

Use-Bounded Strong Reducibilities

by

David R. Bélanger

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Pure Mathematics

Waterloo, Ontario, Canada, 2009

© David R. Bélanger 2009

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

We study the degree structures of the strong reducibilities (\leq_{ibT}) and (\leq_{cl}) , as well as (\leq_{rK}) and (\leq_{wtt}) . We show that any noncomputable c.e. set is part of a uniformly c.e. copy of (\mathbb{Q}, \leq) in the c.e. cl-degrees within a single wtt-degree; that there exist uncountable chains in each of the degree structures in question; and that any countable partially-ordered set can be embedded into the cl-degrees, and any finite partially-ordered set can be embedded into the ibT-degrees. We also offer new proofs of results of Barmpalias [1] and Lewis-Barmpalias [10] concerning the non-existence of cl-maximal sets.

Acknowledgements

This work would not have been possible without my advisor, Barbara F. Csimá, and her endless supply of encouraging words and relevant results. The paper's readers, Rahim Moosa and Ross Willard, greatly influenced (for the better) its eventual content and structure, and their patience through the final revisions has been hagiheroical. Most overdue are my thanks to the family, friends, and colleagues who have helped, for many years and in many ways, keep me from losing my heading or my head.

Contents

- 1 Introduction** **1**
 - 1.1 Other notation and concepts 6
 - 1.2 A note on distinctness 9

- 2 cl-Maximality** **15**
 - 2.1 On the cl degrees 16
 - 2.2 Corollaries 20

- 3 Embedding Linear Orderings** **21**
 - 3.1 Within a single degree 21
 - 3.2 Upper bounds and uncountable chains 23

- 4 Incomparable c.e. Sets** **29**
 - 4.1 Basic techniques and results 29
 - 4.2 Arbitrary countable partially-ordered sets 36
 - 4.3 Compression and c.e. reals 42

- References** **47**

Chapter 1

Introduction

The *ibT* and *cl* *reducibilities* are preorders on 2^ω , the power set of the natural numbers. Each is a strengthened version of the classical Turing reducibility, with a restriction on the computational resources allowed. Although the *ibT* reducibility has long had a place in computability-theoretic arguments, arising naturally as a special case of the Turing reducibility in, for example, certain permitting arguments, it was only recently isolated and named by Soare. The similar *cl* reducibility was devised by Downey, Hirschfeldt, and LaForte as a possible tool (or foundation) for the study and quantification of relative randomness between elements of 2^ω .

For our basic computability-theoretic notation, we conform largely to modern standards as presented in texts such as Cooper's *Computability Theory* [3], Nies's *Computability and Randomness* [13], and Soare's *Computability Theory and Applications* [17]. More specialised background on the *cl* and *rK* degrees can be found in Downey and Hirschfeldt's monograph *Algorithmic Randomness and Complexity* [5] or in their expository paper with Nies and Terwijn [6].

Informally, a **Turing functional** Γ can be thought of as an idealised computer that, given access to some extra information $A \in 2^\omega$, will perform algorithmic operations on its input $n \in \omega$ and, perhaps, halt after some finite number of steps and output a number $k \in \omega$. In this case, we say that $\Gamma^A(n)$ **converges**, written $\Gamma^A(n)\downarrow$, and that $\Gamma^A(n) = k$ —or, more succinctly, $\Gamma^A(n)\downarrow = k$. Otherwise, $\Gamma^A(n)$ **diverges**, denoted $\Gamma^A(n)\uparrow$.

A **partial computable function** is a partial function in $\omega \rightarrow \omega$ of the form $f(n) = \Gamma^\theta(n)$ for some Turing functional Γ , with domain $\text{dom}f = \{n \in \omega : \Gamma^\theta(n) \downarrow\}$. A **(total) computable function** is a partial computable function whose domain is all of ω . It is possible to talk about n -ary partial computable functions, or partial computable functions on a set other than ω , by composing a regular partial computable function with an appropriate effective mapping into or out of ω . A **Gödel numbering** is a computable bijection from some set onto ω . A set or relation is called **computable** if its characteristic function is computable. A set $W \in 2^\omega$ is called **computably enumerable (c.e.)** if it is the domain of some partial computable function f .

We say that a set $A \in 2^\omega$ is **Turing reducible** to a set $B \in 2^\omega$ if there is a Turing functional Γ that can fully decide the elements of A by looking at those of B , *i.e.*, $n \in A \iff \Gamma^B(n) \downarrow = 1$ and $n \notin A \iff \Gamma^B(n) \downarrow = 0$. This is written as $A \leq_T B$. Along with the other reducibilities we shall be studying, the Turing reducibility is both reflexive and transitive, thus forming a pre-ordering on 2^ω . If $A \leq_T B$ and $B \leq_T A$, then A and B are **Turing equivalent**, written $A \equiv_T B$; if $A \not\leq_T B$ and $B \not\leq_T A$, then we write $A \not\equiv_T B$. The central objects of study in computability theory are the partially-ordered **degree** structure that (\leq_T) induces on its equivalence classes, and this structure's restriction to certain subclasses of 2^ω such as the previously-mentioned c.e. sets or the Δ_2 sets, which we define presently:

Definition 1.1. (a) A set A is called a Σ_n^0 set or simply a Σ_n set if there is an $(n+1)$ -ary computable relation R such that

$$k \in A \iff (\exists x_1)(\forall x_2)(\exists x_3) \cdots (Qx_n)[(k, x_1, x_2, \dots, x_n) \in R],$$

where Q is “ \forall ” if n is even and “ \exists ” if n is odd.

(b) A set A is called Π_n^0 or Π_n if its complement \bar{A} is Σ_n .

(c) If A is both Σ_n and Π_n , then it is called Δ_n^0 or Δ_n .

A set is c.e. if and only if it is Σ_1 . This latter, more intuitive, characterisation sheds light on both the c.e. sets and the hierarchy of Σ_n sets.

If a Turing degree \mathbf{a} contains a c.e. set, then we call \mathbf{a} a **c.e. degree**. Similarly, if \mathbf{a} contains a Δ_2 set, then we say that \mathbf{a} is a **Δ_2 degree**.

When $A \leq_T B$ by functional Γ , we may write $A = \Gamma^B$. In this case, each computation $\Gamma^B(n)$ halts after some finite number of steps, and therefore seeks only finitely many elements of B . Define the **use** of the reduction at n , denoted $\text{use } \Gamma^B(n)$, to be the largest such element. Then, $\text{use } \Gamma^B$ is a function in $\omega \rightarrow \omega$. Different pre-orderings and degree structures can be created from the Turing reducibility by weakening the definition (allowing non-determinism, for example) or by strengthening it. The structures we shall be studying are strong forms of the Turing reducibility, usually obtained by placing restrictions on the use function.

Definition 1.2 (Soare [18]). *Let $A, B \in 2^\omega$ be two sets of natural numbers. We say that A is **identity-bounded Turing reducible** to B , written $A \leq_{ibT} B$, if there is a Turing functional Γ such that $A = \Gamma^B$ and, for all n , $\text{use } \Gamma^B(n) \leq n$.*

Definition 1.3 (Downey-Hirschfeldt-LaForte [7]). *We say that A is **computably Lipschitz reducible** to B , written $A \leq_{cl} B$, if there are a Turing functional Γ and a constant $c \in \omega$ such that, for all n , $\text{use } \Gamma^B(n) \leq n + c$.*

The computably Lipschitz reducibility (\leq_{cl}) was first named the **strong weak truth-table reducibility** (\leq_{sw}), and has also been called **linear reducibility** (\leq_ℓ) by Lewis and Barmpalias [10]. The term *computably Lipschitz* was introduced by Barmpalias and Lewis [2] after its behaviour on the Cantor space topology of 2^ω , and this notation is adopted in the Downey and Hirschfeldt's textbook [5]. That the *ibT* and *cl* reducibilities are inherently related is obvious: after all, an *ibT* reduction is exactly a *cl* reduction with constant $c = 0$. One of the goals of this paper is to elucidate the similarities, differences, and interdependencies of these two notions.

An **initial segment** of A is a set of the form

$$A \upharpoonright x := A \cap [0, x] ;$$

we use this here in place of the more traditional notation

$$A \upharpoonright x = A \cap [0, x) .$$

We denote by $2^{<\omega}$ the set of all finite subsets of ω , or, equivalently, the set of all finite binary strings. This set is in computable bijection with the natural numbers, for example, through the Gödel numbering $2^{<\omega} \rightarrow \omega$, $A \mapsto \Sigma\{x : x \in A\}$. Hence we may consider partial computable functions taking 2^ω as domain or codomain.

Definition 1.4 (Downey-Hirschfeldt-LaForte [7]). *We say that A is **relative prefix-free Kolmogorov reducible** to B , written $A \leq_{rK} B$, if, for some $k \in \omega$, there are Turing functionals $\Gamma_0, \dots, \Gamma_k$ such that, for each n , there is a $j \leq k$ with $\Gamma_j^B(n) \downarrow = A \upharpoonright n$ with $\text{use } \Gamma_j^B(n) \leq n$.*

Equivalently, we may gather these Γ_j into a single function, so that $(A \leq_{rK} B)$ iff there exist a binary partial computable function $f(\cdot, \cdot) : 2^{<\omega} \times \omega \rightarrow 2^{<\omega}$ and a constant $k \in \omega$ such that, for all $n \in \omega$, there is a $j \leq k$ with $f(B \upharpoonright n, j) \downarrow = A \upharpoonright n$.

The rK reducibility is thus a non-deterministic version of the ibT reducibility. The cl and rK reducibilities were introduced simultaneously by Downey, Hirschfeldt, and LaForte as possible measures of relative randomness, the idea being that if $A \leq_{cl} B$ or $A \leq_{rK} B$, then B ought to be in some intuitive sense more *random* a set than A . The specific notion they had in mind was a formulation by Martin-Löf [11]. Technical and philosophical aspects of the study of randomness can be found in Downey and Hirschfeldt’s book [5] and van Lambalgen’s dissertation [20], respectively.

We now recall a more classical example of a strong reducibility:

Definition 1.5. *A set A is **weak truth table reducible** to B , written $A \leq_{wtt} B$, if there are a Turing functional Γ and a total computable function $f(x)$ such that, for all n , $\text{use } \Gamma^B(n) \leq f(n)$.*

The weak truth table reducibility was so named because it weakened another classical reducibility, the truth table reducibility (tt). Partly in reaction to the attenuating bonds between names of reducibilities and their actual meaning—as exemplified in the progression from *truth table*, to *weak truth table*, to the *strong weak truth table* introduced a few paragraphs ago—the name **bounded Turing reducibility** (bT), coined by Soare, is gaining some currency as an alternative for “wtt”.

When investigating the properties of (\leq_{ibT}) and (\leq_{cl}) , it is sensible and expedient to look to the vast body of known results on the Turing and weak truth table degrees. For example, a classical result of Spector [19] states that there is a minimal (non-computable) Turing degree, *i.e.*, a noncomputable degree with no noncomputable degree strictly Turing below it. We might ask ourselves the analogous question: Are there minimal *ibT* or *cl* degrees? The answer, after some reflection, will be negative. Here we use a natural pair of counter-examples noted in Barmpalias and Lewis [2]:

Proposition 1.6. *If A is a non-computable set, and if we write $A + 1 = \{x + 1 : x \in A\}$ and $2A = \{2x : x \in A\}$, then $\emptyset <_{ibT} A + 1 <_{ibT} A$ and $\emptyset <_{cl} 2A <_{cl} A$.*

We might similarly wish to seek *maximal* or *complete* degrees. Listing out the algorithms that represent them, we can obtain a computable (though non-injective) enumeration $(\Phi_e)_{e \in \omega}$ of Turing functionals. Perhaps the best-known non-computable set is Turing's **halting set**, defined by $\emptyset' = \{e \in \omega : \Phi_e^\emptyset(e) \downarrow\}$. A relativisation of this construction yields the **Turing jump operator** $'$, by which, if $A \in 2^\omega$, then $A' = \{e \in \omega : \Phi_e^A(e) \downarrow\}$. The jump induced on the Turing degrees is well-defined, and, for any degree \mathbf{a} , it is known that $\mathbf{a} <_T \mathbf{a}'$. Hence there is no maximal Turing degree. When we restrict the ordering to a particular class of sets, however, we may be left with maximal degrees; the well-known Post Theorem, for example, tells us that the degree $\mathbf{0}'$ of the halting set is Turing complete among the Δ_2 sets, that is, $\mathbf{0}'$ is the greatest Δ_2 Turing degree. Restriction along these lines will yield a host of more-or-less natural questions to ask about a degree structure: Yu and Ding [21] settled one such by showing there is no *cl*-complete element among the left-c.e. reals (see Definition 1.12); Barmpalias [1] solved another by proving the non-existence of a *cl*-maximal element among the c.e. sets. Lewis and Barmpalias later showed in [10] that there are no *cl*-maximal sets in general. We shall, in Chapter 2, offer a new proof of this last result, and obtain new corollaries on certain classes, including the Δ_2 sets.

Changing our focus slightly, we might examine the structure under one reducibility within a single degree of another. For example, the *wtt* reducibility is stronger than the Turing, and so each *wtt* degree is contained entirely within a Turing de-

gree. On the other hand, Downey [4] showed that there exist non-trivial **strongly contiguous** c.e. Turing degrees, which contain exactly one wtt-degree. There is no exact analogue of this property between the cl and ibT degrees, or the wtt and cl degrees, as evidenced by the constructions in Proposition 1.6. Applied naively, however, these constructions will generate only linear orderings. In Chapter 3 we investigate what sort of linear orderings can be thus embedded, and in Chapter 4 we do what we can to salvage or forever bury some weakened notion of contiguity among the wtt, cl, and ibT degrees by searching for a degree in which all sets are totally ordered with respect to another reducibility. Our results here are strictly negative; we arrive at a number of results on embeddings of partial orderings into the c.e. cl or ibT degrees, and in particular that each c.e. wtt degree contains two cl-incomparable sets, and each c.e. cl degree contains two ibT-incomparable sets.

