COMPUTABLE ASPECTS OF CLOSED SETS

By

PAUL BRODHEAD

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2008
To my Family, the Artisans of Life
ACKNOWLEDGMENTS

I give thanks, first and foremost, to my Ph.D. advisor Douglas Cenzer, who must be noted in this generation as among the most patient and irrevocably kind of men. His sense of humor also trumps the cloudiest of days, be they even the clouds of hurricanes—which, in fact, occurred as my studies progressed in Florida. These characteristics, combined with my determination, helped me gain an understanding, broad and in-depth, of computability that allowed me to successfully undertake and develop the research presented here. The road to this point began long ago, but reached its pinnacle at Florida under his direction.

The road began at the University of Wisconsin, in my first semester; I was first inspired to pursue mathematics while taking Algebra and Trigonometry. Professor Arnold Johnson made the course enjoyable and challenging, making it unlike any mathematics course I ever took. Him I thank, for as have I always had high goals, his influence was critical as it was at this point in life that I decided that I was to pursue a Ph.D. in mathematics. Later, in my last semester at Wisconsin, he helped me develop a mathematical maturity beyond expectations, as he led me in an undergraduate research project in his area, Algebra. Yet, between these experiences, many others must be thanked.

During my undergraduate years, many inspiring teachers pushed the mathematical envelope, delivering the endeavor of the ages, the enjoyable mental fray of the breaking of mathematical barriers. I thank: Steffen Lempp, Reed Solomon, H. Jerome Keisler, Arnold Miller, Maury Bramson, Steve Bauman, Simon Hellerstein, Daniel Rider, Daniel Shea, and Jennifer Ziebarth. In particular, I thank Steffen Lempp for introducing me to computability theory while I was in his linear algebra class, as it was at this moment that I determined my Ph.D. specialization. My mindset was ever afterwards directed this way. For example, I thank Arnold Miller for his patience in listening to my numerous attempts in using computability theory in his year-long abstract algebra class. I thank H. Jerome Keisler for listening to me in his logic course. I also thank Reed Solomon for helping me even further in his graduate logic course and his subsequent guidance in my first undergraduate research project, the study of Hilbert’s Tenth Problem. I thank for Daniel
Shea (also the undergraduate advisor) for accepting my destined course in mathematics as I sat in his abstract analysis course, vigorously driven under a department mathematics scholarship, when years prior I entered his office whilst still in Algebra and Trigonometry, demanding help in setting up my undergraduate curriculum for an eventual Ph.D. in mathematics.

Upon the closing of my Wisconsin experience, excellence in mathematics, a higher level still, evolved at the University of Puerto Rico-Humacao during the immediate summer. I participated in the NSF-funded undergraduate research experience, the Summer Institute of Mathematics for Undergraduates. I thank: Reinhard Laubenbacher, Abdul Jarrah, Rebecca Garcia, Malarie Cummings, Cora Seidler, Ivelisse Rubio, and Herbert A. Medina. Herbert, Ivelisse, and Reinhard, in particular, helped me to see the urgency in the tackling of the problems at hand. I am greatly appreciative of this help to this day.

My research rigor and core mathematical talent bloomed in graduate school at Florida. I thank: Douglas Cenzer, Bill Mitchell, Jindrich Zapletal, Jean Larson, Alexandre Turull, Paul Robinson, James Keesling, Jonathan King, Scott McCullough, Stephen Summers, and Shari Moscow. I especially thank Alexandre Turull. I had the privilege of having six semesters with him; he helped me to see a problem, its value, and its complexities. I thank my dissertation committee: Douglas Cenzer, Rick Smith, Bill Mitchell, Murali Rao, and Beverly Sanders. I also thank Bernard Mair, Rick Smith, Murali Rao, and Juan Liu; my experiences with them convinced me to move in valuable directions that took me around the globe and beyond.

My collaborators, near and abroad, you I thank: Douglas Cenzer, Angsheng Li, Weilin Li, George Barmpalias, Seyyed Dashti, Rebecca Weber, Jeffrey Remmel, Rod Downey, Noam Greenberg, Keng Meng Ng, and Bjorn Kjos-Hanssen. Many many more do I also thank– I have collaborated with many people, if even for a short time, and their contributions are appreciated. I thank: Peter Hinman, Denis Hirschfeldt, Carl Jockusch, Russell Miller, Johanna Franklin, Bakhadyr Khoussainov, Jan Riemann, Ted Slaman,
André Nies, Robert Soare, Steve Simpson, Antonio Montalbán, Barbara Csima, Joe Miller, Verónica Becher, and Wolfgang Merkle.

I thank the people and institutions who contributed their support through funding: the Southeast Alliance for Graduate Education and the Professoriate, the National Science Foundation, the University of Florida, the Chinese Academy of Sciences, the Association for Symbolic Logic, and others.

Many people deserve credit far beyond the brief, or even omitted, acknowledgement here; I am extremely grateful to you. Of special consideration is my advisor. The magnitude of his support is above any measurable scale. His support in all my endeavors, professionally and beyond, will surely continue to help shape me. This dissertation and the work therein, and the experiences associated with the attainment thereof are a witness to this.

To the many people who helped me along this journey, which is actually just a beginning, you I thank.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACKNOWLEDGMENTS</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>ABSTRACT</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>12</td>
</tr>
<tr>
<td>1.1</td>
<td>General Overview</td>
<td>12</td>
</tr>
<tr>
<td>1.2</td>
<td>Classical Computability</td>
<td>14</td>
</tr>
<tr>
<td>1.3</td>
<td>Closed Sets in Computability</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>EFFECTIVELY CLOSED SETS AND ENUMERATIONS</td>
<td>18</td>
</tr>
<tr>
<td>2.1</td>
<td>Introduction</td>
<td>18</td>
</tr>
<tr>
<td>2.2</td>
<td>The Family of $\Pi^0_1$ Classes</td>
<td>20</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Numberings in the Literature</td>
<td>20</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Equivalence of the Numberings</td>
<td>22</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Equivalence of the Numberings (Alternate Proof)</td>
<td>25</td>
</tr>
<tr>
<td>2.2.4</td>
<td>Injective Computable Numberings</td>
<td>26</td>
</tr>
<tr>
<td>2.2.4.1</td>
<td>Original c.e. sets argument</td>
<td>27</td>
</tr>
<tr>
<td>2.2.4.2</td>
<td>Ordered tuples of disjoint c.e. sets</td>
<td>29</td>
</tr>
<tr>
<td>2.2.4.3</td>
<td>Results for effectively closed sets</td>
<td>32</td>
</tr>
<tr>
<td>2.3</td>
<td>String Verifiable Families of $\Pi^0_1$ Classes</td>
<td>37</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Definition and Examples</td>
<td>37</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Computable and Effective Numberings</td>
<td>39</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Families Containing the Clopen Classes</td>
<td>40</td>
</tr>
<tr>
<td>2.4</td>
<td>Named Families of $\Pi^0_1$ Classes</td>
<td>42</td>
</tr>
<tr>
<td>2.4.1</td>
<td>Homogeneous Classes</td>
<td>42</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Decidable Classes</td>
<td>44</td>
</tr>
<tr>
<td>2.4.2.1</td>
<td>An injective computable numbering (Alternate proof)</td>
<td>44</td>
</tr>
<tr>
<td>2.4.2.2</td>
<td>Trees without dead ends: A numbering result</td>
<td>45</td>
</tr>
<tr>
<td>2.4.2.3</td>
<td>Trees with dead ends: A necessity</td>
<td>47</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Thin and Perfect Thin Classes</td>
<td>49</td>
</tr>
<tr>
<td>2.4.3.1</td>
<td>The Martin–Pour El Construction</td>
<td>49</td>
</tr>
<tr>
<td>2.4.3.2</td>
<td>Non-existence of computable numberings</td>
<td>50</td>
</tr>
<tr>
<td>2.4.4</td>
<td>Small, Very Small, and Nondecidable Classes</td>
<td>51</td>
</tr>
<tr>
<td>2.4.4.1</td>
<td>Numberings and high/noncomputable sets</td>
<td>51</td>
</tr>
<tr>
<td>2.4.4.2</td>
<td>Non-existence of effective numberings</td>
<td>52</td>
</tr>
<tr>
<td>3</td>
<td>RANDOM CLOSED SETS</td>
<td>54</td>
</tr>
<tr>
<td>3.1</td>
<td>Overview</td>
<td>54</td>
</tr>
<tr>
<td>3.2</td>
<td>Effective Randomness of Reals</td>
<td>55</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Introduction</td>
<td>55</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Constructive Martingale Randomness</td>
<td>56</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Prefix-free Randomness</td>
<td>57</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Martin-Löf ($n$-)randomness</td>
<td>58</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.3</td>
<td>Martin-Löf Randomness of Closed Sets</td>
<td>60</td>
</tr>
<tr>
<td>3.3.1</td>
<td>The Hit-or-Miss Topology on C</td>
<td>60</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Toward a Measure</td>
<td>61</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Canonical Coding and Measure</td>
<td>61</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Ghost Coding</td>
<td>64</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Coding Equivalence</td>
<td>65</td>
</tr>
<tr>
<td>3.3.6</td>
<td>Coding and Joins of Closed Sets</td>
<td>66</td>
</tr>
<tr>
<td>3.4</td>
<td>Members of Random Closed Sets</td>
<td>67</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Positive Results</td>
<td>68</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Negative Results</td>
<td>73</td>
</tr>
<tr>
<td>3.5</td>
<td>Measure and Dimension</td>
<td>78</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Measure</td>
<td>78</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Dimension</td>
<td>79</td>
</tr>
<tr>
<td>3.6</td>
<td>Prefix-Free Complexity of Closed Sets</td>
<td>82</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Lower Complexity Bounds</td>
<td>83</td>
</tr>
<tr>
<td>3.6.2</td>
<td>Upper Complexity Bounds</td>
<td>85</td>
</tr>
<tr>
<td>3.7</td>
<td>Other Notions of Randomness for Closed Sets</td>
<td>87</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Randomness with Regular Probability Measures</td>
<td>87</td>
</tr>
<tr>
<td>3.7.2</td>
<td>Randomness with the Inclusion of Trees with Deads Ends</td>
<td>87</td>
</tr>
<tr>
<td>3.8</td>
<td>Random Closed Sets and Effective Capacity</td>
<td>88</td>
</tr>
<tr>
<td>3.8.1</td>
<td>Computable Capacities</td>
<td>88</td>
</tr>
<tr>
<td>3.8.2</td>
<td>Regular Measures and Capacities of Closed Sets</td>
<td>90</td>
</tr>
<tr>
<td>4</td>
<td>RANDOM CONTINUOUS FUNCTIONS</td>
<td>94</td>
</tr>
<tr>
<td>4.1</td>
<td>Overview</td>
<td>94</td>
</tr>
<tr>
<td>4.2</td>
<td>Defining Randomness for Continuous Functions</td>
<td>95</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Representing Functions</td>
<td>95</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Representing Sequences</td>
<td>96</td>
</tr>
<tr>
<td>4.2.3</td>
<td>A Sound Definition</td>
<td>97</td>
</tr>
<tr>
<td>4.3</td>
<td>Random Continuous Functions and Images</td>
<td>99</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Perfect Images, in every instance</td>
<td>99</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Non-injective Images, in every instance</td>
<td>100</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Non-surjective Images, in instances</td>
<td>103</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Images of computable elements</td>
<td>104</td>
</tr>
<tr>
<td>4.4</td>
<td>Random Closed Sets arising from random continuous functions</td>
<td>106</td>
</tr>
<tr>
<td>4.4.1</td>
<td>A Positive Result: Inverse Images of $0^e$</td>
<td>106</td>
</tr>
<tr>
<td>4.4.2</td>
<td>A Negative Result: Images, in general</td>
<td>109</td>
</tr>
<tr>
<td>4.5</td>
<td>Pseudo-Distance Functions</td>
<td>110</td>
</tr>
<tr>
<td>4.6</td>
<td>n-Randomness</td>
<td>111</td>
</tr>
<tr>
<td>4.7</td>
<td>Future Work</td>
<td>112</td>
</tr>
<tr>
<td>5</td>
<td>CONTINUITY OF CAPPING IN $C_{BT}$</td>
<td>113</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>113</td>
</tr>
<tr>
<td>5.2</td>
<td>Continuity Results</td>
<td>114</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Continuity Results in $C$</td>
<td>114</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Continuity Results in $C_{bt}$ and Main Result</td>
<td>115</td>
</tr>
</tbody>
</table>
5.3  Requirements and Strategies ........................................ 118
  5.3.1  The requirements .................................................. 118
  5.3.2  A $P$-strategy ....................................................... 119
  5.3.3  An $R$-strategy ....................................................... 120
  5.3.4  An $S$-strategy ....................................................... 121
5.4  The Priority Tree ......................................................... 125
5.5  The Construction ........................................................ 128
5.6  The Verification ........................................................ 132

REFERENCES ................................................................. 144

BIOGRAPHICAL SKETCH ..................................................... 150
A closed set in \(\{0, 1\}^\mathbb{N}\) may be viewed the set of infinite paths through a tree; a set \(A\) is computable if there is a computer program which halts and gives the correct answer on every query to the membership predicate for \(A\). A numbering, or enumeration, is a map from \(\mathbb{N}\) onto a countable collection of objects; if there is a computable numbering onto a set \(A \subseteq \mathbb{N}\) then we say that \(A\) is computably enumerable (or c.e.). The set of infinite paths through a computable, or equivalently a co-c.e., tree is called an effectively closed set.

In this work, we investigate: (1) numberings for different families of effectively closed sets, (2) notions of randomness for nonempty closed subsets of \(2^\mathbb{N}\), (3) notions of randomness for continuous functions from \(2^\mathbb{N}\) to \(2^\mathbb{N}\), and (4) continuity properties of \(C_{\leq bT}\), the c.e. degrees under the Turing reduction \(\leq_{bT}\) that requires that each use be bounded by a computable function.

(1) Numberings and effectively closed sets. We show that certain families (or classes of families) of effectively closed sets—such as the decidable, homogeneous, thin, small or the entire family of effectively closed sets, or string verifiable families—possess, or do not possess, (injective) computable or effective numberings. This works builds upon the seminal work by Friedberg [46], who constructed an injective numbering of the c.e. sets.

(2) Randomness of closed sets. In the space of closed sets, we give a probability measure and define a version of the Martin-Löf Test for randomness. We show that random closed sets are never effectively closed, but are, on the other hand, always perfect, have measure zero, and have box dimension \(\log_2 \frac{4}{3}\). Every random closed set contains random and non-random elements, but no \(n\)-c.e. elements. We also explore alternate
notions for randomness, such as the problem of compressibility of trees. Finally, we consider the problem of when a randomly chosen closed set meets a closed $Q$; this is the study of capacities.

(3) Randomness of continuous functions. As in (2), we give a probability measure and define a version of the Martin-Löf Test for randomness. We show that the image of a random continuous function is always non-injective and perfect, but not necessarily surjective. Furthermore, computable elements map to random elements. Also, random closed sets arise as inverse images of $0^\omega$, but not, in general, as images. The former motivates a study of pseudo-distance functions. Finally, we consider our results in the context of $n$-randomness.

(4) Continuity properties in $C_{bT}$. We show in $C_{bT}$ that for any $b \neq 0, 0'$, there is an $a > b$ such that for any $x$, $b \land x = 0$ iff $a \land x = 0$. We prove this by first showing that the Seetapun local noncappability theorem in the c.e. Turing degrees [84] also holds in $C_{bT}$. This theorem demonstrates that every $b \neq 0, 0'$ is noncappable with any nontrivial degree below some $a > b$ (i.e. if $x < a$ and $x \land b = 0$ then $x = 0$).
CHAPTER 1
INTRODUCTION

This thesis is an accumulation of my work as a graduate student at the University of Florida. Much of this work is joint and published, or to be published. The citations are listed at the beginning of the appropriate chapters. In this chapter we introduce various notions and expound upon these in later chapters. Each of Chapters 2–5 contains a distinct topic from computability theory.

1.1 General Overview

Computability theory is a field of mathematical logic; the subject captures the precise notion of an algorithmic process towards the study of decidable/undecidable problems in mathematics and nature. Its most notable historical contribution to mathematics is the disruption of Hilbert’s Program by Gödel’s Incompleteness Theorem. More recent work has shown that other mathematical problems are unsolvable, in the sense that no computer algorithm can solve every instantiation of the problem. Examples include Hilbert’s Tenth Problem (to decide whether a given Diophantine equation has solutions), the word problem for groups (to decide whether a given product of generators and their inverses is the identity element of a group defined by a finite set of equations between such products), and the homeomorphy problem (to decide whether the topological spaces defined by a given pair of simplicial complexes are homeomorphic) [32]. Maturation of computability, however, through applications and techniques, has broadened its interactions with other fields, most notably computer science. Each of Chapters 2–5 is a witness to this broadening.

In Chapter 2, we focus on the study of effectively closed sets of binary reals. Thought of as a set representing the solutions to some problem, effectively closed sets characterize many structures in mathematics and computer science. In algebra, for example, they represent the prime ideals of a computable enumerable Boolean algebra or a commutative ring with identity [47]. In graph theory, they represent the set of solutions to many problems with computable graphs, such as Hamiltonian circuits or vertex partitions [20]. In computer science, effectively closed sets arise in the study of non-monotonic logics and
\(\omega\)-languages \([25, 68, 70]\). Given the wide variety of applications, the work in Chapter 2 focuses on methods of enumerating various families of effectively closed sets. The idea is to provide, based on desired properties, complete listings of the objects— in this case, effectively closed sets— representing the set of solutions to problems of a certain type. We show that there is an injective computable enumeration of the entire class of sets, of certain families of string verifiable classes, and of the decidable and homogeneous classes. We also show that no computable enumeration exists for thin, perfect thin, small, very small, or nondecidable classes.

In Chapters 3 and 4, we extend notions related to effective randomness for binary reals, to closed sets (Chapter 3) and continuous functions (Chapter 4); various global properties are obtained. A binary real is effectively random if it is impossible for a computer to find regularity or patterns in it. For a closed set, representing some set of solutions, this means that it is difficult for a computer to precisely obtain or locally describe this set of solutions, given the lack of pattern. In both chapters, we obtain so-called basis and antibasis theorems. For instance, every random closed set contains random and non-random elements, but omits various elements of computability-theoretic interest, such as the properties of being \(f\)-c.e. \((f\) a polynomial), 1-generic, or of incomplete c.e. degree. We also show that concepts from both chapters are closed related; random closed sets arise as inverse images of random continuous functions mapping to \(0^\omega\), but not, in general, as images. Methods employed in all of this work range from techniques in computability and effective randomness, to techniques related to the study of effective Hausdorff dimension and classical probability.

Finally, in connection with the roots of computability theory, in Chapter 5 we focus on the classification of information content by means of Turing degree theory. Mathematical structures or objects are often encoded as sets of natural numbers. Reduction methods, such as the Turing reducibility, allow the information content of these sets to be classified into equivalence classes, called degrees. In Chapter 5, the focus is on the bounded Turing reducibility \([16]\); we show that capping, the operation which takes the meet of a given
noncomputable incomplete c.e. degree with another noncomputable incomplete c.e. degree such that the resulting meet is the degree 0 of the computable sets, is continuous. That is, for any $b \neq 0, 0'$, there is an $a > b$ such that for any $x$, $b \land x = 0$ iff $a \land x = 0$. As this is a three-quantifier statement in the c.e. $bT$-degrees, this result gives insight into the three quantifier theory of the same, whose decidability/undecidability is currently unknown. As an aside, in recent work by Soare, the bounded Turing reducibility with the identity use has led to applications of degree theory to differential geometry [72, 87].

The rest of this chapter is devoted to introducing to basic definitions, terminology, and notations that will be used throughout this entire work. Section 1.2 covers basic notions and notations for topics in classical computability (e.g. computable sets, computably enumerable sets, partial computable functions, Turing degrees). Section 1.3 covers the basics of closed sets in $\mathbb{N}^\mathbb{N}$. As algorithmic randomness is a topic covered only in Chapters 3 and 4, we postpone a general introduction of this topic and provide it in Section 3.2.

### 1.2 Classical Computability

We generally follow the notation of Soare [86] for notions that arise from classical computability. For an in-depth treatment of the basic foundations of the subject, we refer the reader there.

A set $A$ is computable if there is a computer program, or equivalently a computable function, which halts and gives the correct answer on every query to the membership predicate for $A$. It is computably enumerable (c.e.) if the computer program is required to halt only on queries for elements in $A$; this gives rise to the standard example of a c.e. set, namely the halting problem—the set of Turing machines, officially encoded as a set of natural numbers, that halt when given their own binary input. Computably enumerable sets are the domains, therefore, of so-called partial computable functions; we index the partial computable $\{0, 1\}$–valued functions as $\{\phi_e\}_{e \in \omega}$. Partial computable functionals that take natural number ($m$) and real ($x$) inputs are indexed as $\Phi_e$; we will write $\Phi_e^x(m)$ for the result of applying $\Phi_e$ to $m$ and $x$.  

14
Other related notations are standard: $\phi_{e,s}$ denotes that portion $\phi_e$ defined by stage $s$, and $\phi_e(x) \downarrow$ means that $\phi_e$ is defined on $x$ (and $\uparrow$ means undefined). We also index the primitive recursive functions, a smaller class of total functions, as $\{\pi_e\}_{e \in \omega}$. $(\cdot, \cdot) : \omega^2 \to \omega$ is typically a computable bijection such that $(0, 0) = 0$. $\overline{A}$ and $\mathcal{P}(A)$ denote the complement and power set of $A$, respectively. $z = x \oplus y$ is the coding together of two reals $x$ and $y$, so that $z(2n) = x(n)$ and $z(2n + 1) = y(n)$ for all $n$.

In computability, a reduction is a binary relation on subsets of $\mathbb{N}$ that captures a relationship between the information content of two sets. The Turing reduction $\leq_T$ is the main reduction used in computability theory. $A$ is Turing reducible to $B$, written $A \leq_T B$, if membership in $A$ can be determined by a computer algorithm that has full access to the membership predicate for $B$. Intuitively, the information content of $A$ is viewed as computable, or recoverable, from $B$. Furthermore $B$ is viewed as an oracle, in terms of information content, for determining membership in $A$.

Various restrictions on how much oracle information is allowed to be used in determining the membership of a single element have given rise to different kinds of reductions. For example, the bounded Turing reduction $\leq_{bT}$ requires that query to an oracle $B$ for determining membership of $x$ in $A$ use at most $f(x)$ amount of $B$, where $f$, called the use, is bounded by a computable function. The identity bounded Turing reductions insists that the use $f$ be bounded by the identity function.

Reductions often give rise to equivalence classes, called degrees, where two sets are equivalent if they are mutually reducible. The Turing degrees that contain c.e. sets are called c.e. Turing degrees. We denote $\mathcal{C}$ and $\mathcal{C}_{bT}$ as the structures of the c.e. degrees under the Turing reductions and the bounded Turing reductions respectively. The study of the Turing degrees has been one of the major themes in computability research; these degrees capture the structure of the undecidable problems in arithmetic and nature.

### 1.3 Closed Sets in Computability

We generally follow the notation of Cenzer [19] for closed sets: For a finite string $\sigma \in \omega^n$, let $|\sigma| = n$. We let $\emptyset$ denote the empty string, which has length 0. A word $(a)$ of
length 1 is may be identified with the symbol $a$. For two strings $\sigma, \tau$, say that $\tau$ extends $\sigma$ and write $\sigma \sqsubseteq \tau$ if $|\sigma| \leq |\tau|$ and $\sigma(i) = \tau(i)$ for $i < |\sigma|$. Similarly $\sigma \sqsubset x$ for $x \in 2^\mathbb{N}$ means that $\sigma(i) = x(i)$ for $i < |\sigma|$. Let $\sigma \prec \tau$ denote the concatenation of $\sigma$ and $\tau$. Given a finite string $\sigma$, let $I(\sigma)$, or alternatively $[\sigma]$, be the interval of all infinite sequences extending $\sigma$, i.e. $I(\sigma) = \{x \in 2^\mathbb{N} : \sigma \sqsubseteq x\}$. Each such interval is a clopen set and the clopen sets are just finite unions of intervals.

A subset $T$ of $\omega^{<\omega}$ is a tree if it is closed under initial segments. The set $[T]$ of infinite paths through $T$ is defined by $x \in [T] \iff (\forall n)x[n] \in T$, where $x[n] = (x(0), \ldots, x(n-1))$. We say that a tree $T \subseteq \omega^{<\omega}$ and set $[T]$ are clopen if there is a nonempty finite $S \subseteq \omega^{<\omega}$ so that $T = \emptyset$ or $T = \{\sigma : \sigma \sqsubseteq \tau$ or $\tau \sqsubseteq \sigma$ for some $\tau \in S\}$. $P \subseteq \omega^\omega$ is closed if and only if $P = [T]$ for some tree $T$. Now a nonempty closed set $P$ may be identified with a unique tree $T_P = \{\sigma : P \cap I(\sigma) \neq \emptyset\}$. $T_P$ has the distinct property of having no dead ends; that is, if $\sigma \in T_P$ then either $\sigma^0 \in T_P$ or $\sigma^1 \in T_P$.

$P$ is an effectively closed set, or $\Pi_1^0$ class, if $P = [T]$ for some computable tree $T$. Other definitions are equivalent; in particular $P$ is a $\Pi_1^0$ class if and only if $P = [T]$ for some primitive recursive tree $T$ and also if and only if $P = [T]$ for some $\Pi_0^1$ tree $T$. Note that if $P$ is a $\Pi_1^0$ class, then $T_P$ is a $\Pi_0^1$ set, but not in general computable. $P$ is said to be a decidable $\Pi_1^0$ class if $T_P$ is computable. $P$ is said to be a strong $\Pi_0^1$ class, if $T_P$ is a $\Pi_2^0$ set, or equivalently if $P = [T]$ for some $\Delta_2^0$ tree; $P$ is said to be a strong $\Delta_2^0$ class if $T_P$ is $\Delta_0^0$. Thus any $\Pi_1^0$ class is also a strong $\Delta_2^0$ class. Any decidable $\Pi_1^0$ class contains a computable element (in particular the leftmost and rightmost paths) and similarly any strong $\Delta_2^0$ class contains a $\Delta_0^0$ element. On the other hand, there exist $\Pi_1^0$ classes with no computable elements and strong $\Pi_0^1$ classes with no $\Delta_0^0$ elements. A $\Pi_1^0$ class is said to be special if it does not contain a computable member.

A c.e. open set is defined to be the complement of an effective closed set. That is, if $P = [T]$, then $\omega^\omega - P = \bigcup_{(\sigma) \notin T} I(\sigma)$. There is a natural effective enumeration $P_0, P_1, \ldots$ of the $\Pi_1^0$ classes and thus an enumeration of the c.e. open sets. Thus we can say that a sequence $S_0, S_1, \ldots$ of c.e. open sets is effective if there is a computable function, $f$, such
that $S_n = 2^N - P_{f(n)}$ for all $n$. For any c.e. set $W$, we define the c.e. open set generated by $W$ to be

$$\mathcal{O}(W) = \bigcup \{I(\sigma) : \langle \sigma \rangle \in W\}.$$ 

Also let

$$\mathcal{O}(W) \upharpoonright n = \{x \upharpoonright n : x \in \mathcal{O}(W) \text{ and } (\forall j < n) \ x(j) \leq n\}.$$ 

For a detailed development of $\Pi^0_1$ classes, see [20] or [24].
CHAPTER 2
EFFECTIVELY CLOSED SETS AND ENUMERATIONS

The following chapter is joint work with Douglas Cenzer and has been submitted as an article entitled Effectively Closed Sets and Enumerations [13]. A preliminary version of this research was originally presented at the Third International Conference of Computability and Complexity in Analysis in Gainesville, Florida in 2006 by P. Brodhead. This preliminary work was published in the referred conference proceedings as Enumerations of $\Pi^0_1$ Classes: Acceptability and Decidable Classes (P. Brodhead) in hard copy and in Springer Electronic Notes in Theoretical Computer Science, Elsevier Science 167 (2007), 289-301 [12].

Portions of this work were also presented by P. Brodhead at the 2005 SACNAS Conference (October 2005, Denver, CO), 7th Annual Graduate Student Conferences in Logic (April 2006, Madison, WI), the 2007 Association for Symbolic Logic Annual Meeting (March 2007, Gainesville, FL), the 2nd New York Graduate Student Conference in Logic (March 2007, New York, NY), and the 8th Annual Graduate Student Conference in Logic (April 2007, Chicago, IL). Due to inclimate weather, R. Miller presented in place of P. Brodhead at the New York conference.

2.1 Introduction

The general theory of numberings was initiated in the mid-1950s by Kolmogorov, and continued under the direction of Mal’tsev and Ershov [44]. A numbering, or enumeration, of a collection $C$ of objects is a surjective map $F: \omega \to C$. In one of the earliest results, Friedberg constructed an injective computable numbering $\psi$ of the $\Sigma^0_1$ or computably enumerable (c.e.) sets such that the relation “$n \in \psi(e)$” is itself $\Sigma^0_1$. More generally, we will say that a numbering $\psi$ of collection of objects with complexity $\mathcal{C}$ (such as n-c.e., $\Sigma^0_n$, or $\Pi^0_n$) is effective if the relation “$x \in \psi(e)$” has complexity $\mathcal{C}$. We will also consider enumerations where the relation “$x \in \psi(e)$” has a different complexity than $\mathcal{C}$. (For example, there is a c.e., but not computable, numbering of the computable sets.)

A numbering $\mu$ is acceptable with respect to a numbering $\nu$, denoted $\nu \leq \mu$, iff there is a total computable function $f$ such that $\nu = \mu \circ f$. If $\mu$ is acceptable with respect
to all effective numberings, then \( \mu \) is said to be \textit{acceptable}. The ordering \( \leq \) gives rise to an equivalence relation \( \equiv \), and two numberings in the same equivalence class are called \textit{equivalent}. Furthermore, the structure \( \mathcal{L}(C) \) of all numberings of \( C \) modulo \( \equiv \) forms an upper semilattice under \( \leq \). Here injective numberings occur only in the minimal elements and acceptable numberings occur only in the greatest element. In this chapter, we study effective numberings of families of effectively closed sets (i.e. \( \Pi_1^0 \) classes).

Enumerations of \( \Pi_1^0 \) classes were given by Lempp [61] and Cenzer and Remmel [23, 24], where index sets for various families of \( \Pi_1^0 \) classes were analyzed. For a given enumeration \( \psi(e) = P_e \) of the \( \Pi_1^0 \) classes and a property \( R \) of sets, \( \{e : R(P_e)\} \) is said to be an \textit{index set}. For example, \( \{e : P_e \text{ has a computable member}\} \) is a \( \Sigma_0^3 \) complete set. See [20] for many more results on index sets.

Certain types of \( \Pi_1^0 \) classes are of particular interest. Let \( P \) be a \( \Pi_1^0 \) class. We will say that \( P \) is \textit{thin} if for every \( \Pi_1^0 \) subclass \( Q \) of \( P \), there is clopen set \( U \) such that \( Q = U \cap P \).

We say that \( P \) is \textit{homogenous} if, given distinct \( \sigma, \tau \in T_P \) of the same length,

\[ \sigma \upharpoonright i \in T_P \iff \tau \upharpoonright i \in T_P. \]

For \( P \subseteq \{0, 1\}^\omega \), \( P \) is homogeneous if and only if \( P \) is the class of separating sets \( S(A, B) \) for two disjoint c.e. sets \( A, B \), that is,

\[ S(A, B) = \{C \subseteq \omega : A \subseteq C \text{ and } B \cap C = \emptyset\}. \]

\( P \) is \textit{small} if there is no computable function \( \phi \) such that, for all \( n \), \( \text{card}(T_P \cap \omega^{\phi(n)}) \geq n \).

Let \( \psi_P(n) \) be the least \( k \) such that \( \text{card}(T_P \cap \omega^k) \geq n \); then \( P \) is \textit{very small} if the function \( \psi_P \) dominates every computable function \( g \) – that is, \( \psi_P(n) \geq g(n) \) for all but finitely many \( n \).

A numbering \( e \mapsto [T_e] \) of \( \Pi_1^0 \) classes is called a \textit{tree} numbering and written \( e \mapsto T_e \). Numberings based on primitive recursive trees and on \( \Pi_1^0 \) trees are both studied in the literature (see [23, 24, 20]). If the set \( \{(e, \sigma) : \sigma \in T_e\} \) is computable, then the numbering \( \psi(e) = [T_e] \) is said to be a \textit{computable} numbering.
We begin our study with the family of $\Pi_1^0$ classes in Section 2.2. In Sections 2.2.1–2.2.3, several commonly used numberings are studied and shown to be equivalent via a computable permutation. In Section 2.2.4, we give a Friedberg numbering of the $\Pi_1^0$ classes; this motivates a study of a general class of families of $\Pi_1^0$ classes, called *string verifiable families* in Section 2.3. In Section 2.4, we consider named families of $\Pi_1^0$ classes. We obtain positive results for homogeneous can decidable classes in Sections 2.4.1 and 2.4.2. We obtain negative results for thin, perfect thin, small, very small, and nondecidable classes in Sections 2.4.3 and 2.4.4.

### 2.2 The Family of $\Pi_1^0$ Classes

#### 2.2.1 Numberings in the Literature

In this section, we present several different computable numberings of $\Pi_1^0$ classes that have appeared in the literature. We also present an effective, but not computable, numbering. In each case we demonstrate that each provides a complete numbering of the $\Pi_1^0$ classes.

**Numbering 1: Primitive Recursive Functions [23]**

For each $e$, let $\pi_e$ be the $e$th primitive recursive function from $\omega$ to $\omega$ and let

$$\sigma \in U_e \iff (\forall \tau \subseteq \sigma) \pi_e(\langle \tau \rangle) = 1.$$ 

Then $U_e$ is a (uniformly) primitive recursive tree for all $e$ and if $\{\sigma : \pi(\langle \sigma \rangle) = 1\}$ is any primitive recursive tree, then $U_e$ is that tree. Therefore the sequence $U_0, U_1, \ldots$ contains all primitive recursive trees and hence the mapping $\psi_1(e) = [U_e]$ is a computable numbering of the $\Pi_1^0$ classes.

**Numbering 2: Computably Enumerable Sets [20]**

Let

$$\psi_2(e) = \omega^\omega - O(W_e).$$

This is an effective numbering since the relation “$x \in \psi_2(e)$” is $\Pi_1^0$. This can actually be improved to a computable numbering, as follows.
For each $e$, recall that $W_{e,s}$ is the set of elements enumerated into the $e$th c.e. set $W_e$ by stage $s$ and let

$$\sigma \in S_e \iff (\forall \tau \subseteq \sigma) \langle \tau \rangle \notin W_{e,|\sigma|}.$$ 

Then $S_e$ is a (uniformly) primitive recursive tree for all $e$. Let $P = [T]$ be a $\Pi^0_1$ class, where $T$ is a computable tree. It follows that for some $e$,

$$\sigma \in T \iff \langle \sigma \rangle \notin W_e.$$ 

Then $P = [S_e]$. It follows that the sequence $[S_0], [S_1], \ldots$ contains all $\Pi^0_1$ classes and hence the mapping $\psi(e) = [S_e]$ is a computable numbering of the $\Pi^0_1$ classes. It is easy to see that in fact $[S_e] = \psi_2(e)$.

**Numbering 3: Universal $\Pi^0_1$ Relation [52, p.73]**

There is a universal $\Pi^0_1$ relation $U \subseteq \omega \times 2^\omega$ such that if $Q(x)$ is a $\Pi^0_1$ class, then there is an $e \in \omega$ such that $Q = \{x : U(e, x)\}$. $U$ is defined in terms of the Kleen $T$-predicate, so that essentially

$$U(e, x) \iff \Phi^x_e(0) \uparrow.$$ 

Define $\psi_3(e) = \{x : U(e, x)\}$ to obtain an effective numbering.

To see that this is a computable numbering, let

$$\sigma \in R_e \iff \Phi^\sigma_e(0) \uparrow.$$ 

so that $\psi_3(e) = [R_e]$ and the trees $R_e$ are uniformly primitive recursive.

**Numbering 4: The Halting Problem [53]**

Consider the mapping given by

$$\psi_4(e) = \{x : \Phi_e^x(e)(e) \uparrow\}.$$ 

This is a computable numbering, since $\psi_4(e) = [T_e]$, where

$$\sigma \in T_e \iff \Phi_e^\sigma(e) \uparrow.$$
For any computable tree $T$, choose $a$ so that $\Phi^a_\sigma(n)$ converges if and only if $\sigma \in T$. Then

$$\sigma \in T \iff \Phi^a_\sigma(a) \downarrow,$$

so that $[T] = \psi_4(a)$.

**Numbering 5: Total Computable Functions**

Here we will consider an effective, but not computable numbering $\psi$ based on computable trees. This numbering will be used in connection with string verifiable families of classes in Section 2.3.

Let $\psi_5(e) = [T_e]$, where

$$\sigma \in T_e \iff (\forall \tau \leq \sigma)[\phi_e(\langle \tau \rangle) \downarrow \phi_e(\langle \tau \rangle) = 1].$$

This enumeration is uniformly $\Pi^0_1$, but is not computable, since the relation $\phi_e(m) \downarrow$ is c.e. non-computable. Clearly each $\psi_5(e)$ is a $\Pi^0_1$ class. If $\phi_e$ is total and $T$ is a tree such that, for all $\sigma$, we have $\sigma \in T \iff \phi_e(\langle \sigma \rangle) = 1$, then $T_e = T$ and is a $\Pi^0_1$ class. Hence this enumeration has the crucial property that, for every computable tree $T$, there exists $e$ such that $T = T_e$.

### 2.2.2 Equivalence of the Numberings

In this section we show that the computable numberings of section 2.2.1 are mutually equivalent via a computable permutation. Each of these is equivalent to the effective enumeration of section 2.2.1 via a $\Delta^0_3$-permutation. We need the following proposition.

**Proposition 2.2.1.** (a) For each pair $i, j$ with $1 \leq i \leq 5$ and $1 \leq j \leq 4$, there is a computable function $f$ such that $\psi_j = \psi_i \circ f$.

(b) For each $j \leq 5$, there is a $\Delta^0_3$ function $f$ such that $\psi_5 = \psi_i \circ f$.

**Proof.** ($\psi_1 \leq \psi_2$): Use the $S - m - n$ Theorem to define $f$ so that

$$W_{f(e)} = \{n : \pi_e(n) \neq 1\}.$$

Then $\sigma \in U_e \iff \sigma \in S_{f(e)}$. 

22
(ψ₂ ≤ ψ₃): Define f so that, for all m,
\[ \Phi^{x}_{f(e)}(m) = (\text{least } n)\langle x \upharpoonright n \rangle \in W_e. \]
Then \( \psi_2(e) = \psi_3(f(e)) \).

(ψ₃ ≤ ψ₄): Define f so that \( \Phi^x_a(n) = \Phi^x(0) \) for all n. Then \( x \in \psi_3(e) \iff x \in \psi_4(f(e)) \).

(ψ₄ ≤ ψ₁): Recall that \( \psi_4(e) = [\{\sigma : \Phi^{x}_e(\sigma) \uparrow}\} \).
Define the primitive recursive function \( g \) so that for each \( e \),
\[ \pi_{f(e)}(\langle \sigma \rangle) = \begin{cases} 1, & \text{if } \Phi^{x}_e(\sigma) \uparrow; \\ 0, & \text{otherwise}. \end{cases} \]
Then \( \psi_1(e) = \psi_4(g(e)) \).

(ψ₁ ≤ ψ₅): Define the primitive recursive function f such that, for each \( e \),
\[ \phi_{f(e)}(\langle \sigma \rangle) = \begin{cases} 1, & \text{if } (\forall \tau \subseteq \sigma)\pi_e(\langle \tau \rangle) = 1, \\ 0, & \text{otherwise}. \end{cases} \]
Then \( \psi_1(e) = \psi_5(f(e)) \).

