COMPUTABLE REDUCIBILITY OF EQUIVALENCE RELATIONS

by

Marcello Gianni Krakoff

A thesis
submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Mathematics
Boise State University

May 2019

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the thesis submitted by

Marcello Gianni Krakoff

Thesis Title: Computable Reducibility of Equivalence Relations

Date of Final Oral Examination: 12 March 2019

The following individuals read and discussed the thesis submitted by student Marcello Gianni Krakoff, and they evaluated his presentation and response to questions during the final oral examination. They found that the student passed the final oral examination.

Samuel Coskey, Ph.D.	Chair, Supervisory Committee
John D. Clemens, Ph.D.	Co-Chair, Supervisory Committee
Marion Scheepers, Ph.D.	Member, Supervisory Committee

The final reading approval of the thesis was granted by Samuel Coskey, Ph.D., Chair of the Supervisory Committee. The thesis was approved by the Graduate College.

To Ezra.

ACKNOWLEDGMENTS

I would to thank the Boise State University Math department for the summer research fellowship financial support in 2018, as well as the teaching assistantship. I would like to thank Marion Scheepers for serving on the supervisory committee for this thesis. The other math grad students at Boise State for their stimulating if not sometimes distracting conversations. I would express gratitude to Sam Coskey for his guidance throughout this process. John Clemens for always having his office door open and allowing me to bug him with questions. And most of all Kayla, for her patience and love.

ABSTRACT

Computable reducibility of equivalence relations is a tool to compare the complexity of equivalence relations on natural numbers. Its use is important to those doing Borel equivalence relation theory, computability theory, and computable structure theory. In this thesis, we compare many naturally occurring equivalence relations with respect to computable reducibility. We will then define a jump operator on equivalence relations and study proprieties of this operation and its iteration. We will then apply this new jump operation by studying its effect on the isomorphism relations of well-founded computable trees.

TABLE OF CONTENTS

DEDICATION iv

ACKNOWLEDGMENTS v

ABSTRACT vi

1 Introduction 1

1.1 Introduction 1

1.2 Borel Equivalence Relation Theory 2

1.3 Computability Theory 6

1.4 Computable Structure Theory 10

1.5 Trees 12

2 Computability Theoretic Hierarchies 14

2.1 Arithmetical Hierarchy 14

2.2 Hyperarithmetical Hierarchy 16

2.3 Analytical Hierarchy 19

3 Computable Reducibility of Equivalence Relations 25

3.1 Basics 25

3.2 Computably Enumerable Equivalence Relations 28

3.3 Equivalence Relations on c.e. Sets 36

3.4 Relative Computable Reducibility 41

4 A Jump Operator on Equivalence Relations 45

4.1 Friedman-Stanley Jump and its Computable Variant 45

4.2 Benchmark Equivalence Relations 49

4.3 Iterating Computable FS-jump through Computable Ordinals 60

4.4 Computable FS-jump fixed points 61

5 Computable Trees of Computable Ordinal Rank 63

5.1 Isomorphism Relation on Computable Trees 63

5.2 Well-Founded Trees 67

REFERENCES 69

A Forcing 72

CHAPTER 1

INTRODUCTION

1.1 Introduction

Computable reducibility is a notion of reduction of equivalence relations on ω.

Definition 1.1.1. Let E and F be equivalence relations on ω. We say that E is *computably reducible* to F, denoted $E \leqslant F$, if there is a computable function $f : \omega \to \omega$ such that for all $n, n' \in \omega$

$$nEn' \longleftrightarrow f(n)Ff(n').$$

Computable reducibility is of interest to many different fields of mathematics such as descriptive set theory, computability theory, and computable structure theory. To descriptive set theorists, specifically those working in Borel equivalence relation theory, computable reducibility is a natural computable analogue to Borel reducibility. One may investigate analogous equivalence relations on ω that mirror Borel equivalence relations and see to what extent the computable reducibility hierarchy mimics the Borel reducibility hierarchy. To computability theorists it serves as another notion of reducibility to compare the complexity of subsets of natural numbers (or tuples thereof). Of particular interest is the structure of computably enumerable equivalence relations (ceers) under computable reducibility. To computable structure theorists the notion is important in studying equivalence relations on computable structures. For example, one may ask how complicated is the isomorphism relation or how complicated is the computable isomorphism relation for various classes of computable structures such as trees, groups, graphs, etc.

In chapter 1 we will explain the fields for which computable reducibility is of interest as well as build up machinery to be used in this thesis. In chapter 2 we will give a comprehensive outline of the computability theoretic hierarchies of which we will use to classify equivalence relations. In chapter 3 we survey results about computable reducibility in the literature. In chapter 4 we define a jump operator analogous to the Friedman-Stanley jump in Borel equivalence relation theory for equivalence relations on natural numbers. Finally in chapter 5 we look at the isomorphism relation on trees of bounded rank and relate it to the jump operation defined in chapter 4.

1.2 Borel Equivalence Relation Theory

The standard reference for claims and theorems regarding Borel theory presented here is Gao [11] and Kanovei [14].

Definition 1.2.1. A *Polish space* is a complete, separable, metrizable space. A *Borel set* of a topological space is a member of the least σ-algebra containing all the open sets. A function f is a *Borel function* if for every Borel set B, $f^{-1}(B)$ is a Borel set.

Borel equivalence relation theory studies the complexity of equivalence relations on Polish Spaces. For the majority of this paper the polish spaces we will consider are Baire space, ω^ω, or Cantor Space, 2^ω. Let X be a Polish space; a Borel equivalence relation is an equivalence relation that is a Borel subset of $X \times X$ with the inherited product topology. Two Borel equivalence relations may be compared the following notion of reducibility.

Definition 1.2.2. Let X and Y be Polish spaces, with Borel equivalence relations E and F defined on each space respectively. We say that E is *Borel reducible* to F, denoted $E \leqslant_B F$, if there is a Borel function $f : X \to Y$ such that for all $x, x' \in X$

$$xEx' \longleftrightarrow f(x)Ff(x').$$

It is clear from the definitions of computable reducibility and Borel reducibility why computable reducibility is natural to Borel complexity theorists. A computable (Borel) reduction of equivalence relations has two requirements; the forward direction requires that images of elements that were E-equivalent become F-equivalent and the reverse direction requires that images of elements that were E-inequivalent become F-inequivalent. Thus the F equivalence relation forms a set of complete invariants for the E equivalence relation.

One use of Borel equivalence relation theory is the classification of the isomorphism relation on a space of countable models. At a high level, a space of countable models is constructed by encoding each countable model as a real number and taking at the corresponding Polish space of such reals.

Definition 1.2.3. Let $\mathcal{L}$ be a countable relational language, i.e. $\mathcal{L} = \{R_i\}$ for $i \in I$, where I is a countable index set. To each R_i we assign a natural number n_i, the arity of R_i. Let $\mathsf{Mod}(\mathcal{L})$ denote the collection of all countable models of $\mathcal{L}$, where we identify the universe of each model with ω. Thus each element of $\mathsf{Mod}(\mathcal{L})$ may be identified with an element of

$$X_{\mathcal{L}} = \prod_{i \in I} 2^{\omega^{n_i}}.$$

The set $X_{\mathcal{L}}$ is a product of countably many copies of Cantor space, 2^{ω} and thus is a compact Polish space with the product topology. Explicitly, for each $x \in X_{\mathcal{L}}$ let $\mathcal{M}_x \in \mathsf{Mod}(\mathcal{L})$ be a countable model coded by x in the following way; for any $i \in I$ and $(k_1, \dots, k_{n_i}) \in \omega^{n_i}$,

$$R_i^{\mathcal{M}_x}(k_1, \dots, k_{n_i}) \longleftrightarrow x_i(k_1, \dots, k_{n_i}) = 1.$$

The isomorphism relation will be induced by the action of S_∞ (the group of permutations of ω) on $\mathsf{Mod}(\mathcal{L})$ by saying that $g \cdot \mathcal{M} = \mathcal{N}$ if and only if

$$R_i^{\mathcal{N}}(k_1, \dots, k_{n_i}) = R_i^{\mathcal{M}}(g^{-1}(k_1), \dots, g^{-1}(k_{n_i}))$$

for all $i \in I$ and $(k_1, \dots, k_{n_i}) \in \omega^{n_i}$. Thus $g \cdot \mathcal{M} = \mathcal{N}$ if and only if g is an isomorphism from $\mathcal{M}$ onto $\mathcal{N}$. The orbit equivalence relation of this action is exactly the isomorphism relation on $\mathsf{Mod}(\mathcal{L})$ and is denoted $\cong_{\mathcal{L}}$.

For example, consider the space of countable graphs. The language of graphs consists of one binary relation E denoting the edge relation. We may then identify the space of all countable models of graphs in the space

$$G \subset 2^{\omega^2}.$$

Where $x \in G$ if and only if $\mathcal{G}_x \in \mathsf{Mod}(\{E\})$ is a countable graph coded by x. The action by S_∞ on graphs may be thought of as a permutation of the vertices.

We further restrict the space of models in the following way; let φ be an $\mathcal{L}_{\omega_1,\omega}$ sentence (we allow countable conjunctions, disjunctions, and finite quantifiers) in the language $\mathcal{L}$, define the set

$$\mathsf{Mod}(\varphi) = \{\mathcal{M} \in \mathsf{Mod}(\mathcal{L}) \mid \mathcal{M} \vDash \varphi\}.$$

We denote the isomorphism relation on models of φ as $\cong_\varphi$.

Dichotomy theorems are an important tool in Borel equivalence relation theory in order to get non-reduction results.

Theorem 1.2.1. *(Silver) If X is Polish space and E is a coanalytic equivalence relation on X, then either there are countably many E-classes or there is a perfect set of pairwise E-inequivalent elements of X.*

Let X be a set, define the identity equivalence relation $\mathtt{Id}(X)$ as

$$x\mathtt{Id}(X)y \longleftrightarrow x = y.$$

Since every Borel set is coanalytic, it follows that every Borel equivalence relation is subject to Silver's dichotomy theorem. One can use Silver's dichotomy theorem to show that if E is smooth (an equivalence relation E is *smooth* if $E \leqslant_B \mathtt{Id}(2^\omega)$), then exactly one of the following holds: either $E \sim_B \mathtt{Id}(2^\omega)$ or $E \leqslant_B \mathtt{Id}(\omega)$. If E has a perfect set of pairwise inequivalent elements, then $\mathtt{Id}(2^\omega) \leqslant_B E$. If E has countably many classes we may enumerate them in a Borel way and then may map the n^{th}-class to n. Define the equivalence relation E_0 to be the finite difference equivalence relation on 2^ω; formally,

$$xE_0y\longleftrightarrow(\exists m)(\forall n)(n \geqslant m \to x(n) = y(n)).$$

The most famous dichotomy result is called the Glimm-Effros dichotomty theorem.

Theorem 1.2.2. *(Harrington-Kechris-Louveau) If X is a Polish space and E is a Borel equivalence relation on X, then either E is smooth or $E_0 \leqslant_B E$.*

In fact, the Glimm-Effros dichotomy indicates something stronger; in the case that E is not smooth, it gives us that E_0 continuously embeds into E. Proofs of both the Silver and Glimm-Effros dichotomies make vital use of effective methods and the Gandy-Harrington topology with the associated notion of forcing (see [14], Theorem 10.1.1 and Theorem 10.4.1). It was shown in Coskey et al. [6] that there is no analogue to Silver's theorem for computable reducibility. Though it remains open whether there is an analogue to the Glimm-Effros dichotomy for computable reducibility. While there may be no obvious dichotomy theorems for computable reducibility, there is a powerful tool for obtaining non-reducibility results that Borel theory does not have, and that is the appeal to the descriptive complexity of the equivalence relations.

The structure of Borel equivalence relations form a highly non-trivial hierarchy under the notion of Borel reducibility. Computable reducibility compares equivalence relations on ω in the same way but with computable functions instead of Borel functions.

1.3 Computability Theory

The references for most computability theory presented here is Soare [18] and Cooper [5].

We will assume that the reader is familiar with computability up to Turing machines. Since each Turing machine is finitely presented, we may effectively assign to each machine a natural number called a code or index. To each (code for) Turing machine e there is an associated partial function $\varphi_e : \omega \to \omega$ so that $\varphi_e(n) \downarrow = m$ if and only if machine e halts on input n and outputs m. The set of all n such that $\varphi_e(n) \downarrow$ is called the domain of φ_e and is denoted W_e. A function $f : \omega \to \omega$ is called *computable* if it is partial computable and is defined for all $n \in \omega$. A set $A \subset \omega$ is *computable* if its characteristic function χ_A is computable; in other words, χ_A is a total φ_e for some e. A set $A \subset \omega$ is called *c.e.* (computably enumerable) if it is the domain of a partial computable function; in other words, if $A = W_e$ for some index e. The set $K = \{e \mid \varphi_e(e) \downarrow\}$ is c.e. since we may run a machine on the code for itself and, if it halts, we enumerate the index into K; however, if it does not halt in some timely manner we may never know its fate.

There are many notions of reducibility in computability theory. Reducibility compares the complexity or information content with subsets of natural numbers.

Definition 1.3.1. Let $A, B \subset \omega$, we say A is *many-one reducible* to B, written $A \leqslant_m B$ if there is a computable function f such that for all x

$$x \in A \longleftrightarrow f(x) \in B.$$

If f is injective, we will say A is *one-one reducible* to B and write $A \leqslant_1 B$.

For sets A and B we say that A is *Turing reducible* to B, written $A \leqslant_T B$ if there is a program that computes A relative to B, i.e. using B as an oracle. That is, we can ask questions of the form “Is $n \in B$?” when trying to decide if some natural number is in A. We say that $A \equiv_T B$ if $A \leqslant_T B$ and $B \leqslant_T A$. Note that $\equiv_T$ is an equivalence relation and each equivalence class is called a *Turing degree*. The least Turing degree is denoted $\mathbf{0}$ and is the equivalence class of all computable sets. Bold lowercase latin letters will denote Turing degrees. Let A be a set, define $\varphi_e^A(x)$ to be the e^{th} machine run on x with oracle A. Since we can identify subsets of ω with reals, we say a machine has a given real as an oracle. The jump set of A, denoted A', is $\{e \in \omega \mid \varphi_e^A(e) \downarrow\}$. The halting set, denoted K, is identified with the previous set where A is taken to be the empty set.

Next we present several key theorems from computability theory which will be used throughout this paper.

Theorem 1.3.1. *(Padding Lemma) For a given e, there are infinitely many e' so that $\varphi_e = \varphi_{e'}$.*

Proof. Given a φ_e, then for each n there is machine that waits for n steps and then mimics φ_e. □

Theorem 1.3.2. *(s-m-n Theorem) For each m and n, and each φ_e in variables $\overline{y} = (y_1, \ldots, y_m)$ and $\overline{x} = (x_1, \ldots x_n)$, there is a computable injective function f of $m + 1$ variables such that*

$$\varphi_{f(e,\overline{y})}(\overline{x}) = \varphi_e(\overline{y}, \overline{x}).$$

The s-m-n theorem allows us to treat a parameter of a function as fixed and effectively find a new index for a machine.

We will denote the complement of a set A by $\overline{A}$. A subset $A \subset \omega$ is called *simple* if A

is c.e., $\overline{A}$ is infinite and A meets every infinite c.e. set. If A is simple then $\overline{A}$ is called *immune.* Simple sets arose out of Post's attempt to solve the question of whether there are intermediate c.e. degrees. This problem was solved independently by Friedburg and Muchnik using the priority method. Another solution is given by the existence of a low simple set. We will be particularly interested in immunity proprieties later in this paper. The following construction given in Cooper [5] Theorem 6.2.3 gives a proof the existence of a simple set. Notice the construction is effective and thus simple sets are c.e.

