_e$. But if for all k the tree $T^*_{\langle e,k\rangle}$ fails, again by Proposition 2.3.3 each tree gives us a set $\Phi_{f(k)}$ which is at a distance 2^{-k} from Φ^A_e.

The main wrinkle is that Proposition 2.3.3 demands the tree to be partial computable, which $T^*_{\langle e,k\rangle}$ is not, so we will have to find an appropriate substitute (this is Claim 2.3.5.5).□

Proof. Without loss of generality we may assume that C is low. Let $c_0, c_1, c_2, \dots$ be a computable injective enumeration of C. We will construct sequences $T^s_{\langle e,k\rangle}$ of uniformly computable total function trees, and use them to construct the set $A \leq_{\text{T}} C$.

We simultaneously define $T^s_{\langle e,k\rangle}$ and σ_s as follows.

1. For all e, k, and s, set T^s_0 and $T^0_{\langle e,k\rangle+1}$ to be the tree from Lemma 1.1.47.[3]

2. For all e, k, and s, define $T^{s+1}_{\langle e,k\rangle+1}(\emptyset) = T^{s+1}_{\langle e,k\rangle}(0)$, and set $\sigma_s = T^s_s(\emptyset)$.

3. For all e, k, s, and u, once $T^s_{\langle e,k\rangle+1}(u)$ is defined, look for pairs τ_0, τ_1 of extensions of $T^s_{\langle e,k\rangle+1}(u)$ in $T^{s+1}_{\langle e,k\rangle}$ such that $\tau_0, \tau_1 \not\succcurlyeq \sigma_s \restriction c_s$ (these are called *permitted pairs*), take the first found pair with largest $\langle e, k\rangle$-state, and define $T^{s+1}_{\langle e,k\rangle+1}(u * i) = \tau_i$. If no permitted pairs exist, set $T^{s+1}_{\langle e,k\rangle+1}(u * i) = T^s_{\langle e,k\rangle+1}(u * i)$.

3. The choice of the tree from Lemma 1.1.47 will be used on the proof of Claim 2.3.5.8. The other claims would work just fine if we had chosen any other total computable function tree.

Finally, define $A = \lim_s \sigma_s$, and for each e, k, and u, define $T^*_{\langle e,k\rangle+1}(u) = \lim_s T^s_{\langle e,k\rangle}(u)$. (We will prove in Claims 2.3.5.2 and 2.3.5.3 that these two limits indeed exist.) After Claim 2.3.5.5 we will also construct the trees $\hat{T}_{\langle e,k\rangle}$. This finishes the construction. We will show that A satisfies the theorem.

Claim 2.3.5.1. Every $T^{s+1}_{\langle e,k\rangle+1}$ is a total computable function tree refining $T^{s+1}_{\langle e,k\rangle}$.

Proof. That the functions are computable and total follows from the definition, we just have to show that they are still trees refining the previous ones. If there were permitted pairs when defining $T^{s+1}_{\langle e,k\rangle+1}(u * i)$, then it follows directly from the construction that this string equals $T^{s+1}_{\langle e,k\rangle}(\tau)$ for some τ, and that it extends $T^{s+1}_{\langle e,k\rangle+1}(u)$. Assume thus that there were no permitted pairs when defining $T^{s+1}_{\langle e,k\rangle+1}(u * i)$, in which case $T^{s+1}_{\langle e,k\rangle+1}(u * i) = T^{s}_{\langle e,k\rangle+1}(u * i)$.

If $u \neq \emptyset$, this means that there were also no permitted pairs when defining $T^{s+1}_{\langle e,k\rangle+1}(u)$ itself, otherwise $T^{s+1}_{\langle e,k\rangle+1}(u)$ would have been set to one of the sides of that pair, and thus any extension of $T^{s+1}_{\langle e,k\rangle+1}(u)$ would also be permitted. Hence $T^{s+1}_{\langle e,k\rangle+1}(u) = T^{s}_{\langle e,k\rangle+1}(u)$, which is a prefix of $T^{s}_{\langle e,k\rangle+1}(u*i)$. This means that $T^{s+1}_{\langle e,k\rangle+1}(u)$ is a prefix of $T^{s+1}_{\langle e,k\rangle+1}(u*i)$, which is the condition for $T^{s+1}_{\langle e,k\rangle+1}$ to be a function tree. Furthermore, if v satisfies $T^{s+1}_{\langle e,k\rangle+1}(u) = T^{s+1}_{\langle e,k\rangle}(v)$ (which exists by induction on u, as we are showing that $T^{s+1}_{\langle e,k\rangle+1}$ refines $T^{s+1}_{\langle e,k\rangle}$), then no permitted pairs existed when defining the tree $T^{s+1}_{\langle e,k\rangle}$ above $T^{s+1}_{\langle e,k\rangle}(v)$ either, so $T^{s+1}_{\langle e,k\rangle}$ copies $T^{s}_{\langle e,k\rangle}$ above $T^{s+1}_{\langle e,k\rangle}(v)$. In particular, the string $T^{s}_{\langle e,k\rangle+1}(u * i) = T^{s+1}_{\langle e,k\rangle+1}(u * i)$ is contained in $T^{s}_{\langle e,k\rangle}$ (by induction on s) and extends $T^{s}_{\langle e,k\rangle+1}(u) = T^{s+1}_{\langle e,k\rangle+1}(u) = T^{s+1}_{\langle e,k\rangle}(v)$, so it is also copied over to $T^{s+1}_{\langle e,k\rangle}$, which is the condition for $T^{s+1}_{\langle e,k\rangle+1}$ to extend $T^{s+1}_{\langle e,k\rangle}$.

Otherwise, if $u = \emptyset$ but $\langle e,k\rangle \neq 0$, we want to show that $T^{s+1}_{\langle e,k\rangle+1}(i) \succcurlyeq T^{s+1}_{\langle e,k\rangle+1}(\emptyset)$. By definition, $T^{s+1}_{\langle e,k\rangle+1}(\emptyset) = T^{s+1}_{\langle e,k\rangle}(0)$, which must not be an extension of $u_s \upharpoonright c_s$, otherwise we would have permitted pairs when defining $T^{s+1}_{\langle e,k\rangle+1}(i)$. Hence there were also no permitted pairs when defining $T^{s+1}_{\langle e,k\rangle}(0)$, which means $T^{s+1}_{\langle e,k\rangle}(0) = T^{s}_{\langle e,k\rangle}(0)$. Hence, we have

$$T^{s+1}_{\langle e,k\rangle+1}(i) = T^{s}_{\langle e,k\rangle+1}(i) \succcurlyeq T^{s}_{\langle e,k\rangle+1}(\emptyset) = T^{s}_{\langle e,k\rangle}(0) = T^{s+1}_{\langle e,k\rangle}(0) = T^{s+1}_{\langle e,k\rangle+1}(\emptyset).$$

Finally, if $u = \emptyset$ and $\langle e, k\rangle = 0$, we have $T^{s+1}_{\langle e,k\rangle} = T^s_{\langle e,k\rangle}$ by definition, so the previous calculation still applies. $\square$

The following claim will allow us to conclude $A = \lim \sigma_s$ is well-defined, and that C computes it.

Claim 2.3.5.2. $\sigma_{s+1} \succcurlyeq \sigma_s \restriction c_s$.

Proof. By definition, $\sigma_{s+1} = T^{s+1}_{s+1}(\emptyset) = T^{s+1}_s(0)$. If $T^{s+1}_s(0) = T^s_s(0)$, then $\sigma_{s+1} = T^s_s(0) \succcurlyeq T^s_s(\emptyset) = \sigma_s$. Otherwise, there was a change between $T^s_s(0)$ and $T^{s+1}_s(0)$, which may only happen if $T^{s+1}_s(0), T^{s+1}_s(1)$ are a permitted pair, in which case $T^{s+1}_s(0) \succcurlyeq \sigma_s \restriction c_s$ by construction. $\square$

Claim 2.3.5.3. The limit $\lim_s T^s_{\langle e,k\rangle}(u)$ is defined for all e, k, and u.

Proof. Intuitively, once everything else settles, the value of $T^{s+1}_{\langle e,k\rangle+1}(u)$ will only change if this would increase its $\langle e,k\rangle$-state, which may happen at most $2^{\langle e,k\rangle}$ times.

Formally, we proceed by induction on $\langle e,k\rangle$ and on u. Let s be large enough that $T^{s+1}_{\langle e,k\rangle+1}(u)$ has settled, let v satisfy $T^{s+1}_{\langle e,k\rangle+1}(u) = T^{s+1}_{\langle e,k\rangle}(v)$, and assume that it has settled too[4]. Once this happens, we only change the value of $T^{s+1}_{\langle e,k\rangle+1}(u * i)$ for larger s if we find a permitted pair with strictly higher $\langle e,k\rangle$-state, or if the corresponding strings in $T^{s+1}_{\langle e,k\rangle}$ changed. But because $T^{s+1}_{\langle e,k\rangle}(v)$ has settled, by induction the latter only happens if their $(\langle e,k\rangle - 1)$-state increased, which in turn also increases the $\langle e,k\rangle$-state. Hence at this point the pair $T^{s+1}_{\langle e,k\rangle+1}(u * i)$ may change at most $2^{\langle e,k\rangle}$ times, so the limit exists. $\square$

Claim 2.3.5.4. Let P be any infinite path on $T^{s+1}_{\langle e,k\rangle+1}$ or on $T^*_{\langle e,k\rangle+1}$. Then the $\langle e,k\rangle$-states of the prefixes of P are nonincreasing.

4. A priori it is possible that, for each s, we have $T^{s+1}_{\langle e,k\rangle+1}(u) = T^{s+1}_{\langle e,k\rangle}(v_s)$ for a different v_s. But because $T^{s+1}_{\langle e,k\rangle}$ is a total function tree, we have $|v_s| \le |T^{s+1}_{\langle e,k\rangle}(v_s)|$, so there are only finitely many distinct values for v_s. Once $T_{\langle e,k\rangle}$ settles for all of them, by our assumption $T^{s+1}_{\langle e,k\rangle}(v_s)$ all have the same value (i.e. $T^{s+1}_{\langle e,k\rangle+1}(u)$, which has also settled), so there is only value of v.