1.1 Other notation and concepts

We always equate a set A with its characteristic function as an infinite binary sequence:

$$A = (A(0), A(1), A(2), \dots) .$$

When building sets from sets, we usually write them in terms of arithmetic and set operations. For example, we define the **join** operation on the Turing reducibility:

$$A \oplus B = 2A \cup (2B + 1) = \{2n : n \in A\} \cup \{2m + 1 : m \in B\} .$$

When this becomes impossible or unwieldy, we define the characteristic function directly.

Recall that a Gödel numbering is a computable bijection from some countable set to ω . We fix one such numbering $\langle \cdot, \cdot \rangle : \omega \times \omega \rightarrow \omega$ and call it the **pairing function**. Using this, we define the so-called **infinite join** on sets $(A_k)_{k \in \omega}$:

$$\bigoplus_{n \in \omega} A_n = \{ \langle x, n \rangle : x \in A_n \} .$$

Similarly, we fix a computable bijection $\langle \cdot, \cdot, \cdot \rangle : \omega \times \omega \times \omega \rightarrow \omega$, and so on.

We use the symbols \exists^∞ and \forall^∞ to say *for infinitely many* and *for all but finitely many*, respectively. We use μ to represent the partial computable operation of taking the minimum element of a computable set, or, more generally, to denote the minimum of any set.

Recall our effective listing $(\Phi_e)_{e \in \omega}$ of Turing functionals. We shall refer to a such a functional as Φ_e when its index e is important; otherwise, we shall continue to name them Γ , Γ_1 , or Γ_2 . By setting

$$\hat{\Phi}_e^X(y) = \begin{cases} k & \text{if } \Phi_e^X(y) \downarrow = k \text{ with } \text{use } \Phi_e^X(y) \leq y \\ \uparrow & \text{otherwise} \end{cases}$$

we obtain an effective enumeration $(\hat{\Phi}_e)_{e \in \omega}$ of what we will call **ibT functionals**. Clearly, $A \leq_{ibT} B$ if and only if $A = \hat{\Phi}_e^B$ for some e . Altering the definition slightly more,

$$\tilde{\Phi}_e^X(y) = \begin{cases} k & \text{if } \Phi_e^X(y) \downarrow = k \text{ with } \text{use } \Phi_e^X(y) \leq y + e \\ \uparrow & \text{otherwise} \end{cases}$$

gives an enumeration $(\tilde{\Phi}_e)_{e \in \omega}$ of **cl functionals**. Again, if $A \leq_{cl} B$, then there is an e such that $A = \tilde{\Phi}_e^B$. Indeed, if $A = \Phi_i^B$ with use bounded by identity plus c , we can produce (for example, by adding some number of useless operations to the machine) an $e \geq c$ such that Φ_e behaves the same way as Φ_i , giving $A = \tilde{\Phi}_e^B$.

For any Turing functional Γ , any set $B \in 2^\omega$, and any finite number of steps s , we can let Γ_s be the functional describing Γ after s steps of computation:

$$\Gamma_s^B(x) = \begin{cases} \Gamma^B(x) & \text{if the computation halts after fewer than } s \text{ steps} \\ \uparrow & \text{otherwise} \end{cases}$$

One should be able to determine from the context when we are talking about this Γ_s and when we really mean the generic Γ_1 and Γ_2 mentioned earlier. The sequence $(\Gamma_s)_{s \in \omega}$ approximates Γ in the sense that

$$(\forall n \exists s) [\Gamma^B(n) \downarrow \rightarrow \Gamma_s^B(n) \downarrow]$$

and

$$(\forall n \forall s \forall t) [(\Gamma_s^B(n) \downarrow \wedge s \leq t) \rightarrow (\Gamma^B(n) \downarrow \wedge \Gamma_t^B(n) \downarrow \wedge \Gamma^B(n) = \Gamma_t^B(n))] .$$

If $\Phi_e = \Gamma$, we write $\Phi_{e,s} = \Gamma_s$. If $f = \Gamma^\emptyset$ is a computable function, then we write $f_s = \Gamma_s^\emptyset$.

From the enumeration $(\Phi_e)_{e \in \omega}$ of Turing functionals, there is a natural enumeration of the c.e. sets, namely $(W_e)_{e \in \omega}$, where $W_e = \text{dom}(\Phi_e^\emptyset)$, that is, $n \in W_e \iff \Phi_e^\emptyset(n) \downarrow$.

Definition 1.7. *A c.e. approximating sequence is a computable sequence $(A_s)_{s \in \omega}$ of finite sets $A_s \subseteq \{0, 1, \dots, s\}$ such that*

$$(\forall n \forall s)(\forall t \geq s) [n \in A_s \rightarrow n \in A_t] .$$

Any c.e. approximating sequence $(A_s)_{s \in \omega}$ will have a pointwise limit $A = \lim_s A_s$ (in the usual discrete topology on $\{0, 1\}$). If we define a partial computable function f by $f(n) = (\mu s)[n \in A_s]$, then $A = \text{dom} f$, so A is c.e.. On the other hand, for each c.e. W_e and $s \in \omega$, we may write $W_{e,s} = \text{dom}(\Phi_{e,s}^\emptyset) \upharpoonright s$. Then $(W_{e,s})_{s \in \omega}$ forms a c.e. approximating sequence, and $W_e = \lim_s W_{e,s}$. We normally assume, for a c.e. approximation $(A_s)_{s \in \omega}$, that $(\forall s) [|A_{s+1} \setminus A_s| \leq 1]$.

By analogy, we can define a number of other types of sets:

Definition 1.8. *A Δ_2 approximating sequence or computable approximation is a computable sequence $(A_s)_{s \in \omega}$ of finite sets $A_s \subseteq \{0, 1, \dots, s\}$ with a pointwise limit $A = \lim_s A_s$. This A is called a **limit computable** or **computably approximable set**.*

Definition 1.9. *An ω -c.e. approximating sequence is a computable sequence $(A_s)_{s \in \omega}$ of finite sets $A_s \subseteq \{0, 1, \dots, s\}$ such that there exists a total computable function f satisfying:*

$$(\forall n) [|\{s : A_{s+1}(n) \neq A_s(n)\}| \leq f(n)] .$$

The limit $A = \lim_s A_s$ is called an ω -c.e. set. In the case where $f \leq n$ for some constant n , A is called an n -c.e. set. Note that the 1-c.e. sets are exactly the c.e. sets.

With versions of the Shoenfield Limit Lemma [16] and Post's Theorem we can relate these notions back to our reducibilities and justify the name Δ_2 approximating sequence.

Theorem 1.10 (Limit Lemma). (i) A is limit computable $\iff A \leq_T \emptyset'$.

(ii) A is ω -c.e. $\iff A \leq_{wtt} \emptyset'$.

Theorem 1.11 (Post). A is Δ_2 $\iff A \leq_T \emptyset'$.

Definition 1.12. A **c.e. real approximation** is a computable sequence $(A_s)_{s \in \omega}$ of finite sets $A_s \subseteq \{0, 1, \dots, s\}$ with the property that

$$(\forall s \forall n)(\exists m < n)[n \in A_s \setminus A_{s+1} \rightarrow m \in A_{s+1} \setminus A_s] .$$

A c.e. real approximation necessarily has a limit; we can see by induction on n that the initial segment $A_s \upharpoonright n$ changes in at most $2^{n+1} - 1$ stages. This $A = \lim_s A_s$ is called a **(left-) c.e. real** or an **almost computable set**. We normally assume that $|A_{s+1} \setminus A_s| \leq 1$ for all s .

Thinking of A as the real number whose decimal expansion is $0.A(1)A(2)A(3)\dots$, a c.e. real approximation is exactly a computable, increasing sequence of dyadic rationals whose limit is A in the usual norm. The c.e. reals enjoy a position in the study of algorithmic randomness on par with that of the c.e. sets in classical computability theory: a detailed apologia can be found in Downey-Hirschfeldt [5].

1.2 A note on distinctness

We have in a short time defined—formally or not—no fewer than five orderings on 2^ω , namely, the *ibT*, the *cl*, the *rK*, the *wtt*, and the *T* reducibilities. We know that the *wtt* reducibility is strictly weaker than the *T*: an example of a set $A \in 2^\omega$ such that $A \leq_T \emptyset'$ but $A \not\leq_{wtt} \emptyset'$ can be constructed from the *T* and *wtt* versions of the Limit Lemma 1.10. Before immersing ourselves too wholly in the fine points of their structure, we will verify that our objects of study are actually distinct, and decide precisely which are stronger or weaker than which. Most of these questions are settled in, or follow from, the paper in which Downey, Hirschfeldt, and LaForte first introduced *cl* and *rK* [7], as we shall see.

Some relationships are immediate from the mere definitions:

Proposition 1.13.

$$A \leq_{ibT} B \Rightarrow A \leq_{cl} B \Rightarrow A \leq_{utt} B$$

If $A \leq_{cl} B$ via cl functional Γ with use bounded by identity plus c , we can produce, for each subset $\sigma \subseteq \{0, 1, \dots, c-1\}$, an ibT functional Γ_σ such that

$$\Gamma_\sigma^B(n) = \Gamma^{(B \upharpoonright n) \cup \{n+1+x : x \in \sigma\}}(n) .$$

Since there will always be a σ such that $B \cap [n+1, n+c] = \{n+1+x : x \in \sigma\}$, the set $\{\Gamma_\sigma : \sigma \subseteq \{0, 1, \dots, c-1\}\}$ describes an rK reduction from A to B :

Proposition 1.14 (Downey-Hirschfeldt-LaForte [7]).

$$A \leq_{cl} B \Rightarrow A \leq_{rK} B$$

A rather more difficult result relates the rK and T reducibilities. Here we use the proof from [7].

Proposition 1.15 (Downey-Hirschfeldt-LaForte [7]).

$$A \leq_{rK} B \Rightarrow A \leq_T B$$

Proof. Let k be the smallest such that there is an f where $f(\cdot, 0), \dots, f(\cdot, k)$ witness $A \leq_{rK} B$. Assuming that, for a given string $X \in 2^{<\omega}$, no two $f(X, i), f(X, j)$ halt on the same stage s , define a new binary partial computable function g to give these values in order of the functions halting:

$$g(X, y) = \begin{cases} f(X, i) & \text{if } i \leq k \wedge f_s(X, i) \downarrow \wedge f_{s-1}(X, i) \uparrow \wedge |\{\ell \leq k : f_s(X, \ell) \downarrow\}| = y \\ \uparrow & \text{if there is no such } i \end{cases}$$

Then $A \leq_{rK} B$ through g . If there is a largest N such that, for all $j \leq k$, $f(B \upharpoonright N, j) \downarrow$, then for $n > N$, $g(B \upharpoonright n, k)$ is never used in the rK-reduction, allowing us to get an rK-reduction using only $g(B, j)$ such that $j \leq k-1$. This contradicts our choice of k , so there must be infinitely many N such that, for all $j \leq k$, $f(B \upharpoonright N, j) \downarrow$. We may, by waiting, B -compute an ascending sequence $N_0 < N_1 < \dots$ of such N .

Then, for any m , we can B -compute up to $k + 1$ different possibilities, one of which is correct, for $A \parallel N_m$. For each C that is rK-reducible to B through this f and k , we must have $C \parallel N_m = \sigma$ for one of these possibilities σ . If there were at least $k + 2$ such C , there would be an m such that the $C \parallel N_m$ were all different. This is impossible, so there are at most $k + 1$ different C . We can therefore fully B -decide our A by choosing an m such that all $C \parallel N_m$ are different and, with the finite knowledge in $\sigma := A \parallel N_m$, determining $A \parallel N_n$ for all $n > m$ by finding an ℓ large enough that for exactly one $j \leq k$ there is an $i \leq k$ such that

$$A \parallel N_m = f(B \parallel N_n, j) \parallel N_m \text{ and } f(B \parallel N_n, j) = f(B \parallel N_\ell, i) \parallel N_n$$

□

The constructions in Proposition 1.6 establish that ibT , cl , and wtt are not the same; for, if A is not computable, we nonetheless have $A \equiv_{\text{cl}} A + 1$ and $A \equiv_{\text{wtt}} 2A$. The structures of the cl and rK degrees are also different, as guaranteed by, say, Raichev's result in Theorem 3.6 combined with Proposition 2.6.

Downey, Hirschfeldt, and LaForte [7] proved that there were an A and a B such that $A \leq_{\text{rK}} B$ but $A \not\leq_{\text{wtt}} B$, and in particular $A \not\leq_{\text{cl}} B$. We shall exhibit here such an A and B , whose construction introduces a favourite and recurring strategy of ours: that of making two sets sufficiently spread out or shifted to balk a given use-bounded reducibility. We let f be the busy-beaver-style function

$$f(n) = 2 + \max_{e < n} \left(\{0\} \cup \{f(e)\} \cup \{\Phi_e^\emptyset(n) : \Phi_e^\emptyset(n) \downarrow\} \right) .$$

and $C \in 2^\omega$ be such that $C \not\leq_T f$.

Define

$$A = \{f(x) : x \in C\}$$

and

$$B = \{f(x) : x \in \omega\} \cup \{f(x + 1) + 1 : x \in C\} .$$

Then $A \leq_{\text{rK}} B$: for any $n \geq 1$, we can ibT -compute from B the unique ℓ such that $f(\ell) \leq n < f(\ell + 1)$, and we can cl -compute from B the initial segment

$C \uparrow (\ell - 1)$. Hence $A \leq_{rK} B$ by the two cl-functionals

$$\begin{aligned}\Gamma_1^B(n) &= \{f(x) : x \in C \uparrow (\ell - 1)\} \\ \Gamma_2^B(n) &= \{f(x) : x \in C \uparrow (\ell - 1)\} \cup \{f(\ell)\}\end{aligned}$$

where $A \uparrow n = \Gamma_1^B(n)$ if $\ell \notin C$ and $A \uparrow n = \Gamma_2^B(n)$ if $\ell \in C$.