The rest of the proof follows by composition. □

**Theorem 2.2.2.** For any computable numbering φ which is computably equivalent to \( \psi_2 \), there is a computable permutation \( p \) such that \( \psi_2 = \phi \circ p \).

**Proof.** The proof is a modification of an argument due to Jockusch [86, p. 25]. Let \( \psi = \psi_2 \). By assumption, there are computable functions f and g such that \( \psi(f(e)) = \phi(e) \) and \( \phi(g(e)) = \psi(e) \). Since the numbering \( \psi_2 \) is based on an enumeration of the partial computable functions, we can ensure by padding that f is injective. To modify g into an injective function \( g_1 \), it is sufficient to effectively compute from each e an infinite set \( S_e \) of indices such that \( \phi(g(i)) = \phi(e) \) for all \( i \in S_e \). We proceed as follows. Let A and B be computably inseparable c.e. sets and define computable functions k and ℓ such that, for all
That is, we build a tree for $\psi(k(e,m))$ which exactly equals the tree for $\psi(e)$ for strings of length $s$ until $m \in B$, in which case no strings of length $s + 1$ are put into $\psi(k(e,m))$. To build the tree for $\psi(\ell(e,m))$, we put in all strings of length $s$ until $m \in A$, in which case we include only the strings of length $s + 1$ which are in $\psi(e)$.

Now let $C_e = \{k(e,m) : m \in A\}$ and $D_e = \{\ell(e,m) : m \in A\}$ and let $S_e = g(C_e \cup D_e)$. Then for $j = g(i) \in S_e$, it follows from the definition that $\psi(i) = \psi(e)$ and therefore $\varphi(g(i)) = \varphi(g(e))$. We will prove in two cases that either $g(C_e)$ is infinite or $g(D_e)$ is infinite.

**Case I:** Suppose that $\psi(e) \neq \emptyset$ and suppose by way of contradiction that $g(C_e)$ is finite. Then $S = \{m : g(k(e,m)) \in g(C_e)\}$ is a computable set. Now $A \subseteq g(C_e)$ by definition. On the other hand, if $j = g(k(e,m)) \in g(C_e)$ where $m \in A$, then $\varphi(j) = \psi(k(e,m)) = \phi(e) \neq \emptyset$. But for $m \in B$, $\varphi(g(k(e,m))) = \psi(k(e,m)) = \emptyset$, so that $S \cap B = \emptyset$. This contradicts the assumption that $A$ and $B$ are computably inseparable.

**Case II:** Suppose that $\psi(e) = \emptyset$. It follows as in Case I that $g(D_e)$ is infinite.

Thus we may assume without loss of generality that both $f$ and $g$ are one-to-one. Now define a sequence $\{e_n : n \in \omega\}$ and two partitions of $\omega$ as follows. Let $e_0 = 0$ and for each $n$, $e_{n+1}$ is the least $e$ such that $\varphi(e) \neq \varphi(e_i)$ for every $i \leq n$. Let $A_n = \{e : \psi(e) = \psi(e_n)\}$ and $B_n = \{e : \varphi(e) = \psi(e_n)\}$. Then $\omega = \bigcup_n A_n = \bigcup_n B_n$ and each sequence is pairwise disjoint. Furthermore, $f(B_n) \subseteq A_n$ and $g(A_n) \subseteq B_n$. The remainder of the proof follows as in the Myhill Isomorphism Theorem [86, p. 24; Also 5.8, p. 25].

A similar argument shows that if $\varphi$ is a $\Delta^0_3$ numbering of the $\Pi^0_1$ classes, then there is a $\Delta^0_3$ permutation $p$ with $\varphi = \psi_2 \circ p$. It follows that each of the computable numberings
\(\psi_1, \ldots, \psi_4\) are acceptable, that is, they occur in the greatest element of the semilattice \(L(\mathcal{P})\). In the section 2.2.4 we will see that minimal elements exist in the semilattice— that is, injective numberings.

First, however, we provide an alternate proof of Theorem 2.2.2.

2.2.3 Equivalence of the Numberings (Alternate Proof)

At the Third International Conference of Computability and Complexity in Analysis in Gainesville, Florida in 2006, I presented the argument for Theorem 2.2.2 as given above; it is a modification of an argument of Jockusch for the c.e. version. Pleased with the argument’s applicability to \(\Pi^0_1\) classes, I urged Robert Soare, who was present in the audience, to keep the Jockusch argument for the c.e. version in his new and upcoming book, Computability Theory and Applications [88]. If the alternate proof using the recursion theorem for the c.e. version of Theorem 2.2.2 [86, p. 43] could not also be modified to also prove Theorem 2.2.2, then he said he would. We show below that an alternate proof is possible, modifying the recursion theorem argument.

**Theorem 2.2.3.** For any computable numbering \(\varphi\) which is computably equivalent to \(\psi_2\), there is a computable permutation \(p\) such that \(\psi_2 = \varphi \circ p\).

**Alternate Proof.** We proceed, at first, as before. That is, there are computable functions \(f\) and \(g\) such that \(\psi(f(e)) = \varphi(e)\) and \(\varphi(g(e)) = \psi(e)\). Our goal is to modify \(g\) to obtain an injective function \(g_1\). Instead, however, we define \(g_1\) differently, using an auxiliary computable function \(h\) obtained by the Recursion Theorem. We ensure \(h\) satisfies, for all distinct \(i\) and \(j\), (i) \(gh(e, i) \neq gh(e, j)\) and (ii) \(\varphi h(e, i) = \phi_e\).

We now define \(g_1\), given \(h\) satisfying (i) and (ii). Define \(g_1(0) = gh(fg(0), 0)\). To define \(g_1(k + 1)\), note that (i) ensures infinitely many distinct \(gh(e, i)\) for each \(e\). Let \(a_0 = 0\) and \(a_{k+1}\) be the least integer \(i\) such that \(gh(fg(k + 1), i) > g_1(k)\). Define \(g_1(k + 1) = h(fg(k + 1), a_{k+1})\). To see that for all \(e\), \(\psi_{g_1(e)} = \phi_e\), fix \(e\) and note that \(\psi_{g_1(e)} = \psi_{gh(fg(e), a_e)} = \phi_{fg(e), a_e} = \phi_{fg(e)} = \psi_{g(e)} = \phi_e\). To see that \(g_1\) is injective, note that the definition ensures that for all \(k\), \(g_1(k + 1) > g_1(k)\).
We now define $h(e, \bullet)$ for each $e$, by induction, ensuring that (i) and (ii) are satisfied. Define $h(e, 0) = e$. To define $h(e, k + 1)$, use the Recursion Theorem to obtain an $n$ such that
\[
\phi_n(z) = \begin{cases} 
\phi_e(z) & \text{if } g(n) \neq gh(e, i) \ (\forall i \leq k) \\
\text{undefined} & \text{otherwise} 
\end{cases}
\]
Notice that if $g(n) = gh(e, i)$ then since $i \leq k$, by induction $\phi_e = \phi_{h(e, i)} = \psi_{gh(e, i)} = \psi_{g(n)} = \phi_n$, the undefined function. Hence $\phi_e = \phi_n$ in all instances. Since $\{ a : \phi_a = \phi_n \}$ is a productive [86, p. 43], let $p$ be a corresponding productive function. Define $W_{r(x)} = W_x \cup \{ p(x) \}$ and note that each $W_{r^i(n)}$ is a distinct c.e. subset of $A$. Consequently each $\psi_{gr^i(n)} = \phi_{r^i(n)}$ is a distinct partial computable function. Let $n_k$ be the least $i$ such that $r^i(n) \neq h(e, j)$ for all $j \leq k$. Define
\[
h(e, k + 1) = \begin{cases} 
n & \text{if } g(n) \neq gh(e, i) \ (\forall i \leq k) \\
r^{n_k}(n) & \text{otherwise} 
\end{cases}
\]
To see that (i) holds ($gh(e, i) \neq gh(e, j)$ for distinct $i, j$), note that $\psi_{g(n)} = \phi_e \neq \phi_{r^i(n)} = \psi_{gr^i(n)}$ for all $i \geq 1$. Therefore if $g(n) \neq g(e)$ then for all $j$, $h(e, j) \in \{ e \} \cup \{ r^i(n) \}_{i \geq 1}$. Otherwise $h(e, j) \in \{ e \} \cup \{ r^i(n) \}_{i \geq 0}$. In either case, for all distinct $i, j$, $gh(e, i) \neq gh(e, j)$.

To see that (ii) holds ($\phi_{h(e, i)} = \phi_e$ for all $i$), note that $\phi_e = \phi_n$, $h(e, i) \in \{ e \} \cup \{ r^i(n) \}_{i \geq 0 \text{ or } 1}$, and $r^i(n) \in \{ a : \phi_a = \phi_n \}$. So $\phi_e = \phi_{h(e, i)}$ for all $i$. \qed

### 2.2.4 Injective Computable Numberings

In this section, we modify Friedberg’s original argument for injective computable numbering of the c.e. sets, to provide different numbering results needed for $\Pi_0^1$ classes. In Section 2.2.4.1, we present Friedberg’s original argument. In Section 2.2.4.2, we provide an injective computable numbering of disjoint pairs of c.e. sets; this in needed later in Section 2.4.1 in order to provide an injective computable numbering of the homogeneous $\Pi_0^1$ classes. Finally, in Section 2.2.4.3, we construct a computable injective numbering of the $\Pi_0^1$ classes in $2^\omega$; we also provide other injective numbering results for $\Pi_0^1$ classes.
2.2.4.1 Original c.e. sets argument

The following is the original Friedberg argument, with slight modifications in the notation and presentation.

**Theorem 2.2.4** ([46]). There is an injective computable numbering of all c.e. sets.

**Proof.** Let \( \{W_e \}_{e \in \omega} \) be the standard numbering of the c.e. sets. We will construct a sequence of c.e. sets \( \{Y_e \}_{e \in \omega} \) in stages so that \( e \mapsto Y_e \) will be the desired injective numbering. In the construction we will use the notion of one \( Y \)-index \( i \) following a \( W \)-index \( e \) with the idea that in the end \( Y_i \) will equal \( W_e \). At some point, however, we may decide that \( i \) will no longer follow \( e \) again and we will say that \( i \) is released. If \( i \) is never released from following \( e \) then it is said to be a loyal follower and otherwise it is disloyal. Once released, an index remains free and is never again the follower of any \( e \). At any particular stage, any \( Y \)-index that is not following any \( W \)-index is said to be free. Any nonzero \( Y \)-index that has never followed any \( W \)-index is said to be unused.

To ensure that no c.e. set is excluded from the \( Y \)-sets, we will ensure that each \( W_e \) is infinitely often given the opportunity to be followed. To do this, at stage \( s = \langle n_s, e_s \rangle \) all actions in the construction will be taken with respect to \( W_{e_s} \). Assume without loss of generality that \( Y_0 = \emptyset \).

**Construction:** There are three possible cases at each stage \( s \).

**Case 1:** If \( i \) follows \( e_s \) & \( W_{e_s} \cap [0, i - 1] = W_{e_s, s} \cap [0, i - 1] \) (some \( e < e_s \)), release \( i \) and go on to stage \( s + 1 \).

**Case 2:** Suppose Case 1 does not occur. If \( W_{e_s, s} = Y_{i, s-1} \), and either \( i \) follows some \( e < e_s \), \( i = 0 \), \( i \) is free and \( i \leq e_s \), or \( i \) is free and \( i \) was previously displaced (see Case 3) by \( e_s \) and released, then go on to stage \( s + 1 \) without taking any action.

**Case 3:** Suppose that Cases 1 and 2 do not occur. Now ensure that \( e_s \) has a follower. If it does not, choose the lowest unused \( i \neq 0 \) to follow \( e_s \). Now let \( Y_{i, s} = W_{e_s, s} \).

For each \( j \neq i \) such that \( Y_{j, s-1} = W_{e_s, s} \) (in increasing order of \( j \)) put the lowest \( b \) not yet in any \( W \) or \( Y \) into \( Y_j \) and release \( j \) if it is a follower. We say \( e_s \) displaces \( j \).

**Verification:** Given \( e \in \omega \), let \( \hat{e} \) be the least \( k \) so that \( W_k = W_e \). We will show:
(i) \((\forall e)(\exists i) Y_i = W_{\hat{e}}\)

(ii) \(i \neq j \rightarrow Y_i \text{ and } Y_j \text{ are not the same finite sets}\)

(iii) \(i \neq j \rightarrow Y_i \text{ and } Y_j \text{ are not the same infinite sets}\)

**Verification of (i).** Fix \(e\). First note that although \(\hat{e}\) can only have one follower at any particular stage, it cannot have an infinite number of them which are each released at some stage. For example, if \(s\) and \(x\) are sufficiently large, then for all \(j < \hat{e}\), \(W_{j,s} \cap [0, x] \neq W_{\hat{e},s} \cap [0, x]\). Hence release can only occur in Case 1 a finite number of times. Furthermore, Case 2 ensures that release in Case 3 can only occur for any \(s\) when \(e_s < \hat{e}\). Therefore, by the above, if \(i > x\) is follower of \(\hat{e}\) and \(t > s\), then \(Y_{i,t-1} = W_{\hat{e},t-1} \neq W_{e_s,t}\). Hence \(i\) will not be released in Case 3. Therefore release can only occur in Case 3 a finite number of times.

Now let \(s\) be a stage where \(\hat{e}\) never loses a follower. If Case 3 occurs infinitely often after stage \(s\) for \(\hat{e}\), then it has a permanent follower \(i\) so that \(Y_i = W_{\hat{e}}\). Therefore assume Case 3 occurs only finitely often. Since \(\hat{e}\) never loses a follower, Case 1 cannot occur. Hence Case 2 must occur infinitely often. However there are only a finite number of \(i\) such that the ‘If’ clause holds with “\(W_{\hat{e},s} = Y_{i,s-1}\)” in Case 2. (For example, \(\{i : i = 0, i \leq \hat{e}, \text{ or } i \text{ is displaced by } \hat{e}\}\) is a finite set since only a finite number of \(i\) are displaced due to Case 3 occurring only a finite number of times. To see also that \(\{i : i \text{ for some } s, i \text{ follows some } a < \hat{e} \text{ and } "W_{\hat{e},s} = Y_{i,s-1}" \text{ holds}\}\) is finite, note that by the definition of \(\hat{e}\), if \(a < \hat{e}\) then \(W_a \neq W_{\hat{e}}\). So for sufficiently large \(t\), if \(i\) follows \(a\) then \(W_{\hat{e},t} \neq Y_{i,t-1}\).) Therefore Case 2 occurring infinitely often, together with only finite number of \(i\) such that the ‘If’ clause holds with “\(W_{\hat{e},s} = Y_{i,s-1}\)”, implies that there is a single \(i\) such that \(Y_{i,s-1} = W_{\hat{e},s}\) for infinitely many \(s\). Thus \(Y_i = W_{\hat{e}}\).

**Verification of (ii).** First note that at any stage \(s\), if \(W_{e_s} = \emptyset\) then Case 2 ensures that Case 3 will not be reached so that \(e_s\) never receives a follower. Furthermore each \(k \neq 0\) is chosen at some stage \(s\) in Case 3 to follow some \(e\) and from the previous comments, \(Y_{k,s} \neq \emptyset\). Immediately thereafter, \(Y_{k,s}\) is ensured to be distinct from all other \(Y_{\ell,s} (\ell \neq k)\). This also continues to be true at all subsequent stages \(t\) by Case 3.
Now suppose \( i \neq j \). Since we are supposing both \( Y_i \) and \( Y_j \) are finite, there is some \( t \) such that for all \( s > t \), \( Y_{i,s} = Y_i \) and \( Y_{j,s} = Y_j \). As mentioned above \( Y_{i,s} \neq Y_{j,s} \) and therefore \( Y_i \neq Y_j \) as required.

**Verification of (iii).** Assume both \( Y_i \) and \( Y_j \) are infinite and \( i \neq j \). Now \( i \) must eventually follow some \( W \)-index. If \( i \) is disloyal, then after it is released \( Y_i \) can acquire a new member only when \( i \) is displaced. However \( i \) can be displaced only once by each \( e < i \) and never by any \( e \geq i \). It follows that if \( i \) is released then \( Y_i \) only acquires a finite number of elements thereafter, contradicting the fact that it is infinite. This argument shows that both \( i \) and \( j \) are never released. We say that \( i \) and \( j \) are *loyal* followers.

Suppose now that \( i \) and \( j \) loyally follow \( i' \) and \( j' \), respectively. Then \( Y_i = W_{i'} \) and \( Y_j = W_{j'} \). Now \( i' \neq j' \) since a single \( W \)-index cannot have more than one loyal follower. Assume without loss of generality that \( i' < j' \). If \( W_{i'} = W_{j'} \), then for all sufficiently large \( s \), \( W_{i',s} \cap [0, j-1] = W_{j',s} \cap [0, j-1] \) so that Case 1 releases \( j \), a contradiction. Therefore \( Y_i = W_{i'} \neq W_{j'} = Y_j \) as required.

### 2.2.4.2 Ordered tuples of disjoint c.e. sets

In this section we modify the argument of Theorem 2.2.4 to obtain an injective computable numbering of all ordered tuples of disjoint c.e. sets.

**Theorem 2.2.5.** There is an injective computable numbering of all ordered tuples of disjoint c.e. sets.

**Proof.** Let \( \{S_e\}_{e \in \omega} \) be the standard numbering of the c.e. sets. Then \( \langle e, i \rangle \mapsto (S_e, S_i) \) is a computable numbering of all tuples of c.e. sets. Obtain a computable numbering \( \langle e, i \rangle \mapsto (W_e, W_i) \) of all tuples of disjoint c.e. sets as follows. At stage \( s + 1 \), if \( a \in S_{e,s+1} \setminus S_e \) then put \( a \) into \( W_{e,s+1} \) if \( a \not\in W_{i,s} \). If \( b \in S_{i,s+1} \setminus S_i \) then put \( b \) into \( W_{i,s+1} \) if \( b \not\in W_{e,s+1} \). Note that if originally \( S_e \cap S_i = \emptyset \) then \( S_e = W_e \) and \( S_i = W_i \).

Similar to Theorem 2.2.4 (in construction and terminology), we will construct in stages a sequence of pairs of c.e. sets \( \langle Y_e, Y_i \rangle_{\langle e, i \rangle \in \omega} \) so that \( \langle e, i \rangle \mapsto (Y_e, Y_i) \) will be the desired injective numbering. So that no pair is excluded from the \( Y \)-sets, we will ensure that each \( \langle e, i \rangle \) is infinitely often given the opportunity to be followed. To do this, at stage
that release in Case 3 can only occur for any release can only occur in Case 1 a finite number of times. Furthermore, Case 2 ensures $W_s$ put the lowest $W_{\langle e,i \rangle}$ follow some $W_e$ was previously displaced (see Case 3) by $\langle e,i \rangle$, then go on to stage $s + 1$ without taking any action.

Case 3: Suppose Cases 1 and 2 do not occur. Now ensure that $\langle e,i \rangle$ has a follower.

If it does not, choose the least unused $\langle e,i \rangle \neq 0$ to follow $\langle e,i \rangle$. Now let $(Y_{e,s}, Y_{i,s}) = (W_{e,s}, W_{i,s}).$

For each $\langle e,\ell \rangle \neq \langle e,i \rangle$ such that $(Y_{e,s-1}, Y_{i,s-1}) = (W_{e,s}, W_{i,s})$ (in increasing order of $\langle e,\ell \rangle$) put the lowest $b$ not yet in any $W$ or $Y$ into $Y_e$ and release $\langle e,\ell \rangle$ if it is a follower.

We say $\langle e,i \rangle$ displaces $\langle e,\ell \rangle$.

Verification: Given $\langle e,i \rangle \in \omega$, let $\langle e,i \rangle$ be the least $\langle k,\ell \rangle$ so that $(W_k, W_\ell) = (W_e, W_i)$.

We will show:

(i) $\forall \langle e,i \rangle \exists \langle k,\ell \rangle \ (Y_k, Y_\ell) = (W_e, W_i)$

(ii) $[\langle e,i \rangle \neq \langle e,\ell \rangle \ & Y_i = Y_\ell] \rightarrow Y_e$ and $Y_\ell$ are not the same finite sets

(iii) $[\langle e,i \rangle \neq \langle e,\ell \rangle \ & Y_i = Y_\ell] \rightarrow Y_e$ and $Y_\ell$ are not the same infinite sets

(iv) $\forall \langle e,\ell \rangle \ Y_e \cap Y_\ell = \emptyset$

Verification of (i). First note that although $\langle e,\ell \rangle$ can only have one follower at any particular stage, it cannot have an infinite number of them which are each released at some stage. For example, if $s$ and $\langle x,y \rangle$ are sufficiently large then for all $\langle k,\ell \rangle < \langle e,\ell \rangle$, either $W_{k,s} \cap [0, \langle x,y \rangle] \neq W_{e,s} \cap [0, \langle x,y \rangle]$ or $W_{i,s} \cap [0, \langle x,y \rangle] \neq W_{i,s} \cap [0, \langle x,y \rangle]$. Hence release can only occur in Case 1 a finite number of times. Furthermore, Case 2 ensures that release in Case 3 can only occur for any $s$ when $\langle e,i \rangle < \langle e,\ell \rangle$. Therefore, by the
above, if \((a, b) > (x, y)\) is follower of \((e, i)\) and \(t > s\), then \(Y_{a,t-1} = W_{e,t-1} \neq W_{e,s,t}\) or \(Y_{b,t-1} = W_{i,t-1} \neq W_{i,s,t}\). Hence \((a, b)\) will not be released in Case 3. Therefore release can only occur in Case 3 a finite number of times.

Let \(s\) be a stage where \((e, i)\) never loses a follower. If Case 3 occurs infinitely often after stage \(s\) for \((e, i)\), then it has a permanent follower \((e, i)\) so that \((Y_e, Y_i) = (W_e, W_i)\). Therefore assume Case 3 occurs only finitely often. Since \((e, i)\) never loses a follower, Case 1 cannot occur. Hence Case 2 must occur infinitely often.

There are only a finite number of \((e, i)\) such that the ‘If’ clause holds with the equality \(\Lambda(e, i, e, i, s)\) given by “\((Y_{e,s-1}, Y_{i,s-1}) = (W_{e,s}, W_{i,s})\)” in Case 2. For example, \(\{\langle e, i \rangle : \langle e, i \rangle = 0, \langle e, i \rangle \leq \langle e, i \rangle, \text{ or } \langle e, i \rangle \text{ is displaced by } \langle e, i \rangle\}\) is a finite set since only a finite number of \((e, i)\) are displaced due to Case 3 occurring only a finite number of times.

To see also that \(\{\langle e, i \rangle : \text{for some } s, \langle e, i \rangle \text{ follows some } (a, b) < \langle e, i \rangle \text{ and } \Lambda(e, i, e, i, s)\}\) is finite, note that by the definition of \((e, i)\), if \((a, b) < \langle e, i \rangle\), then either \(W_a \neq W_e\) or \(W_b \neq W_i\). So for sufficiently large \(t\), if \((e, i)\) follows \((a, b)\) then \((Y_{e,t-1}, Y_{i,t-1}) \neq (W_{e,t}, W_{i,t})\) and so \(\Lambda(e, i, e, i, t)\) does not hold.

Now Case 2 occurring infinitely often, together with only finite number of \((e, i)\) such that the ‘If’ clause holds with \(\Lambda\), implies that there is a single \((e, i)\) such that \((Y_{e,s-1}, Y_{i,s-1}) = (W_{e,s}, W_{i,s})\) for infinitely many \(s\). Thus \((Y_e, Y_i) = (W_e, W_i)\).

**Verification of (ii).** First note that at any stage \(s\), if \((W_{e,s}, W_{i,s}) = (\emptyset, \emptyset)\) then Case 2 ensures that Case 3 will not be reached so that \((e_s, i_s)\) never receives a follower. Furthermore each \((a, b) \neq (0, 0)\) is chosen at some stage \(s\) in Case 3 to follow some \((e, i)\) and from the previous comments, \((Y_{a,s}, Y_{b,s}) \neq (\emptyset, \emptyset)\). Immediately thereafter, \((Y_{a,s}, Y_{b,s})\) is ensured to be distinct from all other \((Y_{k,s}, Y_{l,s})\) (\((a, b) \neq (k, l)\)). This also continues to be true at all subsequent stages \(t\) by Case 3.

Now suppose \((e, i) \neq (e, i), Y_i = Y_e\), and both \(Y_e\) and \(Y_i\) are finite. There are two cases. Suppose first that both \(Y_i\) and \(Y_e\) are finite. Now there is some \(t\) such that for all \(s > t\), \(Y_{k,s} = Y_k\ (k \in \{e, i, e, i\})\). As shown above \((Y_{e,s}, Y_{i,s}) \neq (Y_{e,s}, Y_{i,s})\) and therefore \((Y_e, Y_i) \neq (Y_e, Y_i)\). Since \(Y_i = Y_e\), it follows that \(Y_e \neq Y_e\).
Suppose now that $Y_i$ and $Y'_i$ are both infinite. Now $\langle e, i \rangle$ must eventually follow some $\langle e', i' \rangle$. If $\langle e, i \rangle$ is ever released then it is free. Thereafter only the first coordinate of $(Y_e, Y_i)$ acquires members so that $Y_i$ is finite, a contradiction. A similar argument holds for $\langle e, \ell \rangle$. Therefore both $\langle e, i \rangle$ and $\langle e, \ell \rangle$ are never released and are *loyal* followers. Suppose that they follow $\langle e', i' \rangle$ and $\langle e', \ell' \rangle$, respectively. Then $(Y_e, Y_i) = (W_{e'}, W_{i'})$ and $(Y_e, Y_i) = (W_e, W_{i'})$. Note that $\langle e', i' \rangle \neq \langle e', \ell' \rangle$ since a single $W$-index cannot have more than one loyal follower. Assume without loss of generality that $\langle e', i' \rangle < \langle e', \ell' \rangle$.

Suppose now that $W_{e'} = W_{e'}$. By assumption, $W_{i'} = Y_i = Y_l = W_{i'}$, Therefore, for all sufficiently large $s$, $W_{e', s} \cap [0, \langle \epsilon, \ell \rangle - 1] = W_{e', s} \cap [0, \langle \epsilon, \ell \rangle - 1]$ and $W_{i', s} \cap [0, \langle \epsilon, \ell \rangle - 1] = W_{i', s} \cap [0, \langle \epsilon, \ell \rangle - 1]$, so that Case 1 releases $\langle \epsilon, \ell \rangle$, a contradiction. Therefore $W_{e'} \neq W_{e'}$ so that also $Y_e = W_{e'} \neq W_{e'} = Y_e$, as required.

**Verification of (iii).** Suppose that $\langle e, i \rangle \neq \langle e, \ell \rangle$, $Y_i = Y_l$, and both $Y_e$ and $Y_{\epsilon}$ are infinite. Now $\langle e, i \rangle$ must eventually follow some $\langle e', i' \rangle$. If $\langle e, i \rangle$ is disloyal, then after it is released $Y_e$ can acquire a new member only when $\langle e, i \rangle$ is displaced. However $\langle e, i \rangle$ can only be displaced once by each $\langle c, d \rangle < \langle e, i \rangle$ and never by any $\langle c, d \rangle \geq \langle e, i \rangle$. It follows that if $\langle e, i \rangle$ is released then $Y_e$ only acquires a finite number of elements thereafter, contradicting the fact that it is infinite. This argument shows that both $\langle e, i \rangle$ and $\langle e, \ell \rangle$ are never released and are therefore loyal followers. Now apply the same argument given in the later part of the verification of (ii) to get that $Y_e \neq Y_{\epsilon}$.

**Verification of (iv).** If $\langle e, i \rangle$ is a loyal follower of some $\langle e', i' \rangle$, then $(Y_e, Y_i) = (W_{e'}, W_{i'})$. Therefore since $\langle e, i \rangle \mapsto (W_e, W_i)$ is an enumeration of disjoint sets, it follows that $Y_e$ and $Y_i$ are disjoint. Otherwise suppose that $\langle e, i \rangle$ is released at some stage $s$. Then $Y_{e,s} \cap Y_{i,s} \cap = \emptyset$ and thereafter only $Y_e$ can acquire new elements not already included in $Y_{i,s}$. Therefore disjointness is preserved and $Y_e \cap Y_i = \emptyset$. 

2.2.4.3 **Results for effectively closed sets**

In this section, we modify the Friedberg argument of Section 2.2.4.1 to construct a computable injective numbering of the $\Pi^0_1$ classes in $2^\omega$. An alternative proof was sketched by Raichev [79]. Afterwards we provide other injective numbering results for $\Pi^0_1$ classes.
Theorem 2.2.6. There is an injective computable numbering of all $\Pi_1^0$ classes in $2^\omega$.

Proof. Let $\{W_e\}_{e \in \omega}$ be the computable enumeration of the nonempty c.e. subsets of $2^{<\omega}$. We will construct a computable numbering $\{Y_e : e \in \omega\}$ in stages $Y_{e,s}$ of a family of c.e. subsets of $2^{<\omega}$ so that $\{O(Y_e)\}_{e \in \omega}$ is an injective numbering of the $\Sigma_1^0$ classes; our construction and terminology will be similar to Theorem 2.2.4.

It is important to note that an injective numbering of the c.e. subsets of $2^{<\omega}$ will not automatically yield an injective numbering of the $\Sigma_1^0$ classes, since each $\Sigma_1^0$ class will equal $O(W)$ for many different c.e. sets $W$. However, if $O(V) \neq O(W)$ for two c.e. sets $V$ and $W$, then there must be some interval $I(\sigma)$ which is included in, say $O(V)$ but not included in $O(W)$ and hence some stage $s$ such that $O(V_s) \upharpoonright s \neq O(W_s) \upharpoonright s$ at stage $s$ and at any later stage.

To ensure that no c.e. set is excluded from the $Y$-sets, we will ensure that each $W_e$ is infinitely often given the opportunity to be followed. To do this, at stage $s = \langle n, e_s \rangle$ all actions in the construction will be taken with respect to $W_{e_s}$. At each stage $s$, we initiate at most one new $Y_i$, so that after stage $s$, we have sets $Y_0, Y_1, \ldots, Y_{k_s}$ for some $k_s \leq s$. Fix $Y_0 = \emptyset$ and $Y_1 = \{\emptyset\}$ so that $O(Y_0) = \emptyset$ and $O(Y_1) = 2^\omega$, respectively.

Construction: There are three possible cases at each stage $s$.

Case 1: If $i$ follows $e_s$ and there exists $e < e_s$ such that $O(W_{e,s}) \upharpoonright (i-1) = O(W_{e_s,s}) \upharpoonright (i-1)$, then release $i$ and go on to stage $s + 1$.

Case 2: Suppose that Case 1 does not occur. If $O(W_{e,s}) = O(Y_{i,s-1})$, and either $i$ follows some $e < e_s$, $i = 0$, $i = 1$, or $i$ is free and either $i \leq e_s$ or $i$ was previously displaced (see Case 3) by $e_s$, then go on to stage $s + 1$ without taking any action.

Case 3: Suppose that Cases 1 and 2 do not occur. Now ensure $e_s$ has a follower. If it does not, choose the least unused $i \neq 0, 1$ to follow $e_s$. Now let $Y_{i,s} = W_{e_s,s}$.

If $Y_{j,s-1}$ for some $j \neq i$ satisfies $O(Y_{j,s-1}) = O(W_{e_s,s})$, then put some $\sigma_j \in 2^{<\omega}$, defined in what follows, into $Y_j$ so that $O(Y_{j,s}) \neq O(W_{e_s,s})$.

Let $E_s = \{j \in \omega : j \neq i \& \; O(Y_{j,s-1}) = O(W_{e_s,s})\}$ be the set of indices of equivalent $Y$-open sets and suppose that $E_s = \{\epsilon_1 < \epsilon_2 < \ldots < \epsilon_{|E_s|}\}$. Now define $\text{Str}(k, s) = \{\sigma \in$
$2^k : \sigma \notin \mathcal{O}(W_{e,s}) \upharpoonright k \}$. Let $\ell(s)$ be the least $k$ such that $|\text{Str}(k,s)| > |E_s|$. Then $\ell(s)$ is the least level of $\mathcal{O}(W_{e,s})$ where there is enough room to give each equivalent $Y_{j,s-1}$ an additional string to distinguish $\mathcal{O}(Y_{j,s-1})$ from $\mathcal{O}(W_{e,s})$. (Notice that $\mathcal{O}(W_{e,s}) \neq 2^\omega$ by Case 2.) Suppose that $\text{Str}(\ell(s), s) = \{ \sigma_1 \prec \sigma_2 \prec \ldots \prec \sigma_{|\text{Str}(\ell(s), s)|} \}$. Now put $\sigma_j$ into $Y_{\ell_j}$ and release $\epsilon_j$ if it is a follower. We say that $\epsilon_j$ is displaced at stage $s$.

**Verification:** Given $e \in \omega$, let $\hat{e}$ be the least $k$ such that $[\mathcal{O}(W_k) = \mathcal{O}(W_e)]$. We will show:

(i) $(\forall e)(\exists i) \ Y_i = W_{\hat{e}}$;

(ii) $i \neq j$ implies that $\mathcal{O}(Y_i)$ and $\mathcal{O}(Y_j)$ are not equal when both are clopen.

(iii) $i \neq j$ implies that $\mathcal{O}(Y_i)$ and $\mathcal{O}(Y_j)$ are not equal when both are not clopen.

**Verification of (i).** Fix $e$. First note that although $\hat{e}$ can have different followers at different stages, it cannot have an infinite number of disloyal followers. That is, if $s$ and $x$ are sufficiently large, then by the definition of $\hat{e}$, for all $j < \hat{e}$, $\mathcal{O}(W_{j,s}) \upharpoonright x \neq \mathcal{O}(W_{\hat{e},s}) \upharpoonright x$. Hence release can only occur in Case 1 a finite number of times. Furthermore, Case 2 ensures that release in Case 3 can only occur for any $s$ when $e_s < \hat{e}$. Therefore, by the above, if $i > x$ is follower of $\hat{e}$ and $t > s$, then $\mathcal{O}(Y_{i,t-1}) = \mathcal{O}(W_{\hat{e},t-1}) \neq \mathcal{O}(W_{e_s,t})$. Hence $i$ will not be released in Case 3. Therefore release can only occur in Case 3 a finite number of times.

Now let $s$ be a stage after which $\hat{e}$ never loses a follower. If Case 3 occurs infinitely often after stage $s$ for $\hat{e}$, then it has a permanent follower $i$ so that $\mathcal{O}(Y_i) = \mathcal{O}(W_{\hat{e}})$. Therefore assume Case 3 occurs only finitely often. Since $\hat{e}$ never loses a follower, Case 1 cannot occur. Thus Case 2 must occur infinitely often. However there are only a finite number of $i$ such that the hypothesis of (ii) holds with $\mathcal{O}(W_{\hat{e},s}) = \mathcal{O}(Y_{i,s-1})$. To see this, consider the three cases. First, each $e < e_s$ has only finitely many followers by the argument above; second, there are only finitely many $i \leq e_s$; and third, only a finite number of $i$ are displaced by $e_s$, due to Case 3 occurring only a finite number of times. This contradiction shows that $\hat{e}$ has a permanent follower, as desired.
Verification of (ii). Suppose that \( U = \mathcal{O}(Y_i) = \mathcal{O}(Y_j) (\neq \emptyset, 2^\omega) \) is clopen and let \( U = \mathcal{O}(W_{\hat{e}}) \). It follows from compactness, that there is some finite \( s \) such that, for all \( t \geq s \), \( \mathcal{O}(Y_{i,t}) \upharpoonright t \mathcal{O}(Y_{j,t}) \upharpoonright t = \mathcal{O}(Y_i) \). It follows from the verification of (i) above that there is a stage \( t > s \) such that Case 3 applies to \( \hat{e} \). But then at least one of \( \mathcal{O}(Y_i), \mathcal{O}(Y_j) \) must change at stage \( t \). This contradiction verifies (ii).

Verification of (iii). Assume both \( \mathcal{O}(Y_i) \) and \( \mathcal{O}(Y_j) \) are not clopen and \( i \neq j \). It follows that \( \mathcal{O}(Y_i) \) must change infinitely often, since of course \( \mathcal{O}(Y_{i,s}) \) is clopen for each \( s \), and similarly for \( \mathcal{O}(Y_j) \). Now \( i \) must eventually follow some \( W \)-index. If \( i \) is ever released, then it is free. Thereafter \( Y_i \) acquires members in Case 3 at stage \( s \) only when \( W_{e,s} = Y_{i,s-1} \).

This implies that Case 2 does not apply at stage \( s \) and thus \( e_s < j \). But each \( e < i \) can only displace \( i \) once, again by the hypothesis of Case 2. Thus if \( i \) is a disloyal follower, then in fact \( \mathcal{O}(Y_i) \) is clopen. Thus we may assume that \( i \) is a loyal follower of \( e \) and \( j \) is a loyal follower of \( e' \). Then \( \mathcal{O}(W_e) = \mathcal{O}(W_{e'}) \) but \( e \neq e' \), since each \( e \) can have at most one loyal follower. Without loss of generality suppose \( e < e' \).

Since \( \mathcal{O}(W_e) = \mathcal{O}(W_{e'}) \), there will be a stage \( s \) large enough so that \( \mathcal{O}(W_e) \upharpoonright (i-1) = \mathcal{O}(W_{e'}) \upharpoonright (i-1) \). Then since \( i \) follows \( e < e' \), \( i \) will be released at stage \( s \), contradicting the assumption that \( i \) is a loyal follower.

This verification completes the proof.

The problem of finding an injective enumeration of the \( \Pi^0_1 \) classes in \( \omega^\omega \) remains. For classes in \( 2^\omega \), we have the following generalization of Theorem 2.2.6. It will be useful later. Let \( C \) be the family of clopen subsets of \( 2^\omega \).

**Theorem 2.2.7.** For any family \( \mathcal{F} \) of \( \Pi^0_1 \) classes in \( 2^\omega \) which has a computable numbering, there is a 1-1 computable numbering of \( C \cup \mathcal{F} \).

**Proof.** Let the computable enumeration \( P_e \) be given. We may assume that \( C \subseteq \mathcal{F} \) by simply enumerating the clopen sets as \( \{Q_{2e} : e < \omega \} \) and letting \( Q_{2e+1} = P_e \). Then the proof of Theorem 2.2.6 produces a 1-1 computable enumeration of \( \mathcal{F} \) as desired.

We have the following corollary from the proof of Theorem 2.2.6.
Corollary 2.2.8. There is an effective numbering of the \( \Pi_1^0 \) classes based on the total computable functions

**Proof.** Modify each c.e. set in the standard numbering to enumerate an element only as long as it is larger than any previously enumerated element. Applying Friedberg’s argument to this class of c.e. sets yields an effective injective numbering \( e \mapsto C_e \) of the computable sets [91]. Furthermore each \( C_e \) still enumerates its elements in increasing order.

Now suppose \( \{\chi_e\}_{e \in \omega} \) is a corresponding set of characteristic functions. One characterization of a \( \Pi_1^0 \) class \( P \) is that \( P = \omega^\omega \setminus \mathcal{O}(W) \) for some computable set \( W \) [20]. As a result, \( e \mapsto \omega^\omega \setminus \mathcal{O}(C_e) = \omega^\omega \setminus \mathcal{O}(\{n : \chi_e(n) = 1\}) \) is an alternative effective numbering based on total computable functions (replacing noneffective Numbering 2).

It is known, for fixed \( n > 0 \), that there is a effective injective numbering of the \( n \)-c.e. sets [50].

**Conjecture 2.2.9.** For each \( n \), there is a numbering \( e \mapsto N_e \) of \( n \)-c.e. sets such that there is an injective computable numbering \( e \mapsto \omega^\omega \setminus \mathcal{O}(N_e) \) of all closed sets of this form.

For \( n = 1 \) the conjecture is given by Theorem 2.2.6.

We next show that Theorem 2.2.6 is not obtainable by any computable procedure that uniformly selects the minimal index of every \( \Pi_1^0 \) class.