Formally, for each n, let u be a string with length n such that either $T^{s+1}_{\langle e,k\rangle+1}(u*0)$ or $T^{s+1}_{\langle e,k\rangle+1}(u*1)$ is a prefix of P, and define ℓ_n to be the $\langle e,k\rangle$-state of the pair $\big(T^{s+1}_{\langle e,k\rangle+1}(u*0), T^{s+1}_{\langle e,k\rangle+1}(u*1)\big)$. (If P is a path on $T^*_{\langle e,k\rangle+1}$, use this tree instead.) Then the sequence ℓ_n is nonincreasing.

Proof. This follows from the fact that we use the same "search bound" s when defining $T^{s+1}_{\langle e,k\rangle+1}(u*i)$ as we did when defining $T^{s+1}_{\langle e,k\rangle+1}(u)$. This means that the search space for permitted pairs to define it is strictly smaller than the search space we used to define $T^{s+1}_{\langle e,k\rangle+1}(u)$ itself. The same applies to the limit trees $T^*_{\langle e,k\rangle+1}$. □

Claim 2.3.5.5. Let P be any infinite path on $T^*_{\langle e,k\rangle+1}$. Then $\big(P \oplus T^*_{\langle e,k\rangle+1}\big)'$ can compute (uniformly in e, k) an index for a partial computable function tree $\hat{T}$ which is a subtree of $T^*_{\langle e,k\rangle+1}$, contains P as a path, and is either $(e, 2^{-k})$-splitting or is without $(e, 2^{-k})$-splits.

In our case we will let $P = A$.

Proof. We will do this by ensuring that all pairs of strings in $\hat{T}$ have the same $\langle e,k\rangle$-state.

Let ℓ_n be the sequence of $\langle e,k\rangle$-states of the prefixes of P, as defined after the statement of Claim 2.3.5.4. This sequence is computable by $P \oplus T^*_{\langle e,k\rangle}$, and by Claim 2.3.5.4 it has a limit ℓ, which can be computed by $\big(P \oplus T^*_{\langle e,k\rangle+1}\big)'$. Let N be the smallest number such that $\ell_n = \ell$ for all $n \geq N$. Define the tree $\hat{T}$ as follows.

Let v satisfy $|v| > N$ and $T^*_{\langle e,k\rangle+1}(v) \preccurlyeq P$ (this is the point beyond which all $\langle e,k\rangle$-states equal ℓ), and set $\hat{T}(\emptyset) = T^*_{\langle e,k\rangle+1}(v)$. For each u, find the smallest s such that the pair $\big(T^{s+1}_{\langle e,k\rangle+1}(v*u*0), T^{s+1}_{\langle e,k\rangle+1}(v*u*1)\big)$ has $\langle e,k\rangle$-state ℓ, and set $\hat{T}(u*i) = T^{s+1}_{\langle e,k\rangle+1}(v*u*i)$. Leave $\hat{T}(u*i)$ undefined if no such s exist. Clearly $\hat{T}$ is partial computable, is a subtree of $T^*_{\langle e,k\rangle+1}$, and an index for it can be computed by $\big(P \oplus T^*_{\langle e,k\rangle+1}\big)'$.

By the definition of $T^{s+1}_{\langle e,k\rangle+1}$, we only change the value between $T^{s}_{\langle e,k\rangle+1}(u)$ and $T^{s+1}_{\langle e,k\rangle+1}(u)$ if that would increase the $\langle e,k\rangle$-state. Hence once the $\langle e,k\rangle$-state reaches ℓ, it will not change anymore. Thus in fact we have that $\hat{T}(u*i)$ is defined and equals $T^*_{\langle e,k\rangle+1}(u*i)$ whenever the $\langle e,k\rangle$-state of the pair $\left(T^*_{\langle e,k\rangle+1}(v*u*0), T^*_{\langle e,k\rangle+1}(v*u*1)\right)$ equals ℓ. Therefore, $\hat{T}$ is indeed a function tree.

And finally, if $2^{-\langle e,k\rangle}$ is contained in the summation that forms ℓ (from Definition 2.3.4), then $\hat{T}$ will be $(e, 2^{-k})$-splitting, the tree will have no $(e, 2^{-k})$-splits otherwise. $\square$

Because $A \upharpoonright c_s = \sigma_s \upharpoonright c_s$ and $\lim_s c_s = \infty$, we have that A is an infinite path in $T^*_{\langle e,k\rangle+1}$. We let $\hat{T}_{\langle e,k\rangle+1}$ be the tree given by Claim 2.3.5.5 with $P = A$. For each e, we now distinguish between two cases: either for some k the tree $\hat{T}_{\langle e,k\rangle+1}$ is $(e, 2^{-k})$-splitting, or this happens for no k.

Claim 2.3.5.6. Suppose that, for some k, the tree $\hat{T}_{\langle e,k\rangle+1}$ is $(e, 2^{-k})$-splitting. Then A is nonuniformly coarsely reducible to Φ_e^A.

Proof. By Claim 2.3.5.5 the tree $\hat{T}_{\langle e,k\rangle+1}$ is partial computable and A is a path on it, so by Proposition 2.3.3 any coarse approximation of Φ_e^A computes A; i.e. $\Phi_e^A \geq_{nc} A$. $\square$

Claim 2.3.5.7. In the second case (where no $\hat{T}_{\langle e,k\rangle+1}$ is $(e, 2^{-k})$-splitting), there exists a sequence B_k of computable sets such that $\overline{d}(\Phi_e^A, B_k) < 2^{-k}$.

Proof. By Proposition 2.3.3, from every tree $\hat{T}_{\langle e,k\rangle+1}$ we can obtain a set B_k such that $\overline{d}(\Phi_e^A, B_k) < 2^{-k}$. $\square$

Note that so far we did not need the hypothesis that C is not computable; we will use it in the following claim, which finishes the proof.

Claim 2.3.5.8. The set A is not coarsely computable.

Proof. Let B be any computable set, and assume by contradiction that B is coarsely equivalent to A. By changing finitely many bits of B we may assume that $d_n(A, B) < 1/5$ for all n.

Define the Turing functional Φ_e as follows. Given the string τ as an oracle, we will essentially copy τ to the output if it is "far enough" from B, and leave Φ_e^τ undefined otherwise. More precisely, interpret τ as a partial function from $\{0,1\}^*$ to $\{0,1\}$ by identifying the number n with the nth string in length-lexicographical order. If x has length m and $\tau(y)\downarrow$ for all y of length m, set $\Phi_e^\tau(x) = \tau(x)$ if $d_m(\tau, B) > 1/4$ and leave $\Phi_e^\tau(x)$ undefined otherwise. For example, this means that Φ_e^A is undefined everywhere, and if $\Phi_e^P(x)$ is defined then it equals $P(x)$. Now we will use Φ_e to do the "dirty work" of diagonalizing for us.

Let T_0 be the tree from Lemma 1.1.47, which is a supertree of every $T^{s+1}_{\langle e,k\rangle+1}$, and hence of every $T^*_{\langle e,k\rangle+1}$ and $\hat{T}_{\langle e,k\rangle+1}$. Call an incomparable pair τ_0, τ_1 of strings in the image of T_0 a *good pair* if there exists a length N such that, if $\hat{\tau}_0 \succcurlyeq \tau_0$ and $\hat{\tau}_1 \succcurlyeq \tau_1$ are strings in the image of T_0 longer than N, then $(\hat{\tau}_0, \hat{\tau}_1)$ has an $(e, 2^{-2})$-split. Because T_0 is total and computable, given τ_0 and τ_1 we can compute a length N that witnesses the fact that (τ_0, τ_1) is a good pair.

We argue that if τ_0, τ_1 are both incomparable and not prefixes of A, then they form a good pair. Indeed, because Φ_e copies the oracle to the output (unless the oracle is "too close" to B), if there existed arbitrarily long pairs $(\hat{\tau}_0, \hat{\tau}_1)$ of non-$(e, 2^{-2})$-splitting extensions of (τ_0, τ_1), by König's Lemma we can find two paths B_0, B_1 extending τ_0, τ_1 respectively such that no pair of prefixes from B_0 and B_1 has an $(e, 2^{-2})$-split. But because A, B_0, B_1 are all paths in T_0 (which obeys Lemma 1.1.47), we know that $d(A, B_0) = d(A, B_1) = d(B_0, B_1) = 1/2$. As $d(A, B) \le 1/5$, we have $d_n(B_0, A) > 1/4$ for all large enough n, so $\Phi_e^{B_0}$ has cofinite domain, and similarly for $\Phi_e^{B_1}$. And because $d_n(B_0, B_1) > 1/4$ for all large enough n, there exists some n such that $d_n(\Phi_e^{B_0}, \Phi_e^{B_1}) > 1/4$ and $\Phi_e^{B_0}, \Phi_e^{B_1}$ are defined for all strings of length n. This n witnesses the fact that some prefix of B_0, B_1 has an $(e, 2^{-2})$-split, which is a contradiction. Hence τ_0, τ_1 form a good pair.

Now we focus on $k = 2$ and work with $T^*_{\langle e,2\rangle+1}$ and $\hat{T}_{\langle e,2\rangle+1}$. Let u_0, u_1 be any two strings such that $\hat{T}_{\langle e,2\rangle+1}(u_0), \hat{T}_{\langle e,2\rangle+1}(u_1)$ are both defined and form a good pair, and let

N be the witness. (The notion of good pair is still defined in terms of T_0, it is just that these two strings are also strings on $\hat{T}_{\langle e,2\rangle+1}$.) Let u be their longest common prefix (then $\hat{T}_{\langle e,2\rangle+1}(u) \preccurlyeq \hat{T}_{\langle e,2\rangle+1}(u_i)$), find v, v_0, v_1 such that $\hat{T}_{\langle e,2\rangle+1}(u) = T^*_{\langle e,2\rangle+1}(v)$ and likewise for u_0, v_1 and u_1, v_1, and let s_0 be large enough that $T^*_{\langle e,2\rangle+1}(w) = T^{s+1}_{\langle e,2\rangle+1}(w)$ for all $s > s_0$ and $w = v, v_0, v_1$.