Theorem 1.3.3. *There exists a simple set A.*

Proof. We computably enumerate A that satisfy the following requirements:

$$\begin{aligned} \mathcal{N}_e: &\quad |A \cap \{0, 1, \ldots, 2e\}| \leqslant e \\ \mathcal{P}_e: &\quad \text{If } W_e \text{ is infinite, then } W_e \cap A \neq \varnothing. \end{aligned}$$

for all $e \in \omega$. If $\mathcal{N}_e$ for all e is satisfied then $|\overline{A} \cap \{0, 1, \ldots, 2e\}| > e$, so $\overline{A}$ is infinite. Also, requirement $\mathcal{P}_e$ is part of the definition of simple set. This is essentially a priority argument with no injury since we have two competing requirements, one of which attempts to put elements into A while the other tries to keep elements out of A. The construction is as follows:

(i) For each *unsatisfied* $\mathcal{P}_e$, *wait* for a stage s at which there is a number $x \in W_{e,s}$ with $x > 2e$.

(ii) If such an x appears, enumerate x into A, at which stage $\mathcal{P}_e$ becomes satisfied.

Lemma 1.3.1. *$\mathcal{P}_e$ is satisfied for each $e \in \omega$.*

Proof. Assume that W_e is infinite. Let s be the least stage at which we get some $x \in W_{e,s}$ with $x > 2e$. By the construction at stage s, $\mathcal{P}_e$ is not already satisfied, which means that it becomes so via part (ii) of the construction at stage s via $x \in W_e \cap A$. □

Lemma 1.3.2. *$\mathcal{N}_e$ is satisfied for each $e \in \omega$.*

Proof. Since we can only enumerate a number x into A with $x \leqslant 2e$ on behalf of some W_i with $i < e$ and for each i at most one such x is enumerated, the lemma follows. □

This proves the existence of a simple set. □

We will often give a description of a machine that computes a reduction since we may, given a description of a machine, effectively find an index for a machine with that description.

1.4 Computable Structure Theory

Computable structure theory investigates the syntactic properties of familiar structures and how they relate to the intrinsic complexity of structures. The complexity is most often measured in Turing degrees. The degree of structure $\mathcal{A}$, denoted $\deg(\mathcal{A})$, is the least upper bound of the degree of the universe, relations, and functions. It follows that a structure $\mathcal{A}$ is computable if its domain is ω (or computable subset thereof) and all functions and relations on $\mathcal{A}$ are computable. For example, a countable linear order is computable if there is a program that can determine, for each pair (a, b), whether $a < b$ or $a > b$.

Definition 1.4.1. Let $\mathcal{L}$ be a computable language and $\mathcal{M}$ be an $\mathcal{L}$-structure. Let $\mathcal{L}_M$ be the language of $\mathcal{M}$ with constants added for each $m \in M$. The *atomic diagram* of $\mathcal{M}$ is the set of all atomic, or negation of atomic, formulas, $\varphi(\overline{m})$ in $\mathcal{L}_M$ such that $\mathcal{M} \vDash \varphi(\overline{m})$.

Definition 1.4.2. A model is said to be computable if its atomic diagram is computable. For a computable model $\mathcal{M}$, an *index* for $\mathcal{M}$ is a number e such that φ_e enumerates the codes for atomic sentences and negated atomic sentences α such that $\mathcal{M} \vDash \alpha$.

For example, let $\mathcal{N} = \langle \omega, +, \cdot \rangle$ be the standard model of arithmetic and $\mathtt{Th}(\mathcal{N}) = \{\varphi \mid \mathcal{N} \vDash \varphi\}$, which is a theory called true arithmetic. The model $\mathcal{N}$ is computable since the structure has computable atomic diagram. The atomic and negated atomic sentences of this language consist of only equality and negated equality. Since addition and multiplication are computable operations, we can check the veracity of each atomic formula of arithmetic. It is important to differentiate between computable and decidable models. A model is decidable if, given any formula in the language, we can tell whether the model satisfies that formula. The model $\mathcal{N}$ is computable, but not decidable by Tarski's theorem. The theory $\mathtt{Th}(\mathcal{N})$ is recursively isomorphic to $\mathbf{0}^{(\omega)}$[1] ([5], Theorem 15.2.1), so it is far from being computable. For each structure, we may define the degree spectrum. This is the set of degrees of structures isomorphic to $\mathcal{A}$, $\mathrm{Spec}(\mathcal{A}) = \{\deg(\mathcal{B}) \mid \mathcal{A} \cong \mathcal{B}\}$. A seminal result in the field is given by the following.

Theorem 1.4.1. *(Knight) The degree spectra of a non-trivial structure is upward closed in the Turing degrees.*

In other words, if A is subset of ω and for a non-trivial structure $\mathcal{M} <_T A$, there is an isomorphic copy of $\mathcal{M}$, denoted $\mathcal{N}$, such that $\mathcal{N} \equiv_T A$. The non-trivialness

[1] $\mathbf{0}^{(\omega)} = \deg(\varnothing^{(\omega)})$ where $\varnothing^{(\omega)} = \{\langle i, n \rangle \mid i \in \varnothing^{(n)}\}$.

condition is that $\mathcal{M}$ and $\mathcal{N}$ are not computably isomorphic. For instance, consider the structure $\langle \omega, \leqslant \rangle$, which is the standard ordering on ω. Let $\mathbf{a}$ be a Turing degree and let $\{a_i \mid i \in \omega\}$ be an $\mathbf{a}$-computable enumeration of some $A \in \mathbf{a}$. Define a new ordering on ω by ordering the evens by placing $2i+1$ in between $2a_i$ and $2a_i+2$. Note that on both ordered structures we can define a successor operation, where $s(x)$ is the $\leqslant$-least element so that $x \leqslant y$. The succesor on the first structure is computable, whereas the successor operation on second structure computes A.

Definition 1.4.3. The *index set* for $\mathcal{M}$ is the set we denote $I(\mathcal{M})$ of all indices of computable isomorphic copies of $\mathcal{M}$. For a class of structures K, closed under isomorphism, the index set $I(K)$ is the set of all indices of computable members of K. We say that K has a *has a computable characterization* if its index set is hyperarithmetical.

In this paper we will be looking at the isomorphism relation on computable trees.

1.5 Trees

A tree $\mathcal{T}$ is a subset of $\omega^{<\omega}$ that is closed under predecessor i.e. if $\tau \in \mathcal{T}$ and $\sigma \leqslant \tau$, then $\sigma \in \mathcal{T}$. The top node will be denoted with $\varnothing$, where $\varnothing$ is its own predecessor. A tree will be computable if its predecessor function is computable. A path through a tree $\mathcal{T}$ is a real x (an element of Baire space) such that every initial segment of x is in $\mathcal{T}$. The body of a tree $\mathcal{T}$ denoted $[\mathcal{T}]$ is the set of paths in $\mathcal{T}$,

$$[\mathcal{T}] = \{x \in \omega^\omega \mid (\forall n)(x \upharpoonright_n \in \mathcal{T})\}.$$

A tree $\mathcal{T}$ is well-founded if and only if $[\mathcal{T}] = \varnothing$.

Computable trees can be effectively coded in several ways; we give one such way. For each $n > 0$ there is a bijection $f : \omega^n \to \omega$. An example of such a function for $n = 2$ is given by

$$f(n, m) = \frac{(n+m)(n+m+1)}{2} + n.$$

Examples of other pairing functions for $n > 2$ can be found by induction on the $n = 2$ case. We denote $f(x_1, \ldots, x_n)$ by $\langle x_1, \ldots, x_n \rangle$. We will use this function to code computable trees in the following way. First, for a computable tree $\mathcal{T}$, find the root. Then label all the children of the root with increasing natural numbers. Do similar inductively for each of the branches coming off each node. For instance, $\{(), (0), (1), (0,0), (0,1), (1,0)\}$ is the tree with two branches coming off the root and the left branch coming off the root has two children and right branch coming of the root has one child. Thusly we may formally identify a tree with sequences that describe its branches. Let tree $\mathcal{T}$ be identified with the following set

$$\{\langle s_0, \ldots, s_n \rangle \mid \text{for all } (s_0, \ldots, s_n) \in \mathcal{T}\}.$$

In this sense a tree will be computable if the set of codes for its branches is a computable set.

CHAPTER 2

COMPUTABILITY THEORETIC HIERARCHIES

2.1 Arithmetical Hierarchy

In the quest to understand the descriptive complexity of equivalence relations on ω, we must have a well-defined hierarchy to classify such relations. We define three hierarchies, each properly extending the previous, the arithmetical, hyperarithmetical, and analytic. A subset A of ω is called arithmetical if it is definable in the language of Peano Arithmetic ($\mathtt{PA}$). The complexity of the class of arithmetical sets is structured into the arithmetical hierarchy. The set Σ^0_0 of computable relations form the base of the arithmetical hierarchy. We say a relation $S(x)$ is Σ^0_1 or computable enumerable (c.e.) if there is a computable relation R such that $S(x) = (\exists y)R(x, y)$. A relation is Π^0_1 (co-c.e.) if it is the negation of a c.e. relation. A relation is Π^0_n if it is the complement of a Σ^0_n relation. The recursive clause of the arithmetical hierarchy states that a relation $S(x)$ is Σ^0_{n+1} when there is a Π^0_n relation R so that $S(x) = (\exists y)R(x, y)$. The Δ^0_n classes are exactly those relations that are simultaneously Σ^0_n and Π^0_n. A result by Kleene states that a relation is computable if it and its complement are both c.e. Thus if a relation has both a Σ^0_1 and a Π^0_1 definition it is computable. So the computable relations are also known as the Δ^0_1 class. The arithmetical hierarchy has many nice closure properties; the Σ^0_n sets are closed under conjunction, disjunction, bounded quantification and existential quantification.

The hyperarithmetical and analytical hierarchy are motivated by the study of subsets natural numbers that are not arithmetical, but are still definable in other ways. One explicit example of a non-arithmetical, reasonably definable set of natural numbers is the set $\{\sharp\varphi \mid \varphi \in \mathtt{Th}(\mathcal{N})\}$, where $\sharp\varphi$ is the Gödel code of φ. This is a direct result of the Tarski's theorem of the undefinability of truth; that is, there is no formula in the language of PA that takes in a Gödel code of a formula and outputs a yes or no answer as to whether that sentence is satisfied in the standard model. The classes $\Sigma^0_n, \Pi^0_n, \Delta^0_n$ and those classes that we will define for the hyperarithmetic and analytic hierarchy are called point classes.

Definition 2.1.1. Let Γ be a point class, say a set A is Γ-complete if $A \in \Gamma$ and for all equivalence relations $B \in \Gamma$, $B \leqslant_m A$.

The arithmetical hierarchy can also be defined by iterating the Turing jump through the finite ordinals. This is justified by Post's Theorem, which is as follows:

Theorem 2.1.1. *(Post) Let $A \subset \omega$ then for all $n \geqslant 0$ the following hold*

(i) $\varnothing^{(n+1)}$ is a Σ^0_{n+1}-complete set;

(ii) $A \in \Sigma^0_{n+1}$ if and only if A is c.e. in $\varnothing^{(n)}$;

(iii) $A \in \Delta^0_{n+1}$ if and only if $A \leqslant_T \varnothing^{(n)}$.

There are two ways to extend the arithmetical hierarchy. One way is to iterate the Turing jump through the computable ordinals using ordinal notations, which gives us the hyperarithmetical hierarchy. The other is to notice that the arithmetical relations have natural number quantifiers, so if we extend this to allow natural number function quantifiers, or equivalently real number quantifiers, we get the analyitcal hierarchy. First we define the hyperarithmetical hierarchy.

2.2 Hyperarithmetical Hierarchy

The standard reference for these topics is Sacks [16], though we follow Ash and Knight [4] which gives a more modern treatment. Ordinal notations are a way to effectively assign natural numbers to computable ordinals. An ordinal is computable if there is a computable well-ordering of natural numbers with that order-type. For example, $\omega \cdot 2$ is a computable ordinal since $0, 2, 4, \ldots, 1, 3, 5, \ldots$ is a computable well-ordering of ω.

Definition 2.2.1. The set $\mathcal{O} \subset \omega$, called Kleene's $\mathcal{O}$, is a system of notations which assigns a natural number a to each computable ordinal α, and we denote $|a|_{\mathcal{O}} = \alpha$:

(i) The ordinal 0 is given the notation 1;

(ii) If a is a notation for α, the 2^a is the notation given to $\alpha + 1$;

(iii) For limit λ, the notations are numbers $3 \cdot 5^e$ such that φ_e is a total computable function with values in $\mathcal{O}$, and α is the least upper bound of $\alpha_n = |\varphi_e(n)|_{\mathcal{O}}$.

There is a natural partial order structure on $\mathcal{O}$. Note that the notations for transfinite computable ordinals are not unique. Since there are only countably many codes, there is a least countable ordinal that does not have a notation, that ordinal is $\omega_1^{\mathbf{ck}}$.

The hyperarithmetical hierarchy is a generalization of the arithmetical hierarchy where the Turing jump is iterated through the computable ordinals. To do this effectively, we must iterate the jump through the notations for computable ordinals.

Definition 2.2.2. Define the sets H_a for $a \in \mathcal{O}$ by transfinite recursion on ordinals $|\alpha|$ as follows:

(i) $H_1 = \varnothing$;

(ii) $H_{2^a} = (H_a)'$;

(iii) $H_{3 \cdot 5^e} = \{\langle u, v\rangle \mid u <_{\mathcal{O}} 3 \cdot 5^e \text{ and } v \in H_u\}$.

Under this system each finite ordinal n has unique notation a such that $H_a = \varnothing^{(n)}$. If $|a| = |b| = \alpha$, where a and b are distinct, then H_a and H_b are distinct sets. Furthermore, since α is infinite, there are infinitely many notations a for α that define infinitely many distinct sets H_a. The key that allows the hierarchy to be well defined is that if $|a| = |b| = \alpha$, then $H_a \equiv_T H_b$. We will prove this in several steps.

Lemma 2.2.1. *(Kleene) There is a partial computable function which assigns to each pair (x, y) of elements of $\mathcal{O}$ such that $x \leqslant_{\mathcal{O}} y$, an index for H_x as a set computable in H_y.*

Proof. If $x, y \in \mathcal{O}$ with $x \leqslant_{\mathcal{O}} y$, then by shortening the stack of 2's in y we arrive at $z \leqslant_{\mathcal{O}} y$ and finite n such that $|z| + n = y$. There are now two cases. The first case is $x = z$, for any $n \in \omega$ we can uniformly find an index for X as a set computable in $X^{(n)}$. In the second case, $|z|$ is a limit ordinal and $x <_{\mathcal{O}} z$. Then we have

$$u \in H_x \longleftrightarrow \langle x, u\rangle \in H_z.$$

So we can find an index for H_x computably in H_z. We may then find an index for H_x as a set computable in H_y by the first case. □

Lemma 2.2.2. *(Kleene) There is a partial computable function f such that for each $a \in \mathcal{O}$, $f(a)$ is an index for $\{b \in \mathcal{O} \mid |b| < |a|\}$as a set computable in $(H_a)'$.*

Proof. The function is defined by computable transfinite recursion on ordinal notations. For case 1 suppose that $a = 1$ and let $f(1)$ be an index for $\{b \in \mathcal{O} \mid |b| < |a|\} = \varnothing$ relative to $\varnothing'$. Now for case 2, suppose that $a = 2^b$; we show how to define $f(a)$ in terms of $f(b)$ already defined. Note that $d \in \mathcal{O}$ and $|d| < |2^b|$ if and only if one of the following holds

(i) $d = 1$;

(ii) d has the form 2^c, where $c \in \mathcal{O}$ and $|c| < |b|$;

(iii) d has the form $3 \cdot 5^e$, where for all n, $\varphi_e(n) \in \mathcal{O}$, $|\varphi_e(n)| < |b|$ and $\varphi_e(n) < \varphi_e(n+1)$.