**Theorem 2.2.10.** There is no computable choice function for indices of \( \Pi_1^0 \) classes. (i.e. a computable function \( f \) such that \( f(e) \) is an index of \( P_e \) and \( P_i = P_e \Rightarrow f(i) = f(e) \))

**Proof.** Suppose that \( f \) exists. Let \( a_0, a_1, \ldots \) be an enumeration of a noncomputable c.e. set \( A \). Define a computable function \( g \) and trees \( T_{g(e)} \) so that if \( |\sigma| = n \), then

\[
\sigma \in T_{g(e)} \iff e \notin \{a_0, \ldots, a_n\}.
\]

Then

\[
P_{g(e)} = \begin{cases} 
\emptyset & \text{if } e \in A \\
2^\omega & \text{otherwise}
\end{cases}
\]

For any \( a \in A, e \in A \leftrightarrow f(g(e)) = f(g(a)) \), making \( A \) computable.
It is still possible, however, that some interesting proper family of $\Pi_1^0$ classes may be enumerated by selecting minimal indices from the enumeration of all $\Pi_1^0$ classes.

### 2.3 String Verifiable Families of $\Pi_1^0$ Classes

In this section we examine families of classes which are deemed to be string verifiable (e.g. decidable or homogeneous classes) or strongly string verifiable (e.g. strongly decidable or strongly homogeneous classes). Any string verifiable family has an effective numbering and any strongly string verifiable family has a computable numbering.

#### 2.3.1 Definition and Examples

First we will define the notions of string verifiable and strongly string verifiable. We also give some examples.

**Notation 2.3.1.** Let $F_0, F_1, \ldots$ be a computable enumeration of the finite subsets of $2^{<\omega}$, that is, for any $n, \sigma \in F_n \iff b_n(\langle \sigma \rangle) = 1$, where $b_n$ is the binary expression for the natural number $n$. Let $E$ denote the family of finite sequences of positive integers of even length. Let $P_0 = [T_0], P_1 = [T_1], \ldots$ be some computable enumeration of the $\Pi_1^0$ classes in $2^{\omega}$.

**Definition 2.3.2** (String Verifiability). (i) A string function is a computable function $f : 2^{<\omega} \rightarrow E$.

(ii) A family $\mathcal{H}$ of trees (or, more generally, of subsets of $2^{<\omega}$) is string verifiable if there is a string function $h : 2^{<\omega} \rightarrow E$ so that for all $T, T \in \mathcal{H}$ if and only if the following condition is satisfied for all $\sigma \in T$, where $h(\sigma) = (m_1, m_2, \ldots, m_{2n})$ and $D_i = F_{m_i}$ for $i = 1, \ldots, n$: There exists $i < n$ such that $D_{2i+1} \subseteq T$ and $T \cap D_{2i+2} = \emptyset$ – that is, $[T] \in S(D_{2i+1}, D_{2i+2})$ (the family of separating sets of $D_{2i+1}$ and $D_{2i+2}$).

**Remark 2.3.3.** Note that the family of trees itself is string verifiable among the family of all subsets of $2^{<\omega}$, via the function $h(\sigma) = (a, b)$, where $F_a = \{ \tau : \tau \sqsupseteq \sigma \}$ and $F_b = \emptyset$.

**Example 2.3.1.** (a) The Homogeneous Trees. A tree $T$ is said to be homogeneous if $(\forall \sigma, \tau \in T)[|\sigma| = |\tau| \Rightarrow (\forall i)(\sigma \upharpoonright i \in T \iff \tau \upharpoonright i \in T)].$
Define the string verification function \( h \) as follows. Let \( A_1, A_2, A_3, A_4 \) enumerate \( \mathcal{P}(\{0, 1\}^{<\omega}) \) and let \( B_1, \ldots, B_{2^|\sigma|} \) enumerate the strings of length \( \sigma \). Let \( h(\sigma) \) enumerate in order the set of \( m_{2(j,k)+1} \) and \( m_{2(j,k)+2} \) for \( 1 \leq j \leq 4 \) and \( 1 \leq k \leq 2^|\sigma| \), where

\[
F_{m_{2(j,k)+1}} = \{ \tau \bar{\cdot} i : i \in A_j, \tau \in B_k \}
\]

and

\[
F_{m_{2(j,k)+2}} = \{ \tau : \tau \notin B_k \} \cup \{ \tau \bar{\cdot} i : i \notin A_j, \tau \in B_k \}.
\]

That is, \( h(\sigma) \) verifies that \( T \) is homogeneous by selecting the unique set \( B_k = \{ |\tau| = |\sigma| \& \tau \in T \} \) and the unique set \( A_j \) such that for \( \tau \in B_k, \tau \bar{\cdot} i \in T_e \iff i \in A_j \).

(b) The Extendible Trees. Recall that a closed set \( P \) is decidable if the \( P = [T] \) for some computable tree \( T \) without dead ends. For the purpose of string verification, let us say that a tree \( T \) is extendible if \( T \) has no dead ends. This means that that for any \( \sigma \in T \), either \( \sigma \bar{\cdot} 0 \in T \) or \( \sigma \bar{\cdot} 1 \in T \). In general, \( Ext(T) = \{ \sigma : I(\sigma) \cap [T] \neq \emptyset \} \) is the set of extendible nodes of \( T \) and \( T \) is extendible if and only if \( T = Ext(T) \).

Thus we let \( h(\sigma) = (m_1, m_2, m_3, m_4) \), such that \( F_{m_1} = \{ \sigma \bar{\cdot} 0 \} \), \( F_{m_2} = F_{m_4} = \emptyset \) and \( F_{m_3} = \{ \sigma \bar{\cdot} 1 \} \). That is, \( h \) verifies that \( T \) has no dead ends by either showing that \( \sigma \bar{\cdot} 0 \in T \) if \( F_{m_1} \subset T \) or that \( \sigma \bar{\cdot} 1 \in T \) if \( F_{m_3} \subset T \).

**Definition 2.3.4.** (a) A \( \Pi_1^0 \) class \( P \) satisfies a finite set of relations \( \mathcal{H}_i \subset \mathcal{P}(2^{<\omega}) \) \((i \leq n)\) if there is a computable tree \( T \) such that \( P = [T] \) and \( \mathcal{H}_i(T) \) for each \( i \leq n \).

(b) A \( \Pi_1^0 \) class \( P \) strongly satisfies a finite set of relations \( \mathcal{H}_i \subset \mathcal{P}(2^{<\omega}) \) \((i \leq n)\) if there is a primitive recursive tree \( T \) such that \( P = [T] \) and \( \mathcal{H}_i(T) \) for each \( i \leq n \).

**Definition 2.3.5.** A family \( \mathcal{F} \) of classes is [strongly] string verifiable (s.v.) if there is some finite set of string verifiable relations so that: \( P \in \mathcal{F} \) if \( P \) [strongly] satisfies these relations.

**Remark 2.3.6.** Note that any string verifiable family of trees contains the empty tree, so that any string verifiable family of \( \Pi_1^0 \) classes contains the empty class. If \( P = \emptyset \), then \( P = [T] \) if and only if \( T \) is finite, so any tree \( T \) with \( P = [T] \) is primitive recursive. So any strongly string verifiable family also contains the empty class.
2.3.2 Computable and Effective Numberings

We now demonstrate that strongly string verifiable and string verifiable families possess computable and effective numberings, respectively.

**Theorem 2.3.7.** (a) Any strongly string-verifiable family of $\Pi^0_1$ classes has a computable numbering.

(b) Any string-verifiable family of $\Pi^0_1$ classes has an effective numbering.

**Proof.** Suppose $F$ is a [strongly] string-verifiable family of $\Pi^0_1$ classes satisfying string verifiable (tree) relations $H_0, H_1, ..., H_m$, with corresponding string functions $h_0, h_1, ..., h_m$. For part (a), let the standard computable enumeration of the $\Pi^0_1$ classes in $2^\omega$ be given by $P_e = [T_e]$, where the sequence $T_e$ is uniformly primitive recursive (for example, the numbering $\psi_2$ given in section 2.2.1). We will define a uniformly computable sequence $S_e$ of trees such that the sequence $Q_e = [S_e]$ enumerates exactly the family of $\Pi^0_1$ classes strongly satisfying $H_0, H_1, ..., H_m$.

For any $\sigma \in \{0,1\}^n$, we determine whether $\sigma \in S_e$ as follows. First check that $\sigma \in T_e$. If so, for each $\tau \in \{0,1\}^n$ and each $i \leq m$, compute $h_i(\tau) = (D_1, D_2, \ldots, D_{2j})$ and determine whether there exists $i < j$ such that $D_{2j+1} \subseteq T_e$ and $D_{2j+2} \cap T_e = \emptyset$. This process is computable since each $D_\ell$ is a canonical finite set. If the answer is yes, for every $\tau \in \{0,1\}^n$, then $\sigma \in S_e$ and otherwise, $\sigma \notin S_e$. It is clear that if $T_e$ satisfies all of the relations $H_0, H_1, ..., H_m$, then $T_e = S_e$. It follows that every $\Pi^0_1$ class in $F$ occurs in the enumeration $Q_e = [S_e]$. On the other hand, if $T_e$ fails any of the relations, then $S_e$ is a finite set and $Q_e = \emptyset$. By assumption, $Q_e \in F$ in this case as well, so that the sequence $\{Q_e : e < \omega\}$ enumerates exactly the family $F$, as desired.

For part (b), let $P_e = \psi_5(e) = T_e$, the uniformly $\Pi^0_1$ enumeration which has the property that every computable tree occurs in the list $\{T_e : e < \omega\}$. We need to do the string verification in a $\Pi^0_1$ fashion and in particular to check that $D_{2j+2} \cap T_e = \emptyset$, which appears to be $\Sigma^0_1$. However, we can simply check that, if $\rho \in D_{2j+2}$ and $\phi_e(\rho) \downarrow$, then $\phi_e(\rho) = 0$. Then the sequence $S_e$ is uniformly $\Pi^0_1$ and, if $T_e$ has characteristic function $\phi_e$ and satisfies the string relations, it follows that $S_e = T_e$. \qed
We now provide some additional examples of string and strongly string verifiable classes.

**Definition 2.3.8.**

(i) A $\Pi^0_1$ class $P$ is *strongly decidable* if there is a primitive recursive tree $T$ with no dead ends such that $P = [T]$.

(ii) A $\Pi^0_1$ class $P$ is *strongly homogeneous* if there is a homogeneous primitive recursive tree $T$ with no dead ends such that $P = [T]$.

From definition 2.3.8, we immediately have the following corollary to Theorem 2.3.7

**Corollary 2.3.9.**

(a) The family of decidable $\Pi^0_1$ classes in $2^\omega$ has an effective numbering and the family of strongly decidable $\Pi^0_1$ classes in $2^\omega$ has a computable numbering.

(b) The family of homogeneous $\Pi^0_1$ classes in $2^\omega$ has an effective numbering and the family of strongly homogeneous $\Pi^0_1$ classes in $2^\omega$ has a computable numbering.

In the following section we will provide results which demonstrate that decidable classes in fact have in fact a 1-1 computable numbering (see Corollary 2.4.2.1). We provide an alternate explicit proof in section 2.4.2.1. For homogeneous classes, a different approach is needed to demonstrate that they possess a 1-1 computable numbering and we do this in section 2.4.1.

### 2.3.3 Families Containing the Clopen Classes

In this section we consider string-verifiable families of classes that contain all clopen classes. We first improve Theorem 2.3.7 to obtain a computable numbering of any string-verifiable family which includes the clopen sets.

**Theorem 2.3.10.** If $\mathcal{F}$ is any string-verifiable family of $\Pi^0_1$ classes, then there is a computable numbering of $\mathcal{C} \cup \mathcal{F}$.

**Proof.** We modify the proof of Theorem 2.3.7 so that when the string-verifiable relations fail, we extend all nodes rather than making them dead ends. Once again, the construction is based on the enumeration $\phi_e$ of the partial computable functions. The construction is in stages, where at stage $s$ we will have

\[ n_{e,s} = \max\{n : (\forall \sigma \in \{0, 1\}^n)\phi_e,\langle \sigma \rangle \downarrow\}, \]
\[ J_{e,s} = \{ \sigma \in \{0,1\}^{n_{e,s}} : \phi_{e,s}(\langle \sigma \rangle) = 1 \}, \]

and
\[ Q_{e,s} = \bigcup J_{e,s} = [S_{e,s}]. \]

Then \( Q_e = \bigcap_s Q_{e,s} = [S_e] \) will be the desired numbering. To ensure that this numbering is computable, we will determine whether \( \sigma \in S_e \) at stage \(|\sigma|\).

For this argument, we assume that \( \phi_e(0) = 1 \) for all \( e \).

**Construction.** At stage 0 we have \( n_{e,0} = 0 \), \( J_{e,0} = S_{e,0} = \{\emptyset\} \) and \( Q_{e,0} = 2^\omega \).

At stage \( s + 1 \), we check to see whether \( \phi_{e,s+1}(\langle \sigma \rangle) \downarrow \) for all \( \sigma \in \{0,1\}^{n_{e,s}+1} \). If not, then \( n_{e,s+1} = n_{e,s} \), \( J_{e,s+1} = J_s \) and \( S_{e,s+1} = S_{e,s} \cup \{ \sigma \upharpoonright i : \sigma \in S_{e,s}, i = 0,1 \} \). If so, then we check to see that \( \phi_{e,s+1} \) is the characteristic function of a tree on \( \{0,1\}^{n_{e,s}+1} \) and we verify the string relations up to \( \{0,1\}^{n_{e,s}+1} \). If this verification fails, then again \( n_{e,s+1} = n_{e,s} \) and \( J_{e,s+1} = J_{e,s} \). In this case, verification will also fail at all future stages, so that \( Q_e = Q_{e,s} \) is a clopen set.

If the tree and string-verification succeed, then \( n_{e,s+1} = n_{e,s} + 1 \), so that \( J_{e,s+1} \subseteq \{0,1\}^{n_{e,s}+1} \) and \( Q_{e,s+1} \) change as indicated above. In this case,
\[ S_{e,s+1} = S_{e,s} \cup \{ \sigma \in \{0,1\}^{s+1} : \sigma \upharpoonright (n_{e,s+1}) \in J_{e,s+1} \}. \]

If \( \phi_e \) is the characteristic function of the computable tree \( T_e \), and if \( P_e = [T_e] \in \mathcal{F} \), then it follows from the construction that \( Q_e = P_e \), so that \( Q_e \in \mathcal{F} \) and furthermore, any \( \Pi^0_1 \) class \( P_e \in \mathcal{F} \) will thereby occur in the numbering. Otherwise, the construction will make \( Q_e \) a clopen set. \( \Box \)

**Corollary 2.3.11.** For any string verifiable family \( \mathcal{F} \) of \( \Pi^0_1 \) classes, there a 1-1 computable numbering of \( \mathcal{C} \cup \mathcal{F} \).

**Proof.** Let \( \mathcal{F} \) be a string verifiable family. Then there is a computable numbering of \( \mathcal{C} \cup \mathcal{F} \) by Theorem 2.3.10. It then follows from Theorem 2.2.7 that there is a 1-1 computable numbering of \( \mathcal{C} \cup \mathcal{F} \). \( \Box \)
Corollary 2.3.12. There a 1-1 computable numbering of any string verifiable family of \( \Pi^0_1 \) classes containing all clopen classes.

Corollary 2.3.13. There a 1-1 computable numbering of the decidable \( \Pi^0_1 \) classes.

2.4 Named Families of \( \Pi^0_1 \) Classes

In this section, we obtain numbering results for various named families that commonly occur in the literature. In the first two sub-sections (2.4.1, 2.4.2), we expand upon the results from section 2.3 and we obtain positive numbering results for the homogeneous and decidable classes. In the two sub-sections (2.4.3, 2.4.4) that follow these, we obtain negative results for the thin, perfect thin, small, very small, and nondecidable classes.

2.4.1 Homogeneous Classes

Homogeneous \( \Pi^0_1 \) classes are a string-verifiable family of \( \Pi^0_1 \) classes. Consequently, by Corollary 2.3.9, they possess an effective numbering. Corollary 2.3.12 falls short of demonstrating that they possess a computable numbering, as clopen sets are not necessarily homogeneous. We provide the necessary argument in Theorem 2.4.3 and show, in fact, that a computable injective numbering exists. We first provide an alternate characterization of the homogeneous classes; they may be viewed separating classes of c.e. sets.

Definition 2.4.1. The separating class \( S(A, B) \) of two sets \( A, B \subset \omega \) is given by

\[
S(A, B) = \{ C \subset \omega : A \subseteq C \text{ and } B \cap C = \emptyset \}.
\]

In what follows, \( S_c(A, B) \) will denote the set of characteristic functions \( f_C \) of \( C \in S(A, B) \).

Theorem 2.4.2. \( P \subseteq 2^\omega \) is a nonempty homogeneous \( \Pi^0_1 \) class iff \( P = S_c(A, B) \) for some disjoint c.e. sets \( A \) and \( B \).

Proof. \((\rightarrow)\) Suppose that \( P \) is a nonempty homogeneous class with homogeneous tree \( T_P \). We will show that \( P = S_c(A, B) \) where \( A = \{ n : 0^n \downarrow 0 \not\in T_P \} \) and \( B = \{ n : 0^n \downarrow 1 \not\in T_P \} \).

To see that \( A \) and \( B \) are c.e., suppose that \( P = [T] \) with \( T \) computable. Then for each \( i \in \{ 0, 1 \} \), \( 0^n \downarrow i \not\in T_P \) iff \( (\exists s \geq (n + 1)) T \cap \{ \sigma : |\sigma| = s \text{ and } 0^n \downarrow i \sqsubseteq \sigma \} = \emptyset \).
To see that $A$ and $B$ are disjoint, suppose that $n \in A$. Since $P$ is not empty, there is some $x \in P$ so that $x \upharpoonright n \in T_P$. Now $0^n \downarrow 0 \notin T_P$ so that $(x \upharpoonright n) \uparrow 0 \notin T_P$. Hence $x(n) \neq 0$ so that $x(n) = 1$. Therefore $(x \upharpoonright n) \uparrow 1 \in T_P$. Since $P$ is homogeneous, $0^n \downarrow 1 \in T_P$ so that $n \notin B$. Hence $A \cap B = \emptyset$.

To see that $S_c(A, B) \subseteq P$, take $f_C \in S_c(A, B)$ for some $C \in S(A, B)$. Then,

$$n \in C \quad \Rightarrow \quad n \notin B \quad \Rightarrow \quad 0^n \downarrow 1 \in T_P \quad \Rightarrow \quad f_C \upharpoonright (n + 1) = (f_C \upharpoonright n) \uparrow 1 \in T_P \quad \text{&} \quad n \notin C \quad \Rightarrow \quad n \notin A \quad \Rightarrow \quad 0^n \downarrow 0 \in T_P \quad \Rightarrow \quad f_C \upharpoonright (n + 1) = (f_C \upharpoonright n) \downarrow 0 \in T_P$$

Since for arbitrary $n$, $(f_C \upharpoonright (n + 1)) \in T_P$, it follows that $f_C \in P$.

To see that $P \subseteq S_c(A, B)$, suppose that $x \in P$ and let $C = \{n : x(n) = 1\}$. It is clear that $x$ is the characteristic function $f_C$ of $C$. It suffices to show that $C \in S(A, B)$ (so that $f_C \in S_c(A, B)$). Suppose first that $n \in A$. Then $0^n \downarrow 0 \notin T_P$ so that $(f_C \upharpoonright n) \uparrow 0 \notin T_P$. Since $x = f_C \in T_P$, it follows that $f_C(n) = 1$. Hence $n \in C$ and $A \subseteq C$. Now suppose $n \in B$. Then $0^n \downarrow 1 \notin T_P$, so $(f_C \upharpoonright n) \downarrow 1 \notin T_P$. As before, $f_C(n) = 0$ so that $n \notin C$. Hence $C \cap B = \emptyset$.

$(\leftarrow)$ Suppose $A$ and $B$ are disjoint c.e. sets and $\{A_s\}_{s \in \omega}$ and $\{B_s\}_{s \in \omega}$ are stage enumerations of $A$ and $B$, respectively. Define $P = \cap_s [T_s]$ where $T_s \subseteq 2^{<\omega}$ is given by $\sigma \in T_s$ iff $(i \in A_s \rightarrow \sigma(i) = 1) \land (i \in B_s \rightarrow \sigma(i) = 0)$. Then $P$ is a $\Pi^0_1$ class and $P = S_c(A, B)$. Note that $T_P = \{\sigma : (\forall i < |\sigma|)[(\sigma(i) = 0 \land i \notin A) \lor (\sigma(i) = 1 \land i \notin B)]\}$ is homogeneous. Furthermore, since $A$ and $B$ are disjoint, $S_c(A, B)$ is nonempty. Hence $S_c(A, B)$ is a nonempty homogeneous class. \qed

**Theorem 2.4.3.** There is an injective computable numbering of the homogeneous $\Pi^0_1$ classes.

**Proof.** We may obtain an injective numbering of all ordered tuples $(A_e, B_e)$ of c.e. sets by Theorem 2.2.4.2. Then by Theorem 2.4.2, $P_e = S(A_e, B_e)$ is an injective numbering of the homogeneous classes. Furthermore, $S(A_e, B_e) = [T_e]$ where

$$\sigma \in T_e \iff (\forall n < |\sigma|)[(n \in A_{e,s} \rightarrow \sigma(n) = 1) \land (n \in B_{e,s} \rightarrow \sigma(n) = 0)]$$

This shows that the numbering is computable. \qed
2.4.2 Decidable Classes

In this section we provide an alternate proof of the existence of injective computable numberings of the decidable classes (see Corollary 2.3.13 for the first proof). We also show that in any computable numbering \( \phi \) of the decidable classes via trees, we can provide a computable numbering \( \hat{\phi} \) of all the trees without dead ends that occur, along with all clopen classes. Finally, we show that some decidable class in class in the numbering \( \phi \) must necessarily have, as a tree, dead ends throughout every occurrence in the numbering.

2.4.2.1 An injective computable numbering (Alternate proof)

Since decidability is string-verifiable and every clopen set is decidable, it follows from Corollary 2.3.12 that the decidable classes have a 1-1 computable numbering.

This result could not be obtained by using the standard numbering of the \( \Pi^0_1 \) classes and modifying each tree as it becomes known that is has a dead end. (For example, simply extend each such node with, say, all ones.) This is because, as a consequence to the following theorem, \( P \) being decidable is insufficient to ensure that the unique tree \( T_P \) without dead ends shows up in a computable tree numbering. The following is an alternate, explicit proof.

Theorem 2.4.4. There is a 1-1 computable numbering of all decidable classes in \( 2^\omega \).

Proof. Let \( e \mapsto W_e \) be the injective effective numbering of the computable sets as given in the proof of Corollary 2.2.8. We will define an effective correspondence between these sets and the nonempty computable trees without dead ends. We will do this through a series of three one-to-one correspondences—namely the correspondences between (1) the subsets of \( \omega \) and \( 2^\omega \), (2) \( 2^\omega \) and \( 3^\omega \), and (3) \( 3^\omega \) and the nonempty trees without dead ends. In a stage construction we will then define at stage \((e, s)\) a tree \( T_{e,s} \) based on \( W_{e,s} \) and the correspondences. Letting \( T_e = \cap_s T_{e,s} \), we will obtain an injective computable numbering \( e \mapsto [T_e] \) of all nonempty decidable \( \Pi^0_1 \) classes. Furthermore \([T_e]\) will correspond to \( W_e \) for each \( e \). Finally, by appending the empty class to the enumeration we obtain the desired result. We now define the correspondences.
The one-to-one correspondence between the subsets $S$ of $\omega$ and $2^\omega$ is given as follows. Each $S$ corresponds to $x_S \in 2^\omega$ given by $x_S(i) = 1$ iff $i \in S$.

The one-to-one correspondence between $2^\omega$ and $3^\omega$ is given as follows. Let $x \in 2^\omega$ and define $a_0 = 1, a_1 = 01$, and $a_2 = 00$. Then $x$ corresponds to the unique sequence $(f_x(i))_{i \in \omega} \in 3^\omega$ where $f_x : \omega \to 3$ and $x = a_{f_x(0)}a_{f_x(1)}a_{f_x(2)} \ldots$

The one-to-one correspondence between the set of all nonempty trees $T \subseteq 2^{<\omega}$ without dead ends and $3^\omega$ is given as follows. Let $T$ be a tree without dead ends and let $\sigma_0 = \emptyset, \sigma_1, \ldots$ be an enumeration of the elements of $T$ in order, first by length and then lexicographically. We define $g = g_T \in 3^\omega$ by recursion as follows. For each $n$, define $g(n) = 2$ if $\sigma_n^0$ and $\sigma_n^1$ are both in $T$, $g(n) = 1$ if $\sigma_n^0 \notin T$ and $\sigma_n^1 \in T$ and $g(n) = 0$ if $\sigma_n^0 \in T$ and $\sigma_n^1 \notin T$. For each such $g$ we now let $T_g$ denote the unique tree without dead ends corresponding to $g$.

At stage $(e, s)$, let $T_{e,s}$ be the clopen tree corresponding to $W_{e,s}$. Then, for fixed $e$, since the elements of $W_e$ are enumerated into the set in increasing order, for all $s$ we have that $T_{e,s+1} \subseteq T_{e,s}$ and each $W_e$ corresponds to $T_e = \cap_s T_{e,s}$. In fact, the finite sets precisely correspond to the clopen trees. Now, $e \mapsto W_e$ injectively numbers all computable sets. Therefore $\{\emptyset\} \cup \{[T_e]\}_{e \in \omega}$ is an ‘injective computable numbering’ of the decidable $\Pi_1^0$ classes. 

2.4.2.2 Trees without dead ends: A numbering result

A class is decidable iff it is the set of infinite paths through computable trees without dead ends. Given any computable numbering of the $\Pi_1^0$ classes via trees, this motivates capturing, through some computable enumeration, those trees in the numbering that have no dead ends. The following theorem demonstrates that this is possible injectively, provided that all clopen trees are also included.

**Theorem 2.4.5.** Suppose $e \mapsto T_e$ is a computable numbering of computable trees in $2^{<\omega}$. Then there is an injective computable numbering of trees consisting precisely of all the $T_i$ that have no dead ends along with all clopen trees.
**Proof.** Assume without loss of generality that \( \{T_e\}_{e \in \omega} \) contains all clopen trees. We will construct in stages, as in the terminology of Theorem 2.2.6, a sequence of *follower* trees \( S_i \). At stage \( i \) we will ensure that we have \( i + 1 \) trees \( S_0, S_1, \ldots, S_i \), constructed up to level \( 2^i \), following trees \( T_{(S_0,k_1)}, \ldots, T_{(S_i,k_i)} \) \((k_i \in \{m,n\})\) which are each pairwise distinct at level \( 2^i \). Also, at stage \( i \), initially some of the \( S_i \) will have the status of being marked \((k_i = m)\) in which case \( S_i \) will continue to follow \( T_{(S_i,m)} \) forever. If not, then \( S_i \) is not marked \((k_i = n)\) and we determine for each \( i \), if \( S_i \) should be marked. If an \( S_i \) needs to be marked then we determine a tree \( T_{(S_i,m)} \) that it shall hereafter follow. Otherwise each \( S_i \) continues to follow \( T_{(S_i,n)} \) and the stage is complete.

**Construction.** *Stage 0.* Find the first tree \( T_i \) such that \( T_i \cap \{0,1\}^{2^0} \neq \emptyset \), denote this tree as \( T_{(S_0,n)} \), and define \( S_0 = T_{(S_0,n)} \cap \{0,1\}^{2^0} \).

*Stage \( j+1 \).* \( S_0, \ldots, S_j \) have already been constructed up to level \( 2^j \) and are already following trees \( T_{(S_0,k_j)}, \ldots, T_{(S_j,k_j)} \). We perform the following two actions at this stage: (1) Construct \( S_0, \ldots, S_j \) up to level \( 2^{j+1} \) by determining the trees \( T_{(S_0,k_{j+1})}, \ldots, T_{(S_j,k_{j+1})} \) they shall follow, and (2) Construct a new tree \( S_{j+1} \) up to level \( 2^{j+1} \).

**Action (1).** Let \( U_{j+1} = \{(S_i,k_j) : k_j = n \text{ and } T_{(S_i,k_j)} \text{ has dead ends at level } 2^{j+1}\} \). All \( S_i \) such that \((S_i,k_j) \notin U_{j+1}\) keep their status as marked or unmarked, so \( k_j = k_{j+1} \), and continue to follow \( T_{(S_i,k_{j+1})} \). Those \( S_i \) such that \((S_i,k_j) \in U_{j+1}\) will hereafter be marked and will now follow the tree \( T_{(S_i,m)} = \{\sigma : \tau \subseteq \sigma \text{ or } \sigma \subseteq \tau \text{ for some } \tau \in T_{(S_i,n)} \text{ of length } 2^{j}\} \). Note that each marked \( S_i \) follows a clopen tree \( T_{(S_i,m)} \).

**Action (2).** Let \((S_{j+1},n)\) be the least \( i \) such that \( T_i \) is distinct from all \( T_{(S_i,k_{j+1})} \) \((i \leq j)\) at level \( 2^{j+1} \) and such that \( T_i \) has no dead ends. Define \( S_{j+1} = T_{(S_{j+1},n)} \cap \{0,1\}^{2^{j+1}} \).

**Verification.** We now verify that: (i) For each \( i \), \( \lim_j T_{(S_i,k_j)} \downarrow = S_i = T_{n_i} \) for some \( T_{n_i} \) without dead ends, (ii) For all \( i \), if \( T_i \) has no dead ends then there is a \( c \) such that \( T_i = S_c \), and (iii) \( i \neq j \) implies that \( S_i \neq S_j \).

Verification of (i). For all \( j \), \( k_j = n \) or \( k_j = m \). Fix \( i \). By Action (2), at stage \( i \), \((S_i,k_i) = (S_i,n)\). By Action (1), \( k_\ell = k_{\ell+1} = n \) for all \( \ell > i \) if \( S_i \) is never marked. If \( S_i \) is marked at stage \( r > i \), then for all \( s \geq r \), \( k_s = k_{s+1} = m \). In either case \( \lim_{j \geq i} k_j \downarrow \) so that
\[ \lim_j (S_i, k_j) \] converges to \((S_i, n)\) or \((S_i, m)\). If it converges to \((S_i, m)\) then \(S_i\) never diverges from following the clopen tree \(T_{(S_i, m)}\). Otherwise \(S_i\) is never marked and continually follows \(T_{(S_i, n)}\). Since it is never marked it means that \(T_{(S_i, n)}\) never has dead ends up to level \(2^r\), for all \(r > i\). So \(T_{(S_i, n)}\) is a tree without dead ends. Either way \(\lim_j T_{(S_i, k_j)} \downarrow = T_{n_i}\) for some tree \(T_{n_i}\) without dead ends. Now for all \(n\), \(S_i \cap \{0, 1\}^n \subseteq T_{(S_i, k_n)} \cap \{0, 1\}^n\). Therefore \(S_i = \lim_j T_{(S_i, k_j)} = T_{n_i}\).

**Verification of (ii).** Let \(T_i\) be a tree without dead ends. There are two cases. If there is a stage \(j\) and a \(c\) such that \(T_i = T_{(S_c, m)}\) at stage \(j\), then by the construction \(T_i = S_c\). If not, let \(\hat{i}\) equal the least \(k\) such that \(T_k = T_i\). Let \(j\) be large enough so that \(T_k\) differs from \(T_e\) at level \(2^j\) for all \(e < \hat{i}\). If at stage \(j\) there already exists a \(c\) such that \(T_{\hat{i}} = T_{(S_c, n)}\) then clearly \(T_i = S_c\). Otherwise, by Action (2), some tree \(S_c\) follows \(T_k\) by no later than stage \(j + \hat{i}\).

**Verification of (iii).** By Action (2), \(S_i\) is distinct from all \(S_j\) \((j < i)\) at level \(2^i\) and from all \(S_j\) \((j > i)\) at level \(2^j\). So \(S_i \neq S_j\) if \(i \neq j\).

### 2.4.2.3 Trees with dead ends: A necessity

In any computable numbering \(\phi\) of \(\Pi^0_1\) classes via trees, some decidable class must necessarily have, as a tree, dead ends throughout every occurrence in the numbering (see Corollary 2.4.7). Two different proofs of this fact may be obtained from Theorems 2.4.6 and 2.4.8 below.

**Theorem 2.4.6.** In any computable numbering of computable trees in \(2^{<\omega}\) there is a computable tree without dead ends outside the image of the numbering.

**Proof.** Let \(\{T_e : e < \omega\}\) be a uniformly computable sequence of trees. Now use a diagonalization argument to construct a tree \(T\) such that for all \(n\), \(T \cap \{0, 1\}^{n+1} \neq T_n \cap \{0, 1\}^{n+1}\), as follows. At stage 0 let \(T \cap \{0, 1\}^0 = \{\emptyset\}\). At stage \(n + 1\) we are given \(T \cap \{0, 1\}^n \neq \emptyset\). Therefore there are at least 2 subtrees of \(\{0, 1\}^{n+1}\) without dead ends extending \(T \cap \{0, 1\}^n\). Define \(T \cap \{0, 1\}^{n+1}\) to be an extension which is different from \(T_n \cap \{0, 1\}^{n+1}\). □
Corollary 2.4.7. For any computable numbering $P_e = [T_e]$ of the $\Pi^0_1$ classes in $2^\omega$, there is a decidable $\Pi^0_1$ class $P$ such that $P \neq [T_e]$ for any $T_e$ without dead ends.

Proof. Let $P = [T]$ where $T$ is the computable tree without dead ends provided by Theorem 2.4.6. Suppose that $P = [T_e]$ for some $e$. Since $T$ has no dead ends, it follows that $T = T_P$ and if $T_e$ also had no dead ends, then $T_e = T_P = T$. But by the construction, $T \cap \{0, 1\}^{e+1} \neq T_e \cap \{0, 1\}^{e+1}$, so that $T \neq T_e$. □

It follows from this corollary that in the standard numbering, $\{e : T_e \text{ has no dead ends}\} \neq \{e : P_e = [T_e] \text{ is decidable}\}$. In fact both have distinct complexities. By Konig’s Lemma, $\text{Ext}(P_e) = \{\sigma \in 2^{<\omega} : I(\sigma) \cap P_e \neq \emptyset\}$ is $\Pi^0_1$. So $\{e : T_e \text{ has no dead ends}\} = \{e : T_e = \text{Ext}(P_e)\}$ is $\Pi^0_1$. However, $\{e : P_e \text{ is decidable}\} = \{e : P_e = [T] \text{ for some computable } T \text{ without dead ends}\} = \{e : (\exists a) \phi_a \text{ is a characteristic function for } \text{Ext}(P_e)\}$ is $\Sigma^0_3$. An alternate proof of Corollary 2.4.7 is as a corollary of the following.

Theorem 2.4.8. For any acceptable numbering $\psi$ of the $\Pi^0_1$ classes,

$$\{e : \psi(e) \text{ is decidable}\} \text{ is } \Sigma^0_3 \text{ complete.}$$

Proof. It suffices to prove this for the standard numbering ($\psi_2$). We will make use of the well-known [86] $\Sigma^0_3$ completeness of $\{e : W_e \text{ is computable}\}$. It is easy to see that $\{e : \psi_2(e) \text{ is decidable}\}$ is $\Sigma^0_3$. For the completeness, define the uniformly computable trees $T_{f(e)}$ so that

(i) $0^n \in T_{f(e)}$ for all $n$;

(ii) $0^n1^s \in T_{f(e)} \iff n \notin W_{e,s}$.

It follows that $0^n1 \in \text{Ext}(T_{f(e)}) \iff n \notin W_e$, so that if $\psi_{f(e)}$ is decidable, then $W_e$ is computable. On the other hand, $\text{Ext}(T_{f(e)}) = \{0^n : n \in \omega\} \cup \{0^n1^s : s \in \omega, n \notin W_e\}$, so that if $W_e$ is computable, then $\psi(f(e))$ is decidable. Thus $W_e$ is computable if and only if $\psi(f(e))$ is decidable. □

Note that in [23], a $\Pi^0_1$ class $P_e = [T_e]$ in the standard numbering is said to be decidable if $T_e$ has no dead ends, which we now see is probably not the right approach.
2.4.3 Thin and Perfect Thin Classes

In the literature, a Martin-Pour El theory is a consistent c.e. propositional theory with additional ‘thinness’ conditions. The conditions imposed have varied depending upon the context and motivation of the authors, but include: (1) few c.e. extensions, (2) essentially undecidable, and (3) well-generated. Some authors have chosen to only impose (1) [21], while others (1) and (2) [19], [24], and finally others (1), (2), and (3) [34], [40], [29]. The complete consistent extensions of these theories correspond to thin, perfect thin (or equivalently, special thin [19]), and homogenous thin classes, respectively. This section is devoted towards demonstrating the nonexistence of computable numberings of the first two cases by modifying the classical Martin-Pour El construction of a perfect thin class. Recently Solomon [89] also modified this theorem to construct a homogeneous thin class and therefore we conjecture that no computable numberings exist for these classes.

2.4.3.1 The Martin–Pour El Construction

Recall that a \( \Pi^0_1 \) class \( P \) is thin if for every \( \Pi^0_1 \) subclass \( Q \subseteq P \), there is a clopen set \( U \) such that \( Q = U \cap P \). It is perfect iff it has no isolated points.

A perfect class may be defined by a function \( g : 2^{<\omega} \to 2^{<\omega} \) such that for all \( \sigma, \tau, \sigma \sqsubseteq \tau \) implies \( g(\sigma) \sqsubseteq \tau \); let us say that \( g \) is extension preserving. Let \( G(x) = \bigcup_n g(x \upharpoonright n) \). Then \( G(2^\omega) \) is a perfect class. If \( g \) is defined in uniformly computable, extension-preserving stages \( g_\epsilon \) (with corresponding \( G : 2^\omega \to 2^\omega \)), so that \( g_\epsilon(\sigma) \sqsubseteq g_{\epsilon+1}(\sigma) \), then we have \( G(2^\omega) = \cap_\epsilon G_\epsilon(2^\omega) \), so that \( G(2^\omega) \) is a \( \Pi^0_1 \) class.

**Theorem 2.4.9** (Martin–Pour-El). For any computable extension-preserving function \( g : 2^{<\omega} \to 2^{<\omega} \), there exists a perfect thin \( \Pi^0_1 \) class \( P \subseteq G(2^\omega) \).

**Proof.** Let \( \{P_e = [T_e] : e \in \omega \} \) be the standard numbering of the \( \Pi^0_1 \) classes and \( \{\phi_e : e \in \omega \} \) be the standard numbering \( \{0,1\} \)-valued partial computable functions. We will construct a computable tree \( S \), corresponding \( \Pi^0_1 \) class \( P = [S] \), and a surjective homeomorphism \( F : 2^\omega \to P \). \( F \) will be constructed by means of an extension-preserving map \( f : 2^{<\omega} \to S \), with corresponding map \( F : 2^\omega \to 2^\omega \) defined by \( F(x) = \bigcup_n f(x \upharpoonright n) \).
We will define $f$ in stages to obtain uniformly computable, extension-preserving functions $f_s$ so that $f = \lim_s f_s$. To ensure that $P$ is thin, we will meet the following requirement for each $e$:

$$\text{Thin}(e) : ((\forall \sigma \in \{0, 1\}^{e+1})(\forall \tau) \ [ (f(\sigma) \in T_e \land \sigma \sqsubseteq \tau) \rightarrow f(\tau) \in T_e]$$

To see that \text{Thin}(e) makes $P$ thin, let $U = \{I(f(\sigma)) : |\sigma| = e + 1 \& f(\sigma) \in T_e\}$ and observe that if $P_e \subseteq P$, then $P_e = P \cap U$.