Because Φ_e^A is undefined everywhere, if σ_0 is a prefix of Φ_e^A and σ_1 is any string, then (σ_0, σ_1) do not $(e, 2^{-2})$-split. As A is a path on $\hat{T}_{\langle e,k\rangle+1}$, this means that there are several non-splitting pairs in this tree, so by Claim 2.3.5.5 we actually know that $\hat{T}_{\langle e,k\rangle+1}$ is without $(e, 2^{-2})$-splits.

When defining $T^{s+1}_{\langle e,2\rangle+1}(v * i)$, once $s > N$, the search space will include pairs of strings $(\hat{\tau}_0, \hat{\tau}_1)$ which extend $\hat{T}_{\langle e,2\rangle+1}(u_0), \hat{T}_{\langle e,2\rangle+1}(u_1)$ in $T^{s+1}_{\langle e,2\rangle+1}$ and which are longer than N. By the previous paragraph, the pair $(T^{s+1}_{\langle e,2\rangle+1}(v * 0), T^{s+1}_{\langle e,2\rangle+1}(v * 1))$ is not $(e, 2^{-2})$-splitting, whence replacing them with any of the pairs of strings $(\hat{\tau}_0, \hat{\tau}_1)$ as above would make them $(e, 2^{-2})$-splitting and increase their $\langle e, k\rangle$-state. The only thing that would prevent us from doing this is if none of these pairs are permitted by C. Therefore, this must be the case; we may thus conclude that $c_s > N$ for all $s > N, s_0$.

Finally, we can compute infinitely many of those lower bounds as follows. Take any three strings u_1, u_2, u_3 such that $\hat{T}_{\langle e,k\rangle+1}(u_i)$ is defined for $i = 1, 2, 3$ and these three strings are pairwise incomparable (infinitely many of those triplets must exist, because if $\hat{T}_{\langle e,k\rangle+1}(w) \preccurlyeq A$ for $w = v, v * 0, v * 0 * 0$ for some v, then $\hat{T}_{\langle e,k\rangle+1}(v * 1)$ and $\hat{T}_{\langle e,k\rangle+1}(v * 0 * 1)$ must also be defined). Look for a witness N that $\hat{T}_{\langle e,k\rangle+1}(u_i), \hat{T}_{\langle e,k\rangle+1}(u_j)$ form a good pair (at most one of these three strings is a prefix of A, so this search will succeed for at least one of these pairs). Then find s_0 as described above; this gives us our lower bound.

Because we can find infinitely many of these lower bounds, that means we can compute C, which is a contradiction. Hence B cannot be a coarse description of A. □

Therefore, the coarse degree of A is nonzero, which finishes the proof. □

We now use this theorem to show that there exists a Δ^0_2 minimal pair for the coarse degrees, giving a positive answer to Open Question 8 from [6].

Corollary 2.3.6. There exists a Δ^0_2 minimal pair for the coarse degrees.

Proof. Let C_1, C_2 be c.e. sets forming a minimal pair for the Turing degrees (see e.g. [17, Theorem X.6.5]). Let $A_1 \leq_T C_1$ and $A_2 \leq_T C_2$ be sets obtained from Theorem 2.3.5. Let G be a Δ^0_2 1-generic set. We have two cases.

If A_1 is not coarsely reducible to G, then they form a minimal pair as follows. If B is coarsely reducible to both A_1 and G, by hypothesis we cannot have B having the same coarse degree as A_1, so there exists a sequence of sets $B_1, B_2, \ldots$ such that $\overline{d}(A, B_k) < 2^{-k}$. Thus by Lemma 1.1.50, the set B is coarsely computable.

The same applies if A_2 is not coarsely reducible to G. And if both are coarsely reducible to G, then each of A_1 and A_2 has minimal degree, so A_1 and A_2 themselves form a minimal pair. □

2.4 Effective Completeness Theorems for Other Asymptotic Notions of Computability

In this section we will talk about variants and limitations of Lemma 1.1.49 for coarse, generic, effective dense, and dense approximations, in this order.

The motivation is the observation that Theorem 2.3.5 can be translated to dense reducibilities with minimal changes, but this is not the case for Corollary 2.3.6 because we used Lemma 1.1.50. In turn, the proof of this lemma [9, Theorem 5.12] uses Lemma 1.1.49. Theorem 2.4.6 is a (partial) version of Lemma 1.1.49 but for the sake of thoroughness we also investigate what happens with the other asymptotic notions of computability.

For convenience, we restate the theorem here:

Lemma 1.1.49 ([9, Corollary 5.11]). Let B be any set and suppose there exists a $\emptyset'$-computable function f such that, for each k, the number $f(k)$ is an index for a computable set $\Phi_{f(k)}$ such that $\overline{d}(B, \Phi_{f(k)}) < 2^{-k}$. Then B is coarsely computable.

We first argue that the $\emptyset'$ in the theorem cannot be improved, in the sense that if we have f be A-computable for some non-Δ^0_2 set A, then the limit set B could be non-coarsely computable.

Proposition 2.4.1. Let A be a non-Δ^0_2 set. Then there exists a set B and an A-computable function f such that $d(B, \Phi_{f(k)}) < 2^{-k}$, but B is not coarsely computable.

Proof. Let $R_n \subseteq \{0,1\}^*$ be the set

$$R_n = 0^n1\{0,1\}^* = \{\sigma \in \{0,1\}^* \mid 0^n1 \preccurlyeq \sigma\}.$$

That is, R_n is the collection of all strings beginning with n zeros followed by a one. These sets form a partition of $\{0,1\}^+ = \{0,1\}^* \setminus \{\varepsilon\}$, and $d(R_n) = 2^{-n-1}$.

Identify A with a subset of $\mathbb{N}$ using the length-lexicographical ordering of $\{0,1\}^*$, and define the set B by

$$B = \bigcup_{n \in A} R_n.$$

We claim that B is not coarsely computable.

If $\hat{B}$ is a coarse description of B, then $\hat{B}'$ computes A as follows. For a fixed n, let q_t be the number

$$q_t = \frac{\left|\{\sigma \in \hat{B} \cap R_n : |\sigma| = t\}\right|}{2^n}.$$

Because $\hat{B}$ is a coarse description of B, we have that $n \in A$ implies $\lim_t q_t = 2^{-n-1}$, and $n \notin A$ implies $\lim_t q_t = 0$. Hence $\hat{B}'$ can compute this limit and figure out whether $n \in A$ or not.

This means that, if B were coarsely computable, then some such $\hat{B}$ would be computable, whence A would be $\hat{B}'$-computable (i.e. Δ^0_2).

Finally, we construct the function f. For each fixed k, the set B_k defined by

$$B_k = \bigcup_{\substack{n \in A \\ n \leq k}} R_n$$

is computable and satisfies $d(B, B_k) < 2^{-k}$, and we can A-computably find a computable index for each B_k. This sequence of indices is the function f. □

For the case of generic computability, because the approximations are not allowed to make mistakes, the situation is significantly more tight: we have to settle for f being computable.

Proposition 2.4.2. Let B be any set and suppose there exists a computable function f such that, for each k, the number $f(k)$ is an index for a partial computable function $\Phi_{f(k)}$ such that $\Phi_{f(k)}(\sigma)\downarrow \Rightarrow \Phi_{f(k)}(\sigma) = B(\sigma)$ and $\underline{d}(\operatorname{Dom} \Phi_{f(k)}) > 1 - 2^{-k}$. Then B is generically computable.

Proof. Define the partial computable function g as follows. Fixed σ, compute $\Phi_{f(k)}(\sigma)[s]$ for all pairs $\langle k, s \rangle$ until one converges, say to x; then define $g(\sigma) = x$. Then $g(\sigma)\downarrow \Rightarrow g(\sigma) = B(\sigma)$, and because $\operatorname{Dom} g \supseteq \operatorname{Dom} \Phi_{f(k)}$ for all k, the domain of g is dense. Thus g is a generic description of B. □

Proposition 2.4.3. Let A be a noncomputable set. Then there exists a set B and an A-computable function f such that $\underline{d}(\operatorname{Dom} \Phi_{f(k)}) > 1 - 2^{-k}$ and $\Phi_{f(k)}(\sigma)\downarrow \Rightarrow \Phi_{f(k)}(\sigma) = B(\sigma)$, but B is not generically computable.

Proof. Let B, B_k and R_k be as in the proof of Proposition 2.4.1. The same B works here, but now if B is generically computable with generic description g, we can compute A by noticing that $\operatorname{Dom} g \cap R_n$ must have density 2^{-k}, and in particular it is nonempty. Thus we can simply locate some $\sigma \in \operatorname{Dom} g \cap R_n$, and we will know $A(n) = g(n)$.

To define f, let $\Phi_{f(k)}(\sigma) = B_k(\sigma)$ if $\sigma \in R_t$ for some $t \leq k$ and leave it undefined otherwise. Then f is A-computable, we have $\Phi_{f(k)}(\sigma) = B_k(\sigma) = B(\sigma)$ if $\sigma \in R_t$ for some $t \leq k$ (which is precisely when $\Phi_{f(k)}(\sigma)$ is defined), and $d(\operatorname{Dom}\Phi_{f(k)}) = 1 - 2^{-k-1}$ by construction. $\square$

For effective dense, the situation is identical.

Proposition 2.4.4. Let B be any set and suppose there exists a computable function f such that, for each k, the number $f(k)$ is an index for a total computable function $\Phi_{f(k)}$ with image contained in $\{0, 1, \square\}$ such that $\Phi_{f(k)}(\sigma)\downarrow \Rightarrow \Phi_{f(k)}(\sigma) = B(\sigma)$ and $\overline{d}(\Phi^{-1}_{f(k)}(\square)) < 2^{-k}$. Then B is effectively densely computable.