To see whether d satisfies (i) is trivial. Assume that $f(b)$ is an index for $\{c \in \mathcal{O} \mid |c| < |b|\}$ relative to $(H_b)'$; we can determine whether $d = 2^c$ satisfies (ii) using H_{2^b}. We can determine whether $d = 3{\cdot}5^e$ satisfies (iii) using $(H_{2^b})'$ since for each n we can determine whether $\varphi_e(n) \in \mathcal{O}$ and $|\varphi_e(n)| < |b|$, using H_{2^b} and we determine $\varphi_e(n) <_{\mathcal{O}} \varphi_e(n+1)$. Now we can find an index for $\{d \in \mathcal{O} \mid |d| < |2^b|\}$ as a set computable in $(H_{2^b})'$ and define that index to be $f(2^b)$. Finally in case 3, $a = 3 \cdot 5^e$. Given the sequence $f(\varphi_e(n))$ for $n < \omega$, we define $f(3 \cdot 5^e)$ as follows. We know:

$$d \in \mathcal{O} \wedge |d| < |3 \cdot 5^e| \longleftrightarrow (\exists n)(d \in \mathcal{O} \wedge |d| < |\varphi_e(n)|).$$

Assuming that $f(\varphi_e(n))$ is an index for $\{d \in \mathcal{O} \mid |d| < |\varphi_e(n)|\}$ as a set computable in $(H_{\varphi_e(n)})'$, we can determine whether d is a notation for an ordinal less that $|\varphi_e(n)|$. The procedure here is uniform in n. Then we determine whether d is in the union using

$(H_{3 \cdot 5^e})'$. It follows that we can computably find an index for $\{d \in \mathcal{O} \mid |d| < |3 \cdot 5^e|\}$ as a set computable in $(H_{3 \cdot 5^e})'$ and define that index to be $f(3 \cdot 5^e)$. □

Corollary 2.2.1. *(Kleene) There are partial computable functions that assign to each $b \in \mathcal{O}$ indices for $\{a \in \mathcal{O} \mid |a| \leqslant |b|\}$ and $\{a \in \mathcal{O} \mid |a| = |b|\}$ as sets computable in $H_{2^{2^b}}$.*

Theorem 2.2.1. *(Kleene) There is a partial computable function f such that for $a, b \in \mathcal{O}$ with $|a| \leqslant |b|$, $f(a, b)$ is an index for H_a as a set computable in H_b.*

Proof. Similar as above, see [4]. □

This theorem gives us that $H_a \leqslant_T H_b$ for notations $|a| \leqslant |b|$. Thus, if equality holds, then $H_a \equiv_T H_b$.

2.3 Analytical Hierarchy

Similarly to iterating the Turing jump through the computable ordinals to extend the arithmetical hierarchy to get the hyperarithmetical hierarchy, we extend by allowing the quantification over natural numbers functions or reals to get the analytical hierarchy. As before with the arithmetical hierarchy, for a point class Γ, we will say a relation R is Γ, if it is defined by a Γ formula.

Definition 2.3.1. Let $R(n, x)$ be a computable relation of natural number n and real x, then

(i) A Σ^1_1 relation is of the form $(\exists x)(\forall n)R(n, x)$;

(ii) A relation is Σ^1_n is of the form $(\exists x)(\forall x_1) \cdots (Q x_{n-1})(\forall k)R(x, x_1, \ldots, x_{n-1}, k)$, where Q is $\exists$ or $\forall$ depending on the parity of n;

(iii) A relation R is Π^1_1 is $\overline{R}$ is Σ^1_1;

(iv) If a relation is both Σ^1_1 and Π^1_1 then it will be Δ^1_1.

In this paper we will be mostly concerned with Σ^1_1 and Π^1_1 relations. We will now prove an important fact that demonstrates the link between the hyperarithmetic and analytical hierarchy, but first we will need an important fact about Kleene's $\mathcal{O}$.

Theorem 2.3.1. *(Kleene) Kleene's $\mathcal{O}$ is a Π^1_1-complete set.*

We will prove this theorem in several steps. First we will define the Kleene-Brouwer ordering on trees. Then we will show that for a Π^1_1 set there is a computable function that assigns codes for well-founded trees on exactly members of that set. Finally, we will show that $\mathcal{O}$ is in fact Π^1_1.

Lemma 2.3.1. *Given an index for a computable linear order $\langle A, < \rangle$, there is a computable function f so that $\{m \mid m < n\}$ is well-ordered if and only if $f(n) \in \mathcal{O}$, and $f(n)$ is the notation for the order-type of $\{m \mid m < n\}$.*

Definition 2.3.2. For a tree $\mathcal{T} \subset \omega^{<\omega}$, define the Kleene-Brouwer ordering of $\mathcal{T}$ by

$$s <_{KB} t \longleftrightarrow t \subset s \vee (\exists i \in \omega)(\forall j < i)(s(j) = t(j) \wedge s(i) < t(i)).$$

Lemma 2.3.2. *If $\mathcal{T} \subset \omega^{<\omega}$ is a tree then $\mathcal{T}$ is well-founded under the Kleene-Brouwer ordering if and only if $\mathcal{T}$ has no path.*

Proof. If $\mathcal{T}$ is well-founded under the Kleene-Brouwer order, then there is no infinite descending sequence in $\mathcal{T}$ so $\mathcal{T}$ can have no path. If there is an infinite descending sequence then there is a sequence

$$\sigma_0 > \sigma_1 > \sigma_2 > \cdots$$

in $\mathcal{T}$ so that, by induction on n, $\sigma_k(n)$ is constant for all sufficiently large k. Thus define

$$\pi(n) = \lim_{k\to\infty} \sigma_k(n),$$

then π is a path in $\mathcal{T}$. □

The next theorem gives that for Π^1_1 sets we can effectively find an index for a computable well-order.

Theorem 2.3.2. *(Kleene-Spector) Let S be Π^1_1, then there is a computable function that assigns to each $n \in \omega$ an index for a linear order $\mathcal{A}_n$ such that*

$$n \in S \longleftrightarrow \mathcal{A}_n \textit{ is a well-ordering.}$$

Proof. By the normal form theorem for Π^1_1 sets then for some c.e. relation $R(n,m)$ we have

$$n \in S \longleftrightarrow (\forall x)(\exists t)R(n, x\restriction_t).$$

Let A_n be the subtree of $\omega^{<\omega}$ consisting of those σ such that if σ has length s, then for all $t \leqslant s$, the pair $(n, \sigma\restriction_t)$ is not in our stage s approximation of $R(n,m)$. Then A_n is a computable subtree of $\omega^{<\omega}$. Moreover,

$$n \in S \longleftrightarrow A_n \text{ has no path.}$$

Let $\mathcal{A}_n = (A_n, <_n)$, where $<_n$ is the restriction of the Kleene-Brouwer ordering A_n. Then by the above lemma

$$A_n \text{ has no path} \longleftrightarrow \mathcal{A}_n \text{ is a well ordering.}$$

Therefore, given n, we can find an effective index for $\mathcal{A}_n$. □

Putting this altogether, we get that if S is Π^1_1 then we can effectively find an index for a computable well-order. Then given a computable well-ordering, we can pass effectively to an element of Kleene's $\mathcal{O}$ and therefore $S \leqslant_m \mathcal{O}$, so Kleene's $\mathcal{O}$ is a Π^1_1-complete set. The crucial link between the hyperarithmetical and analytical hierarchies is that a subset of natural numbers is hyperarithmetic if and only if it is Δ^1_1.

Theorem 2.3.3. *(Kleene) X hyperarithmetic implies X is Δ^1_1.*

Proof. Given a hyperarithmetic X we will produce two predicates, one Π^1_1 and one Σ^1_1, that both express membership in X. First note the each of the following predicates

$$a \in \mathcal{O} \wedge b \in H_a \quad \text{and} \quad a \in \mathcal{O} \wedge b \notin H_a$$

are Π^1_1 since they are the conjunction of all real solutions to the predicate which expresses membership in $\mathcal{O}$. One can show this rigorously through proper coding, see in [16](Lemma II.2.1). Now recall Kleene's T predicate which which expresses the following; for a real x and a code, s, for a finite initial segment of x (representing only a finite amount of an oracle is used in any oracle computation) then for U a computable function

$$\varphi^x_e(n)\downarrow= m \longleftrightarrow (\exists k)(\exists s)(T(s,e,n,k) \wedge U(k) = m).$$

Now let $\varphi(e, a, n, i)$ express the following statement

$$\begin{aligned}(\exists k)(\exists s)(T(s,e,n,k) \wedge \mathrm{lh}(s) = n \wedge U(k) = i \\ \wedge (\forall b < k)(s(b) = 1 \rightarrow a \in \mathcal{O} \wedge b \in H_a) \\ \wedge (\forall b < k)(s(b) = 0 \rightarrow a \in \mathcal{O} \wedge b \notin H_a)).\end{aligned}$$

By the previous remark this formula is Π_1^1, thus if we fix x and e and assume that $X = W_e^{H_a}$, that is, X is enumerated by some partial computable function with hyperarithmetic oracle H_a then we have

$$\begin{aligned}n \in X &\longleftrightarrow \varphi(e,a,n,1) \\ &\longleftrightarrow \neg\varphi(e,a,n,0).\end{aligned}$$

Thus membership in X is expressible as both a Π_1^1 statement and a Σ_1^1 statement and hence is Δ_1^1. □

For the following theorem define the set $\mathcal{O}_b = \{a \in \mathcal{O} \mid |a| < |b|\}$. We will need the following theorem, and though we will not provide a proof, one can be found in ([16], Corollary I.5.6).

Theorem 2.3.4. *(Spector Boundedness Lemma) Let $A \subset \mathcal{O}$ and A be Σ_1^1, then there is an element $b \in \mathcal{O}$ such that $|a| \leqslant |b|$ for all $a \in A$.*

Theorem 2.3.5. *(Kleene) If X is Δ_1^1 then X is hyperarithmetic*

Proof. Let X be Δ_1^1; since $\mathcal{O}$ is Π_1^1-complete there is a computable function g such that for all $n \in \omega$

$$n \in X \longleftrightarrow g(n) \in \mathcal{O}.$$

Define a set A by the following

$$m \in A \longleftrightarrow (\exists n)(n \in X \wedge m = g(n)).$$

Notice that membership in A is Δ^1_1 and hence Σ^1_1. Thus by Spector's boundedness lemma there is $b \in \mathcal{O}$ such that $A \subset \mathcal{O}_b$. Thus we have

$$n \in X \longleftrightarrow g(n) \in \mathcal{O}_b.$$

However, as was showed in an above theorem, the set $\mathcal{O}_b$ is hyperarithmetic, since we can effectively find an index for $\mathcal{O}_b$ in H_{2^b}. Thus X is hyperarithmetic. □

This theorem can also extend to the hyperarithmetic sets over ω^ω (and products thereof). The light face classes, which start with what we call the effective open sets, is some computable enumeration of a countable basis for ω^ω. We close under effective countable union and complements, as done when we define the Borel hierarchy to get Δ^1_1. If we relativize the effective unions to all reals x i.e. $\Delta^1_1(x)$, coupled with the fact that

$$\bigcup_{x \in \omega^\omega} \Delta^1_1(x) = \boldsymbol{\Delta^1_1}$$

we recover a famous theorem of Suslin.

Theorem 2.3.6. *(Suslin) X is $\boldsymbol{\Delta^1_1}$ if and if only X is Borel.*

CHAPTER 3

COMPUTABLE REDUCIBILITY OF EQUIVALENCE RELATIONS

3.1 Basics

Recall the definition of computable reducibility given in section 1.1.

Definition 3.1.1. Let E and F be equivalence relations on ω. We say that E is *computably reducible* to F, denoted $E \leqslant F$, if there is a computable function $f : \omega \to \omega$ such that for all $n, n' \in \omega$

$$nEn' \longleftrightarrow f(n)Ff(n').$$

From now on all equivalence relations will be on ω unless otherwise specified. A computable reduction of equivalence relations is very similar to a many-one reduction in computability theory. The difference is subtle but very important. Given a set of pairs A and B, for a many-one reduction we must specify a computable function that takes in a pair from A and outputs a pair in B and sends pairs in $\overline{A}$ to pairs in $\overline{B}$. For a computable reduction, we specify what the computable function does to a natural number and show that this function preserves the equivalence relation structure. When we write $|E|$ we mean the number of equivalence classes in E.

Theorem 3.1.1. *If $E \leqslant F$, then $|E| \leqslant |F|$.*

Proof. A computable reduction must preserve equivalence classes, thus if $|E| > |F|$, then two E-inequivalent elements must become F-equivalent. This contradicts the definition of computable reducibility. □

Theorem 3.1.2. *If $E \leqslant F$, then $E \leqslant_m F$.*

Proof. Suppose f witnesses $E \leqslant F$, then we have a computable function f so that for all pairs (n, m)

$$(n, m) \in E \longleftrightarrow (f(n), f(m)) \in F.$$

□

To illustrate the two previous theorems, let E be the equivalent mod 3 relation and F be the equivalent mod 2 relation. Since both equivalence relations E and F are finite, non-empty, and computable, they are both many-one equivalent as sets of pairs, that is $E \equiv_m F$. However, there can be no computable reduction from E to F since both equivalence relations are finite and E has more classes than F. A powerful method of obtaining non-computable reducbility results comes from the following theorem.

Theorem 3.1.3. *If Γ is a point class, $B \in \Gamma$ and $A \leqslant_m B$, then $A \in \Gamma$.*

We will give an example of how the previous theorem works. Suppose that $A \leqslant_m B$, and B is Π^0_2. Then there is a computable relation R so that

$$k \in B \longleftrightarrow (\forall n)(\exists m) R(k, n, m).$$

The reduction $A \leqslant_m B$ gives us a function f so that $i \in A$ if and only if $f(i) \in B$. Putting this together we get

$$\begin{aligned} i \in A &\longleftrightarrow f(i) \in B \\ &\longleftrightarrow (\forall n)(\exists m) R(f(i), n, m). \end{aligned}$$

Thus membership in A is defined by a Π_2^0 relation.

To see how this theorem is used, suppose equivalence relation E is computably reducible to some Π_2^0 equivalence relation and suppose that F is a known properly Σ_3^0 equivalence relation. Then by the above theorem $F \nleqslant E$. This makes computable reducibility distinct from Borel reducibility in the following sense. If E and F are Borel equivalence relations then it is possible that $E \leqslant_B F$, even if F has lower rank in the Borel hierarchy than E does.

Definition 3.1.2. An equivalence relation E is Γ-*complete for computable reducibility* if $E \in \Gamma$ for all F, $F \leqslant E$.

Clearly if E is Γ-complete for computable reducibility then E is Γ-complete.

Corollary 3.1.1. *Suppose $A \subset E$ is an equivalence class of E. If $A \in \Gamma$, then E is in some point-class Γ' extending Γ.*

Proof. Note that the identity function is a computable function that witnesses $A \leqslant_m E$. □

This corollary will allow us to get lower bounds on the descriptive complexity of an equivalence relation by looking at the the complexity of its classes.

By properties for computable functions we immediately have that $E \leqslant E$ and if $E_1 \leqslant E_2$ and $E_2 \leqslant E_3$ then $E_1 \leqslant E_3$. Thus $\leqslant$ defines a quasi-order on the set of equivalence relations on ω. So we can define the equivalence relation $E \equiv F$, where we say E is computably bireducible with F if and only if $E \leqslant F$ and $F \leqslant E$. We will call the set of equivalence relations on ω modulo $\equiv$ the computable reducibility hierarchy.

Theorem 3.1.4. *The computable reducibility hierarchy is a partial order.*

This will be proven in the section on c.e. equivalence relations. Throughout this thesis we will show that the computable reducibility hierarchy is rich in structure and complicated to study. So we may consider substructures of the computable reducibility hierarchy, such as the set of computably enumerable equivalence relations modulo $\equiv$.

3.2 Computably Enumerable Equivalence Relations

Definition 3.2.1. An equivalence relation E on ω is called a ceer (computably enumerable equivalence relation) if E is a c.e. subset of ω^2.

Intuitively, if E is a ceer we may enumerate the E-equivalent pairs. We can study the structure of the equivalence classes of ceers under computable reducibility in the same way we might study the Turing degrees. In this section we will investigate the structure of the ceers under computable reducibility. Ceers were first introduced in Gao and Gerdes [12]. A detailed analysis of the structure of ceers under computable reducibility is given in [1], [2], and [3].