**Construction.** Let $f_0 = g$. At stage $t + 1$, we define $f_{t+1}$ as follows. Look for $e < t + 1$, $\sigma \in \{0, 1\}^{e+1}$, and $\tau \sqsupseteq \sigma$ with $|\tau| \leq t + 1$ such that $f_t(\sigma) \in T_e$, but $f_t(\tau) \notin T_e$. If no such $e$, $\sigma$, and $\tau$ exist, then $f_{t+1} = f_t$. Otherwise take the least such $e$ and the lexicographically least $\sigma$ and $\tau$ for that $e$. For all $\rho \in 2^{<\omega}$, let $f_{t+1}(\sigma^\frown \rho) = f_t(\tau^\frown \rho)$; for $\rho \sqsubseteq \sigma$ (with $\rho \neq \sigma$) or $\rho$ incomparable with $\sigma$, let $f_{t+1}(\rho) = f_t(\rho)$.

**Verification.** It is easy to see by induction on $|\sigma|$ that for each $\sigma$, $f_s(\sigma)$ converges to a limit $f(\sigma)$. Then by induction on $e$, each requirement \text{Thin}(e) is satisfied. To see that $f$ is injective, suppose towards a contradiction that $f(\sigma) = f(\tau)$ for $\sigma \neq \tau$. By the construction, $\sigma$ and $\tau$ must be comparable. Assume, without loss of generality, that $\tau = \sigma^\frown \rho (\rho \neq \emptyset)$. By induction it is clear that for all $t$, $f_t(\sigma) \neq f_t(\sigma^\frown \rho) = f_t(\tau)$. Let $F_e(x) = \bigcup_n f_e(x \upharpoonright n)$, so that $P = \bigcap_e F_e(2^\omega)$. Since $f_0 = g$, it follows that $P \subseteq G(2^\omega)$.

**2.4.3.2 Non-existence of computable numberings**

We now modify the Martin–Pour El construction to obtain, as a corollary, non-existence of computable numberings for thin and perfect thin classes.

**Theorem 2.4.10.** Any computable numbering of $\Pi_1^0$ classes in $2^\omega$ of Lebesgue measure zero omits some perfect thin class from its image.

**Proof.** Let $P_e = [T_e]$, where $\{T_e : e \in \omega\}$ is uniformly computable. We will construct a computable extension-preserving function $g : 2^{<\omega} \rightarrow 2^{<\omega}$ such that for all $e$ and all $\sigma \in \{0, 1\}^{e+1}$, $g(\sigma) \notin T_e$. Then letting $G(x) = \bigcup_n g(x \upharpoonright n)$ we will ensure that $G(2^\omega) \cap P_e = \emptyset$. Replacing $f_0$ by $g$ in Theorem 2.4.9, we obtain a perfect thin class $P$ such that $P \cap P_e = \emptyset$ (and hence certainly $P \neq P_e$), for all $e$. 

50
We define $g : 2^{<\omega} \rightarrow 2^{<\omega}$ recursively, as follows. Define $g(\emptyset) = \emptyset$. Then for each $\sigma \in \{0, 1\}^e$, compute the shortest and lexicographically least extension $\tau$ of $g(\sigma)$ such that $\tau \notin T_e$. Since $[T_e]$ has measure zero, it is nowhere dense and thus such a $\tau$ always exists. Then let $g(\sigma \downharpoonright i) = \tau \downharpoonright i$ for $i \in \{0, 1\}$.

**Corollary 2.4.11.** There is no computable numbering of all thin or of all perfect thin $\Pi^0_1$ classes.

**Proof.** All thin classes have Lebesgue measure zero [85]. Therefore if $e \mapsto P_e$ were a numbering of (perfect) thin classes then Theorem 2.4.10 would provide a (perfect) thin class $P$ such that $P \neq P_e$ for all $e$, a contradiction. \qed

### 2.4.4 Small, Very Small, and Nondecidable Classes

Binns defined in [11] the notions of small and very small classes as a means of guaranteeing incompleteness in the lattice of the Medvedev and Muchnik degrees of subsets of $\omega^\omega$. A nonempty $\Pi^0_1$ class $P$ is *small* if there is no computable function $\Phi$ such that for all $n$, $|T_P \cap \omega^{\Phi(n)}| \geq n$. Let $\Psi(n)$ be the least $k$ such that $|T_P \cap \omega^k| \geq n$. A nonempty $\Pi^0_1$ class $P$ is *very small* if the function $\Psi$ dominates every computable function $g$; that is, $\Psi(x) \geq g(x)$ for all but finitely many $x$. Let $A$ be a coinfinite c.e. set, say $\overline{A} = \{a_0 < a_1 < \ldots \}$. Recall that $A$ is *hypersimple* if there is no computable function $f$ such that $f(n) \geq a_n$ for all $n$ and it is *dense simple* if $n \mapsto a_n$ dominates every computable function. In this section we will use these sets to show that no effective numbering exists for the small, very small, or decidable classes.

#### 2.4.4.1 Numberings and high/noncomputable sets

En route to demonstrating our theorem, we now proceed show that there are no effective numberings of the high or of the noncomputable sets. As we shall eventually characterize small an very small classes in terms the degrees of these sets, these results will be crucial to our argument.

First, we modify Shoenfield’s Thickness Lemma [86, p. 131]. Some definitions are needed. For $B \subseteq \omega$, let $B^{\{y\}} = \{(y, z) \in B : z \in \omega\}$ and say that $B$ is piecewise computable
if $B[y]$ is computable for all $y$. For $B \subseteq A \subseteq \omega$, we say that $B$ is a thick subset of $A$ if for all $y$, $B[y] \setminus A[y]$ is finite.

**Lemma 2.4.12** (Thickness Lemma). For any uniformly c.e. sequence $\{W_i : i \in \omega\}$ of noncomputable c.e. sets and any piecewise computable c.e. set $B$, there is a thick c.e. subset $A$ of $B$ so that $W_n \not\leq_T A$ for all $n$.

**Proof.** The proof as in [86] is modified to ensure that the length and restraint functions and the requirements incorporate the pair $\langle i, k \rangle$ in place of the single argument $i$ to make the argument go through with each $W_i$ in conjunction with each functional $\Phi_k$.

We obtain the following corollary.

**Corollary 2.4.13.** For any uniformly c.e. sequence $\{W_n : n \in \omega\}$ of noncomputable c.e. sets, there is a high c.e. set $A$ such that for all $i$, $W_i \not\leq_T A$.

**Proof.** This follows from the modified thickness lemma above by the same argument found in [86, p. 133].

**Corollary 2.4.14.** (a) There is no uniformly c.e. numbering of all high c.e. sets.

(a) There is no uniformly c.e. numbering of all noncomputable c.e. sets.

In fact, it follows that there is no uniformly c.e. numbering of the high or noncomputable c.e. degrees.

### 2.4.4.2 Non-existence of effective numberings

We now proceed characterize the small and very small classes in term of the noncomputable degrees and the high degrees, respectively. The degree of a $\Pi^0_1$ class $P$ is defined to be the degree of $T_P$ and is thus always a c.e. degree (since $T_P$ is a co-c.e. set).

We will use the following two classic results.

(1) [Martin] Any high degree contains a maximal (and hence dense simple) set [86, pp. 211–217].

(2) [Dekker] Any noncomputable c.e. degree contains a hypersimple set [86, p. 81].

**Proposition 2.4.15.** A c.e. degree is high if and only if it contains a very small $\Pi^0_1$ class $P \subseteq 2^{\omega}$.
Proof.  \(\rightarrow\) Suppose \(a\) is high, and let \(A \in a\) be a maximal set, and let \(p\) be the principal function of \(\omega - A\), so that \(p\) dominates every computable function. Now let \(P_A = \{0^\omega\} \cup \{0^n10^\omega : n \notin A\}\). Then \(P_A\) is a \(\Pi_1^0\) class and for each \(n\), the least \(k\) such that \(|T_P \cap \{0,1\}^n| \geq k\) is precisely \(p(n) + 1\) for \(n > 0\) and hence dominates every computable function.

\(\leftarrow\) Let \(a\) be a c.e. degree and suppose that \(T_P \in a\) for some very small \(P\). Then the function \(f(n) = (\text{least } k)[|T_P \cap \{0,1\}^k| \geq n]\), which dominates every computable function, is computable from \(T_P\). It follows from Martin’s Theorem [86, p. 208] that \(T_P\) is high.

\(\square\)

Proposition 2.4.16. A c.e. degree is noncomputable if and only if it contains an infinite, small \(\Pi_1^0\) \(P \subseteq 2^\omega\).

Proof.  \(\rightarrow\) Suppose \(a\) is a noncomputable c.e. degree, let \(A \in a\) be hypersimple, and \(p\) be the principal function of \(\omega - A\), so that \(p\) is not dominated by any computable function. Then the \(\Pi_1^0\) class \(P_A\) as defined in the proof of Proposition 2.4.15 will have degree \(a\) and will be small.

\(\leftarrow\) Suppose that \(P\) is an infinite \(\Pi_1^0\) class and \(T_P\) is computable. Then the function \(g(n) = (\text{least } k)[|T_P \cap \{0,1\}^k| \geq n]\) is computable and it follows that \(P\) is not small. \(\square\)

Theorem 2.4.17. There is no effective (i.e. \(\Pi_1^0\)) numbering of all nondecidable, of all infinite small, or of all very small \(\Pi_1^0\) classes in \(2^\omega\).

Proof.  Suppose, towards a contradiction, that \(\{Q_n = [T_n] : n \in \omega\}\) is an effective numbering of \(\Pi_1^0\) classes such that each \(Q_n\) is nondecidable. Then \(W_n = \{\langle \sigma \rangle : \sigma \notin Ext(T_n)\}\) is a uniformly c.e. numbering of noncomputable c.e. sets. By Corollary 2.4.13, there is a high c.e. set \(A\) such that for all \(n\), \(W_n \not\leq_T A\). Therefore \(A\) is a high degree that contains a (very) small class not amongst the \(Q_i\), a contradiction. \(\square\)
3.1 Overview

The literature abounds with results in algorithmic randomness as pertaining to reals over a finite alphabet, especially within the last few years. Little is known, or even developed, however, with respect to randomness for closed sets of binary reals. This chapter is a first approach in this direction.

In this chapter, we consider a notion of effective (i.e. algorithmic) randomness on the space $\mathcal{C}$ of nonempty closed subsets $P$ of $2^\mathbb{N}$; to accomplish this task, we will need use the definition and machinery of effective randomness for reals, since, through appropriate
coding of closed sets, we will define a closed set to be random iff its code, as a real, is random. (In fact, later in Chapter 4, we will approach a definition of randomness for continuous functions in a similar fashion.) Consequently we begin this chapter with an introduction to algorithmic randomness, including a brief historical background.

More specifically, this chapter is organized as follows. In Section 3.2, we provide an introduction to algorithmic randomness for reals over a finite alphabet. In Section 3.3, we give a probability measure and define a version of the Martin-Löf Test for closed sets, leading to a definition of randomness for closed sets. In Section 3.4, we tackle the question of which types of elements necessarily belong, or do not belong, to random closed sets. For instance, every random closed set contains random and non-random elements, but no \( n \)-c.e. elements. In Section 3.5, we show that random closed sets have measure zero and box dimension \( \log_2 \frac{4}{3} \). In Sections 3.6-3.7, we explore alternate notions for randomness, such as the problem of compressibility of trees. Finally, in Section 3.8, we consider the problem of when a randomly chosen closed set meets a closed \( Q \); this is the study of capacities.

### 3.2 Effective Randomness of Reals

In this section, we present a basic introduction, including a brief historical background, for randomness of reals over a finite alphabet.

#### 3.2.1 Introduction

The study of algorithmic randomness has been of great interest in recent years. The basic problem is to quantify the randomness of a single real number. Early in the last century, von Mises [94] suggested that a random real should obey reasonable statistical tests, such as having a roughly equal number of zeroes and ones of the first \( n \) bits, in the limit. Thus a random real would be *stochastic* in modern parlance. If one considers only *computable* tests, then there are countably many and one can construct a real satisfying all tests.

An early approach to randomness was through betting. Effective betting on a random sequence should not allow one’s capital to grow unboundedly. The betting strategies used
are constructive martingales, introduced by Ville [93] and implicit in the work of Levy [65], which represent fair double-or-nothing gambling.

Martin-Löf [69] observed that stochastic properties could be viewed as special kinds of measure zero sets and defined a random real as one which avoids certain effectively presented measure zero sets; see Section 3.2.4. At the same time Kolmogorov [55] defined a notion of randomness for finite strings based on the concept of incompressibility. A stronger notion of prefix-free complexity was developed by Levin [64], Gács [48] and Chaitin [27] and extended to infinite words.

In the following sections, we formalize the notions of constructive martingale randomness, Martin-Löf randomness, and prefix-free randomness. After their entry into the literature, Schnorr later proved [83] that all of these notions are equivalent; this is a fundamental result in the theory of algorithmic randomness. While these definitions and results are usually given for binary strings and sequences, they carry over to \( k \)-ary strings and sequences as well. See, for example, Calude [17, 18], or Section 3.2.4 below, where we do this for the Martin-Löf definition of randomness.

### 3.2.2 Constructive Martingale Randomness

The betting approach to randomness is formalized as follows.

**Definition 3.2.1** (Ville [93]).

(i) A martingale is a function \( m : k^{<\omega} \to [0, \infty) \) such that for all \( \sigma \in k^{<\omega} \),

\[
m(\sigma) = \frac{1}{k} \sum_{i=0}^{k-1} m(\sigma \upharpoonright i).
\]

(ii) A martingale \( m \) succeeds on \( X \in k^\mathbb{N} \) if

\[
\limsup_{n \to \infty} d(X|n) = \infty.
\]

That is, the betting strategy results in an unbounded amount of money made on the \( k \)-ary infinite sequence \( X \).

(iii) The success set of \( m \) is the set \( S^\infty[m] \) of all sequences on which \( m \) succeeds.

That is, a martingale on \( 2^{<\omega} \) is the capital function of a fair double-or-nothing betting strategy. When working on \( 3^{<\omega} \) the strategy is triple-or-nothing.
Definition 3.2.2. A martingale $m$ is constructive (or effective, or c.e.) if it is lower semi-computable; that is, if there is a computable function $\hat{m} : k^{<\omega} \times \mathbb{N} \rightarrow \mathbb{Q}$ such that

(i) for all $\sigma$ and $t$, $\hat{m}(\sigma, t) \leq \hat{m}(\sigma, t + 1) < m(\sigma)$, and

(ii) for all $\sigma$, $\lim_{t \to \infty} \hat{m}(\sigma, t) = m(\sigma)$.

In other words, $m(w)$ is approximated from below by rationals uniformly in $w$.

Definition 3.2.3 (Constructive Martingale Randomness). A sequence in $k^\mathbb{N}$ is constructive martingale random if no constructive martingale succeeds on it.

Comment 3.2.4 (Nonmonotonic Martingales). Some flexibility may be gained by also considering nonmonotonic martingales; i.e., martingales which bet on the bits of a sequence out of order. While for a monotonic martingale only the amount of the next bet is determined from the bits seen previously, for a nonmonotonic martingale both the amount and the location of the next bet are determined from the bits seen previously (the next bit may precede them, follow them, or lie in the middle). These martingales must obey two rules: the standard fair-betting rule that monotonic martingales obey, and the rule that they never bet on the same bit twice. We refer the reader to Downey and Hirschfeldt [37] for the formal definition.

Although a priori allowing nonmonotonic martingales strengthens the notion of randomness, since more strategies must be defeated, in fact in the c.e. case they are equivalent. Muchnik, Semenov, and Uspensky [71] (Theorem 8.9) show that ML-random sequences defeat all computable nonmonotonic martingales (in fact they show this with respect to general measures, not just the coin-toss measure). The proof does not depend on the computability of the martingale, however; the martingale is used to define a Martin-Löf test which may be enumerated equally well alongside the enumeration of the martingale. Therefore, as defeating all c.e. nonmonotonic martingales is clearly sufficient to be ML-random, the two are equivalent.

3.2.3 Prefix-free Randomness

Prefix-free randomness for reals is defined as follows. A Turing machine $M$ which takes inputs from $A^*$, where $A$ is a finite alphabet, is called prefix-free if it has prefix-free
domain \( \text{dom}(M) \); that is, if \( \sigma \subseteq \tau \) are strings in \( \text{dom}(M) \), then \( \sigma \) must equal \( \tau \). For any finite string \( \tau \), the \textit{prefix-free complexity of \( \tau \) with respect to \( M \)} is

\[
K_M(\tau) = \min\{|\sigma|, \infty : M(\sigma) = \tau\}.
\]

There is a \textit{universal} prefix-free function \( U \) such that, for any prefix-free \( M \), there is a constant \( c \) such that for all \( \tau \)

\[
K_U(\tau) \leq K_M(\tau) + c.
\]

We let \( K(\tau) = K_U(\tau) \) and call it the \textit{prefix-free complexity of \( \tau \)}.

**Definition 3.2.5 (Prefix-free Randomness).** \( x \in \{0,1\}^\omega \) is called prefix-free random if there is a constant \( c \) such that \( K(x|n) \geq n - c \) for all \( n \).

This latter inequality means that the initial segments of \( x \) are not \textit{compressible}.

### 3.2.4 Martin-Löf (\( n \)-)randomness

According to the Martin-Löf definition of randomness, a random real must avoid certain effectively presented measure zero sets. Inherent in the definition, therefore, is the chosen measure being used. Fix an alphabet \( k = \{0,1,\ldots,k-1\} \). We present the definition of a general probability measure on \( k^\omega \), as well as different named measure types. Typically, however, we will make use of the alphabets \( \{0,1\} \) or \( \{0,1,2\} \).

**Definition 3.2.6 (General Probability Measures).** Let \( f : \{0,1,2\}^* \rightarrow [0,1] \) be a function such that \( f(\emptyset) = 1 \) and \( f(\sigma) = \sum_{i=0,1,2} f(\sigma \upharpoonright i) \) for all \( \sigma \). The \( f \)-probability measure \( \nu_f \) is defined so that the \( \nu_f \)-measure of the interval \([\sigma] \) is such that \( \nu_f([\sigma]) = f(\sigma) \).

**Definition 3.2.7 (Different Measures Types).** Let \( f \) and \( \nu_f \) be as in Definition 3.2.6.

(i) \( \nu_f \) is computable if \( f \) is computable.

(ii) \( \nu_f \) is nonatomic, or continuous, if for all \( x \in 3^\mathbb{N} \), \( \nu_f(\{x\}) = 0 \).

(iii) \( f \) and \( \nu_f \) are bounded if \((\exists b,c \in (0,1))(\forall \sigma)(\forall i)[b \cdot f(\sigma) < f(\sigma \upharpoonright i) < c \cdot f(\sigma)]\).

[Note that any bounded measure is also continuous.]

(iv) \( \nu_f \) is regular if \((\exists b_0,b_1,b_2 \in (0,1))(\forall \sigma)f(\sigma \upharpoonright i) = b_i \cdot f(\sigma)\)

Now fix a probability measure \( \mu \). In the literature, \( \mu \) is most typically the Lebesgue measure.
**Definition 3.2.8.** A real \( x \in k^\omega \) is Martin-Löf random if for every effective sequence \( S_1, S_2, \ldots \) of c.e. open sets with \( \mu(S_n) \leq 2^{-n}, x \notin \bigcap_n S_n \).

The latter condition is equivalent to the condition we get if we replace \( 2^{-n} \) with \( q^n \) for a computable sequence \( (q_i) \) of rationals such that \( \lim_i q_i = 0 \). We can also consider an extended definition of Martin-Löf randomness, in terms of \( \Sigma_n^0 \) sets.

**Definition 3.2.9.**

(i) A \( \Sigma_n^0 \) test is a computable collection \( \{V_n : n \in 2^\mathbb{N}\} \) of \( \Sigma_n^0 \) classes such that \( \mu(V_k) \leq 2^{-k} \);

(ii) A real \( \alpha \) is \( \Sigma_n^0 \) random or \( n \)-random if and only if it passes all \( \Sigma_n^0 \) tests (i.e., if \( \{V_n : n \in 2^\mathbb{N}\} \) is a computable collection of \( \Sigma_n^0 \) classes such that \( \mu(V_k) \leq 2^{-k} \), then \( \alpha \notin \bigcap_{n \geq 0} V_n \)).

Thus 1-random reals are just Martin-Löf random reals. See [36] for details on random and \( n \)-random reals. Kurtz [59] and Kautz [54] proved the following result. Let \( \emptyset^{(n)} \) denote the \( n \)-th jump of \( \emptyset \).

**Theorem 3.2.10.** Let \( q \) be a rational number.

(i) For each \( \Sigma_n^0 \) class \( S \), we can uniformly compute from \( q \) and a \( \Sigma_n^0 \) index for \( S \), the index of a \( \Sigma_n^{0^{(n-1)}} \) class \( U \supseteq S \) such that \( U \) is an open \( \Sigma_n^0 \) class and \( \mu(U) - \mu(S) < q \).

(ii) For each \( \Pi_n^0 \) class \( T \), we can uniformly compute from \( q \) and a \( \Pi_n^0 \) index for \( T \), the index of a \( \Pi_n^{0^{(n-1)}} \) class \( V \supseteq T \) such that \( V \) is a closed \( \Pi_n^0 \) class and \( \mu(V) - \mu(T) < q \).

(iii) For each \( \Sigma_n^0 \) class \( S \), we can uniformly compute from \( q \), and a \( \Sigma_n^0 \) index for \( S \) and an oracle for \( \emptyset^{(n)} \), the index of a \( \Pi_n^{0^{(n-1)}} \) class \( V \subseteq S \) such that \( V \) is a closed \( \Pi_n^0 \) class and \( \mu(S) - \mu(V) < q \). Moreover, if \( \mu(S) \) is a real computable from \( \emptyset^{(n-1)} \), then the index for \( V \) can be found computably from \( \emptyset^{(n-1)} \).

(iv) For each \( \Pi_n^0 \) class \( T \), we can uniformly compute from \( q \) and \( \Pi_n^0 \) index for \( T \) and an oracle for \( \emptyset^{(n)} \), the index of a \( \Sigma_n^{0^{(n-1)}} \) class \( U \subseteq T \) such that \( U \) is an open \( \Sigma_n^{0^{(n-1)}} \) class and \( \mu(T) - \mu(U) < q \). Moreover, if \( \mu(S) \) is a real computable from \( \emptyset^{(n-1)} \), then the index for \( U \) can be found computably from \( \emptyset^{(n-1)} \).

**Comment 3.2.11.** It follows that a real is \( n + 1 \)-random if and only if it is \( 1 \)-random relative to \( \emptyset^{(n)} \).
Theorem 3.2.12 (van Lambalgen [92]). The following are equivalent.

1. \( A \oplus B \) is \( n \)-random.
2. \( A \) is \( n \)-random and \( B \) is \( n \)-A-random.
3. \( B \) is \( n \)-random and \( A \) is \( n \)-B-random.
4. \( A \) is \( n \)-B-random and \( B \) is \( n \)-A-random.

3.3 Martin-Löf Randomness of Closed Sets

In this section we define a measure on the space \( C \) of nonempty closed subsets of \( 2^\mathbb{N} \) and use this to define the notion of randomness for closed sets. We then obtain several properties of random closed sets.

3.3.1 The Hit-or-Miss Topology on \( C \)

The standard (hit-or-miss) topology on \( C \) has as a sub-basis, the following two types of sets, where \( Q \) is any closed set: \( V(Q) = \{ K : K \cap Q \neq \emptyset \} \); \( W(Q) = \{ K : K \subseteq Q \} \). A basis for the hit-or-miss topology, then, is formed by taking finite intersections of these.

We now consider a refinement of the sub-basis sets and obtain a basis for the Borel sets. We will use the following notation. \( T^n \) denotes the set \( T \cap \{0, 1\}^n \), and \( T^\leq_n \) denotes the set \( T \cap \{0, 1\}^\leq_n \)

Definition 3.3.1. For any tree \( S \) and any \( n \), define

\[
C_n(S) = \{ Q \in C : S \leq_n = T^\leq_n Q \}
\]

That is, \( C_n(S) \) is the set of closed sets \( Q \in C \) that agree with \( S \) up to level \( n \).

Claim 3.3.2. A basis for the Borel sets is given by the agreement set \( A \):

\[
A = \{ C_n(S) : S \text{ is a tree, } n \in \omega \}.
\]

To see this, first note that any closed set \([T]\) is the decreasing intersection of clopen sets

\[
[T^n] := \bigcup \{ [\sigma] : \sigma \in T^n \}.
\]

Therefore we may rewrite sub-basis elements \( V([T]) \) and \( W([T]) \) as

\[
V([T]) = \cap_n V([T^n]) \quad \text{(by definition)} \quad \text{and}
\]
\( W([T]) = \cap_n W([T^n]) \) (by compactness)

But then,

\[
V([T^n]) = \bigcup \{C_n(S) : S \cap T^n \neq \emptyset \} \quad \text{and} \\
W([T^n]) = \bigcup \{C_n(S) : S^n \subseteq T \}
\]

To see the first equality, for example, note that \( Q \cap [T^n] \neq \emptyset \) if and only if \( Q \in C_n(S) \) for some \( S \) with \( S \cap T^n \neq \emptyset \). The latter equality holds similarly.

### 3.3.2 Toward a Measure

To define \( \mu(V(Q)) \) and \( \mu(W(Q)) \), for some fixed measure \( \mu \) and any closed set \( Q \), it suffices to define \( \mu(V(Q_n)) \) and \( \mu(W(Q_n)) \) for clopen sets \( Q_n \) where \( Q = \cap_n Q_n \). We would simply define \( \mu(V(Q)) = \lim_n \mu(V(Q_n)) \) and \( \mu(W(Q)) = \lim_n \mu(W(Q_n)) \). However, for any clopen \( Q \), \( W(Q) \) is the complement \( V(2^N \setminus Q) \). Hence it furthermore suffices to define \( \mu(V(Q)) \) for clopen sets to get a measure on \( C \).

From the justification of Claim 3.3.2, the latter task may be accomplished by defining a measure on all Borel basis elements, namely the agreement sets \( C_n(S) \). To accomplish this, in the following section we will encode all closed sets \( Q \in C \) with a canonical code \( x_Q \in C \).

Then using the Lebesgue measure on \( 3^N \), we will define a measure on the sets \( C_n(S) \) which, in fact, defines a measure on all closed sets.

### 3.3.3 Canonical Coding and Measure

#### The Canonical Coding

An effective one-to-one correspondence between the space \( C \) and the space \( 3^N \) is defined as follows. Let a closed set \( Q \) be given and let \( T = T_Q \) be the tree without dead ends such that \( Q = [T] \). Define the canonical code \( x = x_Q \in \{0,1,2\}^N \) for \( Q \) as follows. Let \( \lambda = \sigma_0, \sigma_1, \sigma_2, \ldots \) enumerate the elements of \( T \) in order, first by length and then lexicographically. We now define \( x = x_Q = x_T \) by recursion as follows. For each \( n \), \( x(n) = 2 \) if \( \sigma_n^0 \) and \( \sigma_n^1 \) are both in \( T \), \( x(n) = 1 \) if \( \sigma_n^0 \notin T \) and \( \sigma_n^1 \in T \) and \( x(n) = 0 \) if \( \sigma_n^0 \in T \) and \( \sigma_n^1 \notin T \). For example, if \( Q = \{0,1\}^N \), then \( x_Q = (2,2,\ldots) \) and if \( Q = \{y\} \), then \( x_Q = y \). Let \( Q_x \) denote the unique closed set \( Q \) such that \( x_Q = x \).

**Definition 3.3.3 (The Measure).** Define the measure \( \mu^* \) on \( C \) by

\[
\mu^*(\mathcal{X}) = \mu(\{x_Q : Q \in \mathcal{X}\})
\]
where \( \mu \) is the Lebesque measure (i.e. the regular measure \( \mu_d \) with \( b_0 = b_1 = b_3 = \frac{1}{3} \) (see Definition 3.2.7) on \( 3^\mathbb{N} \).

Informally this means that given \( \sigma \in T_Q \), there is probability \( \frac{1}{3} \) that both \( \sigma \downarrow 0 \in T_Q \) and \( \sigma \downarrow 1 \in T_Q \) and, for \( i = 0, 1 \), there is probability \( \frac{1}{3} \) that only \( \sigma \downarrow i \in T_Q \). In particular, this means that \( Q \cap I(\sigma) \neq \emptyset \) implies that for \( i = 0, 1 \), \( Q \cap I(\sigma \downarrow i) \neq \emptyset \) with probability \( \frac{2}{3} \).

**Comment 3.3.4.** At this stage, we have fixed the uniform measure (i.e. all \( b_i = \frac{1}{3} \)) towards defining randomness of closed sets. This allows us to more easily demonstrate the validity of many results. Later, in Section 3.7.1, we will show that the results hold with any regular measure. Proposition 3.3.5, however, demonstrates that the defined measure on \( C \), above, holds for any generalized probability measure \( \mu_d \) (see Definition 3.2.6).

**Justification for the Coding.** Let us also comment briefly on why some other natural representations were rejected. Suppose first that we simply enumerate all strings in \( \{0,1\}^* \) as \( \sigma_0, \sigma_1, \ldots \) and then represent \( T \) by its characteristic function so that \( x_T(n) = 1 \iff \sigma_n \in T \). Then in general a code \( x \) might not represent a tree. That is, once we have \( (01) \notin T \) we cannot later decide that \( (011) \in T \). Suppose then that we allow the empty closed set by using codes \( x \in \{0,1,2,3\}^* \) and modify our original definition as follows. Let \( x(n) = i \) have the same definition as above for \( i \leq 2 \) but let \( x(n) = 3 \) mean that neither \( \sigma_n \downarrow 0 \) nor \( \sigma_n \downarrow 1 \) is in \( T \). Informally, this would mean that for \( i = 0, 1 \), \( \sigma \in T \) implies that \( \sigma \downarrow i \in T \) with probability \( \frac{1}{2} \). The advantage here is that we can now represent all trees. But this is also a disadvantage, since for a given closed set \( P \), there are many different trees \( T \) with \( P = [T] \). The second problem with this approach is that we would have \( [T] = \emptyset \) with positive probability. We briefly return to this subject in Section 3.7.2.

Now recall the definition of a general probability measure on \( 3^\mathbb{N} \) (Definition 3.2.6). Let \( d : \{0,1,2\}^* \rightarrow [0,1] \) be a function such that \( d(\emptyset) = 1 \) and \( d(\sigma) = \sum_{i=0,1,2} d(\sigma \downarrow i) \) for all \( \sigma \). Then \( \mu_d(\sigma) \) is defined to be \( d(\sigma) \). We may now define, for any such \( d \), \( \mu_d^* \) exactly as in Definition 3.3.3. Furthermore \( \mu_d^* \) we be deemed computable if \( d \) is computable.
Proposition 3.3.5. For any \( d \), the measure \( \mu_d^* \) is defined on all Borel sets in the hit-or-miss topology on \( C \). Furthermore, if \( d \) is computable, then \( \mu_d^* \) is computable on the family of clopen sets.

Proof. As discussed in section 3.3.2, it suffices to show that \( \mu_d^*(C_n(S)) \) is defined for all \( C_n(S) \in \mathcal{A} \). Fix a tree \( S \) and suppose that \( \{ \sigma \subseteq \tau : \tau \in S^n \} \) is ordered \( \sigma_0 < \ldots < \sigma_{k(S,n)} \), first by length and then lexicographically. Then it is easy to see that

\[
Q \in C_n(S) \iff x_{[S^n]} \upharpoonright (k(S,n) + 1) \subseteq x_Q
\]

Consequently \( C_n(S) = \bigcap_{\sigma \in S^n} V([\sigma]) \) is clopen in \( C \). Furthermore,

\[
\mu_d^*(C_n(S)) = \mu([x_{[S^n]} \upharpoonright (k(S,n) + 1)]).
\]

This also demonstrates the computability of \( \mu_d^* \). \qed

Definition 3.3.6 (Random Closed Sets). A closed set \( Q \in C \) is (Martin-Löf) random iff its canonical code \( x_Q \) is Martin-Löf random.

This definition clearly relativizes to any oracle in accordance with the definitions of relative randomness in the Cantor space. Since random reals exist, it follows that random closed sets exist. Furthermore, there are \( \Delta^0_2 \) random reals, so we have the following.

Theorem 3.3.7. There exists a random closed set \( Q \) such that \( T_Q \) is \( \Delta^0_2 \). \qed

Note that if \( T_Q \) is \( \Delta^0_2 \), then \( Q \) must contain \( \Delta^0_2 \) elements (in particular the leftmost path). Since there exist strong \( \Pi^0_2 \) classes with no \( \Delta^0_2 \) elements, there are strong \( \Pi^0_2 \) classes \( Q \) such that \( T_Q \) is not \( \Delta^0_2 \). The following lemma will be needed throughout.

Lemma 3.3.8. For any \( Q \subseteq 2^\mathbb{N} \) which is either closed or open,

\[
\mu^*(\{ P : P \subseteq Q \}) \leq \mu(Q).
\]

Proof. Let \( \mathcal{P}_C(Q) \) denote \( \{ P : P \subseteq Q \} \). We first prove the result for nonempty clopen sets \( U \) in place of \( Q \) by the following induction. Suppose \( U = \bigcup_{\sigma \in S} I(\sigma) \), where \( S \subseteq \{0,1\}^n \). For \( n = 1 \), either \( \mu(U) = 1 = \mu^*(\mathcal{P}_C(U)) \) or \( \mu(U) = \frac{1}{2} \) and \( \mu^*(\mathcal{P}_C(Q)) = \frac{1}{3} \).

For the induction step, let \( S_i = \{ \sigma : i \neg \sigma \in S \} \), let \( U_i = \bigcup_{\sigma \in S_i} I(\sigma) \), let \( u_i = \mu(U_i) \) and let
$v_i = \mu^*(P_C(U_i))$, for $i = 0, 1$. Then considering the three cases in which $S$ includes both initial branches or just one, we calculate that

$$\mu^*(P_C(U)) = \frac{1}{3}(v_0 + v_1 + v_0v_1).$$

Thus by induction we have

$$\mu^*(P_C(U)) \leq \frac{1}{3}(u_0 + u_1 + u_0u_1).$$

Now

$$2u_0u_1 \leq u_0^2 + u_1^2 \leq u_0 + u_1,$$

and therefore

$$\mu^*(P_C(U)) \leq \frac{1}{3}(u_0 + u_1 + u_0u_1) \leq \frac{1}{2}(u_0 + u_1) = \mu(U).$$

For a closed set $Q$, let $Q = \bigcap_n U_n$, where $U_n$ is clopen and $U_{n+1} \subseteq U_n$ for all $n$. Then $P \subset Q$ if and only if $P \subseteq U_n$ for all $n$. Thus

$$P_C(Q) = \bigcap_n P_C(U_n),$$

so that

$$\mu^*(P_C(Q)) = \lim_{n \to \infty} \mu^*(P_C(U_n)) \leq \lim_{n \to \infty} \mu(U_n) = \mu(Q).$$

Finally, for an open set $Q$, let $Q = \bigcup_n U_n$ be the union of an increasing sequence of clopen sets $U_n$. Then, by compactness,

$$P_C(Q) = \bigcup_n P_C(U_n),$$

so that

$$\mu^*(P_C(Q)) = \lim_{n \to \infty} \mu^*(P_C(U_n)) \leq \lim_{n \to \infty} \mu(U_n) = \mu(Q).$$

This completes the proof of the lemma.

\[\square\]

3.3.4 Ghost Coding

We wish now to introduce a second method of coding, the \textit{ghost coding}. A ghost code of $Q$ is an infinite ternary string whose terms correspond to all nodes of $2^{<\omega}$ in
lexicographical order. The terms corresponding to the nodes of $Q$’s tree (the “canonical nodes”) agree with the corresponding terms in the canonical code; the remaining “ghost nodes” may hold any values. Ghost codes are non-unique, and every closed set has a non-random ghost code (if the closed set itself is random take the code with ghost nodes all equal to zero, say). This method of coding is more convenient for some purposes; for example, we will use it to show that if $Q_0, Q_1$ are closed sets and $Q = \{0^\infty x : x \in Q_0\} \cup \{1^\infty x : x \in Q_1\}$, $Q$ is random if and only if the $Q_i$ are random relative to each other.

### 3.3.5 Coding Equivalence

The utility of the ghost codes rests on the following correspondence. Recall van Lambalgen’s theorem (Theorem 3.2.12).

**Theorem 3.3.9.** The canonical code of a closed set $Q \subseteq 2^\mathbb{N}$ is random if and only if $Q$ has some random ghost code. Furthermore, for any $y$, the canonical code $r$ is $y$-random if and only if $Q$ has a ghost code which is $y$-random.

**Proof.** ($\leftarrow$) Suppose the canonical code of $Q$ is nonrandom. Then there is a c.e. martingale $m$ that succeeds on it. From any initial segment $\sigma$ of a ghost code $g$ for $Q$, the subsequence $\hat{\sigma}$ of exactly the canonical nodes of $\sigma$ is computable. Therefore it is computable whether the bit of $g$ after $\sigma$ is canonical or ghost. From $m$, define the martingale $m'$ which bets as follows:

$$m'(\sigma^\infty i) = \begin{cases} m(\hat{\sigma}^\infty i) & \text{next bit is a canonical node} \\ m'(\sigma) & \text{next bit is a ghost node.} \end{cases}$$

That is, $m'$ holds its money on ghost nodes and bets identically to $m$ on canonical nodes. It is clear that $m'$ succeeds on the ghost code $g$ and thus $g$ is nonrandom.

($\rightarrow$) Now suppose the canonical code $r$ for $Q$ is random, and let $q$ be an infinite ternary string that is random relative to $r$ (and so by Theorem 3.2.12 $r \oplus q$ is random). We claim the ghost code $g$ obtained by using the bits of $r$ as the canonical nodes and the bits of $q$ in their original order as the ghost nodes is random. It is clear that $g$ is a ghost code for $Q$.

Suppose $m$ is a c.e. martingale that bets on $g$. From $m$ it is straightforward to define a nonmonotonic martingale $m'$ which mimics $m$’s bets exactly but performs them on $r \oplus q$, ...
succeeding whenever \( m \) succeeds. As \( r \) and \( q \) were chosen to be relatively random, this will show \( g \) is random.

As discussed previously, from \( g \upharpoonright n \) it is computable whether \( g(n) \) will be a ghost node or a canonical node, and which position in \( g \) or \( r \) it occupies in either case. Therefore, assuming the bits seen so far may be assembled into an initial segment \( \sigma \) of \( g \), \( m' \) takes the values \( m(\sigma \upharpoonright i), \ i < 3 \), as its bets on the corresponding bit of \( r \) or \( g \), whichever is appropriate. Having seen that bit, then, it can assemble a \((|\sigma| + 1)\)-length initial segment of \( g \) and repeat the process. As \( m' \) makes identical bets to \( m \) and has identical outcomes, since it cannot succeed on \( r \oplus g \), \( m \) cannot succeed on \( g \) and \( g \) is random.

To relativize \((\rightarrow)\), suppose that \( r \) is \( y \)-random, so that \( r \oplus y \) is random by Van Lambalgen’s Theorem 3.2.12. Then in the proof simply choose \( q \) to be random relative to \( r \oplus y \), and then \( g \) will be random relative to \( y \). The other direction relativizes in a straightforward way.

\[ \square \]

3.3.6 Coding and Joins of Closed Sets

The primary purpose of the ghost codes is to remove the dependence on the particular closed set under discussion when interpreting bits of the code as nodes of the tree. This is especially useful when subdividing the tree, as in the following definition.

**Definition 3.3.10.** The tree join of closed sets \( P_0 \) and \( P_1 \) is the closed set

\[
Q = \{0^\upharpoonright x : x \in P_0\} \cup \{1^\upharpoonright x : x \in P_1\}.
\]

Given ghost codes \( r_0, r_1 \) for the \( P_i \), their tree join \( r_0 \oplus r_1 \) is the code for \( Q \) with the corresponding ghost node values.

The standard recursion-theoretic join is defined by

\[
r_0 \oplus r_1 = (r_0(0), r_1(0), r_0(1), r_1(1), \ldots).
\]

We wish to relate the recursion-theoretic join and the tree join.