Proof. Define the total computable function g by

$$g(\sigma) = \begin{cases} \Phi_{f(k)}(\sigma), & \text{if } \Phi_{f(k)}(\sigma) \neq \square \text{ for some } k < |\sigma|; \\ \square, & \text{otherwise.} \end{cases}$$

For a fixed σ, all the $\Phi_{f(k)}(\sigma)$ which are not $\square$ must agree, so g is well-defined, and the search bound $k < |\sigma|$ guarantees that g is computable. Furthermore, by construction, if $g(\sigma) \neq \square$ then $g(\sigma) = B(\sigma)$.

Finally, for each k, the elements $\sigma \in g^{-1}(\sigma)$ are all contained in $\Phi^{-1}_{f(k)}(\square)$, except possibly for the finitely many σ shorter than k. Thus $\underline{d}(g^{-1}(\square)) < 2^{-k}$, whence $g^{-1}(\square)$ is a sparse set. Therefore, g is an effective dense description of B. $\square$

Proposition 2.4.5. Let A be a noncomputable set. Then there exists a set B and an A-computable function f such that $\underline{d}(\Phi^{-1}_{f(k)}(\square)) > 1 - 2^{-k}$ and $\Phi_{f(k)}(\sigma) \neq \square \Rightarrow \Phi_{f(k)}(\sigma) = B(\sigma)$, but B is not generically computable.

Proof. Analogous to the proof of Proposition 2.4.3 $\square$

Finally, for dense computability, the situation is still open (see Open Problem 5.4), but we can provide the following partial result.

Theorem 2.4.6. *Let B be any set and suppose there exists a $\emptyset'$-computable function f such that, for each k, the number $f(k)$ is an index for a partial function $\Phi_{f(k)}$ such that $\overline{d}(\Phi_{f(k)}, \chi_B) < 2^{-k}$. Suppose furthermore that there exists a dense set W such that $W \subseteq$ $\operatorname{Dom}\Phi_{f(k)}$ for all k.*

Then there exists a partial computable dense approximation to B whose domain contains W. In particular, B is densely computable.

Proof. First, we will show that we may assume that f is computable. Let g_i be the partial function defined as follows: given σ, compute $\Phi_{f(i)}(\sigma)[t]$ for all $t > |\sigma|$ in succession until one converges, and let $g_i(\sigma)$ be that value. Then g_i is a partial function uniformly computable in i and σ, and if $|\sigma|$ is large enough then $f(i)[t] = f(i)$ for all $t > |\sigma|$, so g_i and $\Phi_{f(i)}$ disagree at most finitely often. Furthermore, if $\Phi_{f(i)}(\sigma)$ is defined then so is $f(\sigma)$, whence $W \subseteq \operatorname{Dom} g_i$.

We will now construct a partial computable function g which we will later prove to be a dense description of B. Given $i < j \leq m$ and s, say that g_i *trusts* g_j for strings of length m at stage s if

$$d_m(g_{i,s}, g_{j,s}) < 2^{-i+1},$$

where $g_{i,s}$ is the function g_i computed for s steps. (d_m is the "optimistic" definition of distance from 1.1.18.)

Because the $g_{i,s}$ are computable, so is $d_m(g_{i,s}, g_{j,s})$, so we can compute whether g_i trusts g_j or not. Let $k(m, s)$ be the largest $k \leq m$ such that, for all $i < j \leq k$, the function g_i trusts g_j for strings of length m at stage s. Note that $k(m, 0) = m$ for all m, and that the function $s \mapsto k(m, s)$ is nonincreasing. If σ has length m, let s be the first stage at which

$g_{i,s}(\sigma)\downarrow$ for all $i \leq k(m,s)$, and define $g(\sigma)$ by

$$g(\sigma) = g_{k(m,s),s}(\sigma).$$

Leave $g(\sigma)$ undefined if no such s exists.

This finishes the construction of g, which is partial computable and whose domain contains W. We will show that g is a dense description of B.

Fix $\ell \geq 0$; we will show $\overline{d}(g, g_\ell) < 2^{-\ell+4}$. Let N be so large that $d_m(g_i, \chi_B) < 2^{-i}$ for all $i \leq \ell$ and all $m > N$. Because χ_B is total and g_i is an extension of $g_{i,s}$, we also have $d_m(g_{i,s}, \chi_B) < 2^{-i}$ for all s, all $i \leq \ell$, and all $m > N$. Now if $i < j \leq \ell$ we have

$$\begin{aligned} d_m(g_{i,s}, g_{j,s}) &\leq d_m(g_{i,s}, \chi_B) + d_m(\chi_B, g_{j,s}) \\ &\leq d_m(g_i, \chi_B) + d_m(\chi_B, g_j) \\ &< 2^{-i} + 2^{-j} < 2^{-i+1}, \end{aligned}$$

so g_i trusts g_j for strings of length m on every stage s.

Fix $m > N$, and let $\mathcal{D} \subset W$ be the set of disagreements between g_ℓ and g on strings of length m; i.e.

$$\mathcal{D} = \{\sigma \in W : |\sigma| = m \wedge g_\ell(\sigma) \neq g(n)\}.$$

We argue that $|\mathcal{D}| < 2^{m-\ell+3}$.[5]

For $i = \ell+1, \ell+2, \ldots, m$, let $\mathcal{D}_j$ be the elements of $\mathcal{D}$ whose first disagreement with g_ℓ happens in g_j; that is,

$$\mathcal{D}_j = \{\sigma \in \mathcal{D} : g_\ell(n) = g_{\ell+1}(n) = \cdots = g_{j-1}(n) \neq g_j(n)\}.$$

If $\sigma \in \mathcal{D}$ then $\sigma \in W$, so $g_i(\sigma)\downarrow$ for all i. Furthermore, because $\ell \leq k(m,s) \leq m$, we

5. The set $\mathcal{D}$ may not be computable given ℓ but this is not relevant.

have $g(\sigma) = g_j(\sigma)$ for some j between ℓ and m. Therefore, we have $\mathcal{D} = \mathcal{D}_{\ell+1} \cup \cdots \cup \mathcal{D}_m$.

If $\sigma \in \mathcal{D}_j$, let s be the stage at which $g(\sigma)$ is defined. Because $g_\ell(\sigma) = g_{\ell+1}(\sigma) = \cdots = g_{j-1}(\sigma) \neq g_j(\sigma)$ and we know that $k(m,s) \geq \ell$, it must actually be the case that $k(m,s) \geq j$ (otherwise we would have $g(\sigma) = g_i(\sigma)$ for some i satisfying $\ell \leq i < j$, which would make $g(\sigma) = g_\ell(\sigma)$ and thus $n \notin \mathcal{D}$). In particular, g_{j-1} trusts g_j on strings of length m at stage s. But learning that $g_{j-1,s}(\sigma) \neq g_{j,s}(\sigma)$ raises $d_m(g_{j-1,s}, g_{j,s})$, so this may happen at most $2^{m-(j-1)+1}$ times. This means that $|\mathcal{D}_j| \leq 2^{m-(j-1)+1}$.

Adding up the cardinalities of all $\mathcal{D}_j$ thus shows that $|\mathcal{D}| \leq 2^{m-\ell+3}$. That is, $d_m(g_i, g) \leq 2^{-\ell+3}$, so $d_m(g, \chi_B) \leq 2^{-\ell+4}$ for all $m > N$. Because i was arbitrary and the domain of g is dense (because it contains W), this means that $\overline{d}(g, \chi_B) \leq 2^{-\ell+4}$. As ℓ was arbitrary, this shows that g is a dense description of B. □

Proving the theorem without the hypothesis that W is dense is Open Problem 5.4.

The hypothesis that W is dense is needed to go from $d_m(g, \chi_B) < 2^{-\ell+4}$ to $\overline{d}(g, \chi_B) \leq 2^{-\ell+4}$. Philosophically, the function g was defined without any care about whether its domain would be dense or not, but rather we focused on minimizing the disagreements between g and χ_B, We only define $g(\sigma)$ if we can find a long sequence of functions $g_0, g_1, \ldots, g_k$ which trust each other and which are defined on σ. It is just that this will happen in all elements of W, which we demanded to be dense.

The fact that the domain of all the functions $\Phi_{f(k)}$ is dense is not necessary. If we simply had $\overline{d}(g_k, \chi_B) < 2^{-k}$ for some partial computable function g_k, we can make the domain of g_k to be dense by doubling the error of the approximation. Specifically, define $\hat{g}_k$ for all strings of length n by waiting for at least $2^m - 2^{m-k}$ of them to converge for g_k, defining $\hat{g}_k(\sigma) = g_k(\sigma)$ for the ones which converge, and $\hat{g}_k(\sigma) = 0$ for the ones which did not converge. Leave $\hat{g}_k(\sigma)$ undefined if the number of convergences never surpasses $2^m - 2^{m-k}$. Because $d(g_k, \chi_B) < 2^{-k}$, we will define $\hat{g}_k(\sigma)$ for all sufficiently long σ, so the domain of $\hat{g}_k$ is cofinite and thus dense. And when we do define $\hat{g}_k$, we differ from g_k in a fraction of at most 2^{-k} of the strings, so $d(\hat{g}_k, \chi_B) < 2^{-k+1}$.

As a side-note, the theorem above has Lemma 1.1.49 as a corollary, because the domain of g contains W, and $W = \{0,1\}^*$ in this case.

As in the coarse case, we can show that f being $\emptyset'$-computable is the furthest we can go.

Proposition 2.4.7. Let A be a non-Δ^0_2 set. Then there exists a set B and an A-computable function f such that $\underline{d}(\Phi_{f(k)}, \chi_B) < 2^{-k}$ but B is not densely computable.