We will define two operations on ceers that increase the complexity of ceers.

Definition 3.2.2. For a ceer E define the halting jump of E by

$$nE'm \longleftrightarrow n = m \vee \varphi_n(n) \downarrow E\varphi_m(m) \downarrow .$$

That is, n is E'-equivalent to m if and only if $n = m$ or the n^{th} machine run on n halts and the m^{th} machine run on m halts and both outputs are E-equivalent.

Theorem 3.2.1. *[12] If E is a ceer then $E \leqslant E'$.*

Proof. Let f send n to the machine that halts on n and outputs n, thus

$$\begin{aligned} nEm &\longleftrightarrow f(n) = f(m) \vee \varphi_{f(n)}(f(n)) \downarrow E\varphi_{f(m)}(f(m)) \downarrow \\ &\longleftrightarrow f(n)E'f(m). \end{aligned}$$

□

Notice that the halting jump operator does not increase the descriptive complexity of a ceer, that is, E' is still a ceer. If E is computable, then $E < E'$, since if this was not the case we would have a properly c.e. equivalence class mapping into a computable class.

Theorem 3.2.2. *[12] For ceers E and F, if $E \leqslant F$ then $E' \leqslant F'$.*

Proof. Suppose f witnesses $E \leqslant F$. By the s-m-n theorem find a computable injective function g such that for all $n, k \in \omega$ $\varphi_{g(n)}(k) = f(\varphi_n(n))$. Then $\varphi_{g(n)}(g(n))$ halts if and only if $f(\varphi_n(n))$ does, and outputs the same value. The fact that g is injective guarantees the following

$$\begin{aligned} nE'm &\longleftrightarrow n = m \vee \varphi_n(n) \downarrow E\varphi_m(m) \downarrow \\ &\longleftrightarrow n = m \vee f(\varphi_n(n)) \downarrow Ff(\varphi_m(m)) \downarrow \\ &\longleftrightarrow g(n) = g(m) \vee \varphi_{g(n)}(g(n)) \downarrow F\varphi_{g(m)}(g(m)) \downarrow \\ &\longleftrightarrow g(n)F'g(m). \end{aligned}$$

Thus $E' \leqslant F'$. □

The next operation is the join of ceers (though the join can be defined for any equivalence relations on ω). For ceers E and F define the join ceer

$$uE{\oplus}Fv \longleftrightarrow \begin{cases} nEm & \text{if } u = 2n \text{ and } v = 2m, \\ nFm & \text{if } u = 2n+1 \text{ and } v = 2m+1. \end{cases}$$

Theorem 3.2.3. *$E \leqslant E{\oplus}F$ and $F \leqslant E{\oplus}F$.*

Proof. The computable function that maps n to $2n$ witnesses $E \leqslant E{\oplus}F$ and the computable function that maps n to $2n+1$ witnesses $F \leqslant E{\oplus}F$. □

We now classify all computable (ceers) up to computable bireducibility and look at some of the structure of ceers under computable reducbility. Let $\mathtt{Id}_n$ be equivalence relations such that $a\mathtt{Id}_n b$ if and only if $a \equiv b \mod n$.

Lemma 3.2.1. *[6] If ceer E has finitely many equivalence classes then E is computable.*

Proof. Suppose E has n classes, let $i_1, \ldots, i_n$ be a collection of pairwise E-inequivalent elements. We give a procedure to decide aEb for arbitrary a and b. Start enumerating E-equivalent pairs until we find aEi_{j_i} and bEi_{j_2}, then aEb if and only if $j_1 = j_2$. □

Note that E being a ceer is necessary to the proof, since we enumerated the E-equivalent pairs. There are Δ^0_2 equivalence relations with finitely many classes that are not computably bi-reducible with $\mathtt{Id}_n$ for some n.

Theorem 3.2.4. *[6] If E is a ceer with exactly n equivalence classes then E is computably bi-reducible with $\mathtt{Id}_n$.*

Proof. Let $i_1, \ldots, i_n$ be a collection of pairwise E-inequivalent elements. Define function f so that $f(a)$ computes the division algorithm on a by n and maps the remainder r to i_{r+1}. Then f computable and $a\mathtt{Id}_n b$ if and only if a and b have the same remainder when divided by n means that $f(a)Ef(b)$. Thus f witnesses $\mathtt{Id}_n \leqslant E$. Now suppose that E is a ceer with n many classes, then E is computable and we can order the E classes. Then given a, map a to $na + j$ just in case a is in the j^{th} class. Then f is computable and if aEb and a is the j^{th} class then b is as well, then $na + j\mathtt{Id}_n nb + j$. And if a is not E-equivalent to b then they are in different classes say i and j. Then $na + i$ is not $\mathtt{Id}_n$-equivalent to $nb + j$, thus $E \leqslant \mathtt{Id}_n$. □

All ceers with finitely many classes are computably bireducible with $\mathtt{Id}_n$ for some n. Lastly we show the $\mathtt{Id}_n$ for $n < \omega$ form a strict hierarchy.

Theorem 3.2.5. *[6]* $\mathtt{Id}_n < \mathtt{Id}_{n+1}$.

Proof. One can see that $\mathtt{Id}_{n+1} \nleqslant \mathtt{Id}_n$, since under any computable reduction at least two elements that were $\mathtt{Id}_{n+1}$-inequivalent would become $\mathtt{Id}_n$-equivalent. Define the function $f(k)$ that computes the remainder r of k by n and then maps k to $(n+1)k+r$. $\square$

This gives us a strict hierarchy of the ceers with finitely many classes

$$\mathtt{Id}_1 < \mathtt{Id}_2 < \cdots < \mathtt{Id}_n < \cdots .$$

Let $\mathtt{Id}$ be the identity equivalence relation, $x\mathtt{Id}y$ if and only if $x = y$.

Theorem 3.2.6. *[6] For each n, $\mathtt{Id}_n < \mathtt{Id}$.*

Proof. Similar as to the proof that $\mathtt{Id}_n < \mathtt{Id}_{n+1}$. $\square$

Theorem 3.2.7. *[6] For any computable ceer E with infinitely many classes $E \equiv \mathtt{Id}$.*

Proof. To see $E \leqslant \mathtt{Id}$, order the equivalence classes by least element. To do this observe that since E is computable there is algorithm that answers yes or no whether a number is in a class, thus for a given class simply ask if 0 is in the class, if 1 is in the class, and so on. This process is computable and will terminate, and give us the least element in the class. The map the sends a to j just in case a is in the j^{th} class witnesses the reduction. To see $\mathtt{Id} \leqslant E$, enumerate E-inequivalent elements $i_1, i_2, \ldots$ then send a to i_a. $\square$

Note that the above theorem works for infinite Π^0_1 equivalence relations, since we can enumerate E-inequivalent pairs. The computable ceers form an $\omega + 1$ order-type initial segment of structure of the ceers under computable reducibility

$$\mathtt{Id}_1 < \mathtt{Id}_2 < \cdots < \mathtt{Id}_n < \cdots < \mathtt{Id}.$$

To complete the study of the ceers under computable reducibility it remains to classify the non-computable ceers. The structure of ceers under computable reducbility fork at the computable ceers, those which the identity relation is computably reducible to and those that the identity is not computably reducible to, thus we make the following definition.

Definition 3.2.3. A ceer E is called *light* if $\mathtt{Id} \leqslant E$ and E is called *dark* if it has infinitely many equivalence classes and is not light.

One might characterize the light ceers are those which we have an effective enumeration of pairwise E-inequivalent elements.

Definition 3.2.4. Let $A \subset \omega$, define the following equivalence relation

$$nE_A m \longleftrightarrow n = m \vee n, m \in A.$$

Theorem 3.2.8. *[6] There is a dark ceer.*

Proof. Consider the equivalence relation E_A where A is a simple set. Suppose for contradiction that f witnesses $\mathtt{Id} \leqslant E_A$. First we note that if f witnesses $\mathtt{Id} \leqslant E$ for any E then f is injective. Thus there is at most one $n \in \omega$ such that $f(n) \in A$. Suppose $f(n) \in A$ then $f(\omega \setminus \{n\})$ is an infinite c.e. subset of $\overline{A}$. If there is no n such that $f(n) \in A$ then $f(\omega)$ is an infinite c.e. subset of $\overline{A}$. Both of these cases contradict the fact that $\overline{A}$ is immune. □

Dark ceers are not always of the form E_A for A a simple set. For example in [7] they constructed a dark ceer with only finite classes.

Definition 3.2.5. A ceer U is called universal if for all ceers E, $E \leqslant U$.

Theorem 3.2.9. *There is a universal ceer.*

Define the relation enumerated by φ_e in the following way. To each n in the domain of φ_e associate a unique ordered pair (k, m) so that, via the pairing function, $\langle k, m\rangle = n$. Now take the transitive closure of that relation along with the diagonal and for each (k, m) include (m, k), this will be an equivalence relation denoted E_e. All ceers can be defined this way. To see this let E be a ceer then the E-equivalent pairs are enumerated by some φ_e. By the s-m-n theorem we can find an e' so that $\varphi_{e'}$ halts on n if and only if $\langle k, m\rangle = n$ and kEm.

Proof. Define the universal c.e. relation by $\langle e, a\rangle U_{ce}\langle e', a'\rangle$ if and only if $e = e'$ and aE_ea'. The map $a \mapsto \langle e, a\rangle$ witnesses $E_e \leqslant U_{ce}$. □

Observe that for a universal ceer U then $\mathtt{Id} < U$. For a dark ceer E, $E \nleqslant \mathtt{Id}$, for the following reason. Suppose $E \leqslant \mathtt{Id}$, we know that $\mathtt{Id} \nleqslant E$, thus $E < \mathtt{Id}$. But this means that $E \equiv \mathtt{Id}_n$ for some n, but E by definition has infinitely many classes. Thus if $U \leqslant \mathtt{Id}$, then $E \leqslant \mathtt{Id}$, which is a contradiction. Note that that if E is a universal ceer then $E' \equiv E$, thus the halting jump is not a proper jump operator. We will define a proper jump operator for equivalence reltaion on ω in chapter 4.

The next two theorems will show the relationship between computable reducbility and one-one reducibility and ceers of the form E_A where A is a c.e. set.

Theorem 3.2.10. *If A and B are properly c.e. then $E_A \leqslant E_B$ if and only if $A \leqslant_1 B$.*

In [6] A and B are required to be both properly c.e. and in [1] A and B are just required to be c.e. and B infinite.

Proof. Suppose that A and B are infinite c.e. and that $A \leqslant_1 B$ via f. Then f, since it is injective, witnesses the reduction of E_A to E_B since

$$\begin{aligned} nE_Am &\longleftrightarrow n = m \vee n, m \in A \\ &\longleftrightarrow f(n) = f(m) \vee f(n), f(m) \in B \\ &\longleftrightarrow f(n)E_Bf(m). \end{aligned}$$

Now suppose that f witnesses $E_A \leqslant E_B$, we wish to build an injective g that witnesses $A \leqslant_1 B$. Note that either the ran$f \restriction_A \subset B$ or is a singleton. If ran$f \restriction_A$ were a singleton, say $\{n\}$, then the $f^{-1}(\{n\}) = A$ would be computable contrary to A being properly c.e.. Thus ran$f \restriction_A \subset B$ and so f witnesses $A \leqslant_m B$, moreover is already injective on $\overline{A}$. Then we inductively define a function g so that $g(0) = f(0)$ and $g(n+1) = f(n+1)$ if for all $k \leqslant n$, $f(n+1) \neq g(k)$. If $f(n+1) = g(k)$ then we need $f(n+1) \in B$, enumerate B until an element appears not equal to any other $g(k)$ and set $g(n+1)$ to be that number. □

Theorem 3.2.11. *Suppose E is a ceer and A is a c.e. set. If $\mathtt{Id} \leqslant E \leqslant E_A$ then there is a c.e. set B such that $E \equiv E_B$.*

Proof. Let A be c.e. and let f witness $E \leqslant E_A$. Note that the range of f is infinite, thus we can define a computable bijection $f : \text{ran}f \to \omega$. We claim the set B is the range of ran$f \cap A$. First we show that $g \circ f$ witnesses $E \leqslant E_B$

$$\begin{aligned} xEy &\longleftrightarrow f(x)E_Af(y) \\ &\longleftrightarrow f(x) = f(y) \vee f(x), f(y) \in A \\ &\longleftrightarrow f(x) = f(y) \vee f(x), f(y) \in A \cap \operatorname{ran} f \\ &\longleftrightarrow g(f(x)) = g(f(y)) \vee g(f(x)), g(f(y)) \in B \\ &\longleftrightarrow g(f(x))E_Bg(f(y)) \end{aligned}$$

The computable function $h(x) = (\mu y)(g(f(y)) = x)$ witnesses $E_B \leqslant E$.

□

The last theorem, proved by Andrews and Sorbi in [2], gives an idea of the richness the structure of the ceers under computable reducibility.

Theorem 3.2.12. *[2] Let E be a non-universal ceer. There are pairwise incomparable dark ceers E_i for $i \in \omega$ such that for every i and ceer F, $E_i \nleqslant E$ and*

$$F < E_i \rightarrow (\exists n)(F \leqslant \mathtt{Id}_n).$$

3.3 Equivalence Relations on c.e. Sets

In order to prove some results in the next section we give some preliminary results from Coskey et al. [6]. There are many benchmark equivalence relations in Borel equivalence relation theory that can be ported down to the computable world by considering them restricted to indices for c.e. sets. Equality on c.e. sets will be denoted $=^{ce}$, where $e =^{ce} e'$ if and only $W_e = W_{e'}$. In general, if we are considering some equivalence relation E, then when E is only considered restricted to indices for c.e. sets we will denote it by E^{ce}. We will sometimes use e and e' for natural numbers here to emphasize when we are interpreting our natural numbers as indices for c.e. sets.

Definition 3.3.1. For any equivalence relation E, then define E^{ce} to be

$$eE^{ce}e' \longleftrightarrow W_e \; E \; W_{e'}.$$

For example if E is an equivalence relation on reals, then E^{ce} is that same equivalence relation but restricted to c.e. reals.

Theorem 3.3.1. *[6] $=^{ce}$ is a Π^0_2-complete set of pairs.*

Proof. First note that the following formula shows that $=^{ce}$ is Π^0_2

$$e =^{ce} e' \longleftrightarrow (\forall n)((\exists s) n \in W_{e,s} \longleftrightarrow (\exists t) n \in W_{e',t}).$$

To see that $=^{ce}$ is Π^0_2-complete, recall that the set $\texttt{TOT} = \{e \mid W_e = \omega\}$ is Π^0_2-complete ([18], Theorem IV.3.2). $\texttt{TOT}$ is a $=^{ce}$-class. There is m-reduction from $\texttt{TOT}$ to that $=^{ce}$-class by function that, on input e, outputs n just in case φ_e halts on input n. □

Theorem 3.3.2. *[6] If E is a ceer, then $E <=^{ce}$.*

Proof. First note that $=^{ce} \not\leq E$, by descriptive complexity. Each E-class is only c.e. and by the previous theorem $=^{ce}$ has a Π^0_2-complete class. To see there is a reduction of E to $=^{ce}$, notice every E-class is c.e. so let f send each n to the machine e that enumerates $[n]_E$. Then nEm if and only if $[n]_E = [m]_E$. □

Definition 3.3.2. The finite difference, or almost equal, equivalence relation E_0 is given by AE_0B if and only if $A \triangle B$ is finite. Then $eE_0^{ce}e'$ if and only if $W_e \triangle W_{e'}$ is finite.