On the other hand, if $A \leq_{wtt} B$ through a given Turing functional Φ_e and computable bound $g = \Phi_e^\emptyset$, then for $\ell > i$ we have $g(f(\ell)) < f(\ell + 1)$. By definition of B , we know that $B \uparrow (f(\ell + 1) - 1) = B \uparrow (f(\ell) + 1)$. Thus we can compute $A(f(\ell))$ from

$$B \uparrow g(f(\ell)) = B \uparrow (f(\ell) + 1) \equiv_{wtt} \{f(k) : k \leq \ell\} \cup \{f(k) + 1 : k \in C \uparrow (\ell - 1)\}.$$

In particular, given an oracle to f , we can compute $C(\ell)$ from $C \uparrow (\ell - 1)$. Hence $C \leq_T f$, a contradiction.

To resolve the other half of the relationship between rK and wtt, as well as rK and T, will require methods somewhat more sophisticated. We shall tacitly use the technique of *permitting*, introduced in Chapter 4, through one of that chapter's theorems, together with the following result, essentially due to Downey, Hirschfeldt, and LaForte in [7]. The proof here is more basic (as are our intentions) and contains ideas, particularly that of *counting* events by tallying a c.e. approximation, that will resurface in Chapter 4.

Lemma 1.16. *For each c.e. real A , there is a c.e. set W such that $A \equiv_{wtt} W$. Hence, the c.e. real wtt-degrees co-incide with the c.e. wtt-degrees.*

Proof. Take A with c.e. real approximation $(A_s)_{s \in \omega}$. Any initial segment $A \uparrow n$ can change at most $2^{n+1} - 1$ times throughout the approximation, *i.e.*, there are at most $2^{n+1} - 1$ values of s for which $A_{s+1} \uparrow n \neq A_s \uparrow n$. In particular, any given n can enter or leave A no more than 2^{n+1} times. We can use a c.e. set W simply to *count* these changes as they occur: each time the element n enters or leaves A , we add a new element to $W \cap [2^{n+1}, 2^{n+2})$. If we know the final tally $|W \cap [2^{n+1}, 2^{n+2})|$, we can then determine whether $n \in A$ simply by checking the parity.

Formally, define a c.e. approximating sequence:

$$W_0 = \emptyset$$

$$W_{s+1} = W_s \cup \{(\mu m) [m \geq 2^{n+1} \wedge m \notin W_s] : n \in A_{s+1} \setminus A_s \vee n \in A_s \setminus A_{s+1}\}$$

with $W := \lim_s W_s$. If we know an s such that $A_s \upharpoonright n = A \upharpoonright n$, then we know that $W_s \upharpoonright (2^{n+2} - 1) = W \upharpoonright (2^{n+2} - 1)$; hence, taking the A -computable function

$$g(n) = (\mu s) [A_s \upharpoonright n = A \upharpoonright n] ,$$

we can define a functional Γ for $0 \leq k < 2^{n+1}$ by

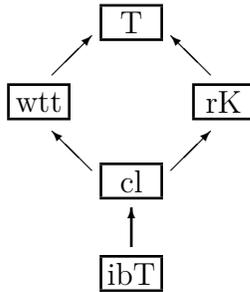
$$\Gamma^A(2^{n+1} + k) = \begin{cases} 1 & \text{if } (\exists_{>k} s < g(n)) [n \in A_{s+1} \setminus A_s \vee n \in A_s \setminus A_{s+1}] \\ 0 & \text{otherwise} \end{cases}$$

to witness $W \leq_{wtt} A$ (since it looks at $A \upharpoonright n$ only to calculate $g(n)$, the use is logarithmic, giving $W \leq_{ibT} A$). On the other hand, we can define a functional Γ_1 by

$$\Gamma_1^W(n) = \begin{cases} 1 & \text{if } |W \cap [2^{n+1}, 2^{n+2}]| \text{ is odd} \\ 0 & \text{otherwise} \end{cases}$$

to witness $A \leq_{wtt} W$. □

If $A \leq_{wtt} B$ were to imply that $A \leq_{rK} B$, then each c.e. real would also be rK-equivalent to a c.e. set; as we shall see in Proposition 4.12, however, there is a c.e. real that is not rK-below any c.e. set. Therefore, wtt and rK are, in this sense, independent of one another. The same argument also shows that $A \leq_T B$ does not imply that $A \leq_{rK} B$.



Chapter 2

cl-Maximality

We begin this section with a proof of a relatively simple fact. The reader will notice that it is nothing but one of the transformations of Proposition 1.6 performed backward. We show its correctness in full to motivate and illuminate our strategy against more exciting problems.

Proposition 2.1. *There are no ibT -maximal sets.*

Proof. Suppose we have some $A \subseteq \omega$. If A is computable, then A is not ibT -maximal since, for example, $A <_{ibT} \emptyset'$. So, assume that A is not computable. Let the set W be A shifted one space to the left as a binary string, namely,

$$W = A \dot{-} 1 = \{a : a + 1 \in A\} .$$

We shall now see that $A <_{ibT} W$.

Claim: $A \leq_{ibT} W$.

Construct a functional Γ such that, on oracle W ,

$$\Gamma^W(x) = \begin{cases} A(0) & \text{if } x = 0 \\ W(x - 1) & \text{otherwise} \end{cases}$$

Then $A = \Gamma^W$ and this reduction is identity-bounded.

Claim: $W \not\leq_{ibT} A$.

Suppose, to the contrary, that $W \leq_{ibT} A$ via functional Γ_1 . Then, for all $x \geq 0$, Γ_1 computes $A(x + 1) = W(x)$ using $A \upharpoonright x$. By induction, then, Γ_1 can compute

any $A(x + 1)$ using only an oracle to $A(0)$. This implies that A is computable, a contradiction.

□

In almost any other context, A and W would be indistinguishable—that is, the sets A and W occur simultaneously in just about any class that we could want. It is clear, for example, that A and W share the same Turing degree. With little effort, we can get a small army of small results.

Corollary 2.2. *There are no ibT -maximal sets when restricted to the classes of c.e. sets, the n -c.e. sets, the ω -c.e. sets, or Δ_2 sets.*

Proof. If $(A_s)_{s \in \omega}$ is a computable approximation for A , then $(A_s \dot{-} 1)_{s \in \omega}$ is a computable approximation for $A \dot{-} 1$. □

2.1 On the cl degrees

To achieve our goal on the ibT degrees, we dropped a finite amount of information—namely, $A(0)$ —from the set A , shifting the rest over to take its place, and then argued that, if there were a machine that could always tell what the next element of our new set would be based only on previous elements, then A must have been computable. Next, we wish to apply the same sort of strategy to the cl -ordering. Because in a cl -reduction the machine may skip any finite number of entries, we must toss out an infinite amount of information—which, in general, we might not be able to recover. In certain cases, however, we can remove a *well-behaved* infinite subset and not feel the loss. We say that a set A is **bi-immune** if neither A nor \overline{A} contains an infinite computable subset.

Lemma 2.3. *If a set A is not bi-immune, then A is not cl -maximal.*

Proof. If A is computable, then $A <_{cl} \emptyset'$. So, assume that A is not computable.

First, say $B \subseteq A$ is an infinite computable subset. Then, we may define a set W by:

$$W(x) = A(x + |B \upharpoonright x|) ,$$

hence W is simply the set A with the subset B deleted and the other elements shifted over in its place. Then A is recoverable from W by the Turing functional:

$$\Gamma^W(x) = \begin{cases} 1 & \text{if } x \in B \\ W(x - |B \upharpoonright x|) & \text{otherwise} \end{cases}$$

Thus $A = \Gamma^W$, and this reduction is identity-bounded. In particular, $A \leq_{cl} W$.

Claim: $W \not\leq_{cl} A$.

Suppose, to the contrary, that $W = \Gamma_1^A$ for some Γ_1 with use bounded by identity plus a constant c . Let N be the $(c+1)$ -th smallest element of B . For each $n \geq N$, $A(n)$ can be computed by Γ using oracles to only $W \upharpoonright (n - c - 1)$; each entry of $W \upharpoonright (n - c - 1)$ can, in turn, be computed by Γ_1 using only $A \upharpoonright ((n - c - 1) + c) = A \upharpoonright (n - 1)$. Hence, iterating and composing Γ and Γ_1 appropriately, we can produce an algorithm to compute any $A(n)$ from $A \upharpoonright (n - 1)$ for any $n \geq N$. Recursing on this algorithm, and including the additional finite information of $A \upharpoonright N$, we can decide effectively whether $n \in A$ for any n . So A is computable, a contradiction.

If \bar{A} has a infinite computable subset B , then we may use it as above to construct W such that $\bar{A} <_{cl} W$. Because $A \equiv_{cl} \bar{A}$, we then get $A <_{cl} W$. \square

This gives immediately an alternate proof of a theorem of Barmpalias:

Corollary 2.4 (Barmpalias [1]). *There are no cl-maximal c.e. sets. That is, for every c.e. set A , there exists a c.e. set W such that $A <_{cl} W$.*

Proof. If A is computable, then we are done. Otherwise, there is an infinite, co-infinite, computable subset $B \subseteq A$ (see, for example, Soare [17]). We can then use the construction in Lemma 2.3 to find a set W such that $A <_{cl} W$.

Claim: W is c.e..

Take a c.e. approximation $(A_s)_{s \in \omega}$ of A . We produced W from A by deleting the elements of B and shifting the rest into their place. In the same way, we can delete B from each A_s to construct a c.e. approximation for W :

$$W_s(x) = A_s(x + |B \upharpoonright x|) .$$

□

There are no n -c.e. bi-immune sets, so this can be generalised:

Corollary 2.5. *There are no cl -maximal n -c.e. sets for any $n \geq 1$.*

Sadly, this precise method can only take us so far: there *do* exist bi-immune sets, even among such relatively tame classes as the ω -c.e. sets. We shall redress the proof by creating two sets U and V such that if $A \not\leq_{cl} U$ then $A <_{cl} V$. The set U will resemble W from Proposition 2.1, except in that rather than shifting the whole sequence, we shift a certain *subsequence* whose entries become more and more spread out. The set V will be made from A by removing this subsequence entirely and shifting the rest to compensate, in the manner of W from Lemma 2.3. This will give a new proof of a result of Lewis and Barmpalias, along with a number of new corollaries:

Proposition 2.6 (Lewis-Barmpalias [10]). *There are no cl -maximal sets.*

Proof. Take a set A and consider it as a binary string. Assume, once again, that A is not computable.

For $a = 0, 1, 2, 3, \dots$ let

$$n_a = \frac{(a+1) \cdot (a+2)}{2} - 1 = -1 + (1 + 2 + 3 + \dots + (a+1)).$$

This gives $n_0 = 0$, $n_1 = 2$, $n_2 = 5$, and so on, with $n_a = n_{a-1} + a + 1$.

Construct a new set U from A by moving back successive n_a :

$$U(x) = \begin{cases} A(n_{a+1}) & \text{if } x = n_a \text{ for some } a \\ A(x) & \text{otherwise} \end{cases}$$

Then we can compute most of A from U by moving each n_a back in the other direction; the single entry $A(0)$ whose information is not present in U can be coded directly into the computation. This reduction is identity-bounded, and, in particular, $A \leq_{cl} U$.

If $U \not\leq_{cl} A$, then $A <_{cl} U$ and we are done; so suppose that $U \leq_{cl} A$, via some functional Γ with use bounded by identity plus constant $c \geq 1$. Construct a new set V from A by deleting each n_a -th entry:

$$V(x) = A(x + |\{n_a : a \in \omega\} \upharpoonright x|) .$$

Claim: $A \leq_{cl} V$.

For any $x \notin \{n_a : a \in \omega\}$, we can just read off

$$A(x) = V(x - |\{n_a : a \in \omega\} \upharpoonright x|) .$$

Given $m = n_a$ with $a \geq c$, we may use Γ to decide $A(n_a) = U(n_{a-1})$ from only the initial segment

$$A \upharpoonright (n_{a-1} + c) \subseteq A \upharpoonright (n_a - 1) ,$$

so, by recursing, all of A can be solved-for using Γ , an identity-bounded oracle to V , and the finite information of $A \upharpoonright n_c$.

Claim: $V \not\leq_{cl} A$.

Suppose that $V \leq_{cl} A$ by Turing functional Γ_1 with use bounded by identity plus d . We shall derive a contradiction by showing A to be computable. Suppose that $m > n_c$ and $m > n_d$. If $m = n_a$ for some a , then we can compute $A(m) = U(n_{a-1})$ using Γ with an oracle to $A \upharpoonright (n_{a-1} + c)$, since $\text{use } \Gamma^U$ is bounded by identity plus c . But we have $n_a > n_c$, giving in particular that $a > c$, so

$$n_{a-1} + c \leq n_a - 1 = m - 1 ;$$

hence Γ can be used to compute $A(m)$ from $A \upharpoonright (m - 1)$.

For all other m , we can apply Γ_1 to obtain $A(m) = V(m - |\{a \in \omega : n_a \leq m\}|)$ from

$$A \upharpoonright (m - |\{a \in \omega : n_a \leq m\}| + d) \subseteq A \upharpoonright (m - 1) ,$$

this time by the fact that $m > n_d$. By applying Γ and Γ_1 repeatedly in this way and hard-coding the finite information in $A \upharpoonright n_c$ and $A \upharpoonright n_d$, we can produce an algorithm to compute A —a contradiction.