Proof. Let B, B_k and R_k be as in the proof of Proposition 2.4.1. The same B and f work here, and if g is a dense description of B, we just need to define $x_t = 1$ if

$$\left|\left\{\sigma \in R_n : |\sigma| = t \wedge g(\sigma)\downarrow = 1\right\}\right| > 2^{-n-2},$$

and $x_t = 0$ if

$$\left|\left\{\sigma \in R_n : |\sigma| = t \wedge g(\sigma)\downarrow = 0\right\}\right| > 2^{-n-2}.$$

The rest of the proof is identical. ∎

CHAPTER 3

ATTRACTIVE AND DISPERSIVE DEGREES

This chapter proves a Kolmogorov-complexity-flavored sufficient condition for a set to be attractive (Theorem 3.3.1), and discusses some consequences of this condition.

3.1 The Gamma Function

For a set A and a Turing degree $\mathbf{b}$, we use d from Definition 1.1.17 to define $\gamma_{\mathbf{b}}(A)$ as

$$\gamma_{\mathbf{b}}(A) = \sup_{B \in \mathbf{b}} 1 - d(A, B).$$

This is a measure of how difficult A is to compute, from the point of view of B. For example, if A is coarsely computable relative to $\mathbf{b}$, then $\gamma_{\mathbf{b}}(A) = 1$ ("$\mathbf{b}$ can tell membership in A asymptotically all the time"), and if $\mathbf{b}$ is 1-random relative to A then $\gamma_{\mathbf{b}}(A) \geq 1/2$ ("$\mathbf{b}$ can guess at least half of the entries of A"). We can "lift" this definition to Turing degrees by taking infimums:

$$\Gamma_{\mathbf{b}}(\mathbf{a}) = \inf_{A \in \mathbf{a}} \gamma_{\mathbf{b}}(A).$$

(We also write $\Gamma_B(A)$ for $\Gamma_{\mathbf{b}}(\mathbf{a})$ where $A \in \mathbf{a}$ and $B \in \mathbf{b}$.)

Intuitively, this asks $\mathbf{b}$ what is the hardest challenge that $\mathbf{a}$ can pose to it, with respect to coarse computations. In fact, we can show that $\Gamma_{\mathbf{b}}(\mathbf{a}) = 1$ if and only if $\mathbf{b} \geq \mathbf{a}$.

This fact can be used to define a metric on Turing degrees, called the Hausdorff distance $H(\mathbf{a}, \mathbf{b})$ between $\mathbf{a}$ and $\mathbf{b}$:

$$H(\mathbf{a}, \mathbf{b}) = 1 - \min\{\Gamma_{\mathbf{b}}(\mathbf{a}), \Gamma_{\mathbf{a}}(\mathbf{b})\}.$$

(The name "Hausdorff distance" is justified because we can show that $H(\mathbf{a}, \mathbf{b})$ equals the classical Hausdorff distance between the closures of $\mathbf{a}$ and $\mathbf{b}$ in the pseudometric space $(2^{\mathbb{N}}, d)$

[10, Proposition 4.15].)

The "1−" is here because $\Gamma_{\mathbf{b}}(\mathbf{a})$ is a "measure of closeness", where $\Gamma_{\mathbf{b}}(\mathbf{a}) = 0$ if $\mathbf{a}$ is "hard" for $\mathbf{b}$, and $\Gamma_{\mathbf{b}}(\mathbf{a}) = 1$ if it is "easy". In fact, H is a metric on the space of Turing degrees. (Analogously we write $H(A, B)$ for $H(\mathbf{a}, \mathbf{b})$ where $A \in \mathbf{a}$ and $B \in \mathbf{b}$, but for sets the function H is just a pseudometric.)

In [15], Monin showed that $\Gamma_{\emptyset}(A)$ can only attain the values 0, 1/2, and 1, and [10, Theorem 4.20] extended this result to all $\Gamma_B(A)$. In fact, we get the following characterization of when $\Gamma_B(A)$ attains each value.

Definition 3.1.1 (see [15, Section 1.3]). A function f is 2^{2^n}-*infinitely often equal* (i.o.e.) relative to A if, for every function $g \leq_{\mathrm{T}} A$ such that $g(n) < 2^{2^n}$ for all n, there exists infinitely many n such that $f(n) = g(n)$. [1]

Theorem 3.1.2 (see [15, 10]). *The following are equivalent.*

- $\Gamma_B(A) > 1/2$.
- $\Gamma_B(A) = 1$.
- $B \geq_{\mathrm{T}} A$.

Theorem 3.1.3 ([15, Theorem 3.11], [10, Theorem 4.20]). *The following are equivalent.*

- $\Gamma_B(A) < 1/2$.
- $\Gamma_B(A) = 0$.
- *A computes a function f which is 2^{2^n} infinitely often equal relative to B.*

1. We could replace the 2^{2^n} with any function F to obtain the definition of F-infinitely often equal, but the only bound we will use in this paper is $F(n) = 2^{2^n}$.

(Note that A is the one computing the function, even though that B is being used as an oracle to try to approach A. Intuitively, the set A can present f as the "challenge", which is "too difficult" for B to compute.)

This immediately implies that $H(\mathbf{a}, \mathbf{b}) \in \{0, 1/2, 1\}$.

3.2 Attractive and Dispersive Degrees

We now define the main object of study of this chapter.

Definition 3.2.1 ([10, Definition 5.10]). A Turing degree $\mathbf{d}$ is called *attractive* if there are measure-1 many degrees $\mathbf{c}$ such that $H(\mathbf{d}, \mathbf{c}) = 1/2$, and is called *dispersive* if there are measure-1 many degrees $\mathbf{c}$ such that $H(\mathbf{d}, \mathbf{c}) = 1$.

Kolmogorov's 0-1 law immediately shows that every degree is either attractive or dispersive.

Proposition 3.2.2 ([10, Proposition 5.11]). The class of attractive degrees is closed upwards, and the class of dispersive degrees is closed downwards.

Proof. A useful trick is to note that, for any noncomputable set A, there are measure-1 many sets B which are 1-random relative to A, and which do not compute A. For any such set B, being random implies that, for every A-computable set $\hat{A}$, we have $d(B, \hat{A}) = 1/2$, thus $\Gamma_B(A) \geq 1/2$; and $B \not\geq_T A$ implies $\Gamma_B(A) < 1$, so for every set A there are measure-1 many sets B for which $\Gamma_B(A) = 1/2$.

Hence, to show that A is attractive, it suffices to show that there are measure-1 many sets B for which $\Gamma_A(B) = 1/2$; conversely, to show that A is dispersive, we must show that there are measure-1 many sets B for which $\Gamma_A(B) = 0$.

So, let A be attractive, and $C \geq_T A$. For any B such that $\Gamma_A(B) = 1/2$ we then also have $\Gamma_C(B) \geq 1/2$. There exists measure-1 many such B. Discarding the measure-0 many such

B that might be computable by C, we get measure-1 many sets B for which $\Gamma_C(B) = 1/2$, which (as the above observation shows) implies that C is also attractive.

That dispersive sets are closed downwards follows by the fact that the two classes are complementary. □

3.3 High Kolmogorov Complexity Implies Attractiveness

We can use Theorem 3.1.3 to prove the following sufficient condition for a degree to be attractive.

Theorem 3.3.1. *Let A be a set such that, for some constant C and all large enough n, we have $K(A \upharpoonright 2^n) > 2K(n) - C$. Then A is attractive.*

That is, $K(A \upharpoonright 2^n)$ may dip below $2K(n)$, but not by much. Observe that this is implied by the stronger condition "$K(A \upharpoonright n) > 2K(n) - C$ for all large enough n", because $K(2^n) = K(n) + O(1)$.

Proof. As noted in the proof of Proposition 3.2.2, it suffices to show that there are measure-1 many sets B such that $\Gamma_A(B) = 1/2$. For B to fail to satisfy $\Gamma_A(B) = 1/2$, unless it is one of the measure-0 many sets below A (in which case we'd have $\Gamma_A(B) = 1$), it must be the case that $\Gamma_A(B) = 0$; i.e., the set B has 2^{2^n}-infinitely often equal degree relative to A. This means that there exists a single function Φ_e^B such that, for every A-computable function f bounded by 2^{2^n}, there exists infinitely many n with $f(n) = \Phi_e^B(n)$.

Fix a Turing functional Φ_e, and let $\mathcal{I}$ be the class of sets B such that Φ_e^B is an i.o.e. function as above. We will show that the measure of $\mathcal{I}$ is zero. For the sake of contradiction, assume that this is not the case. Let $f(n)$ be the first 2^n bits of A, interpreted as the binary expansion of a number. We will find concise descriptions of $f(n)$ (hence of $A \upharpoonright 2^n$)) as follows.

For each n and m, let $\mu(n, m)$ be the measure of all the B such that $\Phi_e^B(n)\downarrow = m$, and let $\kappa(n, m)$ be the smallest k such that $2^{-k} < \mu(n, m)$. We will use the KC Theorem [5, Theorem 3.6.1] We would like to construct a sequence of KC requests of the form $(K(n) + \kappa(n,m), m)$, but the numbers $K(n) + \kappa(n,m)$ are not computable. So we approximate them from above as follows.

For each n and each $m < 2^{2^n}$, simultaneously compute $\Phi_e^\sigma(n)$ for all σ, and search for (Kolmogorov) descriptions τ of n. Take note of the measure of all the σ for which $\Phi_e^\sigma(n)\downarrow = m$. If this measure surpasses 2^{-k} for some integer k, issue the request $(k+|\tau|, m)$, where τ is the shortest description of n found so far. Similarly, if a shorter description τ is found for n, issue the request $(k + |\tau|, m)$ for the same k (i.e. k is the smallest integer such that 2^{-k} is a lower bound to the measure of all σ for which $\Phi_e^\sigma(n)$ has been observed to converge to m).