Theorem 3.3.3. *[6] E_0^{ce} is a Σ^0_3-complete set of pairs.*

Proof. The following formula shows that E_0^{ce} is Σ^0_3. Let $\varphi(m, e, e')$ be a Π^0_2-formula that expresses m is in the symmetric difference of W_e and $W_{e'}$ then

$$eE_0^{ce}e' \longleftrightarrow (\exists n)(\forall m)(\varphi(m, e, e') \to m \leqslant n).$$

To see that E_0^{ce} is Σ^0_3-complete, recall that the set `COF` $= \{e \mid W_e \text{ is cofinite}\}$ is Σ^0_3-complete ([18], Theorem IV.3.5). Any cofinite W_e has a finite symmetric difference with any other cofinite $W_{e'}$, thus E_0^{ce} has `COF` as one of its classes. □

Theorem 3.3.4. *[6] $=^{ce} < E_0^{ce}$.*

Proof. We show first that $=^{ce} \leqslant E_0^{ce}$, given an index e define $f(e)$ as follows. Whenever n is enumerated into W_e then $f(e)$ enumerates codes for the pairs $(n, 0), (n, 1), (n, 2), \ldots$. Then we have W_e and $W_{e'}$ differ if and only if $W_{f(e)}$ and $W_{f(e')}$ differ infinitely often. The descriptive complexity of the equivalence relations prevents the reduction in the other direction since E_0^{ce} has a Σ^0_3-complete class and $=^{ce}$ is Π^0_2. □

If E is a Borel equivalence relation then when restricted to indices for c.e. sets or equivalently c.e. reals, the complexity of E^{ce} directly relates to the complexity of E. At a high level there is an existential quantifier introduced on the inside of the formula defining E. This gives justification for appropriately considering equivalence relations on reals and then considering the c.e. version E^{ce} of E.

Theorem 3.3.5. *Let E be a Borel equivalence relation. Then if E is Π^0_{2n+1} then E^{ce} is Π^0_{2n+2}.*

For this proof will proceed by induction but we will only show the base case since that is where all the work actually happens. The proof makes crucial use of quantifier contraction ([18], Theorem II.1.3), by which a formula that has two universal quantifiers, with appropriate coding, may be expressed as one universal quantifier. The same is true for existential quantifiers.

Proof. We work with the following encoding of c.e. reals on 2^ω into natural numbers. Recall that $\varphi_{e,s}(n)\downarrow$ is a computable relation $R(e,s,n)$ of natural number. A real x is a c.e. real coded by some index e, if the n^{th} digit of x, $x(n)$, is given in the following way:

$$\begin{aligned} x(n) = 1 &\longleftrightarrow n \in W_e \\ &\longleftrightarrow \varphi_e(n)\downarrow \\ &\longleftrightarrow (\exists s)\varphi_{e,s}(n) \\ &\longleftrightarrow (\exists s)R(e,s,n). \end{aligned}$$

Let E be a Π^0_1 Borel equivalence relation defined by

$$E(x,y) \longleftrightarrow (\forall n)(\forall m)S(x,y,n,m).$$

for some computable relation S. For the given formula expressing relationship between coordinates of reals, $x(n)$ and $y(m)$, then $S(x,y,n,m)$ becomes, for some computable R' of natural numbers, an expression for E^{ce} where e and e' code c.e. reals in the above defined way

$$E^{ce}(e,e') \longleftrightarrow (\forall n)(\forall m)(\exists s)(\exists t)R'(e,e',s,t,n,m).$$

By quantifier contraction E^{ce} is Π^0_2. □

Corollary 3.3.1. *(i) If E is Π^0_{2n} then E^{ce} is Π_{2n}.*

(ii) If E is Σ^0_{2n+1} then E^{ce} is Σ_{2n+1}.

(iii) If E is Σ^0_{2n} then E^{ce} is Σ_{2n+1}.

Proof. This is immediate by taking complements when necessary and appropriate use of quantifier contraction. □

For example consider Turing equivalence, $\equiv_T$, as a Borel equivalence relation. Turing equivalence is a Σ^0_3 Borel equivalence relation since

$$\begin{aligned} x \leqslant_T y &\longleftrightarrow (\exists e)(x = W^y_e) \\ &\longleftrightarrow (\exists e)(\varphi^y_e \text{ is total}) \wedge (\forall n)(x(n) = 1 \longleftrightarrow \varphi^y_e(n) \downarrow= 1). \end{aligned}$$

is Σ^0_3 and taking the conjunction of $x \leqslant_T y$ and $y \leqslant_T x$ doesn't add to the descriptive complexity. Thus by the above theorem $\equiv^{ce}_T$, that is

$$e \equiv^{ce}_T e' \longleftrightarrow W_e \equiv_T W_{e'}$$

is a Σ^0_3 equivalence relation on natural numbers.

Let us look at an example that leads to an open problem in computable reducibility. Notice that $\mathtt{Id}(2^\omega) \leqslant_B \equiv_T$ since there is a perfect set of pairwise Turing incomparable reals, this follows immediately from Silver's dichotomy theorem. On the other hand $\equiv_T$ is not a smooth Borel equivalence relation. Since E_0 is a sub equivalence relation of $\equiv_T$, i.e. if xE_0y then $x \equiv_T y$, then if $E_0 \leqslant_B \mathtt{Id}(2^\omega)$ then $\equiv_T \leqslant_B \mathtt{Id}(2^\omega)$. To that end suppose that $E_0 \leqslant_B \mathtt{Id}(2^\omega)$ via f. This would be absolute over some ground model

V. In the forcing extension $V[g]$ adding a Cohen real g the real x such that $f(g) = x$ does not depend on g, since any finite change to g is still generic. But the claim that a real maps to x is absolute so g differs in at most finitely many places from a real in V so it could not have been generic, hence $\equiv_T \not\leqslant_B \mathtt{Id}(2^\omega)$. The analogous theorem for computable reducibility

Theorem 3.3.6. *[6]* $=^{ce} < \equiv_T^{ce}$.

The Glimm Effros dichotomy theorem implies that $E_0 \leqslant \equiv_T$. It was shown by Slaman and Steele in [17] that $\equiv_T$ is not hyperfinite (those equivalence relations who are the countable, increasing union of equivalence relations with all classes finite), and since the Borel equivalence relations E such that $E \leqslant_B E_0$ are precisely the hyperfinite ones [11] (Theorem 7.2.3) we conclude $E_0 <_B \equiv_T$.

Open Question: Are E_0^{ce} and $\equiv_T^{ce}$ comparable with respect to computable reducibility? To mirror their counterparts in Borel equivalence relations we would like $E_0^{ce} < \equiv_T^{ce}$

3.4 Relative Computable Reducibility

We define a new notion of reducibility with respect to an oracle

Definition 3.4.1. Let E and F be equivalence relations on ω, and let $\mathbf{d}$ be a Turing degree, E is $\mathbf{d}$-computably reducible to E, written $E \leqslant_{\mathbf{d}} F$ if there is a $\mathbf{d}$-computable function f such that for all $x, y \in \omega$

$$xEy \longleftrightarrow f(x)Ff(y).$$

This definition was made in passing in Coskey et al [6] and explored in Fokina et al [7]. For every pair of equivalence relations E and F on ω, where $|E| \leqslant |F|$, there is always a degree $\mathbf{d}$ that witnesses a reduction from E to F. In fact, the degrees that compute the reduction exist on a cone above $\deg(E \oplus F)$.

Definition 3.4.2. Let (E, F) be pair of equivalence relations. The *reducibility spectrum* of (E, F), denoted $\mathcal{S}(E, F)$ is the set of Turing degrees $\mathbf{d}$ such that $E \leqslant_{\mathbf{d}} F$. The *degree of reducibility* of (E, F) is the least degree $\mathbf{d} \in \mathcal{S}(E, F)$ if a least degree exists.

For instance if $E \leqslant F$ then the degree of reducibility of (E, F) is $\mathbf{0}$.

Theorem 3.4.1. *Let (E, F) be a pair of equivalences, then $\mathcal{S}(E, F)$ is either empty or upward closed, and if empty then $|E| > |F|$.*

Proof. Assume $|E| \leqslant |F|$, let $\mathbf{a} = \deg(E \oplus F)$ and define f to be the $\mathbf{a}$-computable function such that $f(0) = 0$ and

$$f(x+1) = \begin{cases} f(y) & (\exists x \leqslant y)(y \in [x+1]_E) \\ (\mu z)[(\forall y \leqslant x)(z \notin [f(y)]_F)] & \text{otherwise.} \end{cases}$$

Now since $|E| \leqslant |F|$, then we will never run out of equivalence classes. This f witnesses $E \leqslant_{\mathbf{a}} F$. To see that the spectrum of reducibility is closed upwards it is enough to note that any $\mathbf{a}$-computable function is also $\mathbf{d}$-computable for any $\mathbf{d} \geqslant \mathbf{a}$. □

It should be noted here that computable reducibility of equivalence relations as defined is not directly analogous to Borel reducibility. In fact computable reducibility would be directly analogous to continuous reducibility (replace Borel with continuous in the definition of Borel reducibility), and Borel reducibility would be directly analogous to hyperarithmetic reducibility. But if we were to take the definition of computable reducibility and replace the existence of a computable function with that of a hyperarithmetic function then many equivalence relations on ω would be trivial. For instance, take E and F to be any two arithmetic equivalence relations with $|E| \leqslant |F|$, then since $\deg(E \oplus F)$ is computable in $\mathbf{0}^{(\omega)}$ then there is a $\mathbf{0}^{(\omega)}$-computable function that witnesses $E \leqslant_{\mathbf{0}^{(\omega)}} F$.

It is worth noting that although $|E| \leqslant |F|$ implies there is a non-empty reducibility spectrum for (E, F), the authors of [7] haave shown that there is not always a degree of reducibility. One question that arises is to see if degrees of reducibility exist for nice equivalence relations.

Definition 3.4.3. A subset $A \subset \omega$ is called partial transversal for E if $x, y \in A$, then $\neg(xEy)$ or $x = y$.

Theorem 3.4.2. $\mathbf{d} \in \mathcal{S}(\mathtt{Id}, E)$ *if and only if* $\mathbf{d}$ *computes an infinite partial transversal for* E*.*

Proof. Suppose that $\mathbf{d} \in \mathcal{S}(\mathtt{Id}, E)$, if f is the $\mathbf{d}$-computable function that witnessed $\mathtt{Id} \leqslant E$ then $\mathrm{ran} f$ is an infinite partial transversal for E. For the converse suppose that $\mathbf{d}$ computes an infinite partial transversal for E, call it A. Use the oracle $\mathbf{d}$ to order $A = \{a_0, a_1, \dots\}$. Define the $\mathbf{d}$-computable function f so that $f(n) = a_n$, thus $\mathbf{d} \in \mathcal{S}(\mathtt{Id}, E)$. □

We have shown that $\mathtt{Id} \nleqslant E_A$, where A is a simple set. Let $\mathbf{a} = \deg(\mathbf{A})$, then $\mathtt{Id} \leqslant_{\mathbf{a}} E_A$. First $\mathbf{a}$-computably enumerate $\overline{A}$ then the map which sends i to a_i, witnesses the reduction. The set $\overline{A}$ is an infinite partial transversal for E_A since if

$a_i, a_j \in \overline{A}$ then $a_i = a_j$ if and only if $i = j$, and furthermore there are no $a_i \in A$. Define $E \equiv_{\mathbf{d}} F$ if and only if $E \leqslant_{\mathbf{d}} F$ and $F \leqslant_{\mathbf{d}} E$. Continuing the example we can use $\mathbf{a}$ to decide, for some n, whether $n \in A$ or $n \in \overline{A}$. Thus we can define an $\mathbf{a}$-computable reduction between E_A and $\mathtt{Id}$ as follows

$$f(n) = \begin{cases} 0 & \text{if } n \in A \\ (\mu k \geqslant 1)(\forall m < n)(k > f(m)) & \text{otherwise.} \end{cases}$$

Thus we have that $\mathtt{Id} \equiv_{\mathbf{a}} E_A$.

CHAPTER 4

A JUMP OPERATOR ON EQUIVALENCE RELATIONS

In this chapter we will define a jump operation for equivalence relations on ω analogous to a well-studied jump operator on Borel equivalence relations.

4.1 Friedman-Stanley Jump and its Computable Variant

Definition 4.1.1. Let E be a Borel equivalence relation on a Polish space X. The Friedman-Stanley jump[1] of E, denoted E^+ is the equivalence relation on X^ω defined by

$$(x_n)E^+(y_n) \longleftrightarrow \{[x_n]_E \mid n \in \omega\} = \{[y_n]_E \mid n \in \omega\}.$$

The equivalence relation E^+ is Borel since the jump adds only an $\forall\exists$ quantifier to E as shown

$$(x_n)E^+(y_n)\longleftrightarrow(\forall n)(\exists m)(x_n E y_m) \wedge (\forall m)(\exists n) x_n E y_m.$$

It is immediate that $E \leqslant_B E^+$ since the map that sends x to the constant sequence of x is a reduction. Also we have that if $E \leqslant_B F$ then $E^+ \leqslant_B F^+$. In fact, with quite a bit of work one may show that $E <_B E^+$, which goes through the famous theorem about the non-existence of a Borel diagonalizer.

[1] Kanovei in [14] calls this the countable power operation.

Theorem 4.1.1. *[11] If E is a Borel equivalence relation, then there is no Borel function $f : X^\omega \to X$ with the two following properties: for all $x, y \in X^\omega$*

(i) $xE^+y \to f(x)Ef(y)$

(ii) $(f(x), x_n) \notin E$ *for all* $n \in \omega$.

The are at least two proofs of this, one of which is in [11] which goes through the non-existence of a Borel separator set for certain projective sets. Another, which uses forcing, given by Friedman in [10][2] Proposition C, is a wonderful example of using forcing with the end goal not being an independence result. The gist of the forcing proof is to assume such a function exists with the above properties, extend your given model by adding a generic object and conclude with absoluteness that the generic object must have existed in the ground model. The main contradiction of the proof of $E <_B E^+$ is that a reduction of $E^+ \leqslant_B E$ implies the existence of a Borel diagonalizer for E^{++}.

We define a computable Friedman-Stanley (FS) type jump for equivalence relations on ω. We then see how much this mirrors the structure of Borel equivalence relations with the FS-jump.

Definition 4.1.2. Let E be an equivalence relation on ω. Define the *computable FS-jump of E*, denoted E^+, on ω (thought of as indices for c.e. subsets of ω^2), by

$$eE^+e' \longleftrightarrow \{[\varphi_e(n)]_E \mid n \in W_e\} = \{[\varphi_{e'}(n)]_E \mid n \in W_{e'}\}.$$

[2] This is originally attributed to Leo Harrington.

Formally one may see this as the following

$$\begin{aligned} eE^{+}e' \longleftrightarrow & (\forall n \in W_e)(\exists m \in W_{e'})\varphi_e(n)E\varphi_{e'}(m) \\ & \wedge (\forall m \in W_{e'})(\exists n \in W_e)\varphi_e(n)E\varphi_{e'}(m). \end{aligned}$$

This might be viewed as a more complex variant of the halting jump operator defined for ceers by Gao and Gerdes in [12], which we looked at in section 3.2.

Definition 4.1.3. Let E be an equivalence relation on ω then:

(i) $E^{0+} = E$

(ii) $E^{(n+1)+} = (E^{n+})^{+}$.