**Lemma 3.3.11.** Given two ghost codes \( r_0, r_1 \), the tree join \( r_0 \oplus r_1 \) is random if and only if the recursion theoretic join \( r_0 \oplus r_1 \) is random.
It is clear that there is a computable permutation $\pi$ which uniformly maps any tree join $r_0 \boxplus r_1$ to the recursion-theoretic join $r_0 \oplus r_1$. That is, in $r_0 \oplus r_1$, the entries of $r_0$ and $r_1$ alternate, whereas $r_0 \boxplus r_1$ starts with a 2, followed by blocks from $r_0$ and $r_1$, as follows. First $r_0(0), r_1(0)$, then $r_0(1), r_0(2), r_1(1), r_1(2)$, and continuing with pairs of blocks of size 4, 8 and so on. The result now follows from the Von-Mises–Church–Wald Computable Selection Theorem [94]; the theorem states that, for any random sequence $x$ and any computable 1-1 function $g$, the sequence $z(n) = x(g(n))$ is random.

We now obtain the following corollary of Theorems 3.2.12 and 3.3.9 and Lemma 3.3.11.

**Corollary 3.3.12.** Suppose $P_i, i = 0, 1$, are closed sets with canonical codes $r_i$ and let $P$ be the tree join of $P_0, P_1$. Then $P$ is random if and only if $r_0 \oplus r_1$ is random.

**Proof.** ($\leftarrow$) Suppose that $r_0 \oplus r_1$ is random. Then by Theorem 3.2.12, $r_0$ and $r_1$ are mutually relatively random. By Theorem 3.3.9, $P_0$ has a ghost code $g_0$ which is random relative to $r_1$, and so also vice-versa, and then $P_1$ has a ghost code $g_1$ which is random relative to $g_0$. Again by 3.2.12, the recursion-theoretic join $g_0 \oplus g_1$ is random, so by Theorem 3.3.11 the tree join $g_0 \boxplus g_1$ is also random, and hence $P$ possesses a random ghost code and is random.

($\rightarrow$) Suppose now that $P$ is random, and therefore possesses a random ghost code $g$. The code $g$ may be thought of as a tree join $g_0 \boxplus g_1$, which is therefore random, and so by Theorem 3.3.11, $g_0 \oplus g_1$ is random. By Theorem 3.2.12, the individual codes $g_0, g_1$ are therefore mutually relatively random. Now by the relativated version of Theorem 3.3.9, $r_0$ is random relative to $g_1$. But $r_1$ is computable from $g_1$ and hence $r_0$ is random relative to $r_1$ as well. Similarly, $r_1$ is $r_0$-random and thus, again by 3.2.12, $r_0 \oplus r_1$ is random.

### 3.4 Members of Random Closed Sets

In this section we tackle the question of which types of elements necessarily belong, or do not belong, to random closed sets. The former is addressed in Section 3.4.1 and the latter in Section 3.4.2.
3.4.1 Positive Results

We shall see, as a consequence of Theorem 3.4.19, that every closed set is perfect and contains continuum many elements. In this section, we demonstrate other positive results. For example, every random closed set contains random and nonrandom elements. Other examples abound. We begin with the first example.

**Theorem 3.4.1.** Every random closed set contains a random element.

**Proof.** Suppose that a closed set $Q$ has no random element and consider the following Martin-Löf test on the space $C$:

$$U_i = \{P \mid P \in C \text{ and } P \subseteq V_i\}$$

where $(V_i)$ is a universal Martin-Löf test on the Cantor space. By Lemma 3.3.8, $\mu^*(U_i) \leq \mu(V_i) \leq 2^{-i}$ so that $(U_i)$ is a Martin-Löf test on $C$. But $Q \in \cap_i U_i$, so $Q$ is not random.

As a converse to Theorem 3.4.1 we have the following.

**Theorem 3.4.2.** For any random $r \in 2^{\mathbb{N}}$, there exists a random closed set containing $r$ as a path.

The proof of this theorem was originally given by Joe Miller and Antonio Montalbán and has been subsequently improved thanks to the anonymous referee.

**Proof.** Let $r$ be a random real and let $x$ be the canonical code of an $r$-random closed set. We alter $x$ to the code $x'$ of a closed set guaranteed to contain $r$ but changed as little as possible to achieve that.

To determine $x'(n)$, assume $x' \upharpoonright n$ has been defined. If $x(n) = 2$ or $x(n)$ corresponds to a node not along $r$, set $x'(n) = x(n)$. If $x(n) \in \{0,1\}$ corresponds to $r(k)$, set $x'(n) = r(k)$.

The closed set defined by $x'$ will clearly contain $r$. For a contradiction, assume $x'$ is nonrandom and let $m'$ be a c.e. martingale that succeeds on it. We build a nonmonotonic martingale $m$ to bet on $x \oplus r$. On bits of $x$, $m$ will be a triple-or-nothing martingale; on $r$, it will be double-or-nothing.
First note that from initial segments of $x$ and $r$ we may reconstruct an initial segment of $x'$ computably, and we always know from an initial segment of $x'$ whether the next bit is along $r$ or not, and which bit of $r$ it is. We will construct $m$ so that after every stage of betting (which will be one bet by $m'$ and one or two bets by $m$), the value of $m$ is equal to the value of $m'$. At every stage it will be clear we have revealed enough bits of $x$ and $r$ to reconstruct $x'$ to the needed length.

Suppose inductively $m$ and $m'$ hold equal capital after the stage of betting on the last node of $\sigma \sqsubseteq x'$. If the bit $x'(n)$ following $\sigma$ is not on $r$, $m$ bets identically to $m'$; i.e., $m(x(n) = i) = m'(\sigma \neg i)$ for $i < 3$. In that case $x(n) = x'(n)$ so our inductive hypothesis holds. If $x'(n)$ is on $r$, set $m(x(n) = 2) = m'(\sigma \neg 2)$ and for $i = 0, 1$, set $m(x(n) = i) = \frac{1}{2}[m'(\sigma \neg 0) + m'(\sigma \neg 1)]$. If $x'(n) = 2$, then the capital for both $m$ and $m'$ is $m'(\sigma \neg 2)$, so the inductive hypothesis holds and we proceed to the next stage. Otherwise $m$ bets on $r(k)$ for the appropriate $k$, setting $m(r(k) = i) = m'(\sigma \neg i)$ for $i = 0, 1$. On $r(k)$, the sum of $m$’s capital on each of the two outcomes must average to the previous capital; as the previous capital was $\frac{1}{2}[m'(\sigma \neg 0) + m'(\sigma \neg 1)]$ this clearly holds. By construction $r(k) = x'(n) = i$, so both $m$ and $m'$ now have capital $m'(\sigma \neg i)$ and the inductive hypothesis holds. As $m'$ is c.e., $m$ will also be.

As the values of $m'$ along $x'$ are a subsequence of the values of $m$ along $x \oplus r$, if $m'$ succeeds so does $m$, contradicting our assumption on $x \oplus r$. Therefore $x'$ is the code of a random closed set containing the given random path $r$.

The previous results might suggest that every element of a random closed set is a random real. However, it turns out that every random closed set contains a non-random real.

We need the following classic result of Chernoff [28] (a version of Bernoulli’s Weak Law of Large Numbers) here and also for another theorem to follow. See [67] for an exposition.

**Lemma 3.4.3** (Chernoff). Let $E$ be an event which we will refer to as ‘success’. If $E$ occurs with probability $p$, then for any natural numbers $n$ and any $\varepsilon$ with $0 \leq \varepsilon \leq 1$, the
probability that out of \( n \) mutually independent trials, the number of successes differs from \( pn \) by \( > \varepsilon pn \) is \( \leq 2^{-\varepsilon^2 pn/3} \).

**Theorem 3.4.4.** Every random closed set contains a non-random real; in particular, the leftmost and rightmost paths in a random closed set are not random reals.

**Proof.** We will show that, for a random closed set \( Q \), the leftmost path is not stochastically random, that is, the asymptotic frequency of 0’s is \( \frac{2}{3} \). Since an effectively random real in \( 2^\mathbb{N} \) must have asymptotic frequency of \( \frac{1}{2} \) for 0’s and 1’s, this will suffice to prove that the leftmost path is not random. We define a Martin-Löf test as follows. Fix a rational \( \varepsilon \) such that \( 0 < \varepsilon < 1 \). For each \( n \), let \( S_n \) be the family of closed sets (that is, codes for closed sets) such that the first \( n \) bits of the leftmost path have either \( < \frac{2}{3}(1 - \varepsilon)n \), or \( > \frac{2}{3}(1 + \varepsilon)n \) occurrences of 0. By the definition of our probability measure, we have

\[
\mu^*(S_n) = \sum_{|m - \frac{2}{3}n| > \frac{2}{3} \varepsilon n} \binom{n}{m} \left( \frac{2}{3} \right)^m \left( \frac{1}{3} \right)^{n-m}.
\]

It now follows from Chernoff’s Lemma 3.4.3 that

\[
\mu^*(S_n) \leq 2 e^{-\varepsilon^2 2n/9}.
\]

Thus the measures of the test sets \( S_n \) have effective limit zero. It is easy to see that the sequence \( \{S_n\} \) is computably enumerable. For each \( n \), \( S_n \) is a clopen set and in fact the union of the finite family of intervals \( I(\sigma) \) in \( C \) such that \( \sigma \) codes a tree up to level \( n \) in which the leftmost path has either \( < \frac{2}{3}(1 - \varepsilon)n \), or \( > \frac{2}{3}(1 + \varepsilon)n \) occurrences of 0.

Furthermore, \( S'_n = \bigcup_{p \geq n} S_p \) is also a Martin-Löf test. It follows that for any random closed set \( Q \), and any \( \varepsilon > 0 \), there is an \( n \) such that for all \( m \geq n \), the frequency of 0’s in the first \( m \) bits of the leftmost path is always within \( \varepsilon \) of \( \frac{2}{3} \). Thus the leftmost path is not effectively random.

Recall that the leftmost and rightmost elements of any strong \( \Delta^0_2 \) closed set are \( \Delta^0_2 \). Given Theorems 3.4.1 and 3.4.4, we ask: Does a \( \Delta^0_2 \) random closed set contain a \( \Delta^0_2 \) random path?

**Theorem 3.4.5.** Every random strong \( \Delta^0_2 \) closed set contains a random \( \Delta^0_2 \) real.
**Proof.** Let $Q$ be a random strong $\Delta^0_2$ class. By Theorem 3.4.1, $Q$ contains a random real $x$. Let $P$ be a $\Pi^0_1$ class in the Cantor space which contains only randoms and contains $x$ (this exists since the class of random reals is an effective union of $\Pi^0_1$ classes). Note that $P \cap Q$ is a non-empty strong $\Delta^0_2$ class and it follows that the leftmost path of $P \cap Q$ is a $\Delta^0_2$ real which must be random since it belongs to $P$. \hfill \square

The above theorem does not combine with the low basis theorem to establish the existence of a low random real in any random strong $\Delta^0_2$ class. We can use the low basis theorem, however, to demonstrate the existence of a low random real in any random closed set with low canonical code.

**Theorem 3.4.6.** Every random closed set with low canonical code contains a low random element.

**Proof** (Kjos-Hanssen). Let $Q$ be a random closed set with low canonical code. By Theorem 3.4.1, $Q$ contains a random element $x$. Therefore $x \in 2^\omega \setminus U_n$ for some $n$ and some open $U_n$ from the universal Martin-Löf test. So, in particular, $Q \cap 2^\omega \setminus U_n$ is non-empty. Now $Q \cap 2^\omega \setminus U_n$ is $\Pi^0_1$ relative to $T_Q$. By the low basis theorem, $Q \cap 2^\omega \setminus U_n$ is $\Pi^0_1$ has a member $y$ such that $y' \leq_T T'_Q$. In particular, since $y \in Q \cap 2^\omega \setminus U_n$, it is random. Furthermore, since $T_Q$ is low, $y$ is also low. \hfill \square

It is open whether every random closed set with a $\Delta^0_2$ canonical code has a low random element; we conjecture that the answer is no. In the following section, we will show that there is a random closed set not containing any $\Delta^0_2$ path.

Our next result, Theorem 3.4.8, uses a method which was used in [56] to show that every random real is effectively bi-immune. We first define this latter notion.

**Definition 3.4.7.** (i) A set $A$ is effectively immune if it is infinite and there is a computable function $g(x)$ such that if $W_x \subseteq A$, $|W_x| \leq g(x)$.

(ii) $A$ is effectively bi-immune if $A$ and $\overline{A}$ are both effectively immune.

Note, in particular, that an effectively immune set cannot contain an infinite c.e. set.

**Theorem 3.4.8.** If $P$ is a random closed set then all elements of $P$ are effectively bi-immune.
Proof. Suppose that $P$ is a random closed set and $A \in P$. Let $(U_i)_{i \in \mathbb{N}}$ be a Martin-Löf test (in the space $3^\mathbb{N}$) such that there is a computable function $f$ with the property that if $(V_i)_{i \in \mathbb{N}}$ is the $e$-th Martin-Löf test (under some effective enumeration of all Martin-Löf tests) then for all $e, i$, $V_{f(e,i)}^e \subseteq U_i$ (a standard construction of a universal test gives one with this property). Since $P$ is random, there is some $k$ such that (the canonical code of) $P$ is not in $U_k$; let $U = U_k$. It suffices to find a computable function $g$ such that

$$[A \in P \text{ and } W_x \subseteq A] \Rightarrow |W_x| \leq g(x)$$

for all sets $A$ and all $x$ (the proof that $\overline{A}$ is effectively immune is entirely similar). Let $B_{x,n}$ be $\emptyset$ if $|W_x| \leq n$, and otherwise the class of (canonical codes of) trees which contain a path containing the first $n + 1$ elements in the standard enumeration of $W_x$. Then $(B_{x,n})$ is a uniform double sequence of $\Sigma^0_1$ classes and (by the definition of the probability measure on $3^\mathbb{N}$),

$$\mu^*(B_{x,n}) \leq \left(\frac{2}{3}\right)^{n+1}.$$ 

So for each $x$, $(B_{x,2n})$ is a Martin-Löf test in the space $3^\mathbb{N}$ and from $x$ we can calculate the index of it. Then by using the computable function $f$ mentioned above we get a computable function $g$ such that for all $x$, $B_{x,g(x)} \subseteq U$. This means $g$ satisfies (3–1).

It is well known that effectively immune sets can compute a fixed point free function, so we have the following.

**Corollary 3.4.9.** The paths through a random tree are of fixed point free degree. That is, each of them computes some fixed point free function.

It is known that every real can be computed by some random real. It is not known, however, whether any real can be computed by all the paths of some random closed set. The next theorem, an observation of Ted Slaman at a randomness workshop in Chicago in 2007, is a step in that direction. First, we need the following definitions, which are, in fact, equivalent notions.

**Definition 3.4.10.** (i) A real $x$ is $K$-trivial if $K(x|n) \leq K(n) + c$ for some $c$. 

72
A real $x$ is a base for 1-randomness if there is some $y \geq_T x$ such that $y$ is 1-$x$-random.

**Theorem 3.4.11.** Any set which is computable from all paths of a random closed set is K-trivial.

**Proof.** If $A$ is computable from all paths, then $A$ is computable from the leftmost and rightmost paths. Note that each of these latter paths are each computable from the two halves of the tree. Furthermore these two halves are relatively random to each other. Hence each is random relative to anything the other computes. So half 1, for example, computes $A$ and hence half 2 is random relative to $A$. On the other hand, half 2 also computes $A$. Therefore this is an example of something in the cone above $A$ that is $A$-random. So $A$ is a base for 1-randomness. \qed

Much interesting work has been done on the $K$-trivial reals. Chaitin showed that if $A$ is $K$-trivial, then $A \leq_T 0'$. Solovay constructed a noncomputable $K$-trivial real. Downey, Hirschfeldt, Nies and Stephan [39] showed that no $K$-trivial real is c.e. complete. The notion of a $K$-trivial closed set was introduced in [9]. It was shown in particular that every $K$-trivial class contains a $K$-trivial member, but there exist $K$-trivial $\Pi^0_1$ classes with no computable members.

### 3.4.2 Negative Results

Random closed sets can never contain $n$-c.e., isolated, or 1-generic paths, or paths of incomplete c.e. degree. We build to these facts and prove others along the way.

**Theorem 3.4.12.** Random closed sets contain no computable elements.

**Proof.** For any finite string $\sigma$ of length $n$, the probability that a closed set $Q$ meets $I(\sigma)$ is $(\frac{2}{3})^n$. For a computable real $y$, the sequence $\{Q : Q \cap I(y[\sigma]) \neq \emptyset\}$ thus forms a Martin-Löf test in the space $C$ of closed sets, which shows that $y$ does not belong to any Martin-Löf random closed set. That is, for each $n$, $\{x : Q_x \cap I(y[n]) \neq \emptyset\}$ is a c.e. open set and has measure $\left(\frac{2}{3}\right)^n$ in $\{0,1,2\}^N$, where $Q_x$ is the closed set with code $x$. \qed

We prove an even stronger result in Theorem 3.4.17. First, however, recall that a $\Pi^0_1$ class $P$ is decidable if $T_P$ is decidable. It follows that a nonempty decidable $\Pi^0_1$ class
must contain a computable element (for example, the leftmost path). By Theorem 3.4.12, it follows that no decidable $\Pi^0_1$ class can be random. As every random class contains a random element (Theorem 3.4.1) and has, as we shall show, measure zero (Theorem 3.5.1), the following proposition demonstrates that this extends to arbitrary $\Pi^0_1$ classes.

**Proposition 3.4.13.** If $P$ is a $\Pi^0_1$ class of measure 0, then $P$ has no random elements.

**Proof.** Let $T$ be a computable tree such that $P = [T]$, and for each $n$, let $P_n = \bigcup \{ I(\sigma) : \sigma \in T \cap \{0, 1\}^n \}$. Then $\{P_n\}_{n \in \mathbb{N}}$ is an effective sequence of clopen sets with $P = \bigcap_n P_n$ and $\lim_n \mu(P_n) = \mu(P) = 0$. Furthermore, $\mu(P_n) = 2^{-n}|T \cap \{0, 1\}^n|$; this is a computable sequence. Thus $\{P_n\}_{n \in \mathbb{N}}$ is a Martin-Löf test and $P$ has no random elements.

Alternatively, we can show that no $\Pi^0_1$ class is random through the following stronger result, combined with an appeal to Theorem 3.5.1, from the next section.

**Theorem 3.4.14.** Let $Q$ be a $\Pi^0_1$ class with measure 0. Then no subset of $Q$ is random.

**Proof.** Let $T$ be a computable tree (possibly with dead ends) and let $Q = [T]$. Then $Q = \bigcap_n U_n$, where $U_n = [T_n]$. Since $\mu(Q) = 0$, it follows from Lemma 3.3.8 that $\lim_n \mu^*(\mathcal{P}_C(U_n)) = 0$. But $\mathcal{P}_C(U_n)$ is a computable sequence of clopen sets in $\mathcal{C}$ and $\mu^*(\mathcal{P}_C(U_n))$ is a computable sequence of rationals with limit 0. Thus $\mathcal{P}_C(U_n)$ is a Martin-Löf test, so that for any random closed set, there exists $n$ such that $P \notin \mathcal{P}_C(U_n)$ and hence $P$ is not a subset of $U_n$.

**Corollary 3.4.15.** No $\Pi^0_1$ class can be random.

We now provide an even stronger version of Theorem 3.4.12; we need the following definition.

**Definition 3.4.16 (f-c.e. reals).** For any computable, non-decreasing function $f$, we say that a real $\beta \in \{0, 1\}^\mathbb{N}$ is $f$-c.e. if there exists a computable approximating function $\phi$ such that, for all $i \in \mathbb{N}$,

(i) $\phi(i, 0) = 0$

(ii) $\lim_s \phi(i, s) = \beta(i)$;

(iii) $\{ s : \phi(i, s + 1) \neq \phi(i, s) \}$ has cardinality $\leq f(i)$. 

74
The reals which are $f$-c.e. for some computable function $f$ are part of the well-known Ershov hierarchy \[43, 86\].

**Theorem 3.4.17.** Suppose that $f$ is computable and bounded by a polynomial. Then no random closed set has any $f$-c.e. paths.

**Proof.** Let $f$ be as above, $\beta$ an $f$-c.e. real and $P$ a closed set containing $\beta$. Let $\phi$ be the $f$-approximating function for $\beta$. Also let $M_n \subseteq \{0,1\}^n$ be the set of different $\phi$-approximations to $\beta\upharpoonright n$ during the stages.

A priori, $|M_n|$ is exponential. However, for a fixed $n$, $\beta\upharpoonright n$ can change at most $\sum_{i<n} f(i)$ times, so $|M_n|$ is also bounded by a polynomial, i.e. there is $k \in \mathbb{N}$ such that for almost all $n$, $|M_n| < n^k$. Now let

$$S_n = \bigcup_{\sigma \in M_n} \{P \mid P \in C \& P \cap I(\sigma) \neq \emptyset\}. \quad (3-2)$$

Then $(S_n)$ is a uniformly c.e. sequence of open sets in the space $C$ of closed sets of $2^\mathbb{N}$ and for all $n$, $P \in S_n$. Also for almost all $n$,

$$\mu^*(S_n) \leq \sum_{\sigma \in M_n} \mu^*(\{P \mid P \in C \& P \cap I(\sigma) \neq \emptyset\}) = |M_n| \cdot \left(\frac{2}{3}\right)^n \leq n^k \cdot \left(\frac{2}{3}\right)^n.$$ 

Since $\lim_n [n^k \cdot \left(\frac{2}{3}\right)^n] = 0$ there is a computable subsequence of $(S_n)$ which is a Martin-Löf test and so $P$ is not random.

For any $K$-trivial real $A$ and any unbounded nondecreasing computable function $h$, $A$ is $h$-c.e. (Nies \[75\]). Thus it follows from Theorem 3.4.17 that a random closed set can have no $K$-trivial paths. We observe that Theorem 3.4.17 cannot be extended to $\omega$-c.e. in general, because there are left-c.e. (and hence $\omega$-c.e.) random reals, and by Theorem 3.4.2 each of these belongs to a random closed set.

Also, recall from Corollary 3.4.9 that that paths through a random tree are of fixed point free degree. It is known that fixed point free degrees cannot be 1-generic (see \[42\] for a proof) or incomplete c.e., and that if they are $\Delta_2^0$ they compute a promptly simple set and no pair of them forms a minimal pair (see \[57\]). So we have the following.

**Corollary 3.4.18.** The following hold:
• No path of a random tree is 1-generic.
• No pair of $\Delta^0_2$ paths of a random tree can be a minimal pair.
• Every $\Delta^0_2$ path of a random tree computes a promptly simple set.
• No path of a random tree can have incomplete c.e. degree.

**Theorem 3.4.19.** If $Q$ is a random closed set, then $Q$ has no isolated elements.

**Proof.** Let $Q = [T]$ and suppose by way of contradiction that $Q$ contains an isolated path $x$. Then there is some node $\sigma \in T$ such that $Q \cap I(\sigma) = \{x\}$. For each $n$, let

$$S_n = \{P \in C : |\{\tau \in \{0, 1\}^n : P \cap I(\sigma \upharpoonright \tau) \neq \emptyset\}| = 1\}.$$

That is, $P \in S_n$ if and only if the tree $T_P$ has exactly one extension of $\sigma$ of length $n + |\sigma|$. It follows that

$$|P \cap I(\sigma)| = 1 \iff (\forall n) P \in S_n.$$

Now for each $n$, $S_n$ is a clopen set in $C$ and again by induction, $S_n$ has measure $(\frac{2}{3})^n$. Thus the sequence $S_0, S_1, \ldots$ is a Martin-Löf test. It follows that for some $n$, $Q \notin S_n$. Thus there are at least two extensions in $T_Q$ of $\sigma$ of length $n + |\sigma|$, contradicting the assumption that $x$ was the unique element of $Q \cap I(\sigma)$.

As mentioned previously, it follows that every random closed set is perfect and hence contains continuum many elements.

Next we want to find a random closed set which does not contain a $\Delta^0_2$ path. Now it is easy [20, 24] to construct a strong $\Pi^0_2$ class $P$ of positive measure which contains no $\Delta^0_2$ elements; of course $P$ must contain a random real since it has measure 1. The difficult problem is to construct a random strong $\Pi^0_2$ class with no $\Delta^0_2$ elements. We have the following result in this direction, which yields a random strong $\Delta^0_3$ closed set with no $\Delta^0_2$ elements.

**Theorem 3.4.20.** For any set $A$ there is an $A$-random closed set $Q$ such that $T_Q \leq_T A''$ but $Q$ has no elements $\leq_T A'$.

**Proof.** It is enough if we prove the claim for $A = \emptyset$ because the argument relativises to any oracle $A$ in a straightforward way. For $A = \emptyset$ we use a finite injury construction over
∅' to construct $Q$ with the above properties. In the construction we will $∅'$-approximate the canonical code of a tree $T$ which has no $\Delta^0_2$ paths. To make sure that the tree $T$ is random we fix a $\Pi^0_1$ class $P$ of positive measure in the space $3^N$ (where the code for $T$ lies) which contains only randoms, and we make sure that at every stage our approximation (as a finite ternary string) to $T$’s canonical code can be extended to a path in $P$. Then by compactness the canonical code of our tree will be in $P$ and so the tree will be random.

The changes in the approximations are motivated by the requirements:

$$R_e : \text{if } \Phi^∅'^e \text{ is total then the real it defines is not in } [T].$$

Let $α_s$ be a finite string approximation of the canonical code $α$ we are building. We will have $|α_s| = s$. Strategy $R_e$ will come into power after stage $e$ and will restrain $α$ up to some $r_e \geq e$ (the default value is $r_e[0] = e$). Also it might request some changes in $α$ after the $e$-th bit. We start with $α_0 = ∅$ and at stage $s + 1$, assuming inductively that $α_s \downarrow$ and $[α_s] \cap P \neq ∅$ we ask for the least $i < s$ such that $R_i$ requires attention. This happens if

(i) The longest defined initial segment $τ$ of $Φ^∅'_{i,s+1}$ is larger than ever before;

(ii) there exists $σ \in \{0, 1, 2\}^*$ such that $α_s[(\max_{j<i} r_j[s]) \subseteq σ, I(σ) \cap P \neq ∅, |σ| = s + 1,$ and $τ$ is not consistent with the finite tree with code $σ$.

If there is no such $i$ then we extend $α_s$ by one bit such that $[α_{s+1}] \cap P \neq ∅$. Otherwise we let $α_{s+1} = σ$ and $r_i[s+1] = s + 1$. The construction proceeds in a straightforward way and we can prove inductively that for every $e$, $R_e$ is satisfied, stops requiring attention and $r_e$ reaches a limit. Then the limit $α = \lim_s α_s$ exists and we also have that $α$ is random by compactness. The satisfaction of the requirements comes from a measure-theoretic fact. Consider $R_e$ and inductively assume that after stage $s_e$ no $R_i$ with $i < e$ requires attention. Then $r = \max_{i\leq e} r_i$ will remain constant. Since $P$ contains only randoms and $[α[\max_{i\leq e} r_i] \cap P \neq ∅$, $μ([α[r] \cap P] > 0$

and on the other hand, if $β = Φ_e^∅'$ we have seen that $μ\{γ | γ \in 3^N \text{ and } γ \text{ is the canonical code of a tree which has } β \text{ as a path}\} = 0$. 77
This means that if at stage $s_e$ the requirement $R_e$ is not yet satisfied, it will receive
attention at a later stage and get satisfied permanently.

### 3.5 Measure and Dimension

In this section we show that random closed set have measure zero (Theorem 3.5.1) and box dimension $\log_2 \frac{4}{3}$ (Theorem 3.5.2).

#### 3.5.1 Measure

**Theorem 3.5.1.** If $Q$ is a random closed set, then $\mu(Q) = 0$.

**Proof.** We will show that in the space $C$ of closed sets, the $\mu^*$-probability that a closed set $P$ has Lebesgue measure 0, is 1. This is proved by showing that for each $m$, $\mu(P) \geq 2^{-m}$ with $\mu^*$-probability 0. For each $m$, let

$$S_m = \{ P : \mu(P) \geq 2^{-m} \}.$$

We claim that for each $m$, $\mu^*(S_m) = 0$. The proof is by induction on $m$.

For $m = 0$, we have $\mu(P) \geq 1$ if and only if $P = 2^N$, which is if and only if $x_P = (2, 2, \ldots)$, so that $S_0$ is a singleton and thus has measure 0.

Now assume by induction that $S_m$ has measure 0. Then the probability that a closed set $P = [T]$ has measure $\geq 2^{-m-1}$ can be calculated in two parts.

(i) If $T$ does not branch at the first level, say $T_0 = \{(0)\}$ without loss of generality. Now consider the closed set $P_0 = \{ y : 0^\sim y \in P \}$. Then $\mu(P) \geq 2^{-m-1}$ if and only if $\mu(P_0) \geq 2^{-m}$, which has probability 0 by induction, so we can discount this case.

(ii) If $T$ does branch at the first level, let $P_i = \{ y : i^\sim y \in P \}$ for $i = 0, 1$. Then $\mu(P) = \frac{1}{2}(\mu(P_0) + \mu(P_1))$, so that $\mu(P) \geq 2^{-m-1}$ implies that at least one of $\mu(P_i) \geq 2^{-m-1}$. (Note that the reverse implication is not always true.) Let $p = \mu^*(S_{m+1})$. The observations above imply that

$$p \leq \frac{1}{3}(1 - (1 - p)^2) = \frac{2}{3}p - \frac{1}{3}p^2,$$

and therefore $p = 0$. 

78
To see that a random closed set $Q$ must have measure 0, fix $m$ and let $S = S_m$. Then $S$ is the intersection of an effective sequence of clopen sets $V_\ell$, where for $P = [T]$, 

$$P \in V_\ell \iff \mu([T_\ell]) \geq 2^{-m}.$$ 

Since these sets are uniformly clopen, the sequence $m_\ell = \mu^*(V_\ell)$ is computable. Since $\lim_\ell m_\ell = 0$, it follows that this is a Martin-Löf test and therefore no random set $Q$ belongs to $\bigcap_\ell V_\ell$. Then in general, no random set can have measure $\geq 2^{-m}$ for any $m$.  

### 3.5.2 Dimension

Surprisingly, we can compute the (Kolmogorov) box dimension of a random closed set, and in fact it turns out that all random closed sets have the same dimension. The intuition for this comes from the following lemma. For any function $F$ mapping the space $\mathcal{C}$ of closed sets into $\mathbb{R}$, the expected value of $F$ on $\mathcal{C}$ is the integral $\int F(P)$ with respect to the probability measure $\mu^*$.

**Lemma 3.5.2.** In the space $\mathcal{C}$ of closed sets, the expected cardinality of $\{\sigma \in \{0,1\}^n : Q \cap I(\sigma) \neq \emptyset\}$ is exactly $\left(\frac{4}{3}\right)^n$ for every $n$, where $Q$ is chosen uniformly at random according to $\mu^*$.

**Proof.** Let $S_n = \{\sigma \in \{0,1\}^n : Q \cap I(\sigma) \neq \emptyset\}$, for a randomly chosen $Q$ from $\mathcal{C}$.

The proof is by induction on $n$. For $n = 1$, we have two cases. With probability $\frac{2}{3}$, $\text{card}(S_1) = 1$ and with probability $\frac{1}{3}$, $\text{card}(S_1) = 2$. Thus the expected value is exactly $\frac{4}{3}$. For $n + 1$, there are again two cases. With probability $\frac{2}{3}$, $\text{card}(S_1) = 1$, so that the expected $\text{card}(S_{n+1})$ equals the expected $\text{card}(S_n)$, which is $\left(\frac{4}{3}\right)^n$ by induction. With probability $\frac{1}{3}$, $\text{card}(S_1) = 2$, in which case the expected $\text{card}(S_{n+1})$ is twice the expected $\text{card}(S_n)$, that is, $2\left(\frac{4}{3}\right)^n$. Thus we have the expected value 

$$\text{card}(S_{n+1}) = \frac{2}{3} \left(\frac{4}{3}\right)^n + \frac{1}{3} \cdot 2 \left(\frac{4}{3}\right)^n = \left(\frac{4}{3}\right)^{n+1}.$$ 

\qed
The box dimension of a closed set in the Cantor space, if it exists, is given by the following limit:

$$\dim_B F(Q) = \lim_{n \to \infty} \frac{\log_2(\text{card}(T_Q \cap \{0, 1\}^n))}{n}.$$  

(See [6] for this formulation of the box dimension in $\{0, 1\}^N$.) Now by Lemma 3.5.2, the expected value of $\text{card}(T_Q \cap \{0, 1\}^n)$ for a random closed set $Q$ is $(\frac{4}{3})^n$, which suggests that the box dimension of $Q$ should be $\log_2 \frac{4}{3}$.

**Lemma 3.5.3.** Let $Q$ be a random closed set. Then for any $\varepsilon > 0$, there exists a $m \in \mathbb{N}$ such that, for all $n > m$, $(\frac{4}{3})^n(1 - \varepsilon)^n < \text{card}(T_Q \cap \{0, 1\}^n) < (\frac{4}{3})^n(1 + \varepsilon)^n$.

**Proof.** For each $n$, let $c_n(Q)$, or just $c_n$, denote $\text{card}(T_Q \cap \{0, 1\}^n)$. We will use three applications of Chernoff’s Lemma 3.4.3. First we show that there exists $m$ such that for all $n > m$, $c_{6n} \geq n$. Since the tree $T_Q \cap \{0, 1\}^{6n-1}$ has at least $6n$ nodes, it follows from Chernoff’s Lemma that the number of branching nodes is less than $n$ with probability $\leq 2^{-n/6}$. Thus $c_{6n} < n$ with probability $< 2^{-n/6}$. Then the probability that $c_{6n} < n$ for any $n \geq m$ is less than

$$\sum_{n=m}^{\infty} 2^{-n/6} = \frac{2^{-m/6}}{1 - 2^{-1/6}}.$$  

This provides a computable sequence of clopen sets with measures bounded by a computable sequence with limit zero and hence a Martin-Löf test. It follows that for any random closed set $Q$, there exists $m_0$ such that $c_{6n} \geq n$ for all $n \geq m_0$. Now for $n > m_0$, there are at least $6n^2$ nodes in $T_Q \cap \{0, 1\}^{12n-1} - \{0, 1\}^{6n-1}$, so that again by Chernoff’s Lemma, the probability that $< n^2$ of these are branching nodes is $\leq 2^{-n^2/6}$. It follows as above that there exists $m_1 > 3$ such that $c_{12n} \geq n^2$ for all $n \geq m_1$. Now suppose that $m \geq 12m_1$ and that $12n \leq m < 12(n + 1) < 16n$. Then $n \geq m_1$, so that

$$c_m \geq c_{12n} \geq n^2 > (m/16)^2.$$  

Again by Chernoff’s Lemma, the probability that the number of branching nodes from $T_Q \cap \{0, 1\}^n$ differs from $\frac{1}{3}c_n$ by $> \frac{1}{3}c_n^{-\frac{1}{4}}c_n$ is $< 2^{-\sqrt{n}/9}$. But this is exactly the probability that $c_{n+1}$ differs from $\frac{4}{3}c_n$ by $> \frac{1}{3}c_n^{-\frac{1}{4}}c_n$. For $n > m_1$, we know that $c_n \geq (\frac{n}{16})^2$, so that $\sqrt{c_n} \geq \frac{n}{16}$ and $c_n^{-\frac{1}{4}} \leq \frac{4}{\sqrt{n}}$ and hence $2^{-\sqrt{c_n}/9} \leq 2^{-n/144}$. Thus the probability $p_n$ that $c_{n+1}$
differs from $\frac{4}{3}c_n$ by more than $\frac{c_n}{\sqrt{n}}$ is $<2^{-n/144}$. Then the probability that for any $n \geq m_1$, $c_{n+1}$ differs from $\frac{4}{3}c_n$ by more than $\frac{4}{3\sqrt{n}}c_n$ is bounded by

$$\sum_{n=m}^{\infty} p_n = \sum_{n=m}^{\infty} 2^{-n/144} = \frac{2^{-m/144}}{1 - 2^{-144}}.$$ 

This again provides a Martin-Löf test which shows that for any random closed set $Q$, there exists $m_2$ so that for $n > m_2$,

$$\left(\frac{4}{3}\right) \left(1 - \frac{1}{\sqrt{n}}\right) c_n \leq c_{n+1} \leq \left(1 + \frac{1}{\sqrt{n}}\right) c_n.$$ 

Now given $\varepsilon$, choose $m \geq m_2$ so that $(1 + \frac{1}{\sqrt{m}})^2 < 1 + \varepsilon$ and $1 - \varepsilon < (1 - \frac{1}{\sqrt{m}})^2$.

Then for any $k$,

$$c_m \left(\frac{4}{3}\right)^{2k}(1 - \varepsilon)^k < c_m \left(\frac{4}{3}\right)^{2k}(1 - \frac{1}{\sqrt{m}})^{2k} < c_{m+2k} \left(\frac{4}{3}\right)^{2k}(1 - \frac{1}{\sqrt{m}})^{2k}.$$ 

Now let $k$ be large enough so that

$$(1 - \varepsilon)^{m+k} \leq c_m \leq (\frac{4}{3})^m(1 + \varepsilon)^{m+k}.$$ 

Then the desired inequality

$$\left(\frac{4}{3}\right)^n(1 - \varepsilon)^n < c_n < (\frac{4}{3})^n(1 + \varepsilon)^n.$$ 

will hold for even $n \geq m + 2k$. For odd $n$, this inequality will hold by the inequality (*) above.

**Theorem 3.5.4.** For any random closed set $Q$, the box dimension of $Q$ is $\log_2 \frac{4}{3}$.

**Proof.** Given $\varepsilon > 0$, let $m$ be given by Lemma 3.5.3. Then for $n > m$, we have

$$n \log_2 \frac{4}{3} + n \log_2 (1 - \varepsilon) \leq \log_2 (\text{card}(T_Q \cap \{0, 1\}^n)) \leq n \log_2 \frac{4}{3} + n \log_2 (1 + \varepsilon),$$

so that

$$\log_2 \frac{4}{3} + \log(1 - \varepsilon) \leq \frac{\log_2 (\text{card}(T_Q \cap \{0, 1\}^n))}{n} \leq \log_2 \frac{4}{3} + \log(1 + \varepsilon),$$

and therefore $\dim_B(Q) = \lim_n \frac{\log_2 (\text{card}(T_Q \cap \{0, 1\}^n))}{n} = \log_2 \frac{4}{3}$. 

81
3.6 Prefix-Free Complexity of Closed Sets

In this section, we consider randomness for closed sets in terms of incompressibility of trees. Of course, Schnorr’s theorem tells us that $P$ is random if and only if the code $x_P \in \{0, 1, 2\}^\mathbb{N}$ for $P$ is prefix-free random, that is, $K_3(x_P[n]) \geq n - O(1)$. (Schnorr’s theorem for arbitrary finite alphabets is shown in [18].) Here we write $K_3$ to indicate that we would be using a universal prefix-free function $U : \{0, 1, 2\}^* \rightarrow \{0, 1, 2\}^*$. However, many properties of trees and closed sets depend on the levels $T_n = T \cap \{0, 1\}^n$ of the tree. For example, if $[T_n] = \bigcup\{I(\sigma) : \sigma \in T_n\}$, then $[T] = \bigcap_n [T_n]$ and $\mu([T]) = \lim_{n \to \infty} \mu([T_n])$.

So we want to consider the compressibility of a tree in terms of $K(T_n)$. Now there is a natural representation of $T_n$ as a string of length $2^n$. That is, list $\{0, 1\}^n$ in lexicographic order as $\sigma_1, \ldots, \sigma_{2^n}$ and represent $T_n$ by the string $e_1, \ldots, e_{2^n}$ where $e_i = 1$ if $\sigma_i \in T$ and $e_i = 0$ otherwise. Henceforth we identify $T_n$ with this natural representation.