First, let us show that this is indeed a KC set. The process of searching for better Kolmogorov descriptions of n means that, eventually, we will issue the request $(k+K(n), m)$. All other similar requests $(k + |\tau|, m)$ will have exponentially smaller weight; for a fixed k, m, n, the sum of the weights $2^{-k-|\tau|}$ for all found descriptions τ of n is thus bounded by $2^{-k-K(n)+1}$. (Note that this bound is true even if k changes before we find an optimal τ, as τ is never shorter than $K(n)$.) Similarly, for each n, m, as we observe progressively more σ with $\Phi_e^\sigma(n)$ converging to m, we get a progressively better approximation to $\mu(n,m)$, so eventually the request $(\kappa(n,m) + K(n), m)$ will be issued. By a similar analysis, the sum of the weights $2^{-k-K(n)+1}$ of requests issued for a fixed m, n is bounded by $2^{-\kappa(n,m)-K(n)+2}$. For a fixed n, the numbers $\mu(n,m)$ measure disjoint sets, so their sum is at most 1. Because $2^{-\kappa(n,m)} < \mu(n,m)$, the sum of the $2^{-\kappa(n,m)}$ is also at most 1. Finally, the fact that $\sum_n 2^{-K(n)} < 1$ shows that the collection of requests above is indeed a KC set. Summarizing, we have

$$\begin{aligned}
\sum_{m,n,k,\tau:\text{request } (k+|\tau|,m)} 2^{-k-|\tau|} &\leq \sum_{m,n,k:\text{request } (k+K(n),m)} 2^{-k-K(n)+1} \\
&\leq \sum_{m,n} 2^{-\kappa(n,m)-K(n)+2} \\
&= 4\sum_{n} 2^{-K(n)} \sum_{m} 2^{-\kappa(n,m)} \\
&\leq 4\sum_{n} 2^{-K(n)} \sum_{m} \mu(n,m) \\
&\leq 4\sum_{n} 2^{-K(n)} \\
&\leq 4.
\end{aligned}$$

As an immediate consequence, we have $K(m) \leq \kappa(n,m) + K(n) + O(1)$; in particular, we have $K(f(n)) \leq \kappa(n, f(n)) + K(n) + O(1)$, so (because $f(n)$ is $A \upharpoonright 2^n$ interpreted as a number) there exists a fixed constant C_1 such that $K(A \upharpoonright 2^n) \leq \kappa(n, f(n)) + K(n) + C_1$. We now argue that there are infinitely many n such that $\kappa(n, f(n)) \leq K(n) - C - C_1$.

If this is not the case, then for large enough n, say $n > N$, we have $\kappa(n, f(n)) > K(n) - C - C_1$. In particular,

$$\sum_{n>N} \mu(n, f(n)) \leq 2 \sum_{n>N} 2^{-\kappa(n,f(n))} \leq 2 \sum_{n>N} 2^{-K(n)+C+C_1}.$$

Because $\sum_n 2^{-K(n)} \leq 1$, by making N large enough we can make the rightmost term of this inequality as small as we want; in particular, we can make it smaller than the measure of $\mathcal{I}$ (the class of sets B such that Φ_e^B is an i.o.e. function). However, the leftmost term of this inequality, $\sum_{n>N} \mu(n, f(n))$, is (an upper bound on) the measure of the collection of all B such that $\Phi_e^B(n) = f(n)$ for some $n > N$. This contains all sets of $\mathcal{I}$, which is a contradiction.

Hence, there are infinitely many n such that $\kappa(n, f(n)) \leq K(n) - C - C_1$, which means $K(A \upharpoonright 2^n) \leq 2K(n) - C$. Finally, this contradicts our initial hypothesis that $K(A \upharpoonright 2^n) > 2K(n) - C$ for all large enough n, which means that A is attractive. □

The bound $2K(n) - C$ is pretty much optimal for this proof strategy. The KC requests are (ultimately) of the form $K(n) + \kappa(n, m)$, and we need the facts that $\sum_n 2^{-K(n)}$ and $\sum_m 2^{-\kappa(n,m)}$ are both finite.

A priori, we could replace the $K(n)$ with any other function F such that $\sum_n 2^{-F(n)} < \infty$, but, in order to actually construct the KC set, we need to be able to approximate $F(n)$ from above. In other words, the set $\{(n, k) : F(n) \leq k\}$ is a c.e. set, and we can show that any such F beats K by at most an additive constant (see e.g. [5, Theorem 3.7.8]).

We do have a bit of lee-way in the second term, $\kappa(n, m)$. We bound $\kappa(n, f(n))$ by $F(n) = K(n) - C - C_1$, and again we need the property that $\sum_n 2^{-F(n)} < \infty$. But this part of the argument does not need to be computable, so there are no further restrictions on F. That is, the bound $2K(n) - C$ could be replaced by $K(n) + F(n) - C$ where F is any function satisfying $\sum_n 2^{-F(n)}$. We can still rephrase this in terms of Kolmogorov complexity by using F as an oracle: if we simply let X encode the set $\{\langle n, F(n)\rangle : n \in \mathbb{N}\}$, then $K^X(n) \leq F(n) + O(1)$, so the bound on $K(A \upharpoonright n)$ becomes $K(n) + K^X(n) - C$. We note this improvement in a corollary.

Corollary 3.3.2 (of the proof). For any sets X and A and any constant C, if $K(A \upharpoonright 2^n) \geq K(n) + K^X(n) - C$ for all large enough n, then A is attractive.

The rest of this section discusses some interesting consequences of this theorem.

3.3.1 Genericity and Randomness

It is known that every weakly 2-generic set is dispersive [10, Theorem 5.16], and that every 1-random set is attractive [10, theorem 5.5].

1-random sets are characterized by having $K(A \upharpoonright n) \geq n - O(1)$; i.e. all sets with high Kolmogorov complexity are attractive. Theorem 3.3.1 significantly lowers what "high" means in this case, thus narrowing the gap between genericity and randomness.

Interestingly, it is still open whether every 1-generic set is dispersive. 1-generic sets can only compute infinitely often K-trivial sets (i.e. sets B such that $K(B \upharpoonright n) = K(n) + O(1)$ for infinitely many n), which violate the $2K(n)$ bound from the theorem.

3.3.2 Hausdorff Dimension of the class of dispersive sets

It is known that the class of dispersive sets has measure 0 [10, Corollary 5.8]. Using the point-to-set principle [14, Theorem 1], we can show the following stronger result.

Proposition 3.3.3. Let $\mathfrak{D}$ be the class of dispersive sets. Then the (classical) Hausdorff dimension of $\mathfrak{D}$ is zero.

Proof. The point-to-set principle states that

$$\dim_H(\mathfrak{D}) = \min_{A \subseteq N} \sup_{B \in \mathfrak{D}} \dim^A(B).$$

$\dim_H$ is the classical Hausdorff dimension, and $\dim^A$ is the (relativized) effective Hausdorff dimension.

By the contrapositive of Theorem 3.3.1, for every dispersive set B there are infinitely many n such that $K(B \upharpoonright n) \leq 2K(n)$. This means that

$$\dim(B) = \liminf_n \frac{K(B \upharpoonright n)}{n} = 0.$$

So picking $A = \emptyset$ in the point-to-set principle, we get

$$\begin{aligned}\dim_H(\mathfrak{D}) &\leq \sup_{B\in\mathfrak{D}} \dim^A(B)\\ &\leq \sup_{B\in\mathfrak{D}} \dim(B)\\ &= 0.\end{aligned}$$

Hence $\mathfrak{D}$ has Hausdorff dimension 0. □

3.3.3 Minimal Degrees

Because the class of dispersive degrees is closed downwards, using the following lemma we can show that there exists minimal dispersive degrees.

Lemma 3.3.4 ([10, Theorem 5.15]). Every low c.e. set is dispersive.

Proposition 3.3.5. There exists a dispersive set A which is minimal for Turing degrees.

Proof. Let C be a low c.e. set. By the lemma, the set C is dispersive. Now, by [17, Theorem XI.4.9], the set C computes a set D which has minimal Turing degree. And because dispersive degrees are closed downwards, the set D is also dispersive. □

This argument does not work for attractive degrees, which are closed upwards.

It is known that there are sets A with minimal degree and with effective packing dimension 1 (i.e. $\limsup_n K(A\restriction n)/n = 1$) [4, Theorem 1.1]. However, we would need a lower bound on $K(A\restriction n)$ which is always true, rather than just infinitely often. The problem of finding a set with minimal degree and Hausdorff dimension 1 (i.e. $\liminf_n K(A\restriction n)/n = 1$) is still open, but we can formulate the following easier question.

Open Problem 3.3.6. Are there any sets A with minimal Turing degree which also satisfy $K(A\restriction n) > 2K(n)$ for all n?

CHAPTER 4

FUNCTIONS AND SETS

A common theme in both computability and complexity theory is to only talk about yes/no questions, rather than "function questions". For example, the Satisfiability Problem SAT in complexity theory asks whether a Boolean formula has a satisfying assignment or not, whereas in real-world application we would be more interested in finding such an assignment. The reasoning is that this makes the theory more elegant and comes at no cost to generality. For example, we can find a satisfying assignment to a formula with n variables in polynomial time by performing $O(n)$ queries to an oracle for SAT, so the set problem and the function problem are in the same polynomial class of complexity.

Surprisingly, for the four asymptotic notions of computability studied in this paper, it is still an open problem whether functions are equivalent to sets. In this section we take a small step towards answering this question by showing two small results: first, that a certain class of enumeration operators is unable to show that every uniform degree contains a set; and second, that functions that grow "slowly enough" are equivalent to a set.

Formally, we define coarse, dense, generic, and effective generic reducibilities and degrees for functions in the same way we define for sets, *mutatis mutandis*. The question can then be stated as follows.

Open Problem 4.1. For each of the eight reducibilities (the uniform and nonuniform versions of dense reducibility, generic reducibility, coarse reducibility, and effective dense reducibility), defined for functions, is it true that every degree contains the indicator function of a set?

Every nonuniform degree is a union of uniform degrees, so if the answer to this question is negative, it should be easier to prove so for uniform reducibility. We may simplify the question further, and ask whether there is a single enumeration operator W such that, for all functions f, the set W^f is the indicator function of a set and f and W^f have the same uniform generic degree. (In a sense, this is a stronger uniformity condition.)