□

2.2 Corollaries

We may use the construction in Proposition 2.6, as we did with Lemma 2.3, to build from any computable approximation $(A_s)_{s \in \omega}$ two new computable approximations $(U_s)_{s \in \omega}$ and $(V_s)_{s \in \omega}$ for U and V , respectively. Furthermore, if $(A_s)_{s \in \omega}$ happens to be an ω -c.e. approximating sequence, with computable bound f on the changes, then we can create two new computable functions

$$g(x) = \begin{cases} f(n_{a+1}) & \text{if } x = n_a \text{ for some } a \\ f(x) & \text{otherwise} \end{cases}$$

and

$$h(x) = f(x + |\{n_a : a \in \omega\} \upharpoonright x|) ,$$

bounding the changes on $(U_s)_{s \in \omega}$ and $(V_s)_{s \in \omega}$, respectively. Hence:

Corollary 2.7. *There are no cl-maximal elements among the Δ_2 sets or the ω -c.e. sets.*

We can generalise in another direction by performing the same computable transformation on any computable relation:

Corollary 2.8. *There are no cl-maximal sets among any of the Σ_n^0 , Π_n^0 , or Δ_{n+1}^0 sets, for $n \geq 1$.*

Chapter 3

Embedding Linear Orderings

Given any non-zero cl-degree \mathbf{a} , we can construct, through repeated application of Propositions 1.6 and 2.6, a copy of (\mathbb{Z}, \leq) contained in the cl-degrees, with \mathbf{a} as an element. We might then ask: What other sorts of countable order types can be embedded? We shall answer this question in its greatest generality for certain degrees \mathbf{a} in Chapter 4; for the moment, we restrict ourselves to *linear* order types. This question is settled on the Turing degrees, as every countable linear order type can be embedded into (\mathbb{Q}, \leq) , which can in turn be embedded into the Turing degrees by the Sacks Density Theorem [15]. Similarly, if we can manage to embed \mathbb{Q} into the cl-degrees, our question will be answered.

3.1 Within a single degree

We will push the question of countable linear embeddings as far as we reasonably can by embedding \mathbb{Q} into the cl-degrees into any given nonzero wtt-degree. The sheer strength of the result will lead us to ask what other structures can always be embedded, both globally and within a given degree. This question forms the basis for Section 3.2 and for Chapter 4. For now:

Proposition 3.1. *The rationals \mathbb{Q} can be embedded as a linear ordering into the cl-degrees. In fact, this can be done within any non-computable wtt-degree \mathbf{a} . Moreover, if \mathbf{a} is a c.e. degree, then the image of \mathbb{Q} is uniformly c.e.; that is, their indices*

in the enumeration $(W_e)_{e \in \omega}$ can be found using a single procedure.

Proof. Recall Proposition 1.6, wherein we multiplied a given noncomputable set A by a factor of 2 to *stretch* it and get $A \succ_{cl} 2A$. We could repeat this method for successive powers of two to get an infinite sequence $A \succ_{cl} 2A \succ_{cl} 4A \succ_{cl} 8A \succ_{cl} \dots$, but we can see immediately that every positive natural number can be included: $A \succ_{cl} 2A \succ_{cl} 3A \succ_{cl} 4A \succ_{cl} \dots$. The obvious way, then, to get the order type of \mathbb{Q} would be to multiply A by rational coefficients. While it does not make sense to multiply these sets by a negative number, say, the intuition is largely correct, and, other than some technicalities with rounding, the construction we shall use is essentially that in Proposition 1.6 repeated infinitely many times.

It suffices to embed the open interval $(0, 1) \cap \mathbb{Q}$, which is isomorphic to \mathbb{Q} as a linear ordering. By reversing the relation we may, in fact, embed it backwards, and shall do just that in order to simplify the arithmetic. For each $q \in (0, 1) \cap \mathbb{Q}$, define the corresponding function $f_q : \omega \rightarrow \omega$ by $f_q(x) = \lfloor (1+q) \cdot x \rfloor$. This has a left-inverse $g_q(x) = \left\lceil \frac{x}{1+q} \right\rceil$ (to see this, note that $0 \leq (1+q) \cdot x - f_q(x) < 1$, and hence $0 \leq x - \frac{f_q(x)}{1+q} < 1$). These f_q and g_q are computable, so for any set A , $A \equiv_T f_q(A)$.

Now, if $q < s$ with $q, s \in (0, 1) \cap \mathbb{Q}$, then for all $n \in f_s(\omega)$, $n \in f_s(A) \iff f_q(g_s(n)) \in f_q(A)$. Since $q < s$, we have $f_q(g_s(n)) \leq n$; thus, $f_s(A) \leq_{ibT} f_q(A)$, and in particular $f_s(A) \leq_{cl} f_q(A)$. In order to show that this reduction is strict, we suppose that $f_q(A) \leq_{cl} f_s(A)$ and produce a contradiction by showing that A must be computable. Take Γ and c such that $f_q(A) = \Gamma^{f_s(A)}$ and use $\Gamma^{f_s(A)} \leq id + c$. Notice that, for all n , $A \Vdash n$ is computable via g_q from $f_q(A) \Vdash f_q(n)$, which is computable via Γ from $f_s(A) \Vdash f_q(n) + c$, which is computable via f_s from $A \Vdash g_s(f_q(n) + c)$.

Let $d = \left\lceil \frac{c+1}{s-q} \right\rceil$. Then, for all $n > d$, we have

$$\begin{aligned} f_s(n) - f_q(n) &= \lfloor (1+s) \cdot n \rfloor - \lfloor (1+q) \cdot n \rfloor \geq (1+s) \cdot n - (1+q) \cdot n - 1 \\ &= (s-q) \cdot n - 1 > (s-q) \cdot d - 1 \geq c + 1 - 1 = c. \end{aligned}$$

That is, for all $n > d$, $f_s(n) > f_q(n) + c$. Hence, $A \Vdash g_s(f_q(n) + c)$ is a proper initial segment of $A \Vdash n$. So we can compute $A \Vdash n$ from $A \Vdash (n-1)$ and, as in previous proofs, by iterating this method we can compute all of A from the finite segment $A \Vdash d$. \square

Corollary 3.2. *Any countable linear ordering can be so embedded (perhaps sacrificing uniformity).*

3.2 Upper bounds and uncountable chains

We might similarly wish to ask, given a linear ordering in the cl-degrees, how we can extend that ordering. For example, on the Turing ordering, any countable collection $\{A_n\}_{n \in \omega}$ of sets from 2^ω is bounded above by the infinite join. This bound is not a supremum, however: Kleene and Post [9] used an embedding of (\mathbb{Q}, \leq) into the Turing degrees to see that, if every countable set had a supremum, then using Dedekind cuts one could then construct a copy of (\mathbb{R}, \leq) in the Turing degrees. Such an embedding is impossible, as there are only countably many Turing functionals, and hence at most countably many Turing degrees below a given degree. By Proposition 3.1, the same argument applies to countable collections of cl-degrees.

There is nonetheless some structure to the upper bounds of a sequence of Turing degrees, as can be seen from Spector's remarkable result:

Theorem 3.3 (Kleene-Post-Spector [19]). *Every strictly ascending sequence of Turing degrees $\mathbf{a}_1 <_T \mathbf{a}_2 <_T \mathbf{a}_3 <_T \dots$ has an exact pair \mathbf{b}, \mathbf{c} , that is, one such that $\mathbf{a}_n \leq_T \mathbf{b}, \mathbf{c}$ for each n , and for each degree \mathbf{d} we have $\mathbf{d} \leq_T \mathbf{b} \wedge \mathbf{d} \leq_T \mathbf{c} \Rightarrow (\exists n)[\mathbf{d} \leq_T \mathbf{a}_n]$.*

There are a number of other interesting theorems concerning upper bounds to such a sequence. In order for any of them to have analogues on cl, each cl-ascending sequence must at least be bounded above. We shall show that even this is not the case. The following lemma will be one of our main tools.

Lemma 3.4. *If \mathcal{P} is a non-empty partially-ordered set such that no element is maximal and each ascending sequence $a_1 < a_2 < a_3 < \dots$ has an upper bound, then \mathcal{P} has an uncountable chain.*

Proof. For each element a , choose another element $S(a) > a$. Pick some element a_0 ; then $a_0 < S(a_0) < S(S(a_0)) < \dots$ forms an infinite chain. Now, assume that each chain is countable. Given an infinite chain \mathcal{C} with no maximal element, we

may list its elements c_1, c_2, \dots , in no particular order. Defining $\{m_1 < m_2 < \dots\} = \{n : (\forall k < n)[c_k < c_n]\}$, we take the cofinal sequence \mathcal{D} within \mathcal{C} given by $\mathcal{D} = (c_{m_1} < c_{m_2} < \dots)$. Then \mathcal{D} has an upper bound b in \mathcal{P} , and this b must also bound \mathcal{C} . By Zorn's Lemma, then, \mathcal{P} has a maximal element—a contradiction. \square

This has a famous consequence on the Turing reducibility (and also the wtt reducibility):

Corollary 3.5. *The Turing degrees have an uncountable chain.*

Proof. There is no maximal element because any Turing degree \mathbf{a} is strictly below its jump \mathbf{a}' . Every infinite ascending sequence has an upper bound by Theorem 3.3, and the result follows by Lemma 3.4. \square

In fact, this proves that *any* countable chain in the Turing degrees is part of an uncountable chain. We shall see that this is not true of the ibT or cl degrees. To begin, we introduce a theorem of Raichev.

Theorem 3.6 (Raichev [14]). *There exists an A such that, if $A \leq_{cl} B$, then $A \equiv_{rK} B$. (Indeed, this is true of any A that is ML-random in the sense of Martin-Löf [11].)*

Lewis and Barmpalias [10] later, independently, uncovered a weaker version, with $A \equiv_T B$ in the place of $A \equiv_{rK} B$. They dubbed any A with this property **quasi-maximal**. We can produce two T-incomparable quasi-maximal sets (for example, T-incomparable ML-random sets are known to exist). Such pairs will have no common ibT- or cl-upper bound. We can also combine Theorem 3.6 with Lemma 3.4 to produce a rather indirect proof that there is no analogue to Theorem 3.3 on the cl-degrees:

Corollary 3.7. *Not every ascending sequence in the cl-degrees has an upper bound.*

Proof. Let A be as in Theorem 3.6 and take the partial ordering induced by (\leq_{cl}) on $\mathcal{P}(\geq_{cl} A) := \{B : B \geq_{cl} A\}$. By the construction in Proposition 2.6, $\mathcal{P}(\geq_{cl} A)$ has no maximal elements. If each ascending sequence had an upper bound then, by Lemma 3.4, $\mathcal{P}(\geq_{cl} A)$ would be uncountable. Yet, $\mathcal{P}(\geq_{cl} A) \subseteq \text{deg}_{rK}(A)$ by

Theorem 3.6, and hence $\mathcal{P}(\geq_{cl} A)$ is countable (since the number of possible rK-reductions is countable, each rK degree is countable). \square

For a cl-chain ever to be uncountable, it must climb through uncountably many Turing degrees, all the while managing to avoid the ill-behaved and ill-understood quasi-maximal sets. Challenging though this may seem, there is such a chain; we shall see this first for the simpler ibT reducibility.

Proposition 3.8. *There is an uncountable chain of ibT-degrees.*

Proof. We shall specify a substructure \mathcal{P} of the ibT degrees and verify that it meets the hypotheses of Lemma 3.4. We restrict ourselves to a certain class of very sparse sets, such that none is maximal—using a warped variant of the construction in Proposition 2.1—and that every ascending chain has an upper bound not unlike the infinite join.

For a given set $A \in 2^\omega$, we may decompose it into segments

$$A = (2^0 - 1 + \sigma_0) \cup (2^1 - 1 + \sigma_1) \cup (2^2 - 1 + \sigma_2) \cup \dots$$

with each $\sigma_n \subseteq \{0, 1, \dots, 2^n - 1\}$. That is, A may be written as

$$A = \underbrace{A(0)}_{\sigma_0} \underbrace{A(1)A(2)}_{\sigma_1} \underbrace{A(3)A(4)A(5)A(6)}_{\sigma_2} \underbrace{A(7)A(8)A(9)A(10)A(11)A(12)A(13)A(14)}_{\sigma_3} \dots$$

Defining the maximum of the empty set to be -1 , let

$$\mathcal{P} = \left\{ A \in 2^\omega : A \text{ non-computable, } \lim_n \frac{\max \sigma_n}{2^n} = 0 \right\} .$$

This condition describes sets that are, in a very strong sense, asymptotically non-dense. Such sets do exist: for any non-computable set B , we may define an A by this decomposition:

$$\sigma_n = \begin{cases} \{0\} & \text{if } n \in B \\ \emptyset & \text{if not} \end{cases}$$

Then A is in \mathcal{P} .

We can define a successor function on \mathcal{P} as follows. Given $A \in \mathcal{P}$, let N be such that $(\forall n \geq N) \left[\frac{\max \sigma_n}{2^n} \leq \frac{1}{2} \right]$. Define a leftward shift S on A by

$$S(A) = (2^N - 1 + \sigma_{N+1}) \cup (2^{N+1} - 1 + \sigma_{N+2}) \cup \dots .$$

Then we clearly have $S(A) \in \mathcal{P}$. Our condition on N guarantees that each of the σ_{n+1} we use is of length at most 2^n , so that they never overlap in the expansion of $S(A)$. Hence, for any $n > N$, we can ibT -compute $A \cap [2^n, 2^{n+1}) = 2^n + \sigma_n$ from $S(A) \cap [2^{n-1}, 2^n) = 2^{n-1} + \sigma_n$. As in previous shifting arguments, to have $S(A) \leq_{\text{ibT}} A$ would force A to be computable. Therefore, $A <_{\text{ibT}} S(A)$.