Theorem 4.1.2. *Let E be an equivalence relation on ω, then $E \leqslant E^{+}$.*

Proof. Let f send each e to the code for a machine that halts on all inputs and outputs e. Then

$$\begin{aligned} eEe' \longleftrightarrow & (\forall n \in W_{f(e)})(\exists m \in W_{f(e')})eEe' \\ & \wedge (\forall m \in W_{f(e')})(\exists n \in W_{f(e)})eEe' \\ \longleftrightarrow & (\forall n \in W_{f(e)})(\exists m \in W_{f(e')})\varphi_e(n)E\varphi_{e'}(m) \\ & \wedge (\forall m \in W_{f(e')})(\exists n \in W_{f(e)})\varphi_e(n)E\varphi_{e'}(m) \\ \longleftrightarrow & f(e)E^{+}f(e'). \end{aligned}$$

□

Theorem 4.1.3. *If $E \leqslant F$ then $E^{+} \leqslant F^{+}$.*

Proof. Suppose f witnesses $E \leqslant F$. By the s-m-n theorem find computable injective function g such that for all $e, n \in \omega$, $\varphi_{g(e)}(n) = f(\varphi_e(n))$. Then $\varphi_{g(e)}(n)$ halts if and only if $f(\varphi_e(n))$ does, and outputs the same value.

$$\begin{aligned}
eE^+e' \longleftrightarrow & ((\forall n \in W_e)(\exists m \in W_{e'})\varphi_e(n)E\varphi_{e'}(m) \\
& \wedge (\forall m \in W_{e'})(\exists n \in W_e)\varphi_e(n)E\varphi_{e'}(m)) \\
\longleftrightarrow & ((\forall n \in W_e)(\exists m \in W_{e'})f(\varphi_e(n))Ff(\varphi_{e'}(m)) \\
& \wedge (\forall m \in W_{e'})(\exists n \in W_e)f(\varphi_e(n))Ff(\varphi_{e'}(m))) \\
\longleftrightarrow & ((\forall n \in W_{g(e)})(\exists m \in W_{g(e')})\varphi_{g(e)}(n)F\varphi_{g(e')}(m) \\
& \wedge (\forall m \in W_{g(e')})(\exists n \in W_{g(e)})\varphi_{g(e)}(n)F\varphi_{g(e')}(m)) \\
\longleftrightarrow & g(e)F^+g(e').
\end{aligned}$$

Thus g witnesses $E^+ \leqslant F^+$. □

Corollary 4.1.1. *For all n, $E^{n+} \leqslant E^{(n+1)+}$.*

Proof. Iterating Theorem 4.1.3. □

Corollary 4.1.2. *If $E \equiv F$ then $E^+ \equiv F^+$.*

Theorem 4.1.4. $E^+ \oplus F^+ \leqslant (E \oplus F)^+$.

Proof. By the s-m-n theorem find two injective functions f and g so that for all e and n, $\varphi_{f(e)}(n) = 2\varphi_e(n)$ and $\varphi_{g(e)}(n) = 2\varphi_e(n) + 1$ and using padding we can suppose f, g have disjoint ranges. We will show that the function h defined $h(2e) = 2e$ and $h(2e+1) = g(e)$ gives the reduction. For two even numbers $2e, 2e'$

$$
\begin{aligned}
2eE^{+}{\oplus}F^{+}2e' &\longleftrightarrow eE^{+}e' \\
&\longleftrightarrow (\forall n \in W_e)(\exists m \in W_{e'})(\varphi_e(n)E\varphi_{e'}(m)) \\
&\qquad \wedge (\forall m \in W_{e'})(\exists n \in W_e)(\varphi_e(n)E\varphi_{e'}(m)) \\
&\longleftrightarrow (\forall n \in W_e)(\exists m \in W_{e'})(2\varphi_e(n)E{\oplus}F2\varphi_{e'}(m)) \\
&\qquad \wedge (\forall m \in W_{e'})(\exists n \in W_e)(2\varphi_e(n)E{\oplus}F2\varphi_{e'}(m)) \\
&\longleftrightarrow (\forall n \in W_{f(e)})(\exists m \in W_{f(e')})(\varphi_{f(e)}(n)E{\oplus}F\varphi_{f(e')}(m)) \\
&\qquad \wedge (\forall m \in W_{f(e')})(\exists n \in W_{f(e)})(\varphi_{f(e)}(n)E{\oplus}F\varphi_{f(e')}(m)) \\
&\longleftrightarrow f(e)(E{\oplus}F)^{+}f(e') \\
&\longleftrightarrow h(2e)(E{\oplus}F)^{+}h(2e').
\end{aligned}
$$

By a similar argument one can show

$$2e + 1E^{+}{\oplus}F^{+}2e' + 1 \longleftrightarrow h(2e + 1)(E{\oplus}F)^{+}h(2e' + 1).$$

For $2e$ and $2e' + 1$, these are never $E^{+}{\oplus}F^{+}$ equivalent. Assume that

$$h(2e)(E{\oplus}F)^{+}h(2e' + 1),$$

under the assumptions f, g have disjoint ranges $f(2e) \neq g(2e' + 1)$, this happens because of parity. □

4.2 Benchmark Equivalence Relations

We will examine the computable FS-jump applied to some benchmark equivalence relations on ω. We will investigate the effect of the computable FS-jump has on descriptive complexity. Consider $\mathtt{Id}$, let us calculate $\mathtt{Id}^+$.

$$\begin{aligned} e\mathtt{Id}^+e' \longleftrightarrow & (\forall n \in W_e)(\exists m \in W_{e'})\varphi_e(n)\mathtt{Id}\varphi_{e'}(m) \\ & \wedge (\forall m \in W_{e'})(\exists n \in W_e)\varphi_e(n)\mathtt{Id}\varphi_{e'}(m)) \\ \longleftrightarrow & \{[\varphi_e(n)]_{\mathtt{Id}} \mid n \in W_e\} = \{[\varphi_{e'}(n)]_{\mathtt{Id}} \mid n \in W_{e'}\} \\ \longleftrightarrow & \{\varphi_e(n) \mid n \in W_e\} = \{\varphi_{e'}(m) \mid m \in W_{e'}\} \\ \longleftrightarrow & \mathrm{ran}\varphi_e = \mathrm{ran}\varphi_{e'}. \end{aligned}$$

Thus $e\mathtt{Id}^+e' \longleftrightarrow \mathrm{ran}\varphi_e = \mathrm{ran}\varphi_{e'}$.

Theorem 4.2.1. $=^{ce}$ *is computably bi-reducible with equality of ranges of partial computable functions.*

Proof. For the $\leqslant$ direction, given e let

$$\varphi_{f(e)}(n) = \begin{cases} n & \varphi_e(n)\downarrow, \\ \uparrow & \text{otherwise.} \end{cases}$$

If $e =^{ce} e'$ then φ_e and $\varphi_{e'}$ halt on precisely the same inputs, which implies $\mathrm{ran}\varphi_{f(e)} = \mathrm{ran}\varphi_{f(e')}$. For other direction suppose that $e \neq^{ce} e'$,then there is an m such that $\varphi_e(m)\downarrow$ and $\varphi_{e'}(m)\uparrow$. Which means that there is an $m \in \mathrm{ran}\varphi_{f(e)}$ such that $m \notin \mathrm{ran}\varphi_{f(e')}$, therefore $\mathrm{ran}\varphi_{f(e)} \neq \mathrm{ran}\varphi_{f(e')}$

For the $\geqslant$ direction, given e let

$$\varphi_{f(e)}(n) = \begin{cases} n & (\exists s)(\exists x)\varphi_{e,s}(x)\downarrow= n, \\ \uparrow & \text{otherwise.} \end{cases}$$

If $\text{ran}\varphi_e = \text{ran}\varphi_{e'}$ then $f(e) =^{ce} f(e')$. Conversely, if $\text{ran}\varphi_e \neq \text{ran}\varphi_{e'}$ then there is an m, a stage s, and an x such that $\varphi_{e,s}(x)\downarrow= m$ and for all s such that $\varphi_{e',s}(x)$ halts it never outputs m, therefore $f(e) \neq^{ce} f(e')$. □

For the analogy of this theorem to the FS-jump see ([14] Lemma 5.1.3(i)). The previous theorem gives us a bit of room to be loose with having to consider when we care about the range or domain of a function.

Corollary 4.2.1. $\mathtt{Id}^+ \equiv =^{ce}$.

Theorem 4.2.2. $\mathtt{Id} < \mathtt{Id}^+$.

Proof. By Theorem 4.1.2 we have $\mathtt{Id} \leqslant \mathtt{Id}^+$. We cannot have reduction the other way since by Theorem 3.3.1 $\mathtt{Id}^+$ has a Π^0_2-complete equivalence class while $\mathtt{Id}$ only has computable equivalence classes. □

Now let us compute $(=^{ce})^+$, which is

$$\{[\varphi_e(n)]_{=^{ce}} \mid n \in W_e\} = \{[\varphi_{e'}(n)]_{=^{ce}} \mid n \in W_{e'}\}.$$

First some notation, for a set $A \subset \omega$ define the n^{th} column of A, denoted $A^{[n]}$ to be the set $\{m \mid \langle n, m\rangle \in A\}$.

Definition 4.2.1. The equality on columns equivalence relation is given as follows

$$eE^{ce}_{\mathtt{set}}e' \longleftrightarrow \{W^{[n]}_e \mid n \in \omega\} = \{W^{[n]}_{e'} \mid n \in \omega\}.$$

Informally we see $eE^{ce}_{\texttt{set}}e'$ if and only if W_e and $W_{e'}$ thought of as subsets of ω^2 have the same set of columns.

Theorem 4.2.3. $(=^{ce})^+$ *is computably bireducible with* $E^{ce}_{\texttt{set}}$*.*

Proof. We first show the $\geqslant$ direction. Given e we define a computable function f on pairs $(n,m) \in W_e$ such that

$$f((n,m)) = \begin{cases} 1 & \varphi_{\varphi_e(n)}(m)\downarrow \\ \uparrow & \text{otherwise.} \end{cases}$$

Intuitively we want that the n^{th} column to be $W_{\varphi_e(n)}$. Suppose that $eE^{ce}_{\texttt{set}}e'$ we wish to show

$$\{[\varphi_{f(e)}(n)]_{=^{ce}} \mid n \in W_{f(e)}\} = \{[\varphi_{f(e')}(n)]_{=^{ce}} \mid n \in W_{f(e')}\}.$$

Make the observation that $[\varphi_{f(e)}(n)]_{=^{ce}}$ is the n^{th} column of W_e. Now $eE^{ce}_{\texttt{set}}e'$ have precisely the same set of columns thus equality of the sets of $=^{ce}$-classes holds. If e was not $E^{ce}_{\texttt{set}}$-equivalent to e' then there would be a column in $W_{f(e)}$ or $W_{f(e')}$ that the other one didn't have, thus equality of the $=^{ce}$ classes would not hold.

For the $\leqslant$ direction the same computable function witnesses the reduction. □

Corollary 4.2.2. $\texttt{Id}^{2+}$ *is computably bireducible with* E^{ce}_{set}*.*

In Borel Equivalence relations, the equivalence relation $E_{\texttt{set}}$ is well studied. It is also called F_2 and is Borel bireducible with $=^+$, or the FS-jump of the equality on 2^ω. It is worth noting that $=^{ce}$ is equality on 2^ω restricted to c.e. reals. We have the second computable FS-jump of equality is computably bireducible $E^{ce}_{\texttt{set}}$, since we start the iterated computable FS-jump at $\mathtt{Id}$. The similarities between the FS-jump and the computable FS-jump are still quite apparent.

Theorem 4.2.4. *The set $(=^{ce})^+$ is a Π^0_4-complete set of pairs.*

Proof. There is a computable function f such that $\{W_{f(n)} \mid n \in \omega\}$ consists of exactly the recursive sets ([18], Exercise II.2.11). Recall that the set of indices for computable functions, $\mathtt{Rec}$, is a Σ^0_3-complete set ([18], Corollary IV.3.6). Let A be Π^0_4, we show a reduction to the $(=^{ce})^+$-class of f. For some Σ^0_3 relation R and h a computable function such that

$$\begin{aligned} x \in A &\longleftrightarrow (\forall y) R(x,y) \\ &\longleftrightarrow (\forall y)(h(x,y) \in \mathtt{Rec}). \end{aligned}$$

Now define the partial computable function

$$\varphi_{e(x)}(y) = \begin{cases} f(n) & y = 2n \\ h(x,n) & y = 2n+1. \end{cases}$$

This function witnesses the reduction from A to the $(=^{ce})^+$-class of f. □

Corollary 4.2.3. *E^{ce}_{set} is a Π^0_4-complete set of pairs.*

Corollary 4.2.4. $\mathtt{Id}^+ < \mathtt{Id}^{2+}$.

Theorem 4.2.5. *For a ceer E, E^+ computably reducible to $=^{ce}$. Moreover, if E is light, then $=^{ce} \equiv E^+$.*

Proof. Given e we let f compute the E-saturation of the range of φ_e, i.e. the set $\{x \mid (\exists n) x E \varphi_e(n)\}$. Thus we have

$$\begin{aligned} eE^+e &\longleftrightarrow W_{f(e)} = W_{f(e')} \\ &\longleftrightarrow f(e) =^{ce} f(e'). \end{aligned}$$

If E is light then $\mathtt{Id} \leqslant E$ and by the monotinicity of the computable FS-jump then $\mathtt{Id}^+ \leqslant E^+$. Thus $=^{ce} \leqslant E^+$. □

We will now begin to look at partial results approaching $=^{ce} \leqslant E^+$ for any ceer E. We know that the prototypical example of a dark ceer is the case when E_A for a A a simple set. There are many kinds of immunity[3] properties that subsets of natural numbers can have, for a diagram see ([18], page 211).

Definition 4.2.2. (i) For a finite set $A = \{x_1, \ldots, x_k\}$ where $x_1 < \cdots < x_k$, the canonical index for A is $y = \sum_i 2^{x_i}$. Let D_y denote the finite set with canonical index y.

(ii) A sequence $\{F_n\}$ for $n \in \omega$ of finite sets is a strong (weak) array if there is a computable function f such that $F_n = D_{f(n)}$ ($F_n = W_{f(n)}$), and an array is disjoint if its members are pairwise disjoint.

(iii) A c.e. set A is (hyperhypersimiple) hypersimple if $\overline{A}$ is infinite and there is no disjoint strong (weak) array $\{F_n\}$ for $n \in \omega$ such that $F_n \cap \overline{A} \neq \varnothing$ for all n.

Theorem 4.2.6. *Let A be a non-hyperhypersimple set, then $=^{ce} \equiv E_A^+$.*

[3] Cooper [5] characterizes immunity of set as a sense of difficulty in computably finding infinitely members of a set.

Proof. Let A be a non-hyperhypersimple set, then there is a disjoint strong array $\{B_n\}$ for $n \in \omega$ such that $B_n \cap \overline{A} \neq \varnothing$ for all n. Given e let f enumerate the set $\{B_n \mid n \in W_e\}$. Then since the B_n are disjoint we have $\neg W_{f(e)} E^+_A W_{f(e')}$ if and only if $e \neq^{ce} e'$. □

Theorem 4.2.7. *If E is any ceer with infinitely many classes, then* $\mathtt{Id} \leqslant E^+$.

Proof: The key idea to the proof is to use the fact that E is a ceer to give enough computable information to build a transversal for E^+. Recall by Theorem 3.4.2 that $\mathtt{Id} \leqslant E^+$ if and only if there is a computable enumeration of an infinite transversal of E^+. We'll then use the transversal to find computable function f so that $f(i)$ is a Σ_1-index for a finite set with some element that not E-equivalent with any finite set already defined. To that end, we will use a moving marker construction to meet the following requirement;

$$\mathcal{R}_i : (\forall j < i)(\exists n \in W_{f(i)})(\forall m \in W_{f(j)})(\neg m E n).$$

We do this by enumerating the complement of an infinite partial transversal $B = \bigcup_s B_s$ for E^+. We then let f computably find an index for a finite set $W_{f(i)}$ so that is an element that is not E-equivalent to any $k \in W_{f(j)}$ for $j < i$. We now verify that if each requirement $\mathcal{R}_i$ is met the function f will witness $\mathtt{Id} \leqslant E^+$. If $i = j$, then $W_{f(i)} = W_{f(j)}$ and so $f(i) E^+ f(j)$. If $j < i$, then there will be an $n \in W_{f(i)}$ that is not E-equivalent to any $m \in W_{f(j)}$, and hence $f(i) \not{E}^+ f(j)$. Start with a computable enumeration $\{E_s\}$ for $s \in \omega$ of E, we can safely assume that (i_s, j_s) and (j_s, i_s) are enumerated into E at stage s. Let $\{x^s_i\}$ be a collection of movable markers, that denote the position of the i^{th} element of $\overline{B}_s$ at stage s, i.e. $\overline{B}_s = \{x^s_0 < x^s_1 < \cdots\}$.