We will show that the the answer is negative for the following restricted class of operators.

Definition 4.2. For the purposes of this chapter, define a *simple encoding* to be a function $E : \mathbb{N} \to 2^{\mathbb{N}}$ such that $E(x)$ and $E(y)$ are disjoint if $x \neq y$. For any partial function $f : \mathbb{N} \to \mathbb{N}$, define E_f by

$$E_f = \bigcup_{n \in \operatorname{Dom} f} E(\langle n, f(n) \rangle).$$

For example, if we let $E(x) = \{x\}$, then E_f is the graph of f. The functions $\mathcal{R}$ and $\tilde{\mathcal{R}}$ can also be thought of simple encodings; for example, if we set $E(\langle n, 1 \rangle) = R_n$ and $E(x) = \emptyset$ otherwise, then $E_{\mathbb{1}_A} = \mathcal{R}(A)$ for all sets A.

We have the following.

Theorem 4.3. For each simple encoding E there exists a function f such that f and the indicator function of E_f are not in the same nonuniform coarse, generic, dense, or effectively dense degrees.

Therefore, simple encodings cannot transform a function into a set whilst preserving its degree.

Proof. We will analyze 3 separate cases, according to how the densities of $E(x)$ behaves asymptotically.

Case 1: There are infinitely many n for which $\overline{\rho}(E(\langle n, k \rangle)) > 0$ for some k.

Let $n_0 < n_1 < \cdots$ be an infinite sequence of these numbers n, with corresponding witnesses k_i. We may assume that the set $\mathcal{N} = \{n_0, n_1, \dots\}$ has density 0, as we can replace $\mathcal{N}$ with a sparse subset. For each $\alpha : \mathbb{N} \to \{0, 1\}$ define $f_\alpha : \mathbb{N} \to \mathbb{N}$ by setting $f_\alpha(n_i) = k_i + \alpha(i)$ for each n_i and $f_\alpha(m) = 0$ if $m \notin \mathcal{N}$. We will show that f_α and E_{f_α} are not equivalent for some α.

If $\alpha \neq \beta$, say $\alpha(i) \neq \beta(i)$, then $E_{f_\alpha} \triangle E_{f_\beta}$ contains the set $E(\langle n_i, k_i \rangle)$, which has positive upper density, and thus $E_{f_\alpha} \triangle E_{f_\beta}$ itself has positive upper density. This means that, for each Turing machine e, the function $\Phi_e^{\mathcal{N}}$ can be a dense description of at most one of the sets E_{f_α}.

There are uncountably many such E_{f_α}, but only countably many $\mathcal{N}$-computable dense descriptions, so for at least some α the set E_{f_α} is not densely computable relative to $\mathcal{N}$. However, because $\mathcal{N}$ has density 0, the function f_α itself is effectively densely computable relative to $\mathcal{N}$. This means that even relative to $\mathcal{N}$, the nonuniform coarse, dense, effectively dense, and generic degrees of the function f_α do not contain the indicator function of the set E_{f_α}.

Case 2: There exists some $\varepsilon > 0$ such that, for infinitely many n, there exists some k and j where $\rho_j(E(\langle n, k \rangle)) > \varepsilon$.[1]

Again, let $n_0 < n_1 < \cdots$ be an infinite sequence of such n, with k_i and j_i being the corresponding witnesses, and assume that $\mathcal{N} = \{n_0, n_1, \ldots\}$ is sparse. For $\alpha : \mathbb{N} \to \{0, 1\}$ define f_α as before. Now, suppose that α and β differ at infinitely many places; we claim that $E_{f_\alpha} \triangle E_{f_\beta}$ has positive upper density.

Indeed, the set $E_{f_\alpha} \triangle E_{f_\beta}$ contains infinitely many sets $E(\langle n_i, k_i \rangle)$, with each set satisfying $\rho_{j_i}(E(\langle n_i, k_i \rangle)) > \varepsilon$ for some integer k_i. It is easy to see that if $\rho_{j_i}(E(\langle n_i, k_i \rangle)) > 0$ then the set $E(\langle n_i, k_i \rangle)$ must contain some element smaller than j_i. Since all $E(\langle n_i, k_i \rangle)$ are disjoint, this means that each j_i may only be repeated finitely many times (that is, for each j, the set $\{i \mid j_i = j\}$ is finite). Hence there are infinitely many distinct j_i such that $\rho_{j_i} E(\langle n_i, k_i \rangle) > \varepsilon$ in any infinite subcollection of the j_i. In turn, this means that $\rho_{j_i}(E_{f_\alpha} \triangle E_{f_\beta}) > \varepsilon$ for infinitely many distinct j_i, which shows that the set $E_{f_\alpha} \triangle E_{f_\beta}$ has positive upper density.

Since there are uncountably many α which differ in infinitely many places, there are

1. There is overlap between Case 1 and Case 2, but this is neither harmful nor relevant.

uncountably many E_{f_α} whose pairwise symmetric differences all have positive upper density. Therefore, the same conclusion as in Case 1 applies.

Case 3: Neither Case 1 nor Case 2 applies.

Because we are not in Case 1, there exists some N such that $\overline{\rho}(E(\langle n,k\rangle)) = 0$ for all k and all $n \geq N$.

If $n \geq N$, define ε_n by

$$\varepsilon_n = \sup_{k,j\in\mathbb{N}} \rho_j(E(\langle n,k\rangle)).$$

Because we are not in Case 2, we have $\lim_{n\to\infty} \varepsilon_n = 0$.

Construct the sequences $k_{N,i}, k_{N+1,i}, k_{N+2,i}, \ldots$ for $i = 0, 1$ as follows. First, let $k_{N,0} \neq k_{N,1}$ be arbitrary. Assume we defined $k_{N,i}, k_{N+1,i}, \ldots, k_{n-1,i}$ for $i = 0, 1$, and define

$$A_n = \bigcup_{\substack{N\leq \ell<n \\ i=0,1}} E(\big\langle \ell, k_{\ell,i}\big\rangle).$$

Because A_n is a union of finitely many $E(\langle \ell,k\rangle)$ for $\ell \geq N$, all of which have density 0, we have that $\rho(A_n) = 0$. Therefore, there is some positive M_n such that $\rho_j(A_n) < \varepsilon_n$ for all $j > M_n$. Choose $k_{n,0}$ and $k_{n,1}$ to be any two distinct integers such that neither $E(\big\langle n, k_{n,0}\big\rangle)$ nor $E(\big\langle n, k_{n,1}\big\rangle)$ contains elements smaller than M_n or N; such $k_{n,i}$ exist because the $E(\langle n,k\rangle)$ are all disjoint. This implies that, if $j > M_n$, then

$$\rho_j(A_{n+1}) = \rho_j(A_n) + \rho_j(E(\big\langle n, k_{n,0}\big\rangle)) + \rho_j(E(\big\langle n, k_{n,1}\big\rangle)) < 3\varepsilon_n.$$

Define $A = \bigcup_n A_n$. We may assume that the numbers M_n are increasing. If $M_n \leq j < M_{n+1}$, then

$$\rho_j(A) = \rho_j(A_{n+1}) \leq 3\varepsilon_n.$$

This shows that $\rho(A) = 0$.

Now, finally, given $\alpha : \mathbb{N} \to \mathbb{N}$, define $f_\alpha(n) = 0$ if $n < N$ and $f_\alpha(n) = k_{n,\alpha(n-N)}$

otherwise. For any two α, β, we have $E_{f_\alpha} \triangle E_{f_\beta} \subseteq A$, so all of the E_{f_α} are effectively densely computable relative to A. If α and β disagree on a set with positive upper density, then f_α and f_β cannot be A-densely computed by the same Turing machine. Since there is an uncountable family of functions α which pairwisely disagree on a set with positive upper density, there are functions α such that f_α is not A-densely computable, which in turn mean that there are functions α whose the nonuniform coarse, dense, effectively dense, and generic degrees do not contain the indicator function of the set E_{f_α}. $\square$

On the other direction, using error-correcting codes, we can show that functions that grow "slowly enough" are indeed equivalent to sets.

Theorem 4.4. Let f be an exponentially-bounded function; i.e. there exists a polynomial function p such that $f(n) \leq 2^{p(n)}$ for all n. Then there exists a set A which is equivalent to f, under all eight asymptotic notions of computability (coarse, generic, dense, effective dense; both uniform and nonuniform).

Another way of interpreting the $f(n) \leq 2^{p(n)}$ bound is to think of $f(n)$ as a binary string of length $\log_2 f(n)$; then this just says that the length of $f(n)$ is polynomially-bounded.

Proof. We may assume that $p(n) = n^k$ for some k without loss of generality.

For each n, let $E_n : \{0,1\}^{p(n)} \to \{0,1\}^{5p(n)}$ be a code capable of correcting up to $p(n)$ errors; i.e. if $x \neq y$ then $E_n(x)$ and $E_n(y)$ differ in at least $2p(n)$ entries. The existence of such code can be shown using standard tools from coding theory, or as a direct application of the probabilistic method. (Note that an existence proof is enough, as there are only finitely many functions $\{0,1\}^{p(n)} \mapsto \{0,1\}^{5p(n)}$, so we can just search for a suitable E_n.)

For each n, let τ_n be the $f(n)$-th binary string of length $p(n)$, in lexicographical order, and define A to be the concatenation of all $E_n(\tau_n)$. We claim that A is coarsely equivalent to f (the other notions have analogous proofs).

Let $b(n) = \sum_{i<2^n} 5p(i)$. Since $\sum_{i<m} i^k$ is asymptotically equal to $\frac{n^{k+1}}{k+1}$, for all large enough n, we have $5\frac{2^{nk+n}}{k+2} \leq b(n) \leq 5\frac{2^{nk+n}}{k}$.