Given an ascending chain $A_0 <_{\text{ibT}} A_1 <_{\text{ibT}} A_2 <_{\text{ibT}} \dots$ in \mathcal{P} , with respective decompositions

$$A_k = (2^0 - 1 + \sigma_0^k) \cup (2^1 - 1 + \sigma_1^k) \cup (2^2 - 1 + \sigma_2^k) \cup \dots ,$$

we construct a certain new set A_ω by

$$A_\omega = (2^0 - 1 + \sigma_0^\omega) \cup (2^1 - 1 + \sigma_1^\omega) \cup (2^2 - 1 + \sigma_2^\omega) \cup \dots .$$

For each $n \in \omega$, let $N_n \leq n$ be the largest such that

$$(\forall k < N_n)(\forall \ell \geq n) \left[\frac{\max \sigma_\ell^k}{2^\ell} \leq \frac{1}{2^{N_n}} \right] .$$

Define

$$\sigma_n^\omega = \{m < N_n\} \cup \bigcup_{k < N_n} (N_n + 1 + k \cdot 2^{N_n} + \sigma_{n+1}^k)$$

In other words, σ_n^ω contains N_n in unary, followed by a zero, followed by each of σ_{n+1}^k for $k < N_n$. It is worth noting that the σ_n^ω can spill out of its allotted 2^n spaces for $n \leq 3$. This will not be a problem, however, as we can discount those first few entries.

We clearly have $A_k \leq_{\text{ibT}} A_\omega$ for each k . In particular, A_ω is not computable. As well, for sufficiently large n , $\max \sigma_n^\omega \leq N_n + 1 + N_n \cdot 2^{n-N_n} \leq 2^{n-N_n+1}$, and $N_n \rightarrow \infty$. Hence $\lim_n \frac{\max \sigma_n^\omega}{2^n} = 0$. So A_ω is in \mathcal{P} .

Therefore, \mathcal{P} contains an uncountable chain. □

The construction of A_ω does not actually require that A_1, A_2, \dots be ibT -comparable. Hence on \mathcal{P} it can bound any countable set in a way analogous to the infinite join \bigoplus on the Turing degrees.

Corollary 3.9. *There is an uncountable chain of cl -degrees, and there is an uncountable chain of rK -degrees.*

Proof. Take an uncountable ibT-chain $(A_\alpha)_{\alpha < \aleph_1}$, and let $(\mathbf{a}_\alpha)_{\alpha < \aleph_1}$ be the associated cl- or rK-degrees. We still have a total ordering on (\mathbf{a}_α) , and each degree is countable; so if $(\mathbf{a}_\alpha)_{\alpha < \aleph_1}$ were a countable collection, we must have had a countable number of sets in the first place, a contradiction. \square

Chapter 4

Incomparable c.e. Sets

We wish to continue our examination of the relationships between the ibT , cl , and wtt degrees. We already know a fair bit about linear orderings embedded between them. What about nonlinear ones? On the ibT degrees, for example, which countable partial orderings can be embedded into a single cl degree, and which can be embedded into *any* non-trivial cl degree? We produce a satisfying answer to the first of these questions in Section 4.2. In order to decide whether the second question even makes sense, we first must decide: Does there exist a cl degree in which *all* sets are ibT -comparable? In this chapter, we grope towards a definitive answer.

As usual, it is interesting and useful to consider the restriction to the c.e. sets. In order to construct c.e. sets from other c.e. sets, we make heavy use of the classical techniques of priority and permitting. This chapter will therefore have much more pungent a computability-theoretic flavour than those that came before.

4.1 Basic techniques and results

This section is meant mainly as an exposition of the techniques we shall be using later. Each of the results here will be subsumed under another in Section 4.2 or Section 4.3. The reader should bear in mind that many of the results we shall be proving are very similar to one another, and will have long, boiler-plate verifications

that differ only slightly from theorem to theorem—but that one or two of these proofs is enough to gain a strong intuition for the methods and why they work.

The first technique we introduce is the *finite-injury priority* method. Recall Cantor’s diagonal argument for the existence of multiple infinite cardinalities. Taking a sequence $(r_e)_{e \in \omega}$ of real numbers, we can construct a new number s subject to the requirements

$$\mathcal{N}_e : s \neq r_e$$

by performing a sequence of *diagonalisations* ensuring that the e -th digit $s(e)$ in the decimal expansion of s is not the same as the e -th digit $r_e(e)$ of r_e , by setting $s(e) = 9 - r_e(e)$. Then s is a real number not in the list, leading to the conclusion that the sequence $(r_e)_{e \in \omega}$ of real numbers was not exhaustive.

This argument works out of the box because the diagonalisation neatly sidesteps any possible conflict between different requirements. Each \mathcal{N}_e is allocated one digit in the expansion, and performs its job without worry of external interference. It is a common goal, in computability theory and in other fields, to construct an object satisfying restraints that might interfere with one another. In this case, we assign each of the requirements a unique *priority*, with the understanding that, in the event of a conflict, the higher-priority requirement will take precedence, and with the hope that, in the end, each of the requirements will be satisfied. The first and canonical example of a priority argument is the Friedberg-Muchnik Theorem:

Theorem 4.1 (Friedberg [8]-Muchnik [12]). *There exists a pair U, V of c.e. sets such that $U|_T V$.*

Because it resembles future arguments more closely, we content ourselves with proving a weakened version:

Proposition 4.2. *There exists a pair U, V of c.e. sets such that $U|_{ibT} V$.*

Proof. Let $(\hat{\Phi}_e)_{e \in \omega}$ be an effective enumeration of ibT functionals. In order to have $U \leq_{ibT} V$, there must be a $\hat{\Phi}_e$ such that $U = \hat{\Phi}_e^V$, and similarly for $V \leq_{ibT} U$. Therefore our U and V must satisfy the countable set of requirements:

$$\begin{aligned} \mathcal{N}_{2e} & : U \neq \hat{\Phi}_e^V \\ \mathcal{N}_{2e+1} & : V \neq \hat{\Phi}_e^U \end{aligned}$$

Of these, the requirement \mathcal{N}_0 is considered to have the highest priority, followed by $\mathcal{N}_1, \mathcal{N}_2$, and so on.

Were we to assign to each \mathcal{N}_e a fixed spot in U and V , in the manner of Cantor's diagonalisation, then we should easily run into conflicts. For example, suppose we wish to diagonalise at $U(0)$ against $U = \hat{\Phi}_0^V$ and at $V(1)$ against $V = \hat{\Phi}_0^U$. If at stage s we have determined that $\hat{\Phi}_{0,s}^{U_s}(1) \downarrow = 0$, it will be tempting to enumerate 1 into V and declare requirement \mathcal{N}_1 fulfilled. If we do so, at stage $s + 1$ we may notice that $\hat{\Phi}_{0,s+1}^{V_{s+1}}(0) \downarrow = 0$ and so enumerate 0 into U . But this enumeration would *injure* our previous diagonalisation, re-opening the possibility that $\hat{\Phi}_0^U(1) \downarrow = 1$. We could try again and again to diagonalise, but there is no reason to believe that one of our attempts will escape injury. On the other hand, if we do not enumerate 1 into V but rather wait to see whether $\hat{\Phi}_0^V(0) \downarrow = 0$, we may well be waiting forever.

The solution is, at each stage, to diagonalise against the highest-priority unsatisfied requirement that we can, without disturbing the fulfilled requirements of still higher priority. At each stage s , each requirement \mathcal{N}_k is given a **restraint** $r(k, s)$ such that, if $\ell > k$, then we cannot explicitly act on $U_s \upharpoonright r(k, s)$ or $V_s \upharpoonright r(k, s)$ to satisfy \mathcal{N}_ℓ . This protects the work done to meet requirement \mathcal{N}_k . We view all m in the range $r(k - 1, s) < m \leq r(k, s)$ as candidate witnesses to \mathcal{N}_k . This particular argument will be simple enough that, at any given stage s , the entry $r(k, s)$ is our only candidate witness.

Construction

Assign to each requirement \mathcal{N}_k and each stage s a witness $r(k, s)$. Begin by letting $r(k, 0) = k$ for each $k \in \omega$. Then proceed by stages.

At stage $s + 1$, let i —if it exists—be the smallest natural number smaller than s such that $\hat{\Phi}_{i,s}^{V_s}(r(2i, s)) \downarrow = 0$ and $r(2i, s) \notin U_s$. Similarly, let j be the smallest natural number smaller than s such that $\hat{\Phi}_{j,s}^{U_s}(r(2j + 1)) \downarrow = 0$ with $r(2j + 1, s) \notin V_s$. If i is defined and $i \leq j$ or j is not defined, enumerate $r(2i, s)$ into U_{s+1} . Otherwise, if j is defined, enumerate $r(2j + 1, s)$ into V_{s+1} . Then recalculate the restrictions

for each $e \leq s$:

$$\begin{aligned}
r(0, s+1) &= 0 \\
r(2e+1, s+1) &= (\mu n > r(2e, s+1)) \left[n \notin V_s \vee \hat{\Phi}_{e,s}^{U_{s+1}}(n) \downarrow = 0 \right] \\
r(2e+2, s+1) &= (\mu n > r(2e+1, s+1)) \left[n \notin U_s \vee \hat{\Phi}_{e,s}^{V_{s+1}}(n) \downarrow = 0 \right]
\end{aligned}$$

Verification

We have defined $(r(0, s))_{s \in \omega}$ to be the sequence of all zeroes. We claim that, for any i even or odd, the sequence $(r(i, s))_{s \in \omega}$ is eventually constant. Suppose otherwise. If a sequence of natural numbers is not eventually constant, then it must be increasing at infinitely many stages. Let j be the smallest such that, for infinitely many s , $r(j, s+1) > r(j, s)$. Assume that $j = 2e+1$ is odd, the other case being similar. Choose s_0 such that $s \geq s_0$ implies, for each $k < j$, that $r(k, s+1) = r(k, s)$, and $U_s \Vdash r(j, s) = U_{s+1} \Vdash r(j, s)$, and there is some stage $t > s_0$, $r(j, t+1) > r(j, t)$. This implies that $r(j, s) \in V_{t+1} \setminus V_t$. The only case where we might enumerate $r(j, s)$ into V_{t+1} would be if $\hat{\Phi}_{e,t}^{U_t}(r(j, t)) \downarrow = 0$. But $U_t \Vdash r(j, t) = U_{t+1} \Vdash r(j, t)$, so we must have $\hat{\Phi}_{e,t}^{U_{t+1}}(r(j, t)) \downarrow = 0$, giving $r(j, t+1) \leq r(j, t)$, a contradiction.

Because each $(r(2e+1, s))_{s \in \omega}$ eventually settles, we must have requirement \mathcal{N}_{2e+1} met. If we let $r(2e+1) = \lim_s r(2e+1, s)$, then we know either $r(2e+1) \notin V$ or $\hat{\Phi}_e^V(r(2e+1)) \downarrow = 0$. In the former case, we never needed to perform the diagonalisation, meaning either $\hat{\Phi}_e^V(r(2e+1)) \downarrow = 1$ or $\hat{\Phi}_e^V(r(2e+1)) \uparrow$. In the latter, we must have performed the diagonalisation. The same holds for even requirements \mathcal{N}_{2e+2} . The special case of \mathcal{N}_0 also works, since $V \Vdash 0$ will never change, and so any diagonalisation we may perform at $U(0)$ will remain uninjured. \square

The *permitting* method is common in situations where we want to build a non-computable c.e. set W that is Turing below another non-computable c.e. set A using a c.e. approximation $(A_s)_{s \in \omega}$ of the latter. In the case of *identity-bounded permitting*, the idea is to build a c.e. approximation $(W_s)_{s \in \omega}$ such that $W_{s+1} \Vdash n \neq W_s \Vdash n$ implies $A_{s+1} \Vdash n \neq A_s \Vdash n$. Then, if we know the final value of $A \Vdash n$, we can find s such that $A_s \Vdash n = A \Vdash n$, and hence recover $W \Vdash n = W_s \Vdash n$. Thus $W \leq_{ibT} A$. The tricky part is that we may want W to have some additional distinguishing property P , but we may not be able to impose it more quickly than $(A_s)_{s \in \omega}$ converges and

cuts us off. In this case, we can trace our steps through the approximation $(A_s)_{s \in \omega}$ and our computation of $(W_s)_{s \in \omega}$ and gather enough evidence of failed attempts at P to prove that A was actually computable. (The same sort of argument can be used on Δ_2 sets in general, and with particular success on c.e. reals.)

For the next construction, we combine priority and permitting strategies.

Proposition 4.3. *Each nonzero c.e. wtt-degree contains a pair U, V of c.e. sets such that $U|_{ibT}V$.*

Proof. Suppose A is a non-computable c.e. set with c.e. approximating sequence $(A_s)_{s \in \omega}$. We shall construct two new sets U, V such that $U \equiv_{wtt} A \equiv_{wtt} V$ but $U|_{ibT}V$.

Let $(\hat{\Phi}_e)_{e \in \omega}$ denote a computable enumeration of ibT functionals.

Requirements

$$\begin{aligned} \mathcal{P} : & \quad A \leq_{wtt} U, V \\ \mathcal{R} : & \quad U, V \leq_{wtt} A \\ \mathcal{N}_{2e} : & \quad U \neq \hat{\Phi}_e^V \\ \mathcal{N}_{2e+1} : & \quad V \neq \hat{\Phi}_e^U \end{aligned}$$

We achieve \mathcal{P} by coding a spread-out version of A directly into U and V ; we prevent $U \leq_{ibT} V$ and $V \leq_{ibT} U$ by diagonalising against each such ibT-reduction. We assign priorities to the \mathcal{N}_e in the usual, descending order; \mathcal{R} will be guaranteed through permitting. Note that, among plain c.e. permitting arguments, ours is somewhat non-standard in that, rather than waiting for any $k \leq n$ to enter A before allowing n to enter U or V , we wait for one of a very specific set of k .