It is interesting to note that the code for $T_n$ will have a shorter length than the natural representation. For example, if $[T] = \{y\}$ is a singleton, then $x = y$ and for each $n$, the code for $T_n$ is $x[n]$. If $x$ is the code for the full tree $\{0, 1\}^*$, then $x = (2, 2, \ldots)$ and the code for $T_n$ is a string of $(2^n - 1)$ 2’s, those labels attached to nodes of length $< n$. For the remainder of this section, we will use $T_n$ to mean the natural representation and $x_n$ to mean the code.

**Question.** Is there is a formulation of randomness for closed sets in terms of the incompressibility of $T_n$?

It seems plausible that $P = [T]$ is random if and only if there is a constant $c$ such that $K(T_n) \geq 2^n - c$ for all $n$. However, we will see that this is not possible for any tree. On the one hand, in Section 3.6.1 we achieve a lower bound for incompressibility. That is, we show that if $P = [T]$ is random then there is a constant $c$ such that $K(T_n) \geq (\frac{7}{6})^n - c$ for all $n$. On the other hand, in Section 3.6.2, we see that the $2^n$ is too high of an incompressibility bound since there is some $c$ and some random closed set such that $K(T_n) \leq 2^n - c$ for all $n$. 

82
In a larger sense, we seek a formulation of randomness, in terms of the incompressibility of \( T_n \), for other objects such as \( \Pi_0^0 \) classes or \( \Pi_2^0 \) classes. In the following two sections we consider these questions and achieve some lower and upper bounds for these classes of objects.

### 3.6.1 Lower Complexity Bounds

First we give a lower bound for the prefix-free complexity of a random tree.

**Theorem 3.6.1.** If \( P \) is a random closed set and \( T = T_P \), then there is a constant \( c \) such that \( K(T_n) \geq \left( \frac{7}{6} \right)^n - c \) for all \( n \).

**Proof.** Let \( P = [T] \) be a random closed set. Let \( m \) be given by Lemma 3.5.3, for \( \varepsilon = \frac{7}{6} \), so that for \( n > m \),

\[
\text{card}(T_n) \geq \left( \frac{7}{6} \right)^n.
\]

It follows that the code \( x_n \) for \( T_n \) has length \( \geq \left( \frac{7}{6} \right)^n \). Since \( x \) is random, we know that, for \( n \geq m \),

\[
K_3(x_n) \geq \left( \frac{7}{6} \right)^n - a,
\]

for some constant \( a \). Now we can compute \( x_n \) from \( T_n \), so that

\[
K(T_n) \geq K_3(x_n) - b,
\]

for some constant \( b \). The result now follows.

That is, let \( U \) (mapping \( \{0,1\}^* \) to \( \{0,1\}^* \)) be a universal prefix-free Turing machine and let \( K(T_n) = \min\{|\sigma| : U(\sigma) = T_n\} \). Let \( M \) be a prefix-free machine \( M \) (mapping \( \{0,1\}^* \) to \( \{0,1,2\}^* \)) such that \( M(T_n) = x_n \). Then define \( V \) by

\[
V(\sigma) = M(U(\sigma)).
\]

Then \( K_V(x^n) \leq K(T_n) \), so that for some constant \( e \), \( K_3(x_n) \leq K(T_n) + e \) and hence

\[
K(T_n) \geq K_3(x_n) - e \geq \left( \frac{7}{6} \right)^n - b - e.
\]

\( \square \)
The standard example of a random real, Chaitin’s Ω [27], is a c.e. real and therefore $\Delta_2^0$. Thus there exists a $\Delta_2^0$ random tree $T$ and by Theorem 3.6.1, $K(T_\ell) \geq \left(\frac{7}{6}\right)^\ell - c$ for some $c$.

We have a more modest result for $\Pi_1^0$ classes. That is, there is an effectively closed set with not too much compressibility, in the following sense.

**Theorem 3.6.2.** There is a $\Pi_1^0$ class $P = [T]$ such that $K(T_n) \geq n$ for all $n$.

**Proof.** Recall the universal prefix-free machine $U$ and let $S = \{\sigma \in \text{Dom}(U) : |U(\sigma)| \geq 2^{\vert\sigma\vert}\}$. Then $S$ is a c.e. set and can be enumerated as $\sigma_1, \sigma_2, \ldots$. The tree $T = \bigcap_s T^s$ where $T^s$ is defined at stage $s$. Initially we have $T^0 = \{0, 1\}^*$. We say that $\sigma_t$ requires attention at stage $s \geq t$ when $\tau = U(\sigma_t) = T_n^s$ for some $n$ (so that $|\tau| = 2^n$) and $n \geq |\sigma_t|$. Action is taken by selecting some path $\rho_t \in T_s$ of length $n$ and defining $T^{s+1}$ to contain all nodes of $T^s$ which do not extend $\rho_t$. Then $\tau \neq T_{n+1}^s$ and furthermore $\tau \neq T_{n+1}^r$ for any $r \geq s + 1$ since future action will only remove more nodes from $T_n$.

At stage $s + 1$, look for the least $t \leq s + 1$ such that $\sigma_t$ requires action and take the action described if there is such a $t$. Otherwise, let $T^{s+1} = T^s$.

Let $A$ be the set of $t$ such that action is ever taken on $\sigma_t$. Recall from the Kraft Inequality that $\sum_t 2^{-|\sigma_t|} < 1$. Since $|\rho_t| \geq |\sigma_t|$, it follows that $\sum_{t \in A} 2^{-|\rho_t|} < 1$ as well. Now $\mu([T]) = 1 - \sum_{t \in A} 2^{-|\rho_t|} > 0$ and therefore $[T]$ is nonempty.

It follows from the construction that for each $t$, action is taken for $\sigma_t$ at most once.

Now suppose by way of contradiction that $U(\sigma) = T_n$ for some $\sigma_t$ with $|\sigma| \leq n$. There must be some stage $r \geq t$ such that for all $s \geq r$, $T_n^s = T_n$ and such that action is never taken on any $t' < t$ after stage $r$. Then $\sigma_t$ will require action at stage $r + 1$ which makes $T_{n+1}^r \neq T_n^r$, a contradiction. \hfill $\square$

There is a stronger result for closed $\Pi_2^0$ classes. Namely, there is a closed $\Pi_2^0$ class with the following stronger incompressibility property.

**Theorem 3.6.3.** There is a $\Pi_2^0$ class $P = [T]$ such that $K(T_\ell) \geq 2^{\sqrt{\ell}}$ for all $\ell$. 

84
Proof. We will construct a tree $T$ such that $T_{n,2}$ can not be computed from fewer than $2^n$ bits. We will assume that $U(\emptyset) \uparrow$ to take care of the case $n = 0$. At stage $s$, we will define the (nonempty) level $T_{s,2}$ of $T$, using an oracle for $\emptyset'$.

We begin with $T_0 = \{\emptyset\}^*$. At stage $s > 0$, we consider $D_s = \{\sigma \in \text{Dom}(U) : |\sigma| < 2^s\}$.

Since $U$ is prefix-free, $\text{card}(D_s) < 2^{2s}$. Now there are at least $2^{2^{s+1}}$ trees of height $s^2$ which extend $T_{(s-1),2}$ and we can use the oracle to choose some finite extension $T' = T_{s,2}$ of $T_{(s-1),2}$ such that, for any $\sigma \in D_s$, $U(\sigma) \neq T'$ and furthermore, $U(\sigma) \neq T_r$ for any possible extension $T_r$ with $s^2 \leq r$. That is, since there are $< 2^{2^s}$ finite trees which equal $U(\sigma)$ for some $\sigma \in D_s$, there is some extension $T'$ of $T_{(s-1),2}$ which differs from all of these at level $s^2$. We observe that the oracle for $\emptyset'$ is used to determine the set $D_s$.

At stage $s$, we have ensured that for any extension $T \subseteq \{0,1\}^*$ of $T_{s,2}$, any $\sigma$ with $|\sigma| \leq 2^{s^2}$ and any $n \geq s^2$, $U(\sigma) \neq T_n$. It is immediate that $K(T_n) \geq 2^{\sqrt{n}}$.

3.6.2 Upper Complexity Bounds

In Theorem 3.6.1 we achieved a lower bound of $(\frac{7}{6})^n$ for the prefix-free complexity of a tree $T_P$ of a random closed set $P$. It seems plausible that we might be able to achieve a higher bound of $2^n$. If true this would actually provide and immediate characterization of randomness of closed sets in terms of prefix-free complexity of trees. That is, a closed would be random iff $K(T_n) \geq 2^n - c$ for some constant $c$. (To see this, note that from $x|2^n$ we can compute $T_n$ uniformly so that $K_3(x|2^n) \geq K(T_n) - b$ for some $b$.) However the following theorem provides an upper complexity bound less than $2^n$, refuting such a possibility.

Theorem 3.6.4. For any tree $T \subseteq \{0,1\}^*$, there are constants $k > 0$ and $c$ such that $K(T_\ell) \leq 2^\ell - 2^{\ell-k} + c$ for all $\ell$.

Proof. For the full tree $\{0,1\}^*$, this is clear so suppose that $\sigma \notin T$ for some $\sigma \in \{0,1\}^m$. Then for any level $\ell > m$, there are $2^{\ell-m}$ possible nodes for $T$ which extend $\sigma$ and $T_\ell$ may
be uniformly computed from \( \sigma \) and from the characteristic function of \( T_\ell \) restricted to the remaining set of nodes. That is, fix \( \sigma \) of length \( m \) and define a prefix-free computer \( M \) as follows. The domain of \( M \) is strings of the form \( 0^\ell 1\tau \) where \( |\tau| = 2^\ell - 2^{\ell-m} \). \( M \) outputs the standard representation of a tree \( T_\ell \) such that no extension of \( \sigma \) is in \( T_\ell \) and such that \( \tau \) tells us whether strings not extending \( \sigma \) are in \( T_\ell \). It is clear that \( M \) is prefix-free and we have \( K_M(T_\ell) = \ell + 1 + 2^\ell - 2^{\ell-m} \). Thus \( K(T_\ell) \leq \ell + 1 + 2^\ell - 2^{\ell-m} + c \) for some constant \( c \). Now \( \ell + 1 < 2^{\ell-m-1} \) for sufficiently large \( \ell \) and thus by adjusting the constant \( c \), we can obtain \( c' \) so that

\[
K(T_\ell) \leq 2^\ell - 2^{\ell-m-1} + c'.
\]

The following theorem also refutes the possibility that \( K(T_\ell) > 2^{\ell-c} \) is a characterization of random closed sets in terms of prefix-free randomness. It shows that closed sets with small measure, such as random closed sets which have measure zero (see Theorem 3.5.1), are more compressible.

**Theorem 3.6.5.** If \( \mu([T]) < 2^{-k} \), then there exists \( c \) such that, for all \( \ell \),

\[
K(T_\ell) \leq 2^{\ell-k+1} + c.
\]

**Proof.** Suppose that \( \mu([T]) < 2^{-k} \). Then for some level \( n \), \( T_n \) has \( < 2^{n-k} \) nodes \( \sigma_1, \ldots, \sigma_t \). Now for any \( \ell > n \), \( T_\ell \) can be computed from the fixed list \( \sigma_1, \ldots, \sigma_t \) and the list of nodes of \( T_\ell \) taken from the at most \( 2^{\ell-k} \) extensions of \( \sigma_1, \ldots, \sigma_t \). It follows as in the proof of Theorem 3.6.4 above that for some constant \( c \) and all \( \ell \),

\[
K(T_\ell) \leq 2^{\ell-k} + \ell + 1 + c.
\]

Thus for large enough so that \( \ell + 1 \leq 2^{\ell-k} \), we have

\[
K(T_\ell) \leq 2^{\ell-k+1} + c,
\]

as desired. 

\( \square \)
We conjecture that a bound of \( \left( \frac{4}{3} \right)^n \) would characterize random closed sets in terms of prefix-free complexity. It would suffice, then, to show that \( \left( \frac{4}{3} \right)^n \) is a lower bound and that this bound implies randomness.

We also still seek upper bounds for \( \Pi^0_1 \) or closed \( \Pi^0_2 \) classes towards establishing prefix-free complexity characterizations of these classes. It seems plausible that \( \Pi^0_1 \) classes are more compressible (i.e. necessarily have smaller lower bounds) than random closed sets and we would like to explore this notion further.

### 3.7 Other Notions of Randomness for Closed Sets

Other notions of randomness that depend on different probability measures, or the inclusion of trees with dead ends in the encoding, might also be considered.

#### 3.7.1 Randomness with Regular Probability Measures

For any regular measure \( \nu \), we can define the notion of a \( \nu \)-Martin-Löf test and the resulting notion of a \( \nu \)-Martin-Löf -random (or just \( \nu \)-random) real. It is easy to see that \( \nu \)-random reals exist for any \( \nu \) and hence \( \nu \)-random closed sets exist. The results on ghost codes and joins will hold for any regular measure. The corresponding version of Lemma 3.3.8 will hold if \( \nu \) is regular with \( b_0 \) and \( b_1 \leq \frac{1}{2} \). The proofs of Theorem 3.4.14 and Corollary 3.4.15, that no subset of a measure-zero \( \Pi^0_1 \) class is random, also go through under this assumption.

Some of the results in this chapter may also be obtained for \( \nu_f \) where \( f(\sigma^{\uparrow}i) \leq \frac{1}{2} \) for \( i = 0, 1 \). For example, with respect to \( \nu_f \), a random closed set will have no isolated elements and it will always contain a random element. For any regular measure, either the leftmost or the rightmost path will be nonrandom, since either \( b_0 + b_2 > \frac{1}{2} \) or \( b_1 + b_2 \frac{1}{2} \). The proof of Theorem 3.4.19 that every random closed set has measure 0 seems to require, for \( \nu_f \)-randomness, that \( f(\sigma^{\uparrow}2) \leq \frac{1}{2} \) for all \( \sigma \).

#### 3.7.2 Randomness with the Inclusion of Trees with Deads Ends

Returning to the notion of randomness which allows trees with dead ends, let \( b_3 \) now be the probability that a given node has no extensions and let the probability be regular as above. Then a simple recursion shows the probability \( p \) of a given closed set being
empty satisfies the equation

\[ p = b_3 + (b_0 + b_1)p + b_2p^2. \]

Solving for \( p \), we obtain

\[ (p - 1)(b_2p - b_3) = 0. \]

Thus either \( p = 1 \) or \( p = \frac{b_3}{b_2} \). It follows that if \( b_2 \leq b_3 \), then \( p = 1 \), that is, almost every closed set is empty. Suppose now that \( b_3 < b_2 \) and let \( p_n \) be the probability that a given tree \( T \) has no paths of length \( n \). Then it can be seen by induction that \( p_n \leq \frac{b_3}{b_2} \) for all \( n \). That is, \( p_1 = b_3 \leq \frac{b_3}{b_2} \) and then

\[ p_{n+1} = b_3 + (1 - b_2 - b_3)p_n + b_2p_n^2 \leq \frac{b_3}{b_2}. \]

Hence in this case, the probability that a given closed set is empty is \( \frac{b_3}{b_2} < 1 \). In this case, one could presumably develop a notion of a random tree and a random closed set and explore the properties of random closed sets.

### 3.8 Random Closed Sets and Effective Capacity

In this section we will consider, given a closed set \( Q \), the probability that a randomly chosen closed set meets \( Q \). This probability is given by \( \mu^*(V(Q)) \), where \( V(Q) \) is a sub-basis set for the hit-or-miss topology on \( C \) (as given in section 3.3.1) and \( \mu^* \) is a given probability measure. If we define \( T_d(Q) \) to be precisely \( \mu^*(V(Q)) \), it turns out that \( T_d \) is a (computable) capacity, in the sense defined below; furthermore, the converse also holds. That is, if \( T \) is a (computable) capacity, then there is some probability measure \( \mu^*_d \) for which \( T = T_d \). We then explore the capacities of random and effectively closed sets, under the uniform measure (i.e. \( b_i = \frac{1}{3} \) for all \( i \) in Definition 3.2.6). This is joint work with Douglas Cenzer.

#### 3.8.1 Computable Capacities

**Definition 3.8.1.** A capacity on \( C \) is a function \( T : C \to [0, 1] \) with \( T(\emptyset) = 0 \) such that

(i) \( T \) is monotone increasing, that is,

\[ Q_1 \subseteq Q_2 \implies T(Q_1) \leq T(Q_2). \]
(ii) For \( n \geq 2 \) and any \( Q_1, \ldots, Q_n \in \mathcal{C} \)

\[
T(\bigcap_{i=1}^{n} Q_i) \leq \sum \{ (-1)^{|I|+1} T(\bigcup_{i \in I} Q_i) : \emptyset \neq I \subseteq \{1,2,\ldots,n\} \}.
\]

This is the alternating of infinite order property.

(iii) If \( Q = \cap_n Q_n \) and \( Q_{n+1} \subseteq Q_n \) for all \( n \), then \( T(Q) = \lim_{n \to \infty} T(Q_n) \).

We will assume, unless otherwise specified, that \( T(2^N) = 1 \) for a given capacity \( T \).

**Definition 3.8.2** (Computable Capacities). A capacity \( T \) is computable if it is computable on the family of clopen sets.

It follows that the capacity of any \( \Pi^0_1 \) class is upper semi-computable. Finally, the following notational definition will be used throughout.

**Definition 3.8.3** (\( T_d(Q) \), for \( Q \in \mathcal{C} \)). Suppose \( Q \in \mathcal{C} \). Define \( T_d(Q) := \mu_d^*(V(Q)) \), where \( V(Q) \) is a sub-basis set for the hit-or-miss topology on \( \mathcal{C} \) (as given in section 3.3.1) and \( \mu_d^* \) is a given probability measure.

That is, \( T_d(Q) \) is the probability that a randomly chosen closed set meets \( Q \).

We now show, in the following two theorems, that a computable capacity is always obtainable from, or a consequence of, a computable probability measure \( \mu_d^* \) for some \( d \). The following theorem, in particular, is well-known. For details on capacity and random set variables, see [73].

**Theorem 3.8.4.** If \( \mu_d^* \) is a (computable) probability measure on \( \mathcal{C} \), then \( T_d \) is a (computable) capacity.

**Proof.** This is easily verified. Certainly \( T(\emptyset) = 0 \). The alternating property follows by basic probability. For (iii), suppose that \( Q = \cap_n Q_n \) is a decreasing intersection.

Then by compactness, \( Q \cap K \neq \emptyset \) if and only if \( Q_n \cap K \neq \emptyset \) for all \( n \). Furthermore, \( V(Q_{n+1}) \subseteq V(Q_n) \) for all \( n \). Thus

\[
T(Q) = \mu(V(Q)) = \mu(\cap_n V(Q_n)) = \lim_n \mu(V(Q_n)) = \lim_n T(Q_n).
\]
The computability of $T$ is easily verified. That is, for any clopen set $I(\sigma_1) \cup \cdots \cup I(\sigma_k)$ where each $\sigma_i \in \{0,1\}^n$, we compute the probability distribution for all trees of height $n$ and add the probabilities of those trees which contain one of the $\sigma_i$.

This result has a converse, due to Choquet. See [73] for the general result.

**Theorem 3.8.5.** If $T$ is a computable capacity, then there is a computable measure $\mu^*_{\sigma}$ on the space of closed sets such that $T = T_\sigma$.

**Proof.** Given the values $T(U)$ for all clopen sets $I(\sigma_1) \cup \cdots \cup I(\sigma_k)$ where each $\sigma_i \in \{0,1\}^n$, there is in fact a unique probability measure $\mu_\sigma$ on these clopen sets such that $T = T_\sigma$ and this can be computed as follows.

Suppose first that $T(I(i)) = a_i$ for $i < 2$ and note that each $a_i \leq 1$ and $a_0 + a_1 \geq 1$ by the alternating property. If $T = T_\sigma$, then we must have $d((0)) + d((2)) = a_0$ and $d((1)) + d((2)) = a_1$ and also $d((0)) + d((1)) + d((2)) = 1$, so that $d((2)) = a_0 + a_1 - 1$, $d((0)) = 1 - a_1$ and $d((1)) = 1 - a_0$. This will imply that $T(\tau) = T_\sigma(\tau)$ when $|\tau| = 1$. Now suppose that we have defined $d(\tau)$ and that $\tau$ is the code for a finite tree with elements $\sigma_0, \ldots, \sigma_n = \sigma$ and thus $d(\tau^-i)$ is giving the probability that $\sigma$ will have one or both immediate successors. We proceed as above. Let $T(I(\sigma^-i)) = a_i \cdot T(I(\sigma))$ for $i < 2$. Then as above $d(\tau^-2) = d(\tau) \cdot (a_0 + a_1 - 1)$ and $d(\tau^-i) = d(\tau) \cdot (1 - a_i)$ for each $i$. \qed

### 3.8.2 Regular Measures and Capacities of Closed Sets

In this section all results are with respect to $\mu^*_{\sigma}$ fixed as the uniform measure (i.e. the regular measure with $b_0 = b_1 = b_2 = \frac{1}{3}$; see Definition 3.2.6). With this measure, we will consider the capacities of random closed sets and effectively closed sets. We say that $Q \in \mathcal{C}$ is $\mu^*_{\sigma}$-random if $x_Q$ is (Martin-Löf) random with respect to the measure $\mu_\sigma$.

**Theorem 3.8.6.** For the regular measure $\mu_\sigma$ with $b_i = \frac{1}{3}$, if $R$ is a $\mu^*_{\sigma}$-random closed set, then $T_\sigma(R) = 0$.

**Proof.** Fix $d$ as described above so that $d(\sigma^-i) = d(\sigma) \cdot \frac{1}{3}$ and let $\mu^* = \mu^*_{\sigma}$. We will compute the probability, given two closed sets $Q$ and $K$, that $Q \cap K$ is nonempty. Let

$$Q_n = \bigcup \{I(\sigma) : \sigma \in \{0,1\}^n \land Q \cap I(\sigma) \neq \emptyset\}$$

90
and similarly for $K_n$. Then $Q \cap K \neq \emptyset$ if and only if $Q_n \cap K_n \neq \emptyset$ for all $n$. Let $p_n$ be the probability that $Q_n \cap K_n \neq \emptyset$ for two arbitrary closed sets $K$ and $Q$, relative to our measure $\mu^*$. It is immediate that $p_1 = \frac{7}{9}$, since $Q_1 \cap K_1 = \emptyset$ only when $Q_1 = I(i)$ and $K_1 = I(1 - i)$. Next we will determine the quadratic function $f$ such that $p_{n+1} = f(p_n)$.

There are 9 possible cases for $Q_1$ and $K_1$, which break down into 4 distinct cases.

**Case I:** There are two chances that $Q_1 \cap K_1 = \emptyset$.

**Case II:** There are two chances that $Q_1 = K_1 = I(i)$, so that $Q_{n+1} \cap K_{n+1} \neq \emptyset$ with probability $p_n$.

**Case III:** There are four chances where $Q_1 = 2^N$ and $K_1 = I(i)$ or vice versa, so that once again $Q_{n+1} \cap K_{n+1} \neq \emptyset$ with probability $p_n$.

**Case IV:** There is one chance that $Q_1 = K_1 = 2^N$, in which case $Q_{n+1} \cap K_{n+1} \neq \emptyset$ with probability $1 - (1 - p_n)^2 = 2p_n - p_n^2$. This is because $Q_{n+1} \cap K_{n+1} = \emptyset$ if and only if both $Q_{n+1} \cap I(i) \cap K_{n+1} = \emptyset$ for both $i = 0$ and $i = 1$.

Adding these cases together, we see that

$$p_{n+1} = \frac{6}{9}p_n + \frac{1}{9}(2p_n - p_n^2) = \frac{8}{9}p_n - \frac{1}{9}p_n^2.$$

It follows that the sequence $\langle p_n \rangle$ is computable and we will see that the limit is zero. Let $f(p) = \frac{8}{9}p - \frac{1}{9}p^2$. Elementary calculus shows that $f$ has fixed points at $p = -1$ and $p = 0$ and that for $0 < p < 1$, $0 < f(p) < p$. Since $p_0 = \frac{7}{9}$, it follows that the sequence $\langle p_n \rangle$ is monotonic decreasing. Thus the limit exists and $\lim_n p_n = 0$ (since it must be a fixed point of $f$).

Thus the probability that $Q \cap K \neq \emptyset$ equals $\lim_n p_n = 0$. Next we will obtain a Martin-Löf test to prove our result.

For each $m, n \in \omega$, let

$$A_{m,n} = \{Q : \mu^*(\{K : K_m \cap Q_m \neq \emptyset\}) \geq 2^{-n}\}.$$

Let $C_m$ be the number of trees of height $m$ without dead ends.
For each \( Q \in A_{m,n} \) there are \( 2^{-n}C_m \) possible choices for \( K_m \) such that \( K_m \cap Q_m \neq \emptyset \) and thus at least \( D_{m,n} = (\frac{1}{2})2^{-n}\mu^*(A_{m,n})C_m^2 \) choices for \((K, Q) \in C \times C\) such that \( K_m \cap Q_m \neq \emptyset \) with \( Q \in A_{m,n} \) (since each pair might be counted twice).

Now define a computable sequence \( \langle m_n \rangle_{n \in \omega} \), so that \( p_{m_n} < 2^{-2n-1} \). Then

\[
p_{m_n} \geq \frac{D_{m_n,n}}{C_m^2} = 2^{-(n+1)}\mu^*(A_{m,n}).
\]

It follows that

\[
\mu^*(A_{m,n}) \leq 2^{n+1}p_{m_n} < 2^{n+1}2^{-2n-1} = 2^{-n}.
\]

Letting

\[
S_n = \bigcup_{r>n} A_{m_r,r}
\]

it follows that \( \mu^*(S_n) \leq 2^{-n} \) as well.

Now let \( R \) be a random closed set. The sequence \( \langle S_n \rangle_{n \in \omega} \) is a computable sequence of c.e. open sets with measure \( \leq 2^{-n} \), so that there is some \( n \) such that \( R \notin S_n \). Thus for \( r > n \), \( \mu^*(\{K : K_{m_r} \cap R_{m_r} \neq \emptyset\}) < 2^{-r} \) and it follows that

\[
\mu^*(\{K : K \cap R \neq \emptyset\}) = \lim_n \mu^*(\{K : K_{m_n} \cap R_{m_n} \neq \emptyset\}) = 0.
\]

Thus \( \mathcal{T}_d(R) = 0 \), as desired. \( \square \)

This result seems to depend on the measure. For different regular measures, the capacity of a random closed set can have different values.

**Theorem 3.8.7.** For the regular measure \( \mu_d \) with \( b_i = \frac{1}{3} \), there is a measure zero \( \Pi^0_1 \) class \( Q \) such that \( \mathcal{T}_d(Q) > 0 \).

**Proof.** First let us compute the capacity of \( X_n = \{x : x(n) = 0\} \). For \( n = 0 \), we have \( \mathcal{T}_d(Q_0) = \frac{2}{3} \). Now the probability \( \mathcal{T}_d(X_{n+1}) \) that an arbitrary closed set \( K \) meets \( X_{n+1} \) may be calculated in two distinct cases. Let \( K_n \) be as in the proof of Theorem 3.8.6.

**Case I** If \( K_0 = 2^N \), then \( \mathcal{T}_d(X_{n+1}) = 1 - (1 - \mathcal{T}_d(X_n))^2 \).
Case II If $K_0 = I(i)$ for some $i < 2$, then $T_d(X_{n+1}) = T_d(X_n)$.

It follows that $T_d(X_{n+1}) = \frac{2}{3}T_d(X_n) + \frac{1}{3}(2T_d(X_n) - (T_d(X_n))^2) = \frac{4}{3}T_d(X_n) - \frac{1}{3}(T_d(X_n))^2$. Now the function $f(p) = \frac{4}{3}p - \frac{1}{3}p^2$ has the property that $f(p) > p$ for $0 < p < 1$ and $f(1) = 1$. Since $T_d(X_{n+1}) = f(T_d(X_n))$, it follows that $\lim_n T_d(X_n) = 1$ and is the limit of a computable sequence.

For any $\sigma = (n_0, n_1, \ldots, n_k)$, with $n_0 < n_1 < \cdots < n_k$, similarly define $X_\sigma = \{x : (\forall i < k)x(n_i) = 0\}$. A similar argument to that above shows that $\lim_n T_d(X_\sigma^{-n})/T_d(X_\sigma) = 1$.

Now consider the decreasing sequence $c_k = \frac{2^{k+1}+1}{2^{k+1}}$ with limit $\frac{1}{2}$. Choose $n = n_0$ such that $T_d(X_n) \geq \frac{3}{4} = c_0$ and for each $k$, choose $n = n_{k+1}$ such that $T_d(X_{n_{k+1},\ldots,n_k}) \geq c_{k+1}$. This can be done since $c_{k+1} < c_k$. Finally, let $Q = \bigcap_k X_{(n_0,\ldots,n_k)}$. Then $T_d(Q) = \lim_k T_d(X_{(n_0,\ldots,n_k)}) \geq \lim_k c_k = \frac{1}{2}$.

This result can easily be extended to any bounded measure.
CHAPTER 4
RANDOM CONTINUOUS FUNCTIONS


 Portions of this work were also presented by P. Brodhead at the AMS Fall 2006 Eastern Sectional Meeting (October 2006, Storrs, CT) and the Conference on Logic, Computability, and Randomness (January 2007, Buenos Aires, Argentina).

4.1 Overview

In Chapter 3, we considered a notion of randomness for closed sets. We do the same for continuous functions here. An introduction to randomness for reals is provided in Section 3.2.

This chapter is organized as follows. In Section 4.2, we provide a definition of randomness for continuous functions and show that it is sound. In Section 4.3, we prove various results for (images of) random continuous functions—perfectness, non-injectivity, and instances of non-surjectivity; we also study images of computable elements. In Section 4.4, we tie random closed sets to random closed functions through images: inverse images of $0^\omega$ are random closed sets, but images, in general, are not. Continuing on the theme of inverse images of $0^\omega$, in Section 4.5 we consider pseudo-distance functions. In Section 4.6, we briefly consider how the results of Chapters 3 and 4 can be relativized for $n$-randomness. Finally, in Section 4.7, we describe some directions for future research.
4.2 Defining Randomness for Continuous Functions

A function $F : 2^\mathbb{N} \to 2^\mathbb{N}$ is continuous iff it has a closed graph. It seems reasonable, then, to define continuous function $f$ to be random, iff the its graph $Gr(F) = \{ x \oplus y : y = F(x) \}$ is random. However if $[T]$ is the graph of a function and $\sigma \in T$ has even length, then we must have $\sigma \downarrow 0 \in T$ and $\sigma \downarrow 1 \in T$. This means that the family of closed sets which are the graphs of functions has measure 0 in the space of closed sets and hence a random closed set will not be the graph of a function. We need, therefore, a different method to define randomness for continuous functions. We do this below.

4.2.1 Representing Functions

Given a continuous function $F : 2^\mathbb{N} \to 2^\mathbb{N}$, we are interested in representing it in such a way so as to be able to consider a notion of algorithmic randomness. We show below that any such function $F$ may be represented by infinitely many representing functions of the form $f : \{0, 1\}^* \to \{0, 1, 2\}^*$. This will allow us, in the following section, to be able to represent continuous function as elements of $3^\omega$, so-called representing sequences, and to consider a such a function as random if it possesses a random representing sequence.

Information output, the key to representation. For any continuous function $F$ on $2^\mathbb{N}$ and any $\sigma \in \{0, 1\}^*$, there is a natural number $n$ and binary string $\tau$ of length $n$ such that for all $u \in I(\sigma)$, $F(u)[n] = \tau$. In particular, $F(u)(n) = \tau(n)$ for every such $u$. In general, the length of $\sigma$ may be much larger than $n$, so we may have to extend $\sigma$ by several bits to get uniformity of $F(u)[n+1]$ within the interval around $\sigma$’s extension.

Representing functions. Taking the above into consideration, we may recursively represent any continuous function $F : 2^\mathbb{N} \to 2^\mathbb{N}$ by some function $f : \{0, 1\}^* \to \{0, 1, 2\}^*$ as follows. Suppose $F$ is given. Let $f(\emptyset) = \emptyset$. For $|\sigma| = m + 1$, having defined $f(\sigma[i]) = e_i$ for all $i \leq m$, let $\rho = (n_1, \ldots, n_k)$ be the result of deleting all 2s from $(e_1, \ldots, e_m)$. If for all $u \in I(\sigma)$, $F(u)[k] = \rho^- j$, $j \in \{0, 1\}$, we may let $e_{m+1} = j$. If not we must have $e_{m+1} = 2$; even if so we allow $e_{m+1} = 2$. 
The canonical representation. Notice, from the above, that any continuous \( F \) has infinitely many representing functions \( f : \{0,1\}^* \rightarrow \{0,1,2\}^* \). The representation which uses as few 2s as possible we shall call the canonical representation.

4.2.2 Representing Sequences

We want to code the representing function as an element of \( 3^\mathbb{N} \) to discuss its algorithmic randomness. To do so, first enumerate \( \{0,1\}^* = \{\emptyset\} \) as \( \sigma_0, \sigma_1, \ldots \), ordered first by length and then lexicographically. Thus \( \sigma_0 = (0) \), \( \sigma_1 = (1) \), \( \sigma_2 = (00) \), etc. We define representing sequences below.

**Definition 4.2.1.** (i) \((INF, \text{Rem}_2)\) Let \( INF \) equal the set of \( y \in \{0,1,2\}^n \) such that \( \{n : y(n) \neq 2\} \) is infinite and, for \( y \in INF \), let \( \text{Rem}_2(y) \) be the result of removing from \( x \) all occurrences of 2.

(ii) (Representing functions) A function \( f : \{0,1\}^* \rightarrow \{0,1,2\} \) represents a function \( F : 2^\mathbb{N} \rightarrow 2^\mathbb{N} \) if for all \( x \in 2^\mathbb{N} \), the sequence \( y \), defined by \( y(n) = f(x \upharpoonright n) \) belongs to \( INF \) and \( \text{Rem}_2(y) = F(x) \).

(iii) (Representing sequences) A sequence \( r \in \{0,1,2\}^\mathbb{N} \) represents the continuous function \( F \) (written \( F = F_r \)) if the function \( f_r : \{0,1\}^* \rightarrow \{0,1,2\} \), defined by \( f_r(\sigma_n) = r(n) \), represents \( F \).

(iv) (Labelled \( 2^\omega \)-trees) Given a representing sequence \( r \), the function \( f_r \) gives rise to a labelled \( 2^\omega \)-tree. We attach, or associate, the value of \( f_r(\sigma) \) with each node \( \sigma \).

**Example 4.2.1** (The \( 2^\omega \)-tree for the Identity, A Geometric Intrepretation). The identity function can be represented by placing an \( e \) on any node \( \sigma \) which ends in \( e \). This can also be pictured geometrically as representing the graph of \( F \) as the intersection of a decreasing sequence of clopen subsets of the unit square. Initially the choice of \( f((0)) \) and \( f((1)) \) selects from the 4 quadrants. That is, for example, \( f((0)) = (0) = f((1)) \) implies that the graph of \( F \) is included in the bottom half of the square and \( f((0)) = \emptyset \) and \( f((1)) = (1) \) implies that the graph excludes the lower right hand quadrant. Successive values of \( f \) continue to restrict the graph of \( F \) in a similar fashion.
4.2.3 A Sound Definition

In this section we define a measure $\mu^{**}$ on the space of functions $F : 2^N \to 2^N$ that allows us to define the notion of randomness for functions on $2^N$. In short, a function is random if it possesses a random representing function, or equivalently, a random representing sequence. We will show that no canonical representing function can be random, so that necessarily, the definition of randomness for functions is in this existential format. It may be, however, that no continuous function has a random representing function. We show that this is not so. In fact, we show that every random representing function is continuous. Clearly then, random continuous functions exist and, in fact, $\Delta^0_2$ random continuous functions exist. Therefore the definition is sound and the structure begins to manifest itself as rich.

The Measure for Randomness. The measure which is used to define randomness for continuous functions is the Lebesgue measure on the space $3^N$ of representing sequences. Thus for each new bit of input, there is equal probability $\frac{1}{3}$ that $f_r$ gives a new output of 0 for $F_r$, gives a new output of 1 for $F_r$, or gives no new output for $F_r$. This measure now induces a measure, $\mu^{**}$ say, on the space $\mathcal{F}$ of continuous functions.

Definition 4.2.2. A function $F : 2^N \to 2^N$ is random if there is a representing sequence $r \in 3^N$ for $F$ that is random with respect to the measure $\mu^{**}$.

We first show that every random representing function is continuous. The following lemma is needed.

Lemma 4.2.3. Let $\Sigma$ be a finite set and let $Q \subseteq \Sigma^N$ be a $\Pi^0_1$ class of measure 0. Then no element of $Q$ is Martin-Löf random.

Proof. Let $\Sigma = \{0, 1, 2\}$ without loss of generality. Let $Q = [T]$ where $T \subseteq \{0, 1, 2\}^*$ is a computable tree (possibly with dead ends). For each $n$, let $T_n = T \cap \{0, 1, 2\}^n$ and let

$$Q_n = \bigcup \{I(\sigma) : \sigma \in T_n\}.$$ 

Let $g(n) = \mu(Q_n) = \frac{|T_n|}{3^n}$. Then $g(n)$ is a computable sequence and

$$\lim_{n \to \infty} g(n) = \mu(Q) = 0.$$
This Martin-Löf test shows that $Q$ has no random elements. (As observed by Solovay, it is sufficient to have a computable sequence approaching zero rather than the stricter test with a sequence of measures $g(n) \leq 2^{-n}$.)

**Theorem 4.2.4.** (i) The set of representing functions for total functions has measure one.

(ii) Every random function is continuous.

**Proof.** (i) Let $r \in 3^N$ and suppose that $f_r$ does not represent a total function. Then there is some $x \in 2^N$ and some $\tau \in \{0, 1\}^*$ such that $f_r(x[n]) = \tau$ for almost all $n$. Without loss of generality we may assume that $\tau = \emptyset$. Let $A$ be the set of functions $f : \{0, 1\}^* \to \{0, 1\}^*$ such that $f(\sigma) = \emptyset$ for arbitrarily long strings $\sigma$ and let $p = \mu^{**}(A)$. Then certainly $p \leq \frac{5}{9}$, since if $r(0)$ and $r(1)$ are both in $\{0, 1\}$, then $f_r \notin A$. Considering the 9 cases for the initial choices of $f((0))$ and $f((1))$, we see that

$$p = \frac{4}{9}p + \frac{1}{9}[1 - (1 - p)^2],$$

so that $\frac{1}{9}p^2 + \frac{1}{3}p = 0$, which implies that $p = 0$. (That is, there are 4 cases in which $|f((i))| = 1$ for $i = 0, 1$ so that immediately $f \notin A$, there are 4 cases in which only one of $f((i)) = \emptyset$, in which case the remaining function $g$, defined by $g(\sigma) = f(i^{-} \sigma)$ must be in $A$, and there is one case in which $f((i)) = \emptyset$ for $i = 0, 1$, in which case at least one of the remaining functions must be in $A$.) Consequently, the set of representing functions for total functions has measure one.

(ii) Observe that $A$ is a $\Pi^0_1$ class, since $f_r \in A$ if and only if $(\forall n)(\exists \sigma \in \{0, 1\}^n)f_r(\sigma) = \emptyset$. It follows from Lemma 4.2.3 that no representing function on $2^*$ for a random function on $2^N$ can be in $A$. As all functions representing partial functions on $2^N$ occur in $A$, it follows that every random function is total. Since the graph of a total function is a closed set, it follows that random functions are continuous.

Now the set of Martin-Löf random elements of $\{0, 1, 2\}^N$ has measure one and there exists a $\Delta^0_2$ Martin-Löf real. Hence we have the following.

**Theorem 4.2.5.** There exists a random continuous function which is $\Delta^0_2$ computable.
We also first observe that any continuous function will have a representation which is not random. In fact, the canonical representation itself can never be random.