Construction: At stage $s = 0$ set $B_0 = \varnothing$. At stage $s+1$ assume we have constructed B_s, we consider marker x_i^s and computably check if $(x_i^s, k) \in E_s$ for any $k \in B_s$. If the answer is yes we enumerate it into B_s and set $B_{s+1} = B_s \cup \{x_i^s\}$ and move the markers. Otherwise we computably find an index for $\{x_i^s \mid s \in \omega\} = W_{f(i)}$, and move the markers.

Lemma 4.2.1. *Each $\mathcal{R}_i$, $i \in \omega$ is satisfied.*

Proof. Since we have infinitely many classes for every B_s there will be a stage $t > s$ so that we see an x such that $x \not{E}\ k$ for any $k \in B_s$. □

Lemma 4.2.2. *For each $i \in \omega$, we have $x_i = \lim_s x_i^s$ is finite.*

Proof. Since we have infinitely many classes we eventually see an element that is no E-equivalent to anything that we have seen before. If we enumerate x_i^s at stage s we never move the marker again. Otherwise the marker x_i^s can only be moved for the sake of some x_k^s for $k < i$. This happens at most i many times, so each marker comes to a rest. □

Definition 4.2.3. Let $A \subset \omega$ be infinite, if $A = \{a_0 < a_1 < a_2 < \cdots\}$, define p_A as the function that sends n to a_n. A function f majorizes g if for all x, $f(x) \geqslant g(x)$. A function f majorizes A if it majorizes p_A.

To illustrate an example note that the Post simple set A is not hyperhypersimple, since $\overline{A}$ is majorized by $f(x) = 2x$ ([18], Theorem V.2.3). We can explicitly write down a reduction from $\texttt{Id}$ to E_A^+ as follows. Given e let $W_{f(e)} = [0, 4^e]$ then

$$2^{2e-1} \leqslant |W_{f(e)} \setminus A| \leqslant 2^{2e}.$$

Then $\neg W_{f(e)} E_A^+ W_{f(e')}$ if and only if $e \neq e'$. Thus f witnesses $\texttt{Id} \leqslant E_A^+$.

We will now construct a coinfinite set $\overline{A}$ such that $\texttt{Id} \nleqslant E_A^+$. Note that this construction is necessarily non-effective since we have that $\texttt{Id}$ computably reduces to the jump of any ceer. The construction will essentially be by computable Mathias forcing. Consider the poset of pairs of (s, A), where s is finite and A is infinite. Say that $(s, A) \leqslant (t, B)$ if $t \subset s$, $A \subset B$, and $s \setminus t \subset B$. We might think of the set A as a reservoir from with to extend s.

Lemma 4.2.3. *If $A, B, C \subset \omega$ then at least one of $A \triangle B$, $A \triangle C$, or $B \triangle C$ must be coinfinite.*

Proof. Assume for contradiction that the complements the above symmetric differences are finite. If A, B and C are all finite then the result holds. Without loss of generality consider C to be infinite, then C without $\overline{A \triangle C} \cup \overline{B \triangle C}$ is infinite, but that set is a subset of $\overline{A \triangle B}$, which we assumed to be finite. This contradiction implies that one of the symmetric differences must be coinfinite. □

Theorem 4.2.8. *There exists a set A so that $\texttt{Id} \nleqslant E_A^+$.*

Proof. Consider the poset of conditions described above. We will construct a set A so that $A = \bigcup s$ for all (s, B) that meet the following dense sets (requirements).

$$D_n^0 : \{(s, B) \mid |s| \geqslant n\}$$

$$D_e^1 : \{(s, B) \mid W_e \text{ infinite } \to W_e \cap \overline{B} \neq \varnothing\}$$

$$D_f^2 : \{(s, B) \mid (\exists i \neq j) \operatorname{ran} \varphi_{f(i)} \triangle \operatorname{ran} \varphi_{f(j)} \cap B = \varnothing\}.$$

Notice that if requirements D^0_n and D^1_e for all n and for all e, then A is immune. Then there would be no way to computably enumerate an infinite c.e. subset of A, which is what a reduction of the identity to E^+_A would do. Note that for eE^+_Ae' there are two possibilities, either both $\mathrm{ran}\varphi_e$ and $\mathrm{ran}\varphi_{e'}$ meet A or both miss A and $\mathrm{ran}\varphi_e \triangle \mathrm{ran}\varphi_{e'} \subset A$. If the condition D^2_f is met for each computable f, though we can safely assume that f is injective, then either $\mathrm{ran}\varphi_e$ and $\mathrm{ran}\varphi_{e'}$ both meet or both miss A. So we can make sure that, for every computable function, there are two different e's that both meet A and therefore $\mathtt{Id} \leqslant E_A+$.

Fix an ordering for how to meet the requirements

$$D^0_0 > D^1_0 > D^0_0 > D^0_1 > D^1_1 > D^2_1 > \cdots .$$

To meet the D^0_n requirements, if we are given a (s, B) we can extend to some (s', B') by enumerating n elements of B, then $(s', B') \leqslant (s, B)$. To meet the requirements D^1_e, ask if W_e is infinite, and if it is, find an $n \in W_e$ such that $n > \max(s)$ and enumerate n into s. Call that s' and $B' = B \setminus \{n\}$, then $(s', B') \leqslant (s, B)$. If W_e is finite do nothing. To meet D^2_f, given (s, B), see that there are infinitely many i and j such that

$$(\mathrm{ran}\varphi_{f(i)} \triangle \mathrm{ran}\varphi_{f(j)}) \cap [0, \max(t)] = \varnothing.$$

Fix three such i, j, k. Then by the above lemma at least one the symmetric differences of the ranges of $f(i), f(j)$ and $f(k)$ must be coinfinite. Say $\mathrm{ran}\varphi_{f(i)} \triangle \mathrm{ran}\varphi_{f(j)}$ is coinfinite, then define $B' = B \setminus \mathrm{ran}\varphi_{f(i)} \triangle \mathrm{ran}\varphi_{f(j)}$, and $s = s'$. So $(s', B') \leqslant (s, B)$.

Finally take $A = \bigcup s$ over all (s, B) that meet the above conditions in the given order. And so no computable function f can witness $\texttt{Id} \leqslant E^+_A$. □

The previous theorem gives us that there is an equivalence relation E so that $=^{ce}$ is not computably reducible the computable FS-jump of E^+. The previous construction was not effective so this leads to the following question.

Open Question: What is the descriptive complexity of the set A in the previous proof? The constructive seems to be computable in $\varnothing^{(3)}$, is this optimal?

Conjecture 4.2.1. *If E is a ceer then $E^+ \equiv =^{ce}$.*

Whereas there are equivalence relations E so that $\texttt{Id} \not\leqslant E^+$ the next theorem shows that this is the worst failure.

Theorem 4.2.9. *For any E with infinitely many classes, $\texttt{Id} \leqslant E^{2+}$.*

Proof. Given n, let $f(n)$ be the machine that enumerates all sets with n elements. Since f is a function we clearly have

$$n\texttt{Id}n' \to f(n)E^{2+}f(n').$$

For the converse, we argue the contrapositive, suppose $n \neq n'$, since E has infinitely many classes we are guaranteed to have a set with n' many E-inequivalent elements and thus

$$\{[\varphi_{f(n)}(i)]_{E^+} \mid i \in W_{f(n)}\} \neq \{[\varphi_{f(n')}(i)]_{E^+} \mid i \in W_{f(n')}\}.$$

Thus $\texttt{Id} \leqslant E^{2+}$. □

Corollary 4.2.5. *For any E with infinitely many classes, $\texttt{Id}^{n+} \leqslant E^{(n+2)+}$.*

4.3 Iterating Computable FS-jump through Computable Ordinals

Definition 4.3.1. Iterating the FS-jump through the computable ordinals we make use of Kleene's $\mathcal{O}$. Let E be an equivalence relation on ω, for each $a \in \mathcal{O}$, define E^{a+} by recursion as follows

(i) If $a = 1$, then $E^{a+} = E$.

(ii) If $a = 2^b$, then $E^{a+} = (E^{b+})^+$.

(iii) If $a = 3 \cdot 5^e$, then $E^{a+} = \left(\bigoplus_{n<\omega} E^{\varphi_e(n)+}\right)^+$.

By the above definition this implies $\mathtt{Id} \leqslant E^{a+}$ for any $|a|$ an infinite ordinal. We are particularly interested in the case when $E = \mathtt{Id}$ in the previous definition.

Lemma 4.3.1. *For every $a \in \mathcal{O}$, $E \leqslant E^{a+}$. If $a, b \in \mathcal{O}$ and $a <_{\mathcal{O}} b$ then $E^{a+} \leqslant E^{b+}$.*

Conjecture 4.3.1. *Every hyperarithmetic equivalence relation E on ω, is computably reducible with $\mathtt{Id}^{a+}$ for some computable ordinal α.*

It is not true that every hyperarithmetic equivalence relation is computably bireducible with $\mathtt{Id}^{a+}$ for some a. Let A be a simple set, then E_A is computably incomparable with $\mathtt{Id}$. Also $\mathtt{Id}^+$ cannot computably reduce to E_A because E_A has only c.e. classes and $\mathtt{Id}^+$ has a Π^0_2-complete equivalence class. The iterates in this sense would weakly parameterize the computable reducible hierarchy for hyperarithmetic equivalence relations.

Andrews and Sorbi in [3] have shown that for transfinite jumps of the halting jump operator that there are notations $|a| = |b| = \omega^2$ such that $\mathtt{Id}^a \mid \mathtt{Id}^b$.

A proof of the following theorem will appear in a furture article.

Theorem 4.3.1. *For hyperarithmetic equivalence relations E, $E < E^+$.*

Conjecture 4.3.2. $\mathtt{Id}^{n+}$ *is a Π^0_{2n}-complete set of pairs.*

4.4 Computable FS-jump fixed points

In this section we will show there are Σ^1_1 fixed points of the computable FS-jump. Given a hyperarithmetic equivalence relation on ω it is clear that computable FS-jump preserves the hyperarithmeticity of the equivalence relation, since we are essentially adding two natural number quantifiers.

Definition 4.4.1. A *fixed point* of the computable FS-jump is an equivalence relation E such that $E \equiv E^+$.

In Borel equivalence relation theory the FS-jump has as a fixed point the isomorphism relation on countable graphs, which is Borel complete and $\boldsymbol{\Sigma^1_1}$. This gives more evidence that the computable FS-jump as defined is analogous to the FS-jump in Borel Theory.

For instance, Fokina et al.[9] it was shown that the isomorphism relation on computable trees was Σ^1_1-complete[4] for computable reducibility, we will give a proof of this fact in section 5.1. There are also constructions of a universal Σ^1_1 equivalence relation such as in Fokina [8] akin to the construction of a universal ceer which is given in Coskey et al. [6]. That proof relativizes to arbitrary oracles, if we take z to

[4] This is sometimes called being "on-top for computable reducibility" see Montalbán [15] pg. 15 – 16.

be a Σ^1_1-complete real (identifying subsets of ω with reals) then U^z_{ce} is a universal Σ^1_1 equivalence relation.

Theorem 4.4.1. *Given a Σ^1_1-complete real z, then $(U^z_{ce})^+ \leqslant U^z_{ce}$.*

Proof. Adding natural number quantifiers to a Σ^1_1 relation creates a new Σ^1_1 relation ([13], Lemma 25.2). We are defining a new equivalence relation so we get an Σ^1_1 equivalence relation. Thus $(U^z_{ce})^+$ is a Σ^1_1 equivalence relation. Since U^z_{ce} is Σ^1_1-complete for computable reducibility, then $(U^z_{ce})^+ \leqslant U^z_{ce}$. □

This brings up an important open question in the theory of computable reducibility. In Borel theory the notion of being Borel complete and $\boldsymbol{\Sigma^1_1}$-complete for Borel reducibility are not the same. So far for computable reducibility no such notion has been found to mirror the what it means for a Borel equivlance relation to be Borel complete.

CHAPTER 5

COMPUTABLE TREES OF COMPUTABLE ORDINAL RANK

5.1 Isomorphism Relation on Computable Trees

Computable reducibility allows the analysis of equivalence relations on c.e. structures. Recall that a structure $\mathcal{A}$ is c.e. if there is an index e that enumerates the atomic diagram of $\mathcal{A}$.

Definition 5.1.1. Let $\cong^{ce}_{\mathtt{bin}}$ denote the isomorphism relation on codes for c.e. binary relations. That is, let $e \cong^{ce}_{\mathtt{bin}} e'$ if and only if W_e and $W_{e'}$ thought of as binary relations[1] on ω are isomorphic.

Isomorphism relations on computable structures that are coded by natural numbers are Σ^1_1, since the isomorphism relation can be thought of as being defined a statement of the form "(there exists a function)(something arithmetical)", where the function is appropriately coded by a real. We can use computable reducibility to analyze the complexity of the isomorphism relation on computable structures. As an example of this, we present a proof from [9] that shows that the isomorphism relation on computable trees $\cong_{\mathcal{T}}$ is Σ^1_1-complete for computable reducibility.

[1] If $n \in W_e$ then use the pairing function to think of n as some pair (k, m) where $n = \langle k, m \rangle$.

Definition 5.1.2. Let $\mathcal{T}$ be a subtree of $\omega^{<\omega}$. Define the *tree rank* of $\tau \in \mathcal{T}$ denoted $\mathrm{tr}(\tau)$ by induction

(i) $\mathrm{tr}(\tau) = 0$ if τ has no successor.

(ii) For $\alpha > 0$, $\mathrm{tr}(\tau) = \alpha$ if α is the least ordinal greater than $\mathrm{tr}(\sigma)$ for all immediate successors σ of τ.

(iii) $\mathrm{tr}(\tau) = \infty$ if τ does not have ordinal tree rank.

The *tree rank* of $\mathcal{T}$ is defined to be $\mathrm{tr}(\varnothing)$.

Definition 5.1.3. Let $\mathcal{S}, \mathcal{T} \subset \omega^{<\omega}$ be trees. Define a tree $\mathcal{S} * \mathcal{T}$ in the following way. We think of elements $(\sigma, \tau) \in \mathcal{S} * \mathcal{T}$ as ordered pairs. At level 0 of $\mathcal{S} * \mathcal{T}$, we have $(\varnothing, \varnothing)$. For an element (σ, τ) at level k of $\mathcal{S} * \mathcal{T}$, σ and τ are at level k of $\mathcal{S}$ and $\mathcal{T}$, respectively. The successors (σ', τ') of (σ, τ) are defined in the caseσ' is the successor of $\sigma \in \mathcal{S}$ and τ' is a successor of $\tau \in \mathcal{T}$.

Definition 5.1.4. A computable tree $\mathcal{T} \subset \omega^{<\omega}$ is rank saturated provided that for all $\tau \in \mathcal{T}$

(i) If $\mathrm{tr}(\tau)$ is an ordinal α then for all $\beta < \alpha$, τ has infinitely many successors τ' such that $\mathrm{tr}(\tau') = \beta$.

(ii) If $\mathrm{tr}(\tau) = \infty$, then for all computable β, τ has infinitely many successors τ' such that $\mathrm{tr}(\tau') = \beta$ and τ has infinitely many successors τ' such that $\mathrm{tr}(\tau') = \infty$.

Lemma 5.1.1. *[9] There is a computable rank-saturated tree $\mathcal{T}^{\infty}$ such that tr$(\mathcal{T}^{\infty}) = \infty$.*

Lemma 5.1.2. *[9] If $\mathcal{T}$ is a computable tree, then $\mathcal{T} * \mathcal{T}^{\infty}$ is a computable rank-saturated tree of the same rank as $\mathcal{T}$.*

It is also important to note before the proof that rank saturated trees of infinite or computable ordinal rank are unique up to isomorphism. This is a straight forward proof on by induction on rank.