To govern the finite-injury priority argument, we create, for each \mathcal{N}_e and each stage s , a restraint $r(e, s)$ that shall bound any diagonalisation against \mathcal{N}_e at stage s . If, at stage s , \mathcal{N}_e appears to be satisfied, and \mathcal{N}_i is satisfied and has stopped acting for all $i < e$, subsequent $r(e, t)$ shall be less than or equal to $r(e, s)$. Because we are constructing c.e. approximations, this means that, once satisfied, \mathcal{N}_e will act on its witnesses only a finite number of times, and hence, by induction, any requirement will be injured only a finite number of times.

Our construction is further complicated by the need to provide an adequate number of witnesses, in the appropriate positions, to each \mathcal{N}_e . We are forced to introduce them slowly. Denote the coding position of $A(n)$ in U and V by $\lambda(n)$, so that for all n we have $U(\lambda(n)) = A(n) = V(\lambda(n))$. We insert one witness position (for \mathcal{N}_0) before $\lambda(0)$, two (one for \mathcal{N}_0 and one for \mathcal{N}_1) between $\lambda(0)$ and $\lambda(1)$, \dots , and $n + 2$ (one for each \mathcal{N}_e , $e \leq n + 1$) between $\lambda(n)$ and $\lambda(n + 1)$. A simple calculation gives the closed form $\lambda(n) = \frac{n^2 + 5n + 2}{2}$; for convenience, we also assign $\lambda(-1) = -1$. As for the witnesses, we just allotted one to \mathcal{N}_e between $\lambda(n - 1)$ and $\lambda(n)$ for every $n \geq e$. We write the coding positions for \mathcal{N}_e as the partial function $\Lambda_e : \omega_{\geq e} \rightarrow \omega$, $\Lambda_e(n) = \lambda(n - 1) + e$. Note that $\Lambda_0(0) < \lambda(0) < \Lambda_0(1) < \Lambda_1(1) < \lambda(1) < \Lambda_0(2) < \Lambda_1(2) < \Lambda_2(2) < \lambda(2) < \dots$.

Here are the necessary restraints:

$$\begin{aligned} r(-1, s) &= -1 \\ r(2e, s) &= (\mu k > r(2e - 1, s)) \left[\neg \left(\hat{\Phi}_{e,s}^{V_s}(\Lambda_{2e}(k)) \downarrow = U_s(\Lambda_{2e}(k)) \right) \right] \\ r(2e + 1, s) &= (\mu k > r(2e - 1, s)) \left[\neg \left(\hat{\Phi}_{e,s}^{U_s}(\Lambda_{2e+1}(k)) \downarrow = V_s(\Lambda_{2e+1}(k)) \right) \right] \end{aligned}$$

By convention, before halting, each oracle computation $\Phi_e^X(n)$ will take at least n steps to halt. Hence each $r(\ell, s)$ is defined and $r(\ell, s) \leq s$.

Construction

When $n \in A_s \setminus A_{s-1}$, enumerate $U_s(\lambda(n)) = V_s(\lambda(n)) = 1$.

When such an n enters with $r(2e - 1, s) \leq n < r(2e, s)$, define $U_s(\Lambda_{2e}(n)) = 1 \dot{-} \hat{\Phi}_{e,s}^{V_s}(\Lambda_{2e}(n))$. That is, if $\hat{\Phi}_{e,s}^{V_s}(\Lambda_{2e}(n)) \downarrow = 0$, then enumerate $\Lambda_{2e}(n)$ into U_s , and do nothing otherwise.

When such an n enters with $r(2e, s) \leq n < r(2e + 1, s)$, let $V_s(\Lambda_{2e+1}(n)) = 1 \dot{-} \hat{\Phi}_{e,s}^{U_s}(\Lambda_{2e+1}(n))$.

Verification

Because of the direct coding of A into U and V by λ , \mathcal{P} is immediate. Because n enters U or V only if some $k \leq n$ enters A , \mathcal{R} is also satisfied. U and V are c.e. because they are built by stages, and each entry, being in the image of exactly one of λ and $\{\Lambda_\ell\}$, is changed during at most one stage. We establish \mathcal{N}_e by induction.

For each i , let $r(i) = \lim_s r(i, s)$. We claim that this limit exists for all i . Suppose, for contradiction, that ℓ is the smallest such that $\lim_s r(\ell, s)$ does not exist. Assume further that $\ell = 2e$ is even, the odd case being symmetrical. Let s_0 be such that $s \geq s_0$ implies $(\forall i < \ell)[r(i, s) = r(i)]$ and $A_s \uparrow r(\ell - 1) = A \uparrow r(\ell - 1)$.

Case 1. Suppose there is a $t_0 \geq s_0$ such that $\hat{\Phi}_{e, t_0}^{V_{t_0}}(\Lambda_\ell(r(\ell, t_0))) \downarrow \neq U_{t_0}(\Lambda_\ell(r(\ell, t_0)))$. We shall show that the limit $r(\ell)$ exists. The only way $r(\ell, t)$ can then differ from $r(\ell, t_0)$, with $t > t_0$, is if, at stage t , some new element $n \leq \Lambda_\ell(r(\ell, t_0))$ enters $V_t \setminus V_{t-1}$. Either $n = \lambda(k)$ or $n = \Lambda_i(k)$ for some $i < \ell$, $k \leq r(\ell, t)$; in either case, there is a $k \leq r(\ell, t)$ entering $A_t \setminus A_{t-1}$. We cannot have $k < r(\ell - 1) = r(\ell - 1, t)$, by our choice of s_0 . We thus have $r(\ell - 1, t) \leq k < r(\ell, t)$. Our construction dictates that, at stage t , we diagonalise at $U_t(\Lambda_\ell(k))$; thus, at the next stage, $r(\ell, t + 1) \leq k \leq r(\ell, t)$. This means that the sequence $(r(\ell, s))_{s \geq t_0}$ is decreasing and bounded below by $r(\ell - 1)$; therefore, the limit $r(\ell)$ exists, a contradiction.

Case 2. Suppose next that there is no such t_0 —that is, $(\forall s \geq s_0) [\hat{\Phi}_{e, s}^{V_s}(\Lambda_\ell(r(\ell, s))) \uparrow]$. Again, we wish to show that $r(\ell)$ exists. Say there is some stage $t_1 \geq s_0$ where $(\forall t \geq t_1) [\hat{\Phi}_{e, t}^{V_t}(\Lambda_\ell(r(\ell, t_1))) \uparrow]$. In this case, we know that $(\forall t \geq t_1) [r(\ell, t) \leq r(\ell, t_1)]$. Define the **least modulus** or **settling time** of $(A_s)_{s \in \omega}$ to be the function:

$$\mathbf{m}(n) = (\mu t)(\forall s \geq t) [A \uparrow n = A_s \uparrow n] .$$

Then, for all $t \geq \mathbf{m}(r(\ell, t_1))$, we have $r(\ell, t + 1) = r(\ell, t)$, so that the limit $r(\ell)$ exists, a contradiction.

Case 3. If there are no t_0 or t_1 , then we can define a computable function $f(x) = (\mu s \geq s_1)[r(\ell, s) \geq x]$. Then, $(\forall^\infty x)[\mathbf{m}(x) \leq f(x)]$ —since otherwise we would be performing a diagonalisation after step s_0 , bringing us to Case 1—so that $(\forall^\infty x)[A(x) = A_{f(x)}(x)]$. But then A is computable, a contradiction.

Therefore, each sequence $(r(\ell, s))_{s \in \omega}$ converges. Then, by definition of $r(\ell, s)$, we must have

$$(\forall^\infty s) \left[\neg \left(\hat{\Phi}_{e, s}^{V_s}(\Lambda_\ell(r(\ell))) \downarrow = U_s(r(\ell)) \right) \right] ,$$

and in particular

$$\neg \left(\hat{\Phi}_e^V(\Lambda_\ell(r(\ell))) \downarrow = U(r(\ell)) \right) .$$

Hence, each requirement \mathcal{N}_ℓ is eventually satisfied. \square

This result quickly gives way to a stronger one:

Corollary 4.4. *Within any nonzero c.e. wtt-degree there exist two c.e. sets X, Y such that $X|_{ibT}Y$ but $X \equiv_{cl} Y$.*

Proof. Take the U and V from Proposition 4.3. Let $X = U \oplus V$ —that is, viewed as sets, $X = 2U \cup (2V + 1)$ —and $Y = V \oplus U$. Since U and V are wtt-equivalent, we have $X \equiv_{wtt} U \equiv_{wtt} V \equiv_{wtt} Y$. As well, we know that $X \equiv_{cl} Y$, since, for each n , $X(2n) = Y(2n + 1)$ and $X(2n + 1) = Y(2n)$. Assume that $X \leq_{ibT} Y$. Then there is some Γ that computes $X(2n)$ from $Y \upharpoonright 2n$, and hence there is a Γ_1 that computes $U(n)$ from $V \upharpoonright n$ and $U \upharpoonright (n - 1)$, and hence, iterating Γ_1 on each $k \leq n$, there is a Γ_2 that computes $U(n)$ from just $V \upharpoonright n$. This Γ witnesses $U \leq_{ibT} V$, a contradiction. Hence $X \not\leq_{ibT} Y$, and by a similar argument $Y \not\leq_{ibT} X$. \square

Corollary 4.5. *Within any nonzero c.e. wtt-degree there exist two c.e. sets X, Y such that $X|_{cl}Y$.*

Proof. Again take U and V from Proposition 4.3. Define X by $X(\frac{n \cdot (n+1)}{2}) = U(n)$ for each n and $X(k) = 0$ for all other k . Define Y analogously from V . Then $X \equiv_{wtt} U \equiv_{wtt} V \equiv_{wtt} Y$. Suppose that $X \leq_{cl} Y$ by functional Γ with constant c . Then, for all $n \geq c$, Γ can compute $X(\frac{n \cdot (n+1)}{2})$ from $Y \upharpoonright \frac{n \cdot (n+1)}{2} + c = Y \upharpoonright \frac{n \cdot (n+1)}{2}$. So there is a Γ_1 that can compute $U(n)$ from $V \upharpoonright n$. Iterating this Γ_1 , we can obtain a Γ_2 that ibT -computes all U from V using the finite information in $U \upharpoonright c$ —which contradicts Proposition 4.3. So $X \not\leq_{cl} Y$, and by a similar argument we must have $Y \not\leq_{cl} X$. \square

4.2 Arbitrary countable partially-ordered sets

We now wish to use these techniques to decide what sort of partially-ordered substructures we can find in the cl or the ibT degrees.

Definition 4.6. *If (P, \leq) and (Q, \leq) are partially-ordered sets, then we say $f : P \rightarrow Q$ is a (strong) embedding of P into Q if, for all $p_1, p_2 \in P$, $q_1, q_2 \in Q$, we have $p_1 \leq_P p_2 \iff q_1 \leq_Q q_2$.*

A countable partially-ordered set (P, \leq_P) is called **countably universal** if it contains an order-isomorphic copy of every countable partially ordered set. Clearly, then, if such a set can be embedded into a given degree structure, then *any* countable partially-ordered set can be embedded into that structure. It is possible to produce a computable, countably universal set;

Theorem 4.7. *There exists a computable, countably universal, partially-ordered set (P, \leq_P) —that is, one on which the relation (\leq_P) is computable.*

Hence, to see that any countable partial ordering can be embedded into a degree structure, it suffices to prove that any *computable* partial ordering can so be embedded. It has been observed that any countable partially-ordered set can be embedded into the Turing degrees using a countable collection $\{A_j \in 2^\omega : j \in \omega\}$ of independent sets, *i.e.*,

$$(\forall j) \left[A_j \not\leq_T \bigoplus_{i \neq j} A_i \right].$$

Such a collection of sets was first exhibited by Kleene and Post [9]. An appropriate embedding would then be

$$\begin{aligned} P &\rightarrow 2^\omega \\ q &\mapsto M_q := \bigoplus_{p \leq_P q} A_p \end{aligned}$$

as the computability of the relation (\leq_P) and of the \bigoplus operation guarantee $M_p \leq_T M_q$ if $p \leq_P q$, and our independence condition on $\{A_j\}_{j \in \omega}$ ensures the converse.

This well-known proof carries through immediately on the wtt-degrees, but not on the cl- or ibT-degrees, as \bigoplus tends to spread information too far. The solution is to create sets that are already sufficiently spread out, and replace \bigoplus with \bigcup .

The following result does exactly that to produce a computable, countably universal structure in the ibT degrees. Its immediate consequence, Corollary 4.9, in some ways eclipses our earlier result in Proposition 3.1. It should be noted, however, that this result has some disadvantages, as it does not guarantee that any c.e. set is *part of* such a structure, nor is the proof nearly as elementary.

Proposition 4.8. *If $A \in 2^\omega$ is non-computable and c.e., and (P, \leq_P) is a countable partially-ordered set, then we can embed (P, \leq_P) into the c.e. ibT degrees within the wtt-degree of A .*

Proof. We may assume that $P = \omega$ and that (\leq_P) is computable. We wish to find an infinite collection of ibT-incomparable c.e. sets within the wtt-degree of A . Take a c.e. approximation $(A_s)_{s \in \omega}$ of A .

Our construction here will be a refined version of that in Proposition 4.3 whereby we produce an infinite number of ibT-independent sets, rather than just two. As in the proof of that proposition, here we use a combination of coding and permitting to ensure wtt-equivalence.