**Proposition 4.2.6.** For any continuous function $F$, the canonical representation is not random.

**Proof.** The idea is that whenever the canonical representation labels a node $\sigma$ with 2, then the two labels on the successor nodes $\sigma^0$ and $\sigma^1$ cannot be both 0, or both 1. Thus we have the following Martin-Löf test. Assume by way of contradiction that $r$ is random and canonical. Let $S_e$ be the set of $r \in 3^\mathbb{N}$ such that $r$ has at least $e$ occurrences of 2 and such that, for the first $e$ occurrences of 2 in $r$, the corresponding successor values are not both 0 or both 1. Since $r$ is random, it must have infinitely many occurrences of 2 and since $r$ is canonical, it must belong to every $S_e$. But each $S_e$ is a c.e. open set and has measure $\leq \left(\frac{1}{2}\right)^e$, so that no random sequence can belong to every $S_e$. \qed

The theorem, in fact, demonstrates the need for the existential part of definition of random functions. In the following sections we will obtain some additional properties of random continuous functions.

### 4.3 Random Continuous Functions and Images

#### 4.3.1 Perfect Images, in every instance

In this section we show that all random continuous functions always have perfect images. This is a consequence of the following theorem.

**Theorem 4.3.1.** If $F$ is a random continuous function, then the image $F[2^\mathbb{N}]$ has no isolated elements.

**Proof.** Let $f$ be the random representing function for $F$ and let $Q = F[2^\mathbb{N}]$. Suppose by way of contradiction that $Q$ contains an isolated path $y$. Then there is some finite $\tau \sqsubseteq y$ such that $y$ is the unique element of $I(\tau) \cap Q$. Fix $\sigma$ such that $f(\sigma) = \tau$.

For each $n$, let $S_n$ be the set of all $g \in \mathcal{F}$ such that for all $\rho_1, \rho_2 \in \{0,1\}^n$,

1. $g(\sigma \upharpoonright \rho_1)$ is compatible with $g(\sigma \upharpoonright \rho_2)$,
2. $\tau \sqsubseteq g(\sigma \upharpoonright \rho_1)$, and
3. $\tau \sqsubseteq g(\sigma \upharpoonright \rho_2)$
Then for any each $m < n$ and each $\rho \in \{0, 1\}^m$, we are restricted to at most 7 of the 9 possible choices for $f(\rho^0)$ and $f(\rho^1)$. This same scenario applies for all $\rho \in \{0, 1\}^{n-1}$, so that in general, $\mu(S_n) \leq \left(\frac{7}{9}\right)^{2n-1}$.

Now for each $n$, $S_n$ is a clopen set in $\mathbb{F}$ and thus the sequence $S_0, S_1, \ldots$ is a Martin-Löf test. It follows that for some $n$, $F \notin S_n$. Thus there are two extensions of $\sigma$ of length $n$ which have incompatible images, contradicting the assumption that $y$ was the unique element of $Q \cap I(\tau)$.

It follows that the image of a random continuous function is perfect and has continuum many elements.

### 4.3.2 Non-injective Images, in every instance

In this section we show that no random continuous function is injective. We will use the following theorem en route.

**Theorem 4.3.2.** For any $\sigma \in \{0, 1\}^*$, the probability that the image of a continuous function $F$ meets $I(\sigma)$ is always $> \frac{3}{4}$.

**Proof.** The proof is by induction on $|\sigma|$. Without loss of generality, we assume that $\sigma = 0^n$. For each $n > 0$, let $q_n$ be the probability that $F[2^N]$ meets $I((0^n))$. Let $f$ be the representing function for $F$. For $n = 1$, there are 9 equally probable choices for the pair $f((0))$ and $f((1))$, breaking down into 4 distinct cases.

**Case 1.** If $f((0)) = (1) = f((1))$, then $F[2^N]$ does not meet $I((0))$. This occurs just once.

**Case 2.** If $f((0)) = (0)$ or $f((1)) = (0)$, then $F[2^N]$ meets $I((0))$. This occurs in 5 of the 9 choices.

**Case 3.** If $f((i)) = \emptyset$ and $f((1 - i)) = (1)$, then $F[2^N]$ meets $I((0))$ if and only if $F_{(i)}[2^N]$ meets $I((0))$. This occurs in 2 of the 9 choices, with probability $q_1$.

**Case 4.** If $f((0)) = \emptyset = f((1))$, then $F[2^N]$ meets $I((0))$ if at least one of $F_{(i)}[2^N]$ meets $I((0))$. This occurs in 1 of the choices, with probability $1 - (1 - q_1)^2$. That is, $F[2^N]$ fails to meet $I((0))$ if both $F_{(0)}[2^N]$ and $F_{(0)}[2^N]$ fail to meet $I((0))$. 

100
Putting these cases together, we see that

\[ q_1 = \frac{5}{9} + \frac{2}{9}q_1 + \frac{1}{9}(2q_1 - q_1^2), \]

so that \( q_1 \) satisfies the quadratic equation

\[ x^2 + 5x - 5 = 0. \]

Thus \( q_1 \) is the unique solution in \([0,1]\) of this equation, that is,

\[ q_1 = \frac{\sqrt{45} - 5}{2}, \]

which is indeed > .75.

Now let \( q_n = q \) and let \( q_{n+1} = p \). Once again we consider the 9 initial choices, now breaking down into 6 distinct cases.

**Case 1.** If \( f((0)) = (1) = f((1)) \), then \( F[2^N] \) does not meet \( I((0^{n+1})) \). This occurs just once.

**Case 2.** If \( f((0)) = (0) = f((1)) \), then \( F[2^N] \) meets \( I((0^{n+1})) \) if and only if at least one of \( F(0) \) and \( F(1) \) meets \( I((0^n)) \). This occurs just once, and with probability 

\[ 1 - (1 - q)^2 = 2q - q^2. \]

**Case 3.** If \( f((i)) = (0) \) and \( f((1-i)) = (1) \), then \( F[2^N] \) meets \( I((0^{n+1})) \) if and only if \( F(i)[2^N] \) meets \( I((0^n)) \). This occurs in 2 of the 9 choices, with probability \( q \).

**Case 4.** If \( f((i)) = \emptyset \) and \( f((1-i)) = (1) \), then \( F[2^N] \) meets \( I((0^{n+1})) \) if and only if \( F(i)[2^N] \) meets \( I((0^n)) \). This occurs in 2 of the 9 choices, with probability \( p \).

**Case 5.** If \( f((0)) = \emptyset = f((1)) \), then \( F[2^N] \) meets \( I((0^{n+1})) \) if at least one of \( F(i)[2^N] \) meets \( I((0^{n+1})) \). This occurs just once, with probability \( 1 - (1 - p)^2 \).

**Case 6.** If \( f((i)) = \emptyset \) and \( f((1-i)) = (0) \), then \( F[2^N] \) meets \( I((0^{n+1})) \) if at least one of the following two things happens. Either \( F(i)[2^N] \) meets \( I((0^{n+1})) \), or \( F(1-i)[2^N] \) meets \( I((0^n)) \). This occurs in 2 of the 9 choices, with probability \( 1 - (1 - p)(1 - q) \).
Putting these cases together, we see that
\[ p = \frac{2}{3}p - \frac{1}{9}p^2 - \frac{2}{9}pq + \frac{2}{3}q - \frac{1}{9}q^2, \]
so that \( p = q_{n+1} \) satisfies the equation
\[ p^2 + 3p + 2pq - 6q + q^2 = 0. \]

We note that for \( p = q \), the solutions are \( p = q = 0 \) and \( p = q = \frac{3}{4} \). This explains the value \( \frac{3}{4} \) in the statement of theorem.

Now assume by induction that \( q > \frac{3}{4} \). Suppose by way of contradiction that \( p \leq \frac{3}{4} \). It follows that
\[ \frac{9}{16} + \frac{9}{4} + \frac{3}{2}q - 6q + q^2 \geq 0. \]
Simplifying, this implies that \( 16q^2 - 72q + 45 \geq 0 \). But this factors into \((4q - 3)(4q - 15)\) and is only \( \geq 0 \) when either \( q \leq \frac{3}{4} \) or \( q \geq \frac{15}{4} \). Since the latter is impossible, we obtain the desired contradiction that \( q \leq \frac{3}{4} \).

\[ \blacksquare \]

**Theorem 4.3.3.** No random continuous function is injective.

**Proof.** Let \( p \) be the probability that an arbitrary continuous function \( F \) is injective. It follows from Theorem 4.3.2 that there is a \( \frac{9}{16} \) chance that \( F \) has a zero in \( I(0) \) and also in \( I(1) \), so that \( p \leq \frac{7}{16} \). Now in general, if \( F \) is injective, then it must be injective when restricted to \( I(0) \) and when restricted to \( I(1) \). It follows that \( p \leq p^2 \), which happens only for \( p = 0 \) and \( p = 1 \), given that \( 0 \leq p \leq 1 \). Since \( p \leq \frac{7}{16} \), it follows that \( p = 0 \), as desired. This can be reformulated as a Martin-Löf test as follows. First we observe that \( F \) is injective if and only if, the images of each pair of disjoint intervals \( I(\sigma) \) and \( I(\tau) \) are disjoint. Let
\[ D(\sigma, \tau) = \{ F : F[I(\sigma)] \cap F[I(\tau)] = \emptyset \}. \]
Then \( D(\sigma, \tau) \) is uniformly c.e. since \( F[I(\sigma)] \cap F[I(\tau)] = \emptyset \) if and only if there exists \( n \) such that for all extensions \( \sigma' \) of \( \sigma \) and \( \tau' \) of \( \tau \) of length \( n \), \( f(\sigma') \) and \( f(\tau') \) are incompatible.

Now let
\[ S_m = \{ F : (\forall \sigma, \tau \in \{0, 1\}^m)(\sigma \neq \tau \rightarrow F \in D(\sigma, \tau)) \}. \]
It follows from the observation above that \( F \) is injective if and only if \( F \in \bigcap_m S_m \). The argument above shows that \( \mu^{**}(S_1) \leq \frac{7}{16} \) and that \( \mu^{**}(S_{m+1}) \leq \mu^{**}(S_m)^2 \) and hence
\[
\mu^{**}(S_m) \leq \left( \frac{7}{16} \right)^m .
\]

It follows that \( \{ S_m : m \in \omega \} \) is a Martin-Löf test and therefore no random continuous function may belong to every \( S_m \) and hence no random continuous function can be injective.

4.3.3 Non-surjective Images, in instances

In this section we show that random continuous functions are not necessarily onto.

**Definition 4.3.4** \((F_{\sigma}, \text{the restriction of } F \text{ to } I(\sigma))\). For any function \( F \) on \( 2^\mathbb{N} \) and any \( \sigma \in \{0, 1\}^* \), define the restriction \( F_{\sigma} \) of \( F \) to \( I(\sigma) \) by
\[
F_{\sigma}(x) = F(\sigma \triangledown x).
\]

Clearly any such restriction of a random continuous function will be random, but more can be said. Recall van Lambalgen's theorem, Theorem 3.2.12.

**Proposition 4.3.5.** \( F \) is a random continuous function if and only if the functions \( F(0) \) and \( F(1) \) are relatively random.

**Proof.** Let \( r \) represent \( F \). Suppose first that \( F \) is random. It follows, as in Corollary 3.3.12, that \( F(0) \oplus F(1) \) is random and hence \( F(0) \) and \( F(1) \) are relatively random by van Lambalgen’s theorem.

Next suppose that \( F(0) \) and \( F(1) \) are relatively random and let \( r_i \) represent \( F_i \) for \( i = 0, 1 \). Let \( d \) be any martingale, which we think of as betting on \( r \). Then for \( i = 0, 1 \), we can define a martingale \( d_i \) with oracle \( r_{1-i} \) as follows. We will give the definition for \( d_0 \) and leave \( d_1 \) for the reader. Given \( \sigma = r_0(0), \ldots, r_0(2^p + q - 2) \) where \( 0 \leq q < 2^p \), use \( r_1 \) to compute \( \tau = r(0), \ldots, r(2^{p+1} + q - 2) \) and then define \( d_i \) to bet in the same proportion as \( d \). That is, \( d_i(\sigma \triangledown j)/d_i(\sigma) = d(\tau \triangledown j)/d(\tau) \) for \( j < 3 \). Thus for any node on the left side of the labelled tree for \( F \), \( d_0 \) is making the same bet on the next label that \( d \) would have made, and similarly for \( d_1 \) and the right side.
Since the $F_{(i)}$ are relatively random for $i = 0, 1$, it follows that $d_i$ does not succeed and hence there exist upper bounds $B_i$ for $\{d_i(r_i[n])\}_{n \in \mathbb{N}}$. But it follows from the above definitions of $d_i$ that for any $p$,

$$d(r[2^{p+1} - 2]) = d_0(r_0[2^p - 1]) \cdot d_1(r_1[2^p - 1]).$$

This is because the martingale $d$ alternates using $d_0$ and $d_1$ and the result can be viewed in each alternation as multiplying the capital by some factor. Then in general, for $0 < q \leq 2^p$,

$$d(r[2^{p+1} + q - 2]) = d_0(r_0[2^p + q - 1]) \cdot d_1(r_1[2^p - 1])$$

and

$$d(r[2^{p+1} + 2^p + q - 2]) = d_0(r_0[2^{p+1} - 1]) \cdot d_1(r_1[2^p + q - 1]).$$

It follows that $B_0 \cdot B_1$ is an upper bound for $\{d(r[k]) : k \in \mathbb{N}\}$, so that $d$ does not succeed on $r$.

**Corollary 4.3.6.** A random continuous function is not necessarily onto.

**Proof.** It follows from Proposition 4.3.5 that, for any $\tau \in \{0, 1\}^*$, there is a random continuous function with image $\subseteq I(\tau)$. Thus a random continuous function is not necessarily onto. \hfill \Box

### 4.3.4 Images of computable elements

In this section we will show that the image of computable element under a random continuous function is necessarily non-computable. In fact, it is random. We need the following proposition.

**Proposition 4.3.7.** Suppose $A \subset B$ are two finite sets of symbols. Given $X \in B^\mathbb{N}$, let $\tilde{X} \in A^\mathbb{N}$ be the sequence obtained by deleting all symbols in $B - A$ from $X$. If $X$ is $1$-random, then $\tilde{X}$ is $1$-random.

**Proof.** Given $X$, $\tilde{X}$ as in the proposition, suppose $\tilde{X}$ is not random and let $d$ be a constructive martingale on $A^\mathbb{N}$ that succeeds on $\tilde{X}$. We will construct a martingale $\hat{d}$ on $B^\mathbb{N}$ that succeeds on $\tilde{X}$. Essentially, $\hat{d}$ will keep its capital constant on symbols in $B - A$; it will bet according to $d$, repeating its bets after bits which hold symbols from $B - A$.\hfill 104
Define $\hat{d}(\lambda) = d(\lambda)$, and for $\sigma \in B^*$ and $\tilde{\sigma}$ the corresponding string of $A^*$,

$$
\hat{d}(\sigma^{-}x) = \begin{cases} 
\frac{d(\tilde{\sigma}^{-}x)}{d(\tilde{\sigma})}\hat{d}(\sigma) & x \in A \\
\hat{d}(\sigma) & x \in B - A 
\end{cases}
$$

The function $\hat{d}$ is clearly constructive, since $d$ is. To show $\hat{d}$ is a martingale, consider the sum

$$
\sum_{x \in B} d(\sigma^{-}x) = \sum_{x \in A} \frac{d(\tilde{\sigma}^{-}x)}{d(\tilde{\sigma})}\hat{d}(\sigma) + \sum_{x \in B-A} \hat{d}(\sigma)
$$

$$
= \hat{d}(\sigma)\sum_{x \in A} \frac{d(\tilde{\sigma}^{-}x)}{d(\tilde{\sigma})} + \hat{d}(\sigma)|B - A| = \hat{d}(\sigma)[|A| + |B - A|].
$$

It remains to show that $\hat{d}$ succeeds on $X$. However, that is clear, as on bits which are in $X$ but not $\tilde{X}$, $\hat{d}$ keeps its capital constant, and on bits from $\tilde{X}$, it acts exactly as $d$ would. Therefore since $d$ succeeds on $\tilde{X}$, $\hat{d}$ succeeds on $X$ and $X$ is nonrandom.

It is easy to see that, for any random continuous function $F$ and any computable real $x$, $F(x)$ is not computable. This follows from our next result.

**Theorem 4.3.8.** If $F$ is a random continuous function, then, for any computable real $x$, $F(x)$ is not computable.

**Proof.** Suppose that $F$ is random and let $x$ and $y$ be computable reals. For each $n$, let

$$
S_n = \{G : G(x|n) \prec y[n]\}.
$$

Then $S_0, S_1, \ldots$ is an effective sequence of c.e. open sets in $\mathcal{F}$, and an easy induction shows that $\mu^{\ast\ast}(S_n) = (2/3)^n$. This is a Martin-Löf test and it follows that $F \notin S_n$ for some $n$, so that $F(x) \neq y$.

We now strengthen this result to show that the image of a computable element is random.

**Theorem 4.3.9.** If $F$ is a random continuous function, then, for any computable real $x$, $F(x)$ is a random real.
Proof. Suppose that $F$ is random with representing function $f_r$, let $x$ be a computable real and let $y = F(x)$. Define the computable function $g$ so that, for each $n$, 

$$\sigma_{g(n)} = x[n].$$

By the Von-Mises–Church–Wald Computable Selection Theorem, the subsequence $z(n) = r(g(n))$ is random in $\{0, 1, 2\}^\mathbb{N}$. Now $y = F(x)$ may be computed from $z$ by removing the 2’s. Thus $F(x)$ is random by Proposition 4.3.7.

We note that Fouche [45] has used a different approach to randomness for continuous functions connected with Brownian motion, first presented by Asarin and Prokovskiy [5], and has shown that, under this approach, it is also true that for any random continuous function $F$, $F(x)$ is not computable for any computable input $x$.

It follows that a random function $F$ can never be computably continuous and hence the graph of $F$ is not a $\Pi^0_1$ class.

4.4 Random Closed Sets arising from random continuous functions

4.4.1 A Positive Result: Inverse Images of $0^\omega$

In this section we prove that for any random continuous function $F$, the set $Z(F) = \{x : F(x) = 0\}$ is a random closed set. For any subset $S$ of $\mathcal{C}$, let $Z_S = \{F \in \mathcal{F} : Z(F) \in S\}$.

Lemma 4.4.1. For any open set $S$, $\mu^*(Z_S) \leq \mu^*(S)$.

Proof. It suffices to prove the result for intervals $S = I(\sigma)$. We will show by induction on $|\sigma|$ that $\mu^*(I(\sigma)) = \left(\frac{1}{4}\right)^{|\sigma|}$, whereas of course $\mu^*(I(\sigma)) = \left(\frac{1}{3}\right)^{|\sigma|}$. Recall from Corollary 4.4.3 that $0 \in F[2^\mathbb{N}]$ with probability exactly $\frac{3}{4}$. For $|\sigma| = 1$, there are two distinct cases.

Case I Suppose first that $\sigma = (i)$, where $i \in \{0, 1\}$. Then $F \in Z_S$ if and only if $F$ has a zero in $I((i))$ and has no zero in $I((1 - i))$. Now $F$ has a zero in $I((i))$ if $f((i)) \in \{0, 2\}$ and if the restricted function has a zero, which gives probability $\frac{23}{34} = \frac{1}{2}$. Thus the combined probability that $F \in Z_S$ is $\frac{1}{4}$.

Case II Suppose next that $\sigma = (2)$. Then $F \in Z_S$ if and only if $F$ has zeroes in both $I((0))$ and $I((1))$. It follows from the argument in Case I that $\mu^*(Z_S) = \frac{1}{4}$.
Notice that $Z_{\{\emptyset\}} = \{F : F \text{ has no zeroes}\}$ has positive measure $\frac{1}{4}$ but $\mu^*(\{\emptyset\}) = 0$.

Now suppose $|\sigma| = n$ and let $\tau = \sigma \sim i$; suppose by induction that $\mu^{**}(Z_{I(\sigma)}) \leq \mu^*(I(\sigma))$. Interpret $\tau$ as the code for a (finite) binary tree and let $\rho \in \{0, 1\}^*$ be the terminal node of that tree such that $i$ indicates the branching of $\rho$. Again there are two cases.

**Case I** Suppose first that $i \in \{0, 1\}$. Then $F \in Z_{I(\tau)}$ if and only if $F \in Z_{I(\sigma)}$ and furthermore $F$ has a zero in $I((\rho \sim i))$ and has no zero in $I((\rho \sim 1 - i))$. It follows as above that $\mu^{**}(Z_{I(\tau)}) = \frac{1}{4} \mu^{**}(Z_{I(\sigma)}) = (\frac{1}{4})^{n+1}$.

**Case II** Suppose next that $i = 2$. Then $F \in Z_{I(\tau)}$ if and only if $F$ has zeroes in both $I(\rho \sim 0)$ and $I(\rho \sim 1)$. It follows as above that $\mu^{**}(Z_{I(\tau)}) = \frac{1}{4} \mu^{**}(Z_{I(\sigma)}) = (\frac{1}{4})^{n+1}$.

An arbitrary open set is a disjoint union of intervals and thus the desired inequality can be extended to open sets. \hfill $\square$

**Theorem 4.4.2.** For any random continuous function $G : 2^N \rightarrow 2^N$, the set of zeroes of $G$ is either empty or is a random closed set.

**Proof.** Suppose that $G$ is a random continuous function which has at least one zero, and let $S_0, S_1, \ldots$ be a Martin-Löf test in $C$. Then there is a computable function $\phi$ such that $S_i = \bigcup_n I(\sigma_{\phi(i,n)})$. We may assume without loss of generality $\mu^*(S_i) \leq 2^{-i-2}$ and that each $S_i$ is not clopen and that, for each $i$, the intervals $I(\sigma_{\phi(i,n)})$ are pairwise disjoint. We will define a Martin-Löf test $S'_0, S'_1, \ldots$ in the space $F$ and use the fact that $G$ must satisfy $S'_i \subseteq S_i$ to show that $Z(G)$ satisfies $\{S'_i\}_{i \in \omega}$.

Fix an interval $I(\sigma)$ in $C$ and let $C_\sigma = Z_{I(\sigma)}$. Observe that there is a clopen set $B_\sigma \subseteq 2^N$ and a corresponding finite set $\tau_0, \ldots, \tau_{k-1}$ of strings such that $B_\sigma = \bigcup_{j < k} I(\tau_j)$, associated with $\sigma$ such that, for any $Q \in C$ with code $r$, $r \in I(\sigma)$ if and only if $Q \subseteq B_\sigma$ and $Q \cap I(\tau_j) \neq \emptyset$ for all $j < k$. It follows that $C$ is a difference of $\Pi^1_1$ classes. That is, $F \in C$ if and only if the following two conditions hold.

(i) For each $j$, $F$ has a zero in $I(\tau_j)$; by compactness, this is equivalent to saying that for any $\ell$, there is an extension $\tau \in \{0, 1\}^\ell$ of $\tau_j$ such that $f(\tau) \in \{0, 2\}^{\vert r \vert}$, where $f$ is the function on strings representing $F$.  

\hfill 107
(ii) $F$ has no zeroes outside of $B$. Let $2^N - B = \bigcup_{\tau \in A} I(\tau)$. By compactness, $F$ has no zeroes outside of $B$ if and only if

$$((\exists \ell)(\forall \tau \in A)(\forall \tau' \geq \tau)[|\tau'| = \ell \Rightarrow (\exists m)(f(\tau'[m] = 1)]).$$

(4–1)

Note that the measure of $C_\sigma$ may be computed uniformly from $\sigma$ given the calculation from Corollary 4.4.3 that whenever $f(\sigma) \in \{0, 2\}^{|\sigma|}$, then the probability that $F$ has a zero in $I(\sigma)$ is exactly $\frac{3}{4}$. For each $\sigma$, we will uniformly compute a c.e. open set $S_\sigma \subseteq F$ such that $C_\sigma \subseteq B_\sigma$ and such that $\mu^*(B_\sigma) \leq 2 \cdot \mu^*(C_\sigma)$. There are two stages in the construction of $B_\sigma$.

**Stage I:** Let $U$ be the set of codes $\sigma'$ for partial functions $f'$ such that 4–1 holds with $f'$ in place of $f$, and such that furthermore for every $j$ and $\ell$ such that $f'$ is defined on all length-$\ell$ extensions $\tau$ of $\tau_j$, there is such a $\tau$ with $f'() \in \{0, 2\} \forall \rho \preceq \tau$. It is clear that for any $F \in C_\sigma$, there exists $\sigma' \in U$ with $F \in I(\sigma')$ and hence

$$C_\sigma \subseteq \bigcup\{I(\sigma') : \sigma' \in U\}.$$  

As usual, we may then uniformly compute from $U$ a set $U'$ such that the intervals $I(\sigma')$ for $\sigma' \in U'$ are pairwise disjoint in $F$ and

$$\bigcup\{I(\sigma') : \sigma' \in U\} = \bigcup\{I(\sigma') : \sigma' \in U'\}.$$  

For each $\sigma' \in U'$, let $Q(\sigma') \subseteq I(\sigma)$ be the $\Pi^0_1$ class in $F$ consisting of those extensions of $\sigma'$ which actually have zeroes in each $I(\tau_j)$. Then in fact we have

$$C_\sigma = \bigcup\{Q(\sigma') : \sigma' \in U'\}.$$  

As noted above, we can actually compute the measure $\mu^*(Q(\sigma'))$ uniformly from $\sigma'$ by expressing $Q(\sigma')$ as an effective decreasing intersection of clopen sets. Thus for each $\sigma'$, we can compute a clopen set $B(\sigma')$ such that $Q(\sigma') \subseteq B(\sigma') \subseteq I(\sigma')$ and
\(\mu^{**}(B(\sigma')) \leq 2 \cdot \mu^{**}(Q(\sigma'))\). Let

\[B_\sigma = \bigcup \{B(\sigma') : \sigma' \in U'\}.\]

Then we have \(C_\sigma \subseteq B_\sigma\) and \(\mu^{**}(B_\sigma) \leq \mu^{**}(C_\sigma)\).

Finally, for each \(i\), let

\[S'_i = \bigcup B_{\sigma'_i(i,n)}.\]

Then by Proposition 4.3.7, \(\mu^{**}(S'_i) \leq 2 \cdot \mu^I(S_i) \leq 2^{-i-1}\) and therefore there exists some \(i\) such that \(G \notin S'_i\), since \(F\) is random. But this means that \(Z(G) \notin S_i\) and hence \(Z(F)\) meets the Martin-Löf test. Thus \(Z(F)\) is random, as desired. \(\square\)

### 4.4.2 A Negative Result: Images, in general

In general, the image of a random continuous function need not be a random closed set. To see this, recall the statement of Theorem 4.3.2. That is, given \(\sigma \in \{0,1\}^*\), the probability that the image of a continuous function \(F\) meets \(I(\sigma)\) is always > \(\frac{3}{4}\). We obtain the following corollary.

**Corollary 4.4.3.** For any \(y \in 2^\mathbb{N}\),

(a) \(\mu^{**}(\{F : y \in F[2^\mathbb{N}]\}) = \frac{3}{4}\);

(b) there exists a random continuous function \(F\) with \(y \in F[2^\mathbb{N}]\).

**Proof.** (a) Let \(p\) be the probability that \(y \in F[2^\mathbb{N}]\). It follows that for each \(\sigma \in \{0,1\}^n\), the probability that \(y \in F[I(\sigma)]\), given that \(f(\sigma)\) is consistent with \(y\), also equals \(p\). It follows from the proof of Theorem 4.3.2 that \(p = \frac{3}{4}\).

(b) Since the random continuous functions have measure 1 in \(C(2^\mathbb{N})\), it follows that some random continuous function has \(y\) in the image. \(\square\)

This allows us to demonstrate our result.

**Theorem 4.4.4.** The image of a random continuous function need not be a random closed set.

**Proof.** It was shown in Theorem 3.4.12 that a random closed set has no computable members. Let \(F\) be a random continuous function with \(0^\omega\) in the image, as given by Corollary 4.4.3. Then \(F[2^\mathbb{N}]\) is not a random closed set. \(\square\)
4.5 Pseudo-Distance Functions

In section 4.4.1 we showed that if $\Delta$ is a random continuous function, then $\Delta^{-1}(0^\omega)$ is a random closed set, if it is nonempty. This motivates the study of pseudo-distance functions.

**Definition 4.5.1.** $\Delta : 2^N \to 2^N$ is a pseudo-distance function for $Q \subseteq 2^N$ if $\Delta$ is continuous and $\Delta^{-1}(0^\omega) = Q$.

**Comment 4.5.2** (Background). The name comes from a modification of the distance function $\text{dist}_Q : 2^N \to [0, 1]$ for a closed set $Q$. For $x \in 2^N$, $\text{dist}_Q(x)$ is defined to be $\min\{d(x, y) : y \in Q\}$ where $d$ is a metric on $2^N$ given by

$$d(x, y) = \begin{cases} 0 & \text{if } x = y; \\ 2^{-n} & \text{if } n \text{ is the least such that } x(n) \neq y(n). \end{cases}$$

This may be viewed as a computable mapping from $2^N \times 2^N$ into $2^N$ by representing $0^\omega$ and $2^{-n}$ as $0^n10^\omega$. From this viewpoint, we may view $\text{dist}_Q$ similarly:

$$\text{dist}_Q(x) = \begin{cases} 0^\omega & \text{if } x \in Q; \\ 0^n10^\omega & \text{otherwise, where } n \text{ is the least such that } x[n] \notin T_Q. \end{cases}$$

Every closed set has a characterization in terms of pseudo-distance functions, as follows.

**Theorem 4.5.3.** $Q \subseteq 2^N$ is closed iff there is a pseudo-distance function for $Q$.

**Proof.** First suppose that $Q$ is closed and $Q = [T]$. Define a map $\widehat{\Delta} : 2^{<\omega} \to 2^{<\omega}$ by recursion as follows, with initial mapping $\emptyset \mapsto \emptyset$.

$$\widehat{\Delta}(\sigma^{-i}) = \begin{cases} \widehat{\Delta}(\sigma)\overline{0} & \text{if } \sigma^{-i} \in T; \\ \widehat{\Delta}(\sigma)\overline{1} & \text{otherwise.} \end{cases}$$

Letting $\Delta : 2^\omega \to 2^\omega$ be defined so that $\Delta(X)$ is the unique $Y \in \cap_n[\widehat{\Delta}(X \upharpoonright n)]$, we obtain the required pseudo-distance function.
Now suppose that $\Delta : 2^\omega \to 2^\omega$ is a pseudo-distance function for $Q$. This means that there is some $\hat{\Delta} : 2^{<\omega} \to 2^{<\omega}$ such that $\Delta(X)$ is the unique $Y \in \cap_n [\hat{\Delta}(X \upharpoonright n)]$. Define the required tree $T$ (so that $Q = [T]$) as follows. Put $\sigma \in T$ iff $\hat{\Delta}(\sigma) \subseteq 0^\omega$.

For effectively closed sets, such a characterization in terms of pseudo-distance functions might not seem as immediate, as an effectively closed set may possess a noncomputable distance function. Nonetheless, the we have the following theorem.

**Theorem 4.5.4.** $Q \subseteq 2^N$ is effectively closed iff there is computable pseudo-distance function for $Q$.

**Proof.** The follows as in the proof of Theorem 4.5.3, except that $\hat{\Delta}$, in both directions, is now computable.

It seems plausible that there is a pseudo-distance function characterizaton for random closed sets. Looking first to distance functions, every random closed set posseses the non-random distance function $\text{dist}_Q$. (To see why $\text{dist}_Q$ is non-random, note that if $\sigma \notin T_Q$, then $d$ is constant on the interval $I(\sigma)$.) If a characterization exists, as in the case for effectively closed sets, it appears to be not so immediate. On the other hand, we know, by Theorem 4.4.2, that if $Q$ is a closed set with a random pseudo-distance function, then $Q$ is random. This leads us to the following conjecture.

**Conjecture 4.5.5.** $Q \subseteq 2^N$ is random closed iff there is a random pseudo-distance function for $Q$.

### 4.6 n-Randomness

Recall that a real is $n + 1$-random if and only if it is $1$-random relative to $\emptyset^{(n)}$ (see Remark 3.2.11). Now the analogue of Theorem 3.2.10 also holds for $\{0, 1, 2\}^N$ for our measures $\mu^*$ or $\mu^{**}$. Therefore our approach also allows us to define the notion of $n$-random continuous functions (or $n$-random closed sets), as follows. A continuous function $F : 2^N \to 2^N$ (or closed set) is defined to be $n$-random if and only if it is Martin Löf random relative to $\emptyset^{(n)}$. One can then easily relativize the results of previous sections to obtain similar results for $n$-random continuous functions.
4.7 Future Work

We close this chapter noting that random Brownian motions as studied by Fouche [45] are a special case of random continuous functions on the real line. This is another area of interest for further research. That is, we would like to extend the notion of a random continuous function to functions on the real unit interval \([0, 1]\) and the real line \(\mathbb{R}\) by representing functions again in terms of the images of subintervals. We conjecture that a random continuous real function cannot be left or right computable and in fact, not weakly computable. We also conjecture that a random continuous function is nowhere differentiable.
CHAPTER 5
CONTINUITY OF CAPPING IN $C_{bt}$

The following chapter is joint work with Angsheng Li and Weilin Li and will appear in the Annals of Pure and Applied Logic as an article entitled Continuity of Capping in $C_{bt}$ [16]. For this project, P. Brodhead acknowledges support from the National Science Foundation (under grant number 0714151 as the principal investigator) to conduct this joint work in Beijing during the summer of 2007 as part of the East Asia and Pacific Summer Institutes (EAPSI).

This work was presented by P. Brodhead at the First Joint AMS-NZMS Meeting (December 2007, Wellington, New Zealand).

5.1 Introduction

Given sets $A, B \subseteq \omega$, we say that $A$ is Turing reducible to $B$, if there is an oracle Turing machine $\Phi$ say, such that $A = \Phi^B$ (denoted by $A \leq_T B$). Furthermore, if the bits of oracle queries are bounded by a computable function, then using recent nomenclature from Soare [88] we say that $A$ is bounded Turing reducible to $B$, written $A \leq_{bT} B$. (The literature often refers to this as the weak truth table reducibility, written $\leq_{wtt}$.) A Turing and a bounded Turing (or bT, for short) degree is the equivalence class of a set under the Turing reductions and the bounded Turing reductions respectively. A degree is called computably enumerable (c.e.), if it contains a c.e. set. Let $C$ and $C_{bT}$ be the structures of the c.e. degrees under the Turing reductions and the bounded Turing reductions respectively.

During the past decades, the studies of the structures $C, C_{bT}$ focused on that of the algebraic properties, leading to major achievements such as the decidability results of the $\Sigma_1$-theory of $C$, and the $\Sigma_2$-theory of $C_{bT}$ (Ambos-Spies, P. Fejer, S. Lempp and M. Lerman [3]), and the undecidability results of the $\Sigma_3$-theory of $C$ (Lempp, Nies, and Slaman [63]), and of the $\Sigma_4$-theory of $C_{bT}$ (Lempp and Nies [62]). This progress brings the decidability problems of the $\Sigma_2$-theory of $C$, and the $\Sigma_3$-theory of $C_{bT}$ into sharper focus, for which new ingredients are welcome.
In the recent years, the study of the computably enumerable degrees has focused on Turing definability in the structure $C$. For instance, Slaman asked in 1985 if there are any c.e. degrees that are incomplete and nonzero which are definable in the c.e. degrees $C$. This question of Slaman is still open. A natural approach to this problem is to find some definable substructures of $C$ that have nontrivial minimal/maximal and/or least/greatest members. As a result, topics such as the continuity of the c.e. degrees, started by Lachlan in 1967, have renewed interest.

In this chapter, we demonstrate the continuity of capping in $C_{bT}$. This refutes the existence of a maximal non-bounding degree. It also brings the question of the $\Sigma_3$-theory of $C_{bT}$ into sharper focus, as the statement is one of $\Sigma_3$-complexity.

To motivate these ideas further, we begin with a brief history of relevant continuity results in Section 5.2. This motivates our main result and method of proof, described in Section 5.2.2. The main substance of the proof involves demonstrating that Theorem 5.2.3 holds, that local noncappability holds in $C_{bT}$. Sections 5.3–5.6 are devoted to proving this theorem.

5.2 Continuity Results

The Turing degrees form an upper-semilattice; that is, each pair of elements $a, b$ has a least upper bound (or join) $a \lor b$. A greatest lower bound $a \land b$ may or may not exist. Given a (bounded) Turing degree $a$, we say that $a$ is cappable if a (bounded-) Turing degree $b \neq 0$ exists such that $a \land b = 0$. We say that $a$ is cuppable if there is a degree $b \neq 0'$ such that $a \lor b = 0'$. The study of continuity properties the (bounded) Turing degrees is with respect to meets and joins, and related notions such as capping and cupping.

5.2.1 Continuity Results in $C$

In 1979, Lachlan [60] proved the existence of a non-bounding c.e. degree—namely, a non-computable c.e. degree with no minimal pair below it; Cooper demonstrated in 1974, that no high c.e. degree can be non-bounding [30]. Returning to the Slaman question, Downey, Lempp, and Shore demonstrated in 1993 that Cooper’s non-bounding degree
could be made high₂ [41], leading to the possibility of a maximal non-bounding c.e. degree. Such a degree could be used to show the existence of a discontinuity, which could be used to prove the definibility of a c.e. singleton. (To see the former, note that if \( b > 0 \) is nonbounding, then for any c.e. \( a > b \), there is some minimal pair \( r \land s \) below \( a \). Then either \( b \land r = 0 \) or \( b \land s = 0 \), but neither \( a \land r \) nor \( a \land s \) equals 0.) However, Seetapun refuted this possibility (albeit earlier in 1991), demonstrating the non-existence of a maximal non-bounding degree [84]. Welch proved a complimentary result in 1981: there is no maximal bounding degree, in the sense that for all \( a \neq 0' \), there is are \( b, c \) such that \( b \land c = 0 \) and \( b, c \leq a \) [95].

Continuing with capping results, Harrington and Soare [51] proved in 1989, the nonexistence of maximal minimal pairs— that is, for any non-trivial minimal pair \((a, b)\) of c.e. Turing degrees \( a, b \), there exists a c.e. Turing degree \( c > a \) such that \((c, b)\) is still a minimal pair. Seetapun [84] showed an even stronger result, the continuity of capping: for any c.e. Turing degree \( b \neq 0, 0' \), there exists a c.e. Turing degree \( a > b \) such that for any c.e. Turing degree \( x \), if \( x \leq a \), then \( a \land x = 0 \) if and only if \( b \land x = 0 \).

Ambos-Spies, Lachlan, and Soare [4] proved the dual case of the Harrington and Soare’s result: for any non-trivial splitting \( x, y \) of \( 0' \), there exists a c.e. degree \( a < x \) such that \( a \lor y = 0' \). Cooper and Li [31] showed the dual of the Seetapun theorem, that for any c.e. Turing degree \( b \neq 0, 0' \), there exists a c.e. Turing degree \( a < b \) such that for any c.e. Turing degree \( x \), \( x \lor a = 0' \) if and only if \( b \lor x = 0' \), answering Lachlan’s major subdegree problem.

5.2.2 Continuity Results in \( C_{bT} \) and Main Result

There are few results in the topic of definability in the c.e. bT-degrees, \( C_{bT} \). For instance, we know nothing about the Slaman problem as described in Section 5.1 or the characterization of definable ideals in \( C_{bT} \). As a matter of fact, little is known of the continuity properties of the c.e. bT-degrees. An interesting partial result was given by Stob [90]: in both \( C \) and \( C_{bT} \), there are c.e. degrees \( a_0, a_1 > 0 \), with \( a_0 \) being the unique complement of \( a_1 \) in the interval \([0, a_0 \lor a_1]\), such that if \( b < a_0 \lor a_1 \) is cappable with
any $x < a_0$ (i.e. $b \land x = 0$), then $x < a$. This can be interpreted as a continuity or discontinuity result in both $C$ and $C_{bT}$.

The main result of this chapter is the following theorem, an analogue of Seetapun continuity result.

**Theorem 5.2.1** (Continuity of Capping in $C_{bT}$). For any c.e. bT-degree $b \neq 0, \varnothing$, there is a c.e. bT-degree $a > b$ such that for any c.e. bT-degree $x$, $b \land x = 0 \leftrightarrow a \land x = 0$.