Let g be a coarse approximation to f, and construct B in the same way we constructed A. Fix N so that $d_n(f,g) < \varepsilon$ if $n > N$. This means that at most $\varepsilon 2^n$ of the numbers in $[2^n, 2^{n+1})$ are incorrect, and each error makes at most $5p(2^{n+1})$ of the bits of B in the region $[b(n), b(n+1))$ to be incorrect. Hence at most $5 \cdot \varepsilon 2^{nk+n}$ of these bits are incorrect. Now if m satisfies $[2^m, 2^{m+1}) \cap [b(n), b(n+1)) \neq \emptyset$, these errors will contribute to $d_m(A,B)$, so that $2^m d_m(A,B) < 10 \cdot \varepsilon 2^{nk+k}$. But by the observation above, if n is large enough, we must have $2^m > b(n-1) > 5\frac{2^{nk+n-k-1}}{k+2}$, whence

$$d_m(A,B) < \frac{10 \cdot \varepsilon 2^{nk+n}}{5 \cdot 2^{nk+n-k-1}/(k+2)} = (k+2)2^{k+2}\varepsilon.$$

Since, for all ε, the inequality above is true for all large enough m, we conclude that $d(A,B) = 0$; i.e. B is a coarse description of A, whence A is coarsely reducible to f.

In the other direction, let B be a coarse approximation of A. Construct g by "undoing" the construction of A: for each n, let $c(n) = \sum_{i<n} 5p(i)$, let σ_n be the string $B(c(n))B(c(n)+1)\cdots B(c(n+1)-1)$ (i.e. the bits of B between $c(n)$ and $c(n+1)$), and let $g(n)$ be the element of $\{0,1\}^{p(n)}$ which minimizes the distance between σ_n and $E_n(g(n))$. We will prove that g is a coarse approximation of f.

Indeed, if $2^n d_n(f,g) > j$ (i.e. at least j bits between $[2^n, 2^{n+1})$ differ), then at least $jp(2^n)$ bits of B between $c(2^n)$ and $c(2^{n+1})$ differ from A. Let m be the largest number such that $2^m < c(2^n)$, so that the interval $[2^m, 2^{m+1})$ intersects $[c(2^n), c(2^n+1))$. Because $c(x)$ is asymptotically equal to $x^{k+1}/(k+1)$, we can show that $2^{m+k+2} > c(2^{n+1})$, so at least one of the intervals

$$[2^m, 2^{m+1}), [2^{m+1}, 2^{m+2}), \ldots, [2^{m+k+1}, 2^{m+k+2})$$

contains at least $jp(n)/(k+2)$ of the differences between A and B. Hence if i is this interval then

$$\begin{aligned} d_{m+i}(A,B) &> \frac{jp(2^n)/(k+2)}{2^{m+k+2}} \\ &= \frac{j2^{nk}}{(k+2)2^{m+k+2}} \\ &> \frac{j2^{nk}}{(k+2)2^{k+2}c(2^n)} \\ &> \frac{k}{(k+2)2^{k+2}} \cdot \frac{j2^{nk}}{2^{nk+n}} \\ &= \frac{k}{(k+2)2^{k+2}} \cdot j2^{-n}. \end{aligned}$$

In terms of $d_n(f,g)$, this means that if $d_n(f,g) > \varepsilon$, then $d_{m+i}(A,B) > \varepsilon \cdot \frac{k}{(k+2)2^{k+2}}$ for some $i < k+2$. So if f and g are not coarsely equivalent, neither are A and B, a contradiction. Hence f and g are coarsely equivalent, whence f is coarsely reducible to A. $\square$

CHAPTER 5

OPEN PROBLEMS

This chapter summarizes the open problems derived from the theorems in the previous chapters.

As mention in Section 4, it is not known whether sets are equivalent to functions. We restate this question here, for completeness.

Open Problem 4.1. For each of the eight reducibilities (the uniform and nonuniform versions of dense reducibility, generic reducibility, coarse reducibility, and effective dense reducibility), defined for functions, is it true that every degree contains the indicator function of a set?

The statement of Theorem 2.1.4 raises the following question.

Open Problem 5.1. Theorem 2.1.4 requires the sets to be 2-random. Can this condition be improved to 1-random?

In Igusa's proof of Theorem 1.3.1, we could have replaced "undefined" with $\square$ throughout, resulting in the theorem that there are no minimal pairs for relative effectively dense computability. Similarly, the same modifications applied to Theorem 2.1.4 yields the result that if A and B are 2-random then A and B do not form a minimal pair for effectively dense reducibility. However, the existence of minimal pairs is still open.

Open Problem 5.2. Do there exists minimal pairs for the effective dense degrees (both uniform and nonuniform)?

In Section 1.1.6, we provided examples of sets which are densely computable but neither coarsely nor generically computable (Example 1.1.29), which are coarsely computable but not densely computable (Example 1.1.25) and vice-versa (Example 1.1.26), and which are coarsely and generically computable but not effectively densely computable (Example 1.1.31). These non-implications between these asymptotic notions of computability entails non-implications in the associated reducibilities. For example, if A is the set from Example 1.1.29 then $A \leq_{\text{ud}} \emptyset$, but $A \not\leq_{\text{nc}} \emptyset$ and $A \not\leq_{\text{ng}} \emptyset$. However, the other directions are still open.

Open Problem 5.3. Does $A \leq_{\text{ned}} B$ implies $A \leq_{\text{nc}} B$ or $A \leq_{\text{ng}} B$? Does $A \leq_{\text{nc}} B$ and $A \leq_{\text{ng}} B$ implies $A \leq_{\text{nd}} B$? What about uniform reducibilities?

We also ask about the limits of Theorem 2.4.6.

Open Problem 5.4. Is it possible to prove Theorem 2.4.6 without the hypothesis that the intersection of the domains of the $\Phi_{f(k)}$ is dense?

We finish by restating this open problem from Chapter 3.

Open Problem 3.3.6. Are there any sets A with minimal Turing degree which also satisfy $K(A \upharpoonright n) > 2K(n)$ for all n?

REFERENCES

[1] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2016.

[2] Eric P. Astor, Denis R. Hirschfeldt, and Carl G. Jockusch. Dense computability, upper cones, and minimal pairs. *Computability*, 8(2):155–177, June 2019. doi:10.3233/COM-180231.

[3] Warwick de Launey and Daniel M. Gordon. On the density of the set of known Hadamard orders. *Cryptography and Communications*, 2(2):233–246, May 2010. doi:10.1007/s12095-010-0028-9.

[4] Rod Downey and Noam Greenberg. Turing degrees of reals of positive effective packing dimension. *Information Processing Letters*, 108(5):298–303, 2008. doi:10.1016/j.ipl.2008.05.028.

[5] Rodney G. Downey and Denis R. Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer New York, New York, NY, 2010. doi:10.1007/978-0-387-68441-3_2.

[6] Denis R. Hirschfeldt. A minimal pair in the generic degrees. *The Journal of Symbolic Logic*, 85(1):531–537, 2020. doi:10.1017/jsl.2019.77.

[7] Denis R. Hirschfeldt. Minimal pairs in the generic degrees, 2020. URL: https://www.youtube.com/watch?v=B0mLss1jbo8.

[8] Denis R. Hirschfeldt, Carl G. Jockusch, Rutger Kuyper, and Paul E. Schupp. Coarse reducibility and algorithmic randomness. *The Journal of Symbolic Logic*, 81(3):1028–1046, 2016. doi:10.1017/jsl.2015.70.

[9] Denis R. Hirschfeldt, Carl G. Jockusch, Timothy H. McNicholl, and Paul E. Schupp. Asymptotic density and the coarse computability bound. *Computability*, 5(1):13–27, February 2016. doi:10.3233/COM-150035.

[10] Denis R. Hirschfeldt, Carl G. Jockusch, and Paul E. Schupp. Coarse computability, the density metric, Hausdorff distances between Turing degrees, perfect trees, and reverse mathematics. *Journal of Mathematical Logic*, 2023. URL: https://arxiv.org/abs/2106.13118, doi:10.1142/S0219061323500058.

[11] Gregory Igusa. Nonexistence of minimal pairs for generic computability. *The Journal of Symbolic Logic*, 78(2):511–522, 2013. doi:10.2178/jsl.7802090.

[12] Carl G. Jockusch and Paul E. Schupp. Generic computability, Turing degrees, and asymptotic density. *Journal of the London Mathematical Society*, 85(2):472–490, 2012. doi:10.1112/jlms/jdr051.

[13] Carl G. Jockusch and Paul E. Schupp. *Asymptotic Density and the Theory of Computability: A Partial Survey*, pages 501–520. Springer International Publishing, Cham, 2017. doi:10.1007/978-3-319-50062-1_30.

[14] Jack H. Lutz and Neil Lutz. Algorithmic information, plane Kakeya sets, and conditional dimension. *ACM Trans. Comput. Theory*, 10(2), May 2018. doi:10.1145/3201783.

[15] Benoit Monin. An answer to the gamma question. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 730–738, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3209108.3209117.

[16] Piergiorgio Odifreddi. *Classical Recursion Theory: The Theory of Functions and Sets of Natural Numbers, Vol. 1.* Studies in Logic and the Foundations of Mathematics, Vol. 125. North Holland, paperback edition, 2 1992.

[17] Piergiorgio Odifreddi. *Classical Recursion Theory: The Theory of Functions and Sets of Natural Numbers, Vol. 2.* Studies in Logic and the Foundations of Mathematics, Vol. 143. North Holland, hardcover edition, 9 1999.

[18] Hartley Rogers, Jr. *Theory of recursive functions and effective computability (Reprint from 1967).* MIT Press, Cambridge, MA, USA, 1987.

[19] Robert Irving Soare. *Turing Computability: Theory and Applications*. Theory and Applications of Computability. Springer-Verlag Berlin Heidelberg, 2016.

[20] Charles Edmund Michael Yates. Initial segments of the degrees of unsolvability part ii: Minimal degrees. *The Journal of Symbolic Logic*, 35(2):243–266, 1970.