Theorem 5.1.1. *[9] $\cong_{\mathcal{T}}$ is Σ^1_1-complete for computable reducibility.*

Proof. Let E be a Σ^1_1 equivalence relation on ω. We will show that there is computable sequence of trees $(\mathcal{T}_n)$ for $n \in \omega$ such that for all $n, m \in \omega$

$$mEn \longleftrightarrow \mathcal{T}_m \cong \mathcal{T}_n.$$

Recall that if A is Σ^1_1 subset of ω then there exists a uniformly computable sequence of computable $(\mathcal{T}_n)$ for $n \in \omega$ such that

$$n \in A \longleftrightarrow \mathcal{T}_n \text{ has an infinite path.}$$

Thus since E is Σ^1_1 there exists a uniformly computable sequence of trees $(\mathcal{T}_{m,n})$ for $m, n \in \omega$ such that

$$\neg mEn \longleftrightarrow \mathcal{T}_{m,n} \text{ is well founded.}$$

Say that $\neg mEn$ is *witnessed by stage* $\alpha < \omega_1^{\mathsf{ck}}$ if and only if $\mathcal{T}_{m,n}$ has tree rank less than α.

We want an appropriate class of trees to use for the reduction. The strategy to build $(\mathcal{T}_n)$ for $n \in \omega$ is as follows. Uniformly in m, n we build a computable tree $\mathcal{T}^*_{m,n}$ with the following properties

(i) $\mathcal{T}^*_{m,n} \cong \mathcal{T}^*_{n,m}$

(ii) $mEn \to \mathcal{T}^*_{m,n} \cong \mathcal{T}^\infty$, where $\mathcal{T}^\infty$ is the rank-saturated tree with an infinite path

(iii) $\neg mEn \to \mathcal{T}^*_{m,n} \cong \mathcal{T}^\alpha$, where $\mathcal{T}^\alpha$ is the rank-saturated tree of rank α, for α least such that for all $m' \in [m]_E$ and $n' \in [n]_E$ the relation $\neg m'En'$ is witnessed by stage α.

For every $m, n \in \omega$ uniformly and effectively construct a new tree $\mathcal{T}'_{m,n}$ in the following way. Let σ_i for $i \in \omega$ be an effective enumeration of $\omega^{<\omega}$. Suppose $\sigma_s = (a_0, \dots, a_{l_s})$, then under the s^{th} node on level 1 of $\mathcal{T}'_{m,n}$ we put the tree $P_s = \mathcal{T}_{m,a_0} * \mathcal{T}_{a_0,a_1} * \dots * \mathcal{T}_{a_{l_s},n}$, identifying the top node P_s with s. Then

$$\text{tr}(\mathcal{T}'_{m,n}) = \sup\{\text{tr}(P_s) + 1 \mid s \in \omega\}.$$

We show that the trees $\mathcal{T}'_{m,n}$ still preserve the property of nEm if and only if $\mathcal{T}'_{m,n}$ has an infinite path. If mEn, then $\mathcal{T}_{m,n}$ has infinite path i.e. $\text{tr}(\mathcal{T}_{m,n}) = \infty$. Under P_0, the indexed node of the empty sequence, in $\mathcal{T}'_{m,n}$ we have the tree $\mathcal{T}_{m,n}$, thus $\text{tr}(\mathcal{T}'_{m,n}) = \infty$. If $\neg mEn$, then for every $\sigma = (a_0, \dots, a_l)$, $\text{tr}(\mathcal{T}_{m,a_0} * \mathcal{T}_{a_0,a_1} * \dots * \mathcal{T}_{a_l,n})$ is a computable ordinal. To see this, fix $n, m \in \omega$ such that $\neg mEn$. For every finite sequence σ_s consider the corresponding tree $P_s = \mathcal{T}_{m,a_0} * \mathcal{T}_{a_0,a_1} * \dots * \mathcal{T}_{a_{l_s},n}$. Consider the function F from the set of finite sequences into the set of computable well founded trees such that $F(s)$ is the code of P_s. The function F is hyperarithmetical, its domain is computable. By Spector Bounding ([16]), there is a computable bounded on the range of F. Therefore, $\mathcal{T}'_{m,n}$ has rank α for some computable α. We now show everything we have done is well-defined. To that end, note that for all $m' \in [m]_E$ and $n' \in [n]_E$, we show that we get the same bound α. Let $m'Em$, $n'En$ and let β be the computable bound on the ranks of trees constructed using finite sequences starting with m' and ending with n'. Let $P_s = \mathcal{T}_{m,a_0} * \mathcal{T}_{a_0,a_1} * \dots * \mathcal{T}_{a_{l_s},n}$ be as above. Then

$\mathrm{tr}(\mathcal{T}_{m',m} * P_s * \mathcal{T}_{n,n'}) = \mathrm{tr}(P_s)$, thus $\alpha \leqslant \beta$. Similarly one can show that $\beta \leqslant \alpha$.

Let $\mathcal{T}^*_{m,n} = \mathcal{T}'_{m,n} * \mathcal{T}^\infty$. The tree $\mathcal{T}^*_{m,n}$ is a computable rank-saturated tree, $\mathrm{tr}(\mathcal{T}^*_{m,n}) = \mathrm{tr}(\mathcal{T}'_{m,n})$, and the construction is uniform.

Now we build the desired sequence $(\mathcal{T}_n)$ for $n \in \omega$. Take the tree $\mathcal{T}$ consisting exactly of the sequences $(m, m, \ldots, m)$ of length $i \leqslant m$ for $m \in \omega$. Now fix n and for every m, attach $\mathcal{T}^*_{m,n}$ to the m^{th} leaf of $\mathcal{T}$. The resulting tree is $\mathcal{T}_n$. We claim that the resulting sequence $(\mathcal{T}_n)$ for $n \in \omega$ witnesses the reduction. Suppose that nEm,

(i) For every k in the E-class of n and m,the tree rank of $\mathcal{T}'_{k,m}$ and $\mathcal{T}'_{k,n}$ are infinite. Thus $\mathcal{T}^*_{k,m} \cong \mathcal{T}^*_{k,n} \cong \mathcal{T}^\infty$ and so $\mathcal{T}_m \cong \mathcal{T}_n$.

(ii) for every k not in the E-class of m the tree rank of $\mathcal{T}'_{k,m}$ and $\mathcal{T}'_{k,n}$ are some fixed computable ordinal α. Thus $\mathcal{T}^*_{k,m} \cong \mathcal{T}^*_{k,n} \cong \mathcal{T}^\alpha$, and so $\mathcal{T}_m \cong \mathcal{T}_n$.

Finally, if $\neg mEn$, then $\mathcal{T}^*_{m,m} \cong \mathcal{T}^\infty$, so on the m^{th} leaf of $\mathcal{T}_m$ we have $\mathcal{T}^\infty$. While $\mathcal{T}^*_{m,n} \cong \mathcal{T}^\alpha$, and therefore $\mathcal{T}_m \ncong \mathcal{T}_n$. □

Corollary 5.1.1. $\cong_{\mathcal{T}}$ *is a fixed point for the computable FS-jump.*

5.2 Well-Founded Trees

We continue our investigation of the relationship between the FS-jump and the computable FS-jump by analyzing analogies between a theorem relating the isomorphism relation on countable trees and the FS-jump. First some preliminaries

Definition 5.2.1. Let Tr be the Polish space of all countable trees. Define the set $\texttt{WF} \subset \mathrm{Tr}$ to be the set of all well-founded trees.

The set $\mathtt{WF}$ is a non-Borel $\mathbf{\Pi}^1_1$ subset of Tr, since it of the form there is does not exist a order preserving map (which is coded by a real) from ω into some tree $\mathcal{T}$.

Definition 5.2.2. The height of a tree $\mathcal{T}$, denoted $||\mathcal{T}||$ is the least ordinal such that there is an order preserving map from $\mathcal{T}$ into the ordinals. Define the rank of a tree $||\mathcal{T}||$, denoted $\mathrm{tr}(\mathcal{T})$, as $\mathrm{tr}(\mathcal{T}) = ||\mathcal{T}|| + 1$. Let $\mathtt{WF}_\alpha$ be the set of trees of rank less than or equal to α.

For each $\alpha < \omega_1$ the set $\mathtt{WF}_\alpha$ is Borel. Let $\cong_\alpha$ denote the isomorphism relation on $\mathtt{WF}_\alpha$.

Definition 5.2.3. The iterated FS-jump of equality is defined by transfinite recursion on the countable ordinals as follows

(i) $=^{0+} = \mathtt{Id}(\omega^\omega)$

(ii) $=^{(\alpha+1)+} = (=^{\alpha+})^+$

(iii) For limit λ, $=^{\lambda+} = \Pi_{\alpha>\lambda} =^{\alpha+}$.

The following theorem gives the relationship between iterates of the FS-jump and the isomorphism relation on $\mathtt{WF}_\alpha$.

Theorem 5.2.1. *For each $\alpha < \omega_1$, $\cong_{3+\alpha}$ is Borel bireducible with $=^{\alpha+}$.*

Proof. See [11] Theorem 13.2.5. □

In a forthcoming work we hope to provide a computable version of this theorem.

REFERENCES

[1] Uri Andrews, Steffen Lempp, Joseph S. Miller, Keng Meng Ng, Luca San Mauro, and Andrea Sorbi. Universal computably enumerable equivalence relations. *J. Symb. Log.*, 79(1):60–88, 2014.

[2] Uri Andrews and Andrea Sorbi. Joins and meets in the structure of Ceers. *arXiv e-prints*, page arXiv:1802.09249, February 2018.

[3] Uri Andrews and Andrea Sorbi. Jumps of computably enumerable equivalence relations. *Ann. Pure Appl. Logic*, 169(3):243–259, 2018.

[4] C. J. Ash and J. Knight. *Computable structures and the hyperarithmetical hierarchy*, volume 144 of *Studies in Logic and the Foundations of Mathematics.* North-Holland Publishing Co., Amsterdam, 2000.

[5] S. Barry Cooper. *Computability theory.* Chapman & Hall/CRC, Boca Raton, FL, 2004.

[6] Samuel Coskey, Joel David Hamkins, and Russell Miller. The hierarchy of equivalence relations on the natural numbers under computable reducibility. *Computability*, 1(1):15–38, 2012.

[7] Ekaterina Fokina, Dino Rossegger, and Luca San Mauro. Measuring the complexity of reductions between equivalence relations. *arXiv e-prints*, page arXiv:1806.10363, June 2018.

[8] Ekaterina B. Fokina and Sy-David Friedman. On Σ^1_1 equivalence relations over the natural numbers. *Math. Log. Q.*, 58(1-2):113–124, 2012.

[9] Ekaterina B. Fokina, Sy-David Friedman, Valentina Harizanov, Julia F. Knight, Charles McCoy, and Antonio Montalbán. Isomorphism relations on computable structures. *J. Symbolic Logic*, 77(1):122–132, 2012.

[10] Harvey Friedman. On the necessary use of abstract set theory. *Adv. in Math.*, 41(3):209–280, 1981.

[11] Su Gao. *Invariant descriptive set theory*, volume 293 of *Pure and Applied Mathematics (Boca Raton).* CRC Press, Boca Raton, FL, 2009.

[12] Su Gao and Peter Gerdes. Computably enumerable equivalence relations. *Studia Logica*, 67(1):27–59, 2001.

[13] Thomas Jech. *Set theory.* Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2003. The third millennium edition, revised and expanded.

[14] Vladimir Kanovei. *Borel equivalence relations*, volume 44 of *University Lecture Series.* American Mathematical Society, Providence, RI, 2008. Structure and classification.

[15] Antonio Montalbán. Computability theoretic classifications for classes of structures. In *Proceedings of the International Congress of Mathematicians—Seoul 2014. Vol. II*, pages 79–101. Kyung Moon Sa, Seoul, 2014.

[16] Gerald E. Sacks. *Higher recursion theory.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1990.

[17] Theodore A. Slaman and John R. Steel. Definable functions on degrees. In *Cabal Seminar 81–85*, volume 1333 of *Lecture Notes in Math.*, pages 37–55. Springer, Berlin, 1988.

[18] Robert I. Soare. *Recursively enumerable sets and degrees.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1987. A study of computable functions and computably generated sets.

APPENDIX A

FORCING

Forcing was originally introduced by Paul Cohen to show that the continuum hypothesis was not provable in ZFC. We give a brief treatment here for reference.

Definition A.0.1. Given model $\mathcal{M}$ (usually countable and transitive) of set theory a notion of forcing is a poset $\mathbb{P} = \langle P, \leqslant \rangle$ such that $\mathbb{P} \in \mathcal{M}$. The elements of P are called conditions. If $p \leqslant q$, then p is said to be stronger than q.

Definition A.0.2. A set $D \subset P$ is called dense if for all $p \in P$ there is a $q \in D$ so that $q \leqslant p$. A filter G on P is a subset P such that the following hold

(i) For all $p \in G$ and for all $q \in P$, if $p \leqslant q$ then $q \in G$.

(ii) For all $p, q \in G$ there exists $r \in G$ so that $r \leqslant p$ and $r \leqslant q$.

A filter G is called generic if for all dense sets D in P, then $G \cap D \neq \varnothing$.

Forcing a Cohen Real

Let $\mathcal{M} \vDash \mathtt{ZF}$ be countable and transitive. Let $P = 2^{<\omega}$, where $\sigma \leqslant \tau$ if and only if $\sigma \supset \tau$. We think of the conditions of P as finite partial functions from ω to 2. This notion of forcing is called Cohen forcing. Let $G \subset P$ be a generic filter. For compatible $p, q \in G$ we have they agree on the intersection or domain, thus $p \leqslant q$ or $q \leqslant p$. It also follows that there is an r such that $r \leqslant p$ and $r \leqslant q$. Thus G is linear ordered. Let $x_G = \bigcup G$. We must show that x_G is typical in the following way. It is possible that x_G have a small domain, but requiring that G meet every dense set allows us make x_G different from every real in the ground model $\mathcal{M}$. For $n \in \omega$ let $D_n = \{p \in P \mid n \in \mathrm{dom}(p)\}$. We can meet these dense sets since if $n \notin \mathrm{dom}(p)$, then we can extend p by defining it for n, and thus meeting D_n. For a real $h : \omega \to 2$ define $E_h = \{p \in P \mid (\exists n)(n \in \mathrm{dom}(p))(p(n) \neq h(n))\}$. We can meet this dense set since for given undefined $p(n)$ define it to be 1 if $h(n) = 0$ and to be 0 if $h(n) = 1$. Then every condition sees that x_g is not equal to any other real h.

Forcing in Computability Theory

Forcing is used in computability theory to construct subsets of natural numbers that have certain properties. It is equivalent to forcing a Cohen real except that we don't require our filter to meet every dense set, only certain arithmetically definable ones.

Definition A.0.3. A set $A \subset \omega$ is called n-generic if for every Σ^0_n set $S \subset 2^{<\omega}$ either

(i) $(\exists \tau \subset A)(\tau \in S)$

(ii) $(\exists \tau)(\forall \sigma \supset t)(\sigma \notin S)$

If (i) holds we say that A meets S, and if (ii) holds then we say A misses S.

Thus we are saying here that A is n-generic if it meets or avoids every Σ^0_n set of strings.

Theorem A.0.1. *There is a* 1*-generic set* $A \leqslant_T \varnothing'$.

Proof. We'll construct strings σ_i so that for each i we have σ_i meeting W_i or avoiding W_i, and let $A = \bigcup_i \sigma_i$. At stage $s = 0$ set $\sigma_0 = \varnothing$. At stage $s = i + 1$ ask the question

$$(\exists t \subset \sigma_i)(\tau \supset \tau' \text{ for some } \tau' \in W_i).$$

If there is such a τ pick the least one and set $\sigma_{i+1} = \tau$. As a result $\sigma_{i+1} \in W_i$ and satisfies (i) of the definition. If there is no such τ, then define $\sigma_{i+1} = \sigma_i \frown 0$. As a result every extension of σ_{i+1} misses W_i and satisfies (ii) of the definition. Finally it is clear that A as defined misses or meets every c.e. set and is thus 1-generic. To see that this construction is computable in $\varnothing'$, notice that our question at each stage is Σ^0_1. □