To control ibT reducibility, we partition ω into an infinite number of computable subsets S and $\{T_i\}_{i \in \omega}$. We code a $W \subseteq S$ such that $W \equiv_{wtt} A$, and within each of the T_i we build a corresponding $C_i \subseteq T_i$ such that, for each i , $W \cup C_i \not\leq_{ibT} W \cup \bigcup_{j \neq i} C_j$. These sets can therefore be used to build the desired copy of (P, \leq_P) . If we define $B = W \cup \bigcup_{p \in P} T_p$, these requirements can be summed up:

Requirements

$$\begin{aligned} \mathcal{P} : & \quad A \leq_{wtt} B \cap S \\ \mathcal{R} : & \quad B \leq_{wtt} A \\ \mathcal{N}_{\langle p, e \rangle} : & \quad B \cap (T_p \cup S) \neq \hat{\Phi}_e^{B \setminus T_p} \end{aligned}$$

Suppose we have satisfied all these requirements. For any $q \in P$, let $M_q = B \cap (S \cup \bigcup_{p \leq_P q} T_p)$. If $q \leq_P r$, then $\{p : p \leq_P q\}$ is a computable set, so $M_q \leq_{ibT} M_r$ through the restriction map. On the other hand, if $q \not\leq_P r$, we have $M_r \leq_{ibT} (B \setminus T_q)$ by the restriction map, giving $M_q \not\leq_{ibT} M_r$ by requirements $\{\mathcal{N}_{\langle q, e \rangle} : e \in \omega\}$. As well, because $B \cap S$ is wtt-computable from each M_q , and M_q is wtt-computable from B by computable restriction, if \mathcal{P} and \mathcal{R} are met, then we must have $M_q \equiv_{wtt} A$ for each $q \in P$.

Then $q \mapsto M_q$ is an embedding, as desired.

Construction

Much like in the proof of Proposition 4.3, we wish to have coding locations $\lambda(x)$ for A and diagonalisation witnesses $\Lambda_{\langle p, e \rangle}(x)$ such that

$$\Lambda_0(0) < \lambda(0) < \Lambda_0(1) < \Lambda_1(1) < \lambda(1) < \dots$$

It is enough to let $\lambda(x) = \frac{(x+2)(x+3)}{2} - 2$ and, for $x \geq \langle p, e \rangle$, let $\Lambda_{\langle p, e \rangle}(x) = \frac{(x+1)(x+2)}{2} + \langle p, e \rangle - 1$. Then we have our computable partition:

$$\begin{aligned} S &= \lambda(\omega) \\ T_p &= \bigcup_{e \in \omega} \Lambda_{\langle p, e \rangle}(\omega) \end{aligned}$$

Assign beginning restraints:

$$\begin{aligned} r(-1, 0) &= 0 \\ r(\langle p, e \rangle, 0) &= 0 \text{ for each } e \geq 0, p \in \omega \end{aligned}$$

At each stage $s + 1$:

Step 1. Take $n \in A_{s+1} \setminus A_s$. There is at most one such n ; if no such n exists, skip to Step 3. Otherwise, find $\ell = \langle p, e \rangle$ such that $r(\ell - 1, s) \leq n < r(\ell, s)$ and $\ell < n$. If no such ℓ exists, proceed to Step 2. Otherwise, enumerate $\Lambda_\ell(n)$ into $B_{s+1} \cap T_p$ for diagonalisation.

Step 2. Enumerate $\lambda(n)$ into $B_{s+1} \cap S$ for coding.

Step 3. Recalculate the restraints for each $\ell = \langle p, e \rangle \leq s$:

$$\begin{aligned} r(-1, s+1) &= 0 \\ r(\ell, s+1) &= (\mu k > r(\ell - 1, s+1)) \left[\neg \left(\hat{\Phi}_{e,s}^{B_{s+1} \setminus T_p}(\Lambda_\ell(k)) \downarrow = B_{s+1}(\Lambda_\ell(k)) \right) \right] \end{aligned}$$

Verification

We defined $(r(-1, s))_{s \in \omega}$ to be the constant sequence of all zeroes. For each ℓ , let $r(\ell) = \lim_s r(\ell, s)$. We claim that each of these limits exists. Suppose that $\ell = \langle p, e \rangle$ is the smallest for which the limit does not exist. Choose s_0 large enough that $i < \ell$ and $s \geq s_0$ imply $r(i, s) = r(i)$ and $A_{s_0} \upharpoonright r(\ell - 1) = A \upharpoonright r(\ell - 1)$.

Case 1. Suppose there is a $t_0 \geq s_0$ such that $\hat{\Phi}_{e,t_0}^{B_{t_0} \setminus T_p}(\Lambda_\ell(r(\ell, t_0))) \downarrow \neq B_{t_0}(\Lambda_\ell(r(\ell, t_0)))$. We shall show that the limit $r(\ell)$ exists. The only way $r(\ell, t)$ can then differ from $r(\ell, t_0)$, with $t > t_0$, is if, at stage t , some new element $n \leq \Lambda_\ell(r(\ell, t_0))$ enters $B_t \setminus B_{t-1}$. Either $n = \lambda(k)$ or $n = \Lambda_i(k)$ for some $i < \ell$, $k \leq r(\ell, t)$; in either case, there is a $k \leq r(\ell, t)$ entering $A_t \setminus A_{t-1}$. We cannot have $k < r(\ell - 1) = r(\ell - 1, t)$, by our choice of s_0 . We thus have $r(\ell - 1, t) \leq k < r(\ell, t)$. Then, at stage t , we diagonalise at $B_t(\Lambda_\ell(k))$; thus, at the next stage, $r(\ell, t+1) \leq k \leq r(\ell, t)$. This

means that the sequence $(r(\ell, s))_{s \geq t_0}$ is decreasing and bounded below by $r(\ell - 1)$; therefore, the limit $r(\ell)$ exists, a contradiction.

Case 2. Suppose that there is no such t_0 . We wish to show that $r(\ell)$ exists. Say there is some stage $t_1 \geq s_0$ where $(\forall t \geq t_1) [\hat{\Phi}_{e,t}^{B_t \setminus T_p}(\Lambda_\ell(r(\ell, t_1))) \uparrow]$. In this case, we know that $(\forall t \geq t_1) [r(\ell, t) \leq r(\ell, t_1)]$. Then, for all $t \geq \mathbf{m}(r(\ell, t_1))$, we have $r(\ell, t + 1) = r(\ell, t)$, so that the limit $r(\ell)$ exists, a contradiction.

Case 3. If there are no t_0 or t_1 , then we can define a computable function $f(x) = (\mu s \geq s_1)[r(\ell, s) \geq x]$. Then, $(\forall^\infty x)[\mathbf{m}(x) \leq f(x)]$, so that $(\forall^\infty x)[A(x) = A_{f(x)}(x)]$. But then A is computable, a contradiction.

Therefore, each sequence $(r(\ell, s))_{s \in \omega}$ converges. Then, by definition of $r(\ell, s)$, we must have

$$(\forall^\infty s) \left[\neg \left(\hat{\Phi}_{e,s}^{B_s \setminus T_p}(\Lambda_\ell(r(\ell))) \downarrow = B_s(r(\ell)) \right) \right] ,$$

and in particular

$$\neg \left(\hat{\Phi}_e^{B \setminus T_p}(\Lambda_\ell(r(\ell))) \downarrow = B(r(\ell)) \right) .$$

Hence, each requirement \mathcal{N}_ℓ is eventually satisfied.

We satisfy \mathcal{R} by permitting and \mathcal{P} by coding into S . □

Spreading the sets out further, as in Corollary 4.5, gives a similar result on the cl-degrees:

Corollary 4.9. *If $A \in 2^\omega$ is non-computable and c.e., and (P, \leq_P) is a countable partially-ordered set, then we can embed (P, \leq_P) into the c.e. cl degrees within the wtt-degree of A .*

Proof. Take the sets S , T_p , and B constructed in Proposition 4.8. Create new sets

$$S^* = \left\{ \frac{n \cdot (n + 1)}{2} : n \in S \right\}$$

$$T_p^* = \left\{ \frac{n \cdot (n + 1)}{2} : n \in T_p \right\}$$

$$B^* = \left\{ \frac{n \cdot (n + 1)}{2} : n \in B \right\}$$

Then we have, for every $\langle e, p \rangle$ and every cl-functional $\tilde{\Phi}_e$,

$$\begin{aligned} A &\leq_{cl} B^* \cap S^* \\ B^* &\leq_{cl} A \\ B^* \cap (T_p^* \cup S^*) &\neq \tilde{\Phi}_e^{B^* \setminus T_p^*} \end{aligned}$$

giving the desired construction. \square

To get Corollary 4.4 from Proposition 4.3, we simply interleaved the two sets U and V to get a cl-equivalence. We can do the same to any finite number of sets built in Proposition 4.8:

Corollary 4.10. *If $A \in 2^\omega$ is non-computable and c.e., and (P, \leq_P) is a finite partially-ordered set, then we can embed (P, \leq_P) into the c.e. ibT degrees within a single cl-degree inside the wtt-degree of A .*

Proof. Suppose $P = n = \{0, 1, \dots, n-1\}$. Construct B exactly as in Proposition 4.8. For each $q < n$, let $M_q = B \cap (S \cup \bigcup_{p < n, p \leq_P q} T_p)$. Then, as before, $p \leq_P q$ if and only if $M_p \leq_{ibT} M_q$.

Weave these sets together as a sort of n -tuple join, with conspicuous blank spaces every $(n+1)$ -th entry:

$$X = ((n+1) \cdot M_0 + 1) \cup ((n+1) \cdot M_1 + 2) \cup \dots \cup ((n+1) \cdot M_{n-1} + n)$$

and fill those spaces in n different ways:

$$\begin{aligned} N_0 &= (n+1) \cdot M_0 \cup X \\ N_1 &= (n+1) \cdot M_1 \cup X \\ &\vdots \\ N_{n-1} &= (n+1) \cdot M_{n-1} \cup X \end{aligned}$$

Then $N_0 \equiv_{cl} N_1 \equiv_{cl} \dots \equiv_{cl} N_{n-1}$, and each is wtt-equivalent to A . Moreover, $M_p \leq_{ibT} M_q$ if and only if $N_p \leq_{ibT} N_q$, in the obvious way. \square

4.3 Compression and c.e. reals

In general, it is easy to construct different sorts of c.e. reals, and Δ_2 sets in general, from a c.e. approximation. The relaxed requirements of an approximating sequence give us a good deal of freedom, in particular allowing us to compress the information from a segment of size $2^n - 1$ from a c.e. set A to segment n of a c.e. real B as follows. Identify a finite binary string σ of length n with the natural number $\sum_{i=0}^{n-1} \sigma(i) \cdot 2^i$. Given a c.e. set A with approximating sequence $(A_s)_{s \in \omega}$, and any choice of n , construct a new sequence $(B_s)_{s \in \omega}$ by letting $B_s \upharpoonright n$ be the size $|A_s \upharpoonright (2^n - 1)|$ as a binary number of length n . Then $(B_s)_{s \in \omega}$ is a c.e. real approximation whose limit B encodes $A \upharpoonright 2^n$ in a way dual to our “counting” construction in Lemma 1.16. Encoding more and more segments of A can result in a c.e. real B from which A is ibT-computable, but which still has room for coding.

To demonstrate, we construct a structure of Δ_2 sets.

Corollary 4.11. *If $A \in 2^\omega$ is non-computable and c.e., and (P, \leq_P) is a finite partially-ordered set, then we can embed (P, \leq_P) into the Δ_2 ibT-degrees within the cl-degree of A .*

Proof. We may assume that the set P is $\{0, 1, 2, \dots, n - 1\}$. Further assume that $n \geq 2$, the other cases being trivial. Take a c.e. approximation $(A_s)_{s \in \omega}$. Take an enumeration $(\hat{\Phi}_e)_{e \in \omega}$ of ibT functionals. Let $\langle \cdot, \cdot \rangle$ be a Gödel numbering from $P \times \omega \rightarrow \omega$.

We compress A to give room for our diagonalisations. Since we want consecutive strings of n witnesses, we’ll need to compress segments by more than n positions. It will be sufficient, if somewhat wasteful, to take segments of size 3^n .

For each $p = 0, 1, \dots, n - 1$, let $T_p = \{x \cdot 3^n + p\}_{x \in \omega}$. Define $\Lambda_p(x) = x \cdot 3^n + p$. Let $S = \{x \cdot 3^n + k : n \leq k < 3n\}$ be the coding locations. The intervals are large enough, since $\log_2 3 < 2$. Then the T_p and S do not form a partition of ω , but they are disjoint, computable sets, which is all we need.

Requirements

$$\begin{aligned}
\mathcal{P} : & \quad A \leq_{cl} B \cap S \\
\mathcal{R} : & \quad B \leq_{cl} A \\
\mathcal{N}_{\langle p, e \rangle} : & \quad B \cap (T_p \cup S) \neq \hat{\Phi}_e^{B \setminus T_p}
\end{aligned}$$

Using the S and T_p as before, we can then make an embedding of the finite lattice (P, \leq) into the cl-degree of A .

Because we wish to satisfy \mathcal{R} through permitting, we'll need, between each two witnesses for $\mathcal{N}_{\langle p, e \rangle}$, some maximum distance c to bound the use of the cl-reduction. Because there are infinitely many e , this will cause some complications: we must re-use the same witnesses for different e . The weaker requirements of a Δ_2 approximation allow us to reverse a diagonalisation when a small element enters A . Though our argument will be slightly more complicated, we shall see in the verification that, as in previous arguments, a c.e. set A that does not eventually allow us to satisfy each $\mathcal{N}_{\langle p, e \rangle}$ must be computable.

Construction

Assign beginning restraints:

$$\begin{aligned}
r(-1, 0) &= 0 \\
r(\langle p, e \rangle, 0) &= 0 \text{ for each } e \geq 0, p \in \omega
\end{aligned}$$

At each stage $s + 1$:

Step 1. Take $x \in A_{s+1} \setminus A_s$. There is at most one such x ; if no such x exists, skip to Step 3. Otherwise, find numbers m, k such that $x = m \cdot 2^n + k$ with $0 \leq k < 3^n$. Find $\ell = \langle p, e \rangle$ such that $r(\ell - 1, s) < n < r(\ell, s)$ and $\ell < m$. If no such ℓ exists, proceed to Step 2. Otherwise, let $\Lambda_p(m) = 1 \dot{-} \hat{\Phi}_{e,s}^{B_{s+1} \cap T_p}$ for diagonalisation.

Step 2. Code $|A_s \cap [(m + 2) \cdot 3^n, (m + 3) \cdot 3^n]|$ into $B_s \cap [m \cdot 3^n + n, m \cdot 3^n + 3n]$.

Step 3. Recalculate the restraints for each $\ell = \langle p, e \rangle \leq s$:

$$\begin{aligned}
r(-1, s + 1) &= 0 \\
r(\ell, s + 1) &= (\mu k > r(\ell - 1, s + 1)) \left[k \geq s \vee \neg \left(\hat{\Phi}_{e,s}^{B_{s+1} \setminus T_p}(\Lambda_p(k)) \downarrow = B_{s+1}(\Lambda_p(k)) \right) \right]
\end{aligned}$$

Verification

The verification will be similar to previous arguments. For each ℓ , we let $r(\ell)$ be the limit of $(r(\ell, s))_{s \in \omega}$. Supposing that $\ell = \langle p, e \rangle$ is the smallest for which this limit does not exist, choose a stage s_0 after which, for all smaller $i < \ell$ and all subsequence $s \geq s_0$, we have $r(i, s) = r(i)$ and

$$A_s \uparrow ((r(\ell - 1) + 2) \cdot 3^n) = A \uparrow ((r(\ell - 1) + 2) \cdot 3^n) .$$

Case 1. Suppose there is a $t_0 \geq s_0$ such that $\hat{\Phi}_{e, t_0}^{B_{t_0} \setminus T_p}(\Lambda_p(r(\ell, t_0))) \downarrow \neq B_{t_0}(\Lambda_p(r(\ell, t_0)))$. We shall show that the limit $r(\ell)$ exists. The only way $r(\ell, t)$ can then differ from $r(\ell, t_0)$, with $t > t_0$, is if, at stage t , some new element $m \leq \Lambda_p(r(\ell, t_0))$ enters $B_t \setminus B_{t-1}$. Either $m = \Lambda_p(k)$ for some $i < \ell$, $k \leq r(\ell, t)$ or m is entering to code some change in $A \uparrow (3^n \cdot (r(\ell, t) + 2))$; in either case, there is a $k \leq r(\ell, t)$ entering $A_t \setminus A_{t-1}$. We cannot have $k < r(\ell - 1) = r(\ell - 1, t)$, by our choice of s_0 . We thus have $r(\ell - 1, t) \leq k < r(\ell, t)$. Then, at stage t , we diagonalise at $B_t(\Lambda_p(k))$; thus, at the next stage, $r(\ell, t + 1) \leq k \leq r(\ell, t)$. This means that the sequence $(r(\ell, s))_{s \geq t_0}$ is decreasing and bounded below by $r(\ell - 1)$; therefore, the limit $r(\ell)$ exists, a contradiction.

Case 2. Suppose that there is no such t_0 . We wish to show that $r(\ell)$ exists. Say there is some stage $t_1 \geq s_0$ where $(\forall t \geq t_1) [\hat{\Phi}_{e, t}^{B_t \setminus T_p}(\Lambda_p(r(\ell, t_1))) \uparrow]$. In this case, we know that $(\forall t \geq t_1) [r(\ell, t) \leq r(\ell, t_1)]$. Then, for all $t \geq \mathbf{m}(r(\ell, t_1))$, we have $r(\ell, t + 1) = r(\ell, t)$, so that the limit $r(\ell)$ exists, a contradiction.

Case 3. If there are no t_0 or t_1 , then we can define a computable function $f(x) = (\mu s \geq s_1) [r(\ell, s) \geq x]$. Then, $(\forall^\infty x) [\mathbf{m}(x) \leq f(x)]$, so that $(\forall^\infty x) [A(x) = A_{f(x)}(x)]$. But then A is computable, a contradiction.

Therefore, each sequence $(r(\ell, s))_{s \in \omega}$ converges. Then, by definition of $r(\ell, s)$, we must have

$$(\forall^\infty s) \left[\neg \left(\hat{\Phi}_{e, s}^{B_s \setminus T_p}(\Lambda_p(r(\ell))) \downarrow = B_s(r(\ell)) \right) \right] ,$$

and in particular

$$\neg \left(\hat{\Phi}_e^{B \setminus T_p}(\Lambda_p(r(\ell))) \downarrow = B(r(\ell)) \right) .$$

Hence, each requirement \mathcal{N}_ℓ is eventually satisfied.

We satisfy \mathcal{R} by permitting and \mathcal{P} by coding into S . □

Finally, we show that, although it is easy to find Δ_2 sets cl-equivalent to a given c.e. set, and in contrast to the result in Lemma 1.16, there is a Δ_2 set—a c.e. real, in fact—strictly cl-above *every* c.e. set.

Proposition 4.12. *There is a c.e. real A such that, for each c.e. set W , $W \leq_{ib\Gamma} A$ but $A \not\leq_{rK} W$. In particular, $W <_{ib\Gamma} A$, $W <_{cl} A$, and $W <_{rK} A$.*

Proof. Let $(W_e)_{e \in \omega}$ be an effective enumeration of c.e. sets. Let $(f_i)_{i \in \omega}$ be an effective listing of binary partial computable functions in $\omega \times \omega \rightarrow \omega$, and for each i let $(f_{i,s})_{s \in \omega}$ be the sequence of s -step approximations to f_i . Let $\langle \cdot, \cdot, \cdot \rangle : \omega \times \omega \times \omega \rightarrow \omega$ be a ternary Gödel numbering such that, for any a, b, c , we have $\langle a, b, c \rangle \geq a, b, c$.

Requirements

$$\begin{aligned} \mathcal{P}_e : & & W_e \leq_{ib\Gamma} A \\ \mathcal{N}_{\langle e, i, k \rangle} : & A \text{ is not } rK\text{-below } W_e \text{ via } f_i(\cdot, 0), \dots, f_i(\cdot, k) \end{aligned}$$

In order to code each of the c.e. sets into A , we shall introduce them one by one and code longer and longer (and hence more compressed) segments into A . That is, we shall start by coding an initial segment of W_0 into A ; then a longer segment of W_0 and a segment of W_1 ; then, longer still, W_0 , W_1 , and W_2 ; and so on. Because the length of a coded segment in A is logarithmic in that of the original segment in W_e , it will not be difficult to choose an appropriate length at each step to fit in each W_e and have some space left over to use for the $\mathcal{N}_{\langle e, i, k \rangle}$.

The way we fulfill requirement $\mathcal{N}_{\langle e, i, k \rangle}$ is by simple diagonalisation. We reserve k entries of A for this purpose, say $A \cap [n, n+k)$, and make sure that, if $f_i(W_e \upharpoonright (n+k-1), j) \downarrow = \sigma$, then $A(n+j) \neq \sigma(n+j)$. We can ensure that this is a c.e. real by requiring that $W_e \upharpoonright (n+k-1)$ be coded into $A \upharpoonright (n-1)$ —so that whenever one of our diagonalisations is injured by a change to W_e , there will be a change to the coding somewhere in $A \upharpoonright (n-1)$, allowing us to reset $A \cap [n, n+k)$ and perform each diagonalisation again.

More precisely, when $n \geq e$, we can use $A \cap [2^n + e \cdot (n+2), 2^n + e \cdot (n+2) + n+2)$ to code $W_e \upharpoonright 2^{n+2} - 1$. We then have the leftover space in $A \cap [2^n + (n+1) \cdot (n+2), 2^{n+1})$

for other coding. Supposing that we want this last interval to have length at least n , we need $n \geq 6$.

Construction

For each e , let $(W_{e,s})_{s \in \omega}$ be the natural enumeration of W_e .

At stage $s \geq 6$:

Step 1. For each n, e with $e \leq n \leq s$, code

$$A_s \cap [2^n + e \cdot (n + 2), 2^n + e \cdot (n + 2) + n + 2) = |W_{e,s} \uparrow 2^{n+2} - 1| .$$

Step 2. For each $n \leq s$, take e, i, k such that $n = \langle e, i, k \rangle$. For each $j \leq k$, let

$$A_s(2^n + (n + 1) \cdot (n + 2) + j) = \begin{cases} 1 & \text{if } f_{i,s}(W_{e,s} \uparrow (2^{n+1} - 1), j) \downarrow = \sigma \\ & \text{with } \sigma(2^n + (n + 1) \cdot (n + 2) + j) = 0 \\ 0 & \text{otherwise} \end{cases}$$

Let $A = \lim_s A_s$.

Verification

If an element is removed from A at stage s in step 1, then, by place-value, some smaller element must at the same time enter A . If an element is removed in step 2, then there must have been some injury of the form $W_{e,s} \uparrow (2^{n+2} - 1) \neq W_{e,s-1} \uparrow (2^{n+2} - 1)$ to the corresponding computation, in which case—because $e \leq \langle e, i, k \rangle = n$ —a smaller element was added to $A_s \cap [2^n + e \cdot (n + 2), 2^n + e \cdot (n + 2) + n + 2)$ in step 1. Therefore, A is indeed a c.e. real.

Requirement \mathcal{P}_e is met simply enough: for each e, m , with $m \geq 2^{6+2} = 256$, let $\ell = \lceil \log_2(m + 1) \rceil$. Then $m \leq 2^\ell - 1$; if $e \leq \ell - 2$, we can compute $W_e(m)$ simply by running the approximation $(W_{e,s}(m))_{s \in \omega}$ and waiting for an s with $|W_{e,s} \uparrow (2^\ell - 1)| = A \cap [2^{\ell-2} + e \cdot (\ell - 2 + 2), 2^{\ell-2} + e \cdot (\ell - 2 + 2) + \ell - 2 + 2)$. Hence $W_e(m)$ is computed from $A \uparrow 2^{\ell-1}$, and $2^{\ell-1} \leq 2^{\log_2(m+1)} - 1 \leq m$, so this is an ibT reduction.

For each $n = \langle e, i, k \rangle$, and any j such that $0 \leq j \leq k$, we know by Step 2 of the construction that $A \uparrow (2^{n+1} - 1) \neq f_i(W_e \uparrow (2^{n+1} - 1), j)$. Hence \mathcal{N}_n is met. \square

References

- [1] George Barmpalias. Computably enumerable sets in the solovay and the strong weak truth table degrees. In S. Barry Cooper, Benedikt Löwe, and Leen Torenvliet, editors, *CiE*, volume 3526 of *Lecture Notes in Computer Science*, pages 8–17. Springer, 2005.
- [2] George Barmpalias and Andrew E. M. Lewis. The ibT degrees of computably enumerable sets are not dense. *Ann. Pure Appl. Logic*, 141(1-2):51–60, 2006.
- [3] S. Barry Cooper. *Computability theory*. Chapman & Hall/CRC, Boca Raton, FL, 2004.
- [4] R. G. Downey. Δ_2^0 degrees and transfer theorems. *Illinois J. Math.*, 31(3):419–427, 1987.
- [5] Rod Downey and Denis R. Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer-Verlag, 2010 (To appear).
- [6] Rod Downey, Denis R. Hirschfeldt, André Nies, and Sebastiaan A. Terwijn. Calibrating randomness. *Bull. Symbolic Logic*, 12(3):411–491, 2006.
- [7] Rod G. Downey, Denis R. Hirschfeldt, and Geoff LaForte. Randomness and reducibility. *J. Comput. System Sci.*, 68(1):96–114, 2004.
- [8] Richard M. Friedberg. Two recursively enumerable sets of incomparable degrees of unsolvability (solution of Post’s problem, 1944). *Proc. Nat. Acad. Sci. U.S.A.*, 43:236–238, 1957.
- [9] S. C. Kleene and Emil L. Post. The upper semi-lattice of degrees of recursive unsolvability. *Ann. of Math. (2)*, 59:379–407, 1954.

- [10] Andrew E. M. Lewis and George Barmpalias. Randomness and the linear degrees of computability. *Ann. Pure Appl. Logic*, 145(3):252–257, 2007.
- [11] Per Martin-Löf. The definition of random sequences. *Information and Control*, 9:602–619, 1966.
- [12] A. A. Muchnik. Negative answer to the problem of reducibility of the theory of algorithms. *Dokl. Akad. Nauk SSSR*, 108:194–197, 1956. (in Russian).
- [13] André Nies. *Computability and Randomness*. Oxford University Press, 2009.
- [14] Alexander Raichev. *Relative Randomness via rK -Reducibility*. PhD thesis, University of Wisconsin–Madison, 2006.
- [15] Gerald E. Sacks. The recursively enumerable degrees are dense. *Ann. of Math. (2)*, 80:300–312, 1964.
- [16] J. R. Shoenfield. On degrees of unsolvability. *Ann. of Math. (2)*, 69:644–653, 1959.
- [17] Robert I. Soare. *Computability Theory and Applications*. Springer-Verlag Monograph, to appear.
- [18] Robert I. Soare. Computability theory and differential geometry. *Bull. Symbolic Logic*, 10(4):457–486, 2004.
- [19] Clifford Spector. On degrees of recursive unsolvability. *Ann. of Math. (2)*, 64:581–592, 1956.
- [20] Michiel van Lambalgen. *Random sequences*. PhD thesis, University of Amsterdam, 1987.
- [21] Liang Yu and Decheng Ding. There is no SW -complete c.e. real. *J. Symbolic Logic*, 69(4):1163–1170, 2004.