For this, it suffices to prove Theorem 5.2.3 below, an analog of the Seetapun local noncappability theorem for the c.e. Turing degrees [84].

**Definition 5.2.2** (Local Noncappability). A degree $b \neq 0$ is locally non-cappable if there is some $a > b$ such that for all $x < a$, if $x \land b = 0$ then $x = 0$.

**Theorem 5.2.3** (Local Noncappability in $C_{bT}$). For any c.e. bT-degree $b$, if $b \neq 0, \varnothing$, then there is a c.e. bT-degree $a > b$ such that if $x \leq a$ is noncomputable, then $x \land b \neq 0$.

**Proof of Theorem 5.2.1.** Assuming Theorem 5.2.3, we can see Theorem 5.2.1. Given $b$, let $a$ be the degree in Theorem 5.2.3. For a fixed $x \in C_{bT}$, by Theorem 5.2.3, we consider only the case where $x \not\leq a$. Clearly if $x \land a = 0$, then $x \land b = 0$. Assume $a \land x \neq 0$.

We can choose a c.e. bT-degree $y$ such that $y \neq 0$ and $y \leq a, x$. Therefore $0 < y \leq a$ and by Theorem 5.2.3, we have $y \land b \neq 0$, so that $x \land b \neq 0$. Theorem 5.2.1 follows.

As a consequence of Theorem 5.2.1, no maximal non-bounding degrees exist in the c.e. bT-degrees. To see this, suppose $b \neq 0$ is a degree which bT-bounds no minimal pairs in $C_{bT}$, and let $a > b$ be the degree in Theorem 5.2.1. We claim that there are no bT-minimal pairs below $a$. Suppose to the contrary that $x, y$ is a minimal pair below $a$. Since $a \land x = x \neq 0$ and $a \land y = y \neq 0$, we can choose nonzero $x_1$ to be below both $x$ and $b$, and nonzero $y_1$ below both $y$ and $b$. Then $(x_1, y_1)$ is a minimal pair below $b$, a contradiction. An alternative approach is to use the fact that a maximal non-bounding c.e. degree is equivalent to a non-bounding degree which is not locally non-cappable [49, 84]. Consequently, by Theorem 5.2.3, no maximal non-bounding degree can exist.

Our approach to the proof of theorem 5.2.3 is similar to Seetapun’s approach for the c.e. Turing degrees, but it is non-obvious due to the computable bounds of oracle query
bits in both the conditions and conclusions of requirements. That is, bT-reductions are stronger than Turing reductions. So when we require the reductions being built to be bT-reductions, we must satisfy stronger conditions and, in this sense, the problem becomes harder to solve. For example, A. Li, W. Li, Y. Pan, and L. Tang [66] have shown that the solution to the major sub-degree problem in $C_{bT}$ (i.e. the dual to the continuity problem) is completely different from the result for $C$ (see Cooper and Li [31]). They show that the statement of the solution in $C$ fails badly in $C_{bT}$: there exist c.e. bT-degrees $a, b$ such that $0 < a < 0'$, and for any c.e. bT degree $x$, $b \lor x = 0'$ if and only if $x \geq a$.

Our approach might not be the only one. Klaus Ambos-Spies proved that for any c.e. set, its Turing degree is cappable in the Turing degrees iff its bT-degree is cappable in the bT-degrees [2]; we thank an anonymous referee for pointing this out. Therefore, another possible approach might be to prove, if possible, that for any two c.e. sets, their Turing degrees form a minimal pair in the Turing degrees iff their bT-degrees form a minimal pair in the bounded Turing degrees. As consequence, our continuity result would immediately follow from Seetapun’s continuity result. We comment that although our continuity proof might be non-obvious, from the above perspective, oftentimes bT-degrees can be handled much more easily than Turing degrees [1]. Ambos-Spies provides various examples [1]. For example, density of the c.e. bT-degrees can be proved by a finite injury priority argument, whereas the same result requires an infinite injury argument for the c.e. Turing degrees.

The rest of this chapter is devoted to proving Theorem 5.2.3, the main result. In section 5.3, we formulate the conditions of the theorem by requirements; in section 5.4, we arrange all strategies to satisfy the requirements on the nodes of a tree, or more precisely, the priority tree $T$. In section 5.5, we use the priority tree to describe a stage-by-stage construction of the objects we need. Finally, in section 5.6 we verify that the construction in section 5.5 satisfies all of the requirements, finishing the proof of the theorem.

Our notation and terminology are standard and generally follow Soare [86]. During the course of a construction, notations such as $A, \Phi$ are used to denote the current approximations to these objects, and if we want to specify the values immediately at
the end of stage $s$, then we denote them by $A_s, \Phi[s]$ etc. For a partial computable (p.c., or for simplicity, also a Turing) functional, $\Phi$ say, the use function is denoted by the corresponding lower case letter $\phi$. The value of the use function of a converging computation is the greatest number which is actually used in the computation. For a Turing functional, if a computation is not defined, then we define its use function $= -1$.

During the course of a construction, whenever we define a parameter, $p$ say, as fresh, we mean that $p$ is defined to be the least natural number which is greater than any number mentioned so far. In particular, if $p$ is defined afresh at stage $s$, then $p > s$.

### 5.3 Requirements and Strategies

In this section we provide the requirements and strategies for proving Theorem 5.2.3. We restate it here for convenience.

**Theorem 5.2.3** (Local Noncappability in $C_{bT}$). For any c.e. bT-degree $b$, if $b \neq 0, 0'$, then there is a c.e. bT-degree $a > b$ such that if $x \leq a$ is noncomputable, then $x \wedge b \neq 0$.

#### 5.3.1 The requirements

Given a c.e. set $B$, we will build a c.e. set $A$ to satisfy the following requirements:

- $P_e : A \neq \Psi_e(B) \vee K \leq_{bT} B$
- $R_e : X_e = \Phi_e(A, B) \rightarrow (\exists$ c.e. $D_e)[D_e \leq_{bT} X_e, B \& (\forall i)S_{e,i}]$
- $S_{e,i} : D_e \neq \lambda_i \vee X_e \leq_T \emptyset \vee B \leq_T \emptyset$

where $e, i \in \omega, \{(\Phi_e, \Psi_e, X_e) : e \in \omega\}$ is an effective enumeration of all triples ($\Phi, \Psi, X$) of all bounded Turing (bT, for short) reductions $\Phi, \Psi$, and of all c.e. sets $X$; $\{\lambda_i : i \in \omega\}$ is an effective enumeration of all partial computation functions $\lambda$; and $K$ is a fixed creative set. $D_e$ for all $e$, are c.e. sets built by us.

Let $a, b, x, d$ be the bT-degrees of $A \oplus B, B, X, D$, respectively. By the $P$-requirements, $a > b$ (unless $b$ was already the degree of $0'$), and by the $R$-requirements, if $x \leq a$ there is a $d$ below both $x$ and $b$ such that $d \neq 0$ unless either $x = 0$ or $b = 0$. Therefore the requirements are sufficient to prove the theorem.
Before describing the strategies, we introduce some conventions of the bounded Turing reductions. We will assume that for any given bounded Turing reduction $\Phi$ or $\Psi$, the use functions $\phi$ and $\psi$ will be increasing in arguments.

5.3.2 A $\mathcal{P}$-strategy

A $\mathcal{P}$-strategy will try to satisfy a $\mathcal{P}$-requirement, $\mathcal{P}$ say (we drop the index in the following discussion). We use a node on a tree, $\gamma$ say, to denote a $\mathcal{P}$-strategy. It aims to ensure that if $A = \Psi(B)$, then there is a bounded Turing reduction $\Delta$ such that $\Delta(B) = K$. Therefore the $\mathcal{P}$-strategy $\gamma$ will try to build a bounded Turing reduction $\Delta$. $\Delta$ will be built by an $\omega$-sequence of cycles $k$. Each cycle $k$ of $\gamma$ will be responsible for defining $\Delta(B; k)$ as follows: first $\gamma$ chooses a fresh witness $a(k)$ and waits for a stage, $v$ say, at which we have $\Psi(B; a(k)) \downarrow = 0 = A(a(k))$. When this occurs, we define $\Delta(B; k)$ to be $K(k)$ with use function $\delta(k) = \psi(a(k))$. Since $\psi$ is partial computable, so is the use function $\delta$ of $\Delta$. We will always assume that whenever $B$ changes below the $\delta$-use, $\delta(k)$ say, the corresponding computation $\Delta(B; k)$ becomes undefined simultaneously. Suppose that at a later stage $s > v$, $k$ is enumerated into $K$, and $B$ has not changed since $\Delta(B; k)$ was last created, then $\Delta(B; k) \neq K(k)$. In this case, we enumerate $a(k)$ into $A$ so that an inequality $\Psi(B; a(k)) \neq A(a(k))$ is created. The key point is that, if $\Delta(B; k) \neq K(k)$ is a permanent inequality, so is $\Psi(B; a(k)) = 0 \neq 1 = A(a(k))$.

The $\mathcal{P}$-strategy $\gamma$ will start cycles $k$ in increasing order of $k$. Cycle $k$ acts only if the following conditions occur:

1. For all $k' < k$, $\Delta(B; k') \downarrow = K(k')$.
2. Either $a(k) \uparrow$, or $\Delta(B; k) \uparrow$ and $\Psi(B; a(k)) = A(a(k))$, or $\Delta(B; k) \downarrow = 0 \neq 1 = K(k)$.

As we have seen in the above analysis, if there is a permanent inequality between $\Delta(B)$ and $K$, there is a corresponding permanent inequality between $\Psi(B)$ and $A$. Since $\Delta$ is a bounded Turing reduction (with use bound $\delta$), we have that if $\Delta$ is built infinitely many times, then $\Delta(B)$ is total, and $\Delta(B) = K$. Hence $K$ is $bT$-reducible to $B$. Suppose this never occurs, then the $\mathcal{P}$-strategy $\gamma$ acts only finitely many times, which will be denoted by 1. Therefore a $\mathcal{P}$-strategy $\gamma$ has only one possible outcome 1, unless $K \leq_{bT} B$. 
5.3.3 An $\mathcal{R}$-strategy

Before describing the $\mathcal{R}$-strategy, we introduce a convention of the bounded Turing reduction $\Phi$. We assume that for any $x$ and any $s$, if $x$ enters $X$ at stage $s$, then $\Phi(A, B; x)[s] \downarrow = 1$.

Given an $\mathcal{R}$-requirement, $\mathcal{R}$ say, we define the length function of agreement as usual. That is to say: At stage $s$, the length function of agreement $\ell$ between $\Phi(A, B)$ and $X$ is defined as the largest $x$ such that $\Phi(A, B)$ and $X$ agree on all values $y < x$:

$$\ell = \ell(X, \Phi(A, B))[s] = \max \{x : (\forall y < x)[\Phi(A, B; y)[s] \downarrow = X(y)[s]\}$$

Stage $s$ is said to be $\mathcal{R}$-expansionary if the length function of agreement increases; that is, if for all $v < s$, $\ell[v] > \ell[s]$. At $\mathcal{R}$-expansionary stages, an $\mathcal{R}$-strategy builds bounded Turing reductions $\Theta(X), \Xi(B)$ (with use functions $\theta, \xi$, respectively) so that for the least undefined $x < \ell$:

$$\Theta(X, x) \downarrow = D(x) \text{ with } \theta(x) = x \text{ and } \Xi(B, x) \downarrow = D(x) \text{ with } \xi(x) = \phi(x) \quad (5-1)$$

where $\phi$ is the use of $\Phi(A, B)$, and $D$ is a c.e. set, whose elements are enumerated into it by lower priority $\mathcal{S}$-strategies associated with $\mathcal{R}$, to satisfy the $\mathcal{R}$-requirement.

Suppose that $\alpha$ is an $\mathcal{R}$-strategy. As above, the use functions $\theta$ and $\xi$ of the bounded Turing reductions $\Theta$ and $\Xi$ built by $\alpha$ are the identity function and $\phi$ respectively, so that both $\Theta$ and $\Xi$ are bounded Turing reductions. (Notice that $\Phi(A, B)$ is a bounded Turing reduction, so that the use $\phi$ is a partial computable function.)

To satisfy $\Theta(X) = D$ and $\Xi(B) = D$, the $\mathcal{R}$-strategy $\alpha$ will impose the following constraints on all $\mathcal{S}$-strategies with the same global index as the $\mathcal{R}$-strategy $\alpha$:

For any $s$, and any $d$, $d$ is allowed to be enumerated into $D_\alpha$ at stage $s$ only if both $\Theta_\alpha(X; d)$ and $\Xi_\alpha(B; d)$ are undefined during stage $s$.

We assume that for the bounded Turing reductions $\Theta$ and $\Xi$, any computation will automatically become undefined, whenever the oracle changes below the corresponding use.
By the building of $\Theta_\alpha$ and $\Xi_\alpha$, and by the constraints of $\alpha$, we have that if $\Theta_\alpha$ and $\Xi_\alpha$ are built infinitely many times, then both $\Theta_\alpha(X)$ and $\Xi_\alpha(B)$ are total, and both equal $D_\alpha$. Hence $R$ is satisfied.

Therefore the key point towards the satisfaction of $R$ is that if there are infinitely many $R$-expansionary stages, then both $\Theta_\alpha$ and $\Xi_\alpha$ are built infinitely many times.

We thus define the possible outcomes of the $R$-strategy $\alpha$ by

$$0 <_L 1$$

to denote infinite and finite expansionary stages respectively.

### 5.3.4 An $S$-strategy

Suppose that we want to satisfy an $S$-requirement, $S_{e,i}$ say. For simplicity, we use $R$ and $S$ to denote $R_e$ and $S_{e,i}$ respectively. Suppose that $\alpha$ and $\beta$ are the $R$- and $S$-strategies respectively. Let $\alpha^\downarrow(0) \subseteq \beta$.

$\beta$ attempts to find some $d$ such that $\lambda(d) \downarrow = 0$ with an expectation of enumerating $d$ into $D$ to create an inequality $\lambda(d) = 0 \neq 1 = D(d)$. However $\beta$ can enumerate a number $d$ into $D_\alpha$ at a stage, $s$ say, only if both $\Theta_\alpha(X;d)$ and $\Xi_\alpha(B;d)$ are undefined during stage $s$ as required by the $R$-strategy $\alpha$. Therefore, $\beta$ will prepare a sequence of possible candidates $c$'s such that

- $\lambda(c) \downarrow = 0 = D(c)$,
- $\Theta_\alpha(X;c) \downarrow$, and
- $\Xi_\alpha(B;c) \downarrow$.

For the largest $c$, we build a partial computable $f_\beta$ as follows:

- for every $y \leq \xi_\alpha(c)$, if $f_\beta(y)$ is undefined, then define $f_\beta(y) = B(y)$.
- define $d(\beta) = c$, and set $c$ to be undefined, which allows us to define a larger $c$.

Suppose that there is an error between $f_\beta$ and $B$, in the sense that there is a $y$ such that $f_\beta(y) \downarrow = 0 \neq 1 = B(y)$ occurs at a stage, $v$ say. Then we open an $A$-gap:

- build a partial computable function $g_\beta$ to simulate $X_\alpha$ as follows: for every $x \leq d(\beta)$, if $g_\beta(x)$ is undefined, define $g_\beta(x) \downarrow = X_\alpha(x)$,
• set $f_\beta$ to be totally undefined (the $f_\beta$ proves wrong, so it is cancelled),
• drop the A-restraint by defining $r^A(\beta) = -1$, and
• create a link $(\alpha, \beta)$.

[Notice that at stage $v$, $\Xi_\alpha(B; d(\beta))$ is undefined due to the $B$-change in the domain of $f_\beta$. We regard this as a $B$-permission for the enumeration of $d(\beta)$ into $D_\alpha$. This $B$-permission will be kept until the current link $(\alpha, \beta)$ is either travelled or cancelled so that, in either case, the link is removed.]

Suppose that $\beta$ creates a link $(\alpha, \beta)$ at stage $v$. Then the link $(\alpha, \beta)$ will be travelled at the next $\alpha$-expansionary stage $s > v$. Now we consider two cases:

**Case 1.** There is an error between $g_\beta$ and $X_\alpha$.

In this case, there is an $x \leq d(\beta)$ which has entered $X_\alpha$ since stage $v$. Therefore $\Theta_\alpha(X_\alpha; d(\beta))$ is currently undefined. Together with the condition that $\Xi_\alpha(B; d(\beta)) \uparrow$, found at the stage we created the current link $(\alpha, \beta)$, $\beta$ is qualified to enumerate $d(\beta)$ into $D_\alpha$. $\mathcal{S}$ is satisfied by $\lambda(d(\beta)) = 0 \neq 1 = D_\alpha(d(\beta))$.

**Case 2.** Otherwise, we know that $g_\beta$ is correct during the gap. Therefore we preserve $g_\beta$ on its domain until $\beta$ opens another A-gap. For this, we implement:

• for every $y \leq \phi(d(\beta))$, if $f_\beta(y)$ is undefined, then define $f_\beta(y) = B(y)$, and
• define the A-restraint $r^A(\beta)$ of $\beta$ to be $\phi(d(\beta))$.

[Notice that although we have A-restraint at this stage, $X_\alpha$ may change due to a $B$-change below $\phi(d(\beta))$. The definition of $f_\beta$ at this stage allows us to immediately open an A-gap once such a $B$-change occurs, in which case, we do not increase the domain of $g_\beta$ but resume with the old candidate $d(\beta)$.]

Therefore the $\mathcal{S}$-strategy $\beta$ is a gap/cogap strategy. It will build partial computable functions $f_\beta$ and $g_\beta$ and will proceed as follows:

1. Define a possible candidate $c(\beta)$ as fresh.
2. (Building $f_\beta$) Wait for a stage $v$ at which
   • $\lambda(c(\beta)) \downarrow = 0 = D_\alpha(c(\beta))$,
   • $\Theta_\alpha(X_\alpha; c(\beta)) \downarrow = 0 = D_\alpha(c(\beta))$, and
• $\Xi_\alpha(B; c(\beta)) \downarrow = 0 = D_\alpha(c(\beta))$.

Then for $c = c(\beta)$,

• for every $y \leq \xi_\alpha(c)$, if $f_\beta(y) \uparrow$, then define $f_\beta(y) = B(y)$,

• define $d(\beta) = c$,

• set $c(\beta)$ to be undefined, and go back to step 1.

3. (Creating a link $(\alpha, \beta)$) Let $s$ be the current stage. Suppose that there is a $b$ in the domain of $f_\beta$ that enters $B$ at stage $s$. Notice that the domain of $f_\beta$ is precisely everything $\leq \xi_\alpha(d(\beta)) = \phi_\alpha(d(\beta))$, so that $\Xi_\alpha(B; d(\beta))$ becomes undefined at stage $s$. Then:

• for every $x \leq d(\beta) = \theta_\alpha(d(\beta))$, if $g_\beta(x)$ is undefined, define it to be $X_\alpha(x)$,

• define the $A$-restraint of $\beta$ by $r^A(\beta) = -1$,

• set $f_\beta$ to be totally undefined, and

• create a link $(\alpha, \beta)$.

4. (Travelling the link $(\alpha, \beta)$) We travel the link $(\alpha, \beta)$ at the next $\alpha$-expansionary stage $t > s$. There are two cases:

Case 4a. (Successful closure) There is an $x$ such that $g_\beta(x) \downarrow = 0 \neq 1 = X_\alpha(x)$. (This $x$ must enter $X_\alpha$ since the current link $(\alpha, \beta)$ was created.) Then:

• enumerate $d(\beta)$ into $D_\alpha$, and stop.

Case 4b (Unsuccessful Closure) Otherwise, then

• for every $y \leq \phi(d(\beta))$, if $f_\beta(y)$ is undefined, then define $f_\beta(y) = B(y)$,

• define an $A$-restraint of $\beta$ by $r^A(\beta) = \phi_\alpha(d(\beta))$.

The Possible Outcomes

We consider the following cases.

Case 1. Case 4a occurs at some stage $t$.

In this case, $\lim_s d(\beta)[s] \downarrow = d(\beta) < \omega$, and $\lambda(d(\beta)) = 0 \neq 1 = D_\alpha(d(\beta))$ is created. $S$ is satisfied.

Case 2. Otherwise, and Case 4b occurs infinitely many times.
Notice that \( g_\beta \) is never set to be totally undefined, and that for a fixed number \( d \), \( \phi(d) \) is a fixed number, so that \( B \) changes below \( \phi(d) \) only finitely many times, and so that Step 3 occurs with the same \( d \) only finitely many times. Since Case 4b occurs infinitely many times, we have that \( d(\beta)[s] \) will be unbounded over the course of the construction, and that whenever Step 3 occurs, we build \( g_\beta \) on the initial segment of the current \( d(\beta) \). Therefore \( g_\beta \) is built as a computable function.

For an arbitrarily given \( x \), we prove \( g_\beta(x) \downarrow = X_\alpha(x) \). Let \( s \) be the stage at which \( g_\beta(x) \) is created. Suppose that \( s_i \) are all stages \( s' \geq s \) at which Step 3 of \( \beta \) occurs, and that for each \( s_i, t_i \in (s_i, s_{i+1}) \) is the stage at which the link \((\alpha, \beta)\) created at stage \( s_i \) is travelled through Case 4b.

By the choice of \( s_i, s_0 = s \). Since Case 4b occurs at stage \( t_0 \), and \( t_0 \) is \( \alpha \)-expansionary, we have that

(i) \( g_\beta(x) = X_\alpha[s_0](x) \) (\( \alpha \) will never be visited at stage \( s_0 \)).

(ii) For any \( s \in [s_0, t_0], g_\beta(x) = X_\alpha[s](x) \).

(iii) \( g_\beta(x) = \Phi_\alpha(A, B; x)[t_0] = X_\alpha[t_0](x) \).

By the \( A \)-restraint \( r^A(\beta)[t_0] \), and the convention of \( \Phi \), we have that for any \( t \in [t_0, s_1] \),

(iv) \( g_\beta(x) = \Phi_\alpha(A, B; x)[t] = X_\alpha[t](x) \).

Suppose by induction that for \( n \), we have that

(A) For any \( s \in [s_n, t_n], g_\beta(x) = X_\alpha[s](x) \).

(B) \( g_\beta(x) = X_\alpha(x)[t_n] = \Phi_\alpha(A, B; x)[t_n] \),

(C) For any \( t \in [t_n, s_{n+1}], g_\beta(x) = \Phi_\alpha(A, B; x)[t] = X_\alpha[t](x) \), and

(D) \( g_\beta(x) = X_\alpha[s_{n+1}](x) \).

By (C), (D) for \( n \), and by the choice of \( t_{n+1} \), (A) holds for \( n + 1 \). By (A) for \( n + 1 \), and the choice of \( t_{n+1} \), we have (B) for \( n + 1 \). By (B) for \( n + 1 \), by the \( A \)-restraint at stage \( t_{n+1} \), and by the convention of \( \Phi \), (C) holds for \( n + 1 \). (D) for \( n + 1 \) follows from (C) for \( n + 1 \) and the assumption that \( \alpha \) is not visited at stage \( s_{n+1} \).

[Remark. We will arrange the construction so that an \( A \)-gap can be opened only at odd stages, and that no \( R \)-strategies can be visited at these stages. This means that no
for any $\mathcal{R}$-strategy $\alpha$ can receive elements at odd stages, since we assume that $X_\alpha$ is enumerated only at stages at which $\alpha$ is visited.]

Therefore for any $s \geq s_0$, either $g_\beta(x) = X_\alpha(x)[s]$ or $g_\beta(x) = \Phi_\alpha(x)[s]$. Since $\Phi_\alpha(A, B; x)$ equals $X_\alpha(x)$, we have that $g_\beta(x) = X_\alpha(x)$.

This is a global win for the requirement $\mathcal{R}$, so that we don’t consider any other $S$-requirements with the same global index with the requirement $\mathcal{R}$.

**Case 3.** Otherwise, and Step 2 occurs infinitely many times.

Since step 3 occurs only finitely many times, $f_\beta$ is set to be totally undefined only finitely many times. Let $f$ be the final version of $f_\beta$. By the assumption that step 2 occurs infinitely many times, $f$ is built as a computable function.

By the choice of $f$, for any $x$, once $f(x)$ is created, we have that $f(x) = B(x)$. $B$ is computable, contradicting the assumption of the theorem. So we assume that this case will never occur.

**Case 4.** Otherwise, then by the strategy, we have that $\lim_s c(\beta)[s] \downarrow = c(\beta) < \omega$ exists, $c(\beta) \notin D_\alpha$, and $\lambda(c(\beta)) = 0$ never occurs. Therefore $\lambda(c(\beta)) \neq 0 = D_\alpha(c(\beta))$. $S$ is satisfied again.

We use $d$, $g$, and $w$ to denote the possible outcomes of $\beta$ corresponding to case 1, case 2, and case 4 respectively. To guarantee that the true outcome will be the one on the leftmost visited path, we define the priority ordering of the possible outcomes as follows:

$$d <_L g <_L w$$

With this ordering, we notice that case 3, in case it happens, will be an outcome between $g$ and $w$.

### 5.4 The Priority Tree

In this section, we will build a priority tree of strategies $T \subset \Lambda^{<\omega}$ with $\Lambda = \{0, 1, d, g, w\}$. Note that there are infinitely many copies of a fixed computable function $\lambda_i$ in $\{\lambda_i : i \in \omega\}$. Therefore, to satisfy a fixed $\mathcal{R}_e$ requirement, it suffices to satisfy all
\( S_{e,i} \) with \( i \geq e \). Let \( P < R \) denote that the priority ranking of \( P \) is higher than that of \( R \).

Also let \( <_L \) be a left-to-right ordering of the nodes on the priority tree, as given below.

**Definition 5.4.1.**  (i) Define a priority ranking of the requirements so that, \( \forall e \in \omega \):

\[
P_e < R_e < S_{0,e} < S_{1,e} < \ldots < S_{e,e} < P_{e+1}
\]

(ii) The possible outcome of a \( P \)-strategy is only 1.

(iii) The possible outcomes of an \( R \)-strategy are 0 \( <_L 1 \).

(iv) The possible outcomes of an \( S \)-strategy are \( d <_L g <_L w \).

**Definition 5.4.2.** Given a node \( \xi \), we say that:

(i) \( P_e \) is satisfied at \( \xi \) if there is a \( P_e \)-strategy \( \gamma \subset \xi \)

(ii) \( R_e \) is satisfied at \( \xi \) if either

- there is some \( R_e \)-strategy \( \alpha \) such that \( \alpha^\langle 1 \rangle \subseteq \xi \), or
- there is some \( S_{e,i} \)-strategy \( \beta \) (for some \( i \)) such that \( \beta^\langle g \rangle \subseteq \xi \)

(iii) \( R_e \) is active at \( \xi \) if \( R_e \) is not satisfied at \( \xi \) and there there is an \( R_e \)-strategy \( \alpha \) with \( \alpha^\langle 0 \rangle \subseteq \xi \), such that there is no \( S_{e,i'} \)-strategy \( \beta' \) with \( \alpha^\langle 0 \rangle \subseteq \beta' \subset \beta''^\langle g \rangle \subseteq \xi \) for any \( e' < e \).

In this case, \( \alpha \) is unique, and we say that \( R_e \) is active at \( \xi \) via \( \alpha \).

(iv) \( S_{e,i} \) is satisfied at \( \xi \) if \( R_e \) is satisfied at \( \xi \) or \( R_e \) is active at \( \xi \) via \( \alpha \), say, and there is an \( S_{e,i} \)-strategy \( \beta \) such that \( \alpha^\langle 0 \rangle \subseteq \beta \subset \xi \).

We now define the priority tree \( T \) inductively as follows.

**Definition 5.4.3 (Priority Tree).**  (i) Define the root node \( \emptyset \) to be a \( P_0 \)-strategy.

(ii) The immediate successors of a node are the possible outcomes of the corresponding strategy.

(iii) A node \( \xi \), say, will work on the highest priority ranking requirement which is not satisfied and not active at \( \xi \).

**Definition 5.4.4.** The index \( I(\xi) \) of a node \( \xi \) is the index of the requirement on which the node acts. For example, if \( \xi \) is an \( R_e \)- or \( P_e \)-strategy, we define \( I(\xi) = e \), and if \( \xi \) is an \( S_{e,i} \)-strategy, we define \( I(\xi) = (e,i) \).

As usual, the priority tree \( T \) will have the following properties.
Proposition 5.4.5. Let \( f \) be an infinite path through \( T \). Then for any requirement \( \mathcal{X} \), there is a node \( \xi_0 \subset f \) such that either (i) or (ii) below holds,

(i) \( \mathcal{X} \) is satisfied at \( \xi \) for any \( \xi \) with \( \xi_0 \subset \xi \subset f \).

(ii) \( \mathcal{X} \) is active at \( \xi \) for any \( \xi \) with \( \xi_0 \subset \xi \subset f \).

Proof. Let \( \{ X_i : i \in \omega \} \) be the priority ranking of the requirements so that \( X_i < X_{i+1} \) for all \( i \). We prove this by induction. Assume that the proposition holds for all \( X_i \) with \( i \leq n \).

Let \( \mathcal{X} = \mathcal{X}_{n+1} \). Fix \( \xi' \subset f \) so that the proposition holds for all \( X_i \) with \( i \leq n \). Let \( \xi_0 \) be an \( \mathcal{X} \)-strategy such that \( \xi' \subset \xi_0 \subset f \). Note that \( \xi_0 \) must exist by the priority ranking of the requirements unless \( \mathcal{X} \) already satisfies Proposition 5.4.5 with the given \( \xi' \). There are three cases.

**Case 1.** \( \xi_0 \) is a \( \mathcal{P} \)-strategy.

The only outcome of a \( \mathcal{P} \)-strategy is 1. Therefore, necessarily \( \xi_0 \langle 1 \rangle \subset f \in [T] \). So the proposition holds with \( \xi'_0 = \xi_0 \langle 1 \rangle \).

**Case 2.** \( \xi_0 \) is a \( \mathcal{R}_{i,e} \)-strategy for some \( e \).

If \( \xi_0 \langle 1 \rangle \subset f \), then the proposition holds with \( \xi'_0 = \xi_0 \langle 1 \rangle \). Otherwise \( \xi'_0 = \xi_0 \langle 0 \rangle \subset f \). Now if \( \mathcal{R}_i \) (\( i < e \)) is satisfied at all \( \xi \) with \( \xi_0 \subset \xi \subset f \), then by the priority ranking of the requirements, there is no \( \mathcal{S}_{i,e} \)-strategy \( \beta \supset \xi_0 \langle 0 \rangle \). Otherwise, by assumption \( \mathcal{R}_i \) is active at all \( \xi \) with \( \xi_0 \subset \xi \subset f \). So the outcome of \( \mathcal{S}_{i,e} \) for each such \( \mathcal{R}_i \) must necessarily be \( d \) or \( w \). Hence \( \xi_1 = \xi_0 \langle 0 \rangle \Lambda \subset f \) where \( \Lambda \) is the concatenated outcomes of all \( \mathcal{S}_{i,e} \)-strategies (\( i < e \)) such that \( \mathcal{R}_i \) is active at all \( \xi \) with \( \xi_0 \subset \xi \subset f \).

Now \( \xi_1 \langle k \rangle \subset f \) for some \( k \in \{ d, w, g \} \). If \( k = g \) then the proposition holds with \( \xi'_0 = \xi_1 \langle g \rangle \). If \( \xi_i = \xi_1 \langle i \rangle \subset f \) (\( i \in \{ d, w \} \)) then by the same reasoning above concerning the \( \mathcal{R}_j \) (\( j < e \)), \( \mathcal{R}_e \) is active at \( \xi \) as long as \( \xi_i \subset \xi \subset f \).

**Case 3.** \( \xi_0 \) is an \( \mathcal{S}_{e,i} \)-strategy.

By the priority ranking of the requirements and by the assumption on the \( \mathcal{X}_i \) (\( i \leq n \)), it follows the \( \mathcal{R}_e \) is active at all \( \xi \) with \( \xi_0 \subset \xi \subset f \). Therefore either \( \xi_w = \xi_0 \langle w \rangle \subset f \) or \( \xi_d = \xi_0 \langle d \rangle \subset f \). By the assumption on \( \mathcal{R}_e \), the proposition holds with \( \xi'_0 = \xi_k \) with appropriate choice of \( k \) so that \( \xi_k \subset f \). □
Definition 5.4.6. If \( \beta \) is an \( S_{e,i} \)-strategy for some \( e,i \), then define \( \text{top}(\beta) \) to be the longest \( R_{e} \)-strategy \( \alpha \) such that \( \alpha^{\uparrow}(0) \subseteq \beta \).

5.5 The Construction

Our construction will perform different actions at even and odd stages. At even stages, strategies on the tree will act to satisfy the requirements.

Suppose that \( B \) is enumerated at odd stages only, and that at every odd stage, there is exactly one element that enters \( B \). Given a \( B \)-permission in the construction, we want to open \( A \)-gaps for as many \( S \)-strategies as possible. This allows us to specify a \( P \)-strategy so that we enumerate its witness into \( A \). So we will ensure that \( A \)-restraints drop at odd stages, and also that \( A \) is only enumerated at odd stages.

During the course of the construction, we may initialize a node, \( \xi \) say, which means that all the actions taken by \( \xi \) previously, are cancelled, or set to be totally undefined. Precisely, if an \( R \)-strategy \( \alpha \) is initialized, then both \( \Theta_\alpha \) and \( \Xi_\alpha \) are set to be totally undefined, \( D_\alpha \) is set to be the empty set \( \emptyset \), and all links associated with \( \alpha \) are cancelled. If an \( S \)-strategy \( \beta \) is initialized, then both \( g_\beta \) and \( f_\beta \) are set to be totally undefined, parameters \( d(\beta) \) and \( c(\beta) \) are both set to be undefined, and any link associated with \( \beta \) is cancelled. If a \( P \)-strategy \( \gamma \) is initialized, then \( \Delta_\gamma \) is set to be totally undefined, and all witnesses of \( \gamma \) are cancelled.

Notice that an \( S \)-strategy \( \beta \) opens its \( A \)-gap, exactly at stages at which an error between \( f_\beta \) and \( B \) occurs, which gives a \( B \)-permission for its current candidate \( d(\beta) \). Our problem is to make sure that there are infinitely many stages at which all the \( S \)-strategies on the true path (or the current approximation of the true path) drop their \( A \)-restraints simultaneously.

Given a node \( \xi \), suppose that \( \beta_1 \subset \beta_2 \subset \cdots \subset \beta_n \) are all \( S \)-strategies \( \beta \) with \( \beta^\uparrow(g) \subseteq \xi \). To guarantee that if \( \xi \) is a \( P \)-strategy, then there are infinitely many stages at which for all \( i = 1,2,\cdots,n \), the \( A \)-restraints \( r^A(\beta_i) \) of \( \beta_i \) drop to \( -1 \) infinitely often, proceed as follows. Let \( f_i \) be \( f_{\beta_i} \) for all \( i = 1,2,\cdots,n \).

We will arrange the building of various \( f_{\beta_i} \), such that: for any \( s \),
1. for any $i$, if $f_i[s]$ is empty, then the current $A$-restraint $r^A(\beta_i)$ is $-1$, and
2. for any $i < j$, if both $f_i[s]$ and $f_j[s]$ are not empty, then $\text{dom}(f_i) \supseteq \text{dom}(f_j)$,
   where $\text{dom}(f)$ is the domain of $f$.

Our construction will ensure that for any $s$, at the end of stage $s$, the two properties above hold for all nodes $\xi$.

Using these properties, we know that whenever we find an error between $f_n$ and $B$, the same error occurs for $f_i$ for all $i < n$, allowing us to open $A$-gaps for the $S$-strategies $\beta_i$ simultaneously, except for those $\beta_i$'s which are already in their $A$-gaps. Therefore, if $\beta_n$ opens an $A$-gap at stage $s$, then the current stage $s$ is in the $A$-gap of $\beta_i$ for all $i \in \{1, 2, \cdots, n\}$.

We are ready to describe the stage-by-stage construction.

**Definition 5.5.1.** (The Construction) The construction will be defined as follows.

**Stage** $s = 0$. Initialize every node $\xi$, and set $A = \emptyset$.

**Stage** $s = 2n + 1$. Let $b$ be the number that enters $B$ at stage $s$.

Run the following procedure:

1. Let $\beta$ be the $<\text{-minimal}$ and $\subset\text{-maximal}$ $S$-strategy, if it exists, such that $f_\beta(b) \downarrow$.
   Suppose that $\beta_1 \subset \beta_2 \subset \cdots \subset \beta_{n-1}$ are all nodes $\beta'$ such that $\beta' \langle g \rangle \subseteq \beta$. Let $\beta = \beta_n$.

2. Initialize all nodes $\xi$ with $\beta_n \langle g \rangle < L \xi$.

3. In increasing order of $i$, for $\beta_i$, and $\alpha_i = \text{top}(\beta_i)$, if $f_{\beta_i} \neq \emptyset$, then:
   - for every $x \leq d(\beta_i)$, if $g_{\beta_i}(x)$ is undefined, then define $g_{\beta_i}(x) = X_{\alpha_i}(x)$,
   - create a link $(\alpha_i, \beta_i)$,
   - set $r^A(\beta_i) = -1$, and
   - set $f_{\beta_i}$ to be totally undefined.

We say that a $P$-strategy $\delta$ requires attention at stage $s$ if:
   - There is some $x$ such that $\Delta_\delta(B; x) \downarrow \neq K(x)$ and $a_\delta(x) \notin A$;
   - For all $\beta$ with $\beta \langle g \rangle \subset \delta$, $r^A(\beta) = -1$ hold during stage $s$.

4. If there is a $P$-strategy which requires attention at stage $s$, then:
   - let $\gamma$ be the $<\text{-least}$ such $P$-strategy,
let $k$ be the least $x$ such that $\Delta_\gamma(B;x) \downarrow \neq K(x)$ and $a_\gamma(x) \not\in A$,

- enumerate $a_\gamma(k)$ into $A$, and

- initialize all nodes $\xi$ with $\xi > \gamma$, and go to stage $s + 1$.

5. Otherwise, then go to stage $s + 1$.

**Stage** $s = 2n + 2$. We first specify the root node to be eligible to act at substage $t = 0$. At each substage $t$, we allow the strategy which is eligible to act at this substage to take action, and then either close the current stage or specify a new node to be eligible to act at the next substage of stage $s$.

**Substage** $t$. Let $\xi$ be the node which is eligible to act at substage $t$ of stage $s$. If $\xi$ has length $s$, then initialize all nodes $\xi' \not\in \xi$, and close the current stage. Otherwise, there are three cases corresponding to different types of strategy $\xi$.

**Case 1.** $\xi = \gamma$ is a $\mathcal{P}$-strategy. Then run the following:

**Program** $\gamma$: $\gamma$ will build a bounded Turing reduction $\Delta_\gamma$, and define witnesses $a_\gamma(k)$.

For simplicity, we drop the subscription $\gamma$ in the description of the program.

1. If there is an $n$ such that $a(n)$ is defined, and $l(\Psi_\gamma(B), A) \not\exists a(n)$, then let $\gamma^\langle 1 \rangle$ be eligible to act next (i.e. at substage $t + 1$ of stage $s$).

2. Otherwise, let $k$ be the least $x$ such that $\Delta(B;x)$ is undefined. Then:
   - if $a(k) \downarrow$, then define $\Delta(B;k) = K(k)$ with $\delta(k) = \psi(a(k))$,
   - otherwise, then define $a(k)$ to be fresh, and
   - initialize all nodes $\xi > \gamma$, and go to stage $s + 1$.

**Case 2.** $\xi = \alpha$ is an $\mathcal{R}_e$-strategy for some $e$. Run the following

**Program** $\alpha$:

1. If $s$ is not $\alpha$-expansionary, let $\alpha^\langle 1 \rangle$ be eligible to act next.

2. Otherwise, and there is a link $(\alpha, \beta)$ which was created and has never been cancelled or travelled. Let $\beta_0$ be the $<\text{-}\text{least such } \beta$, and let $\beta_0$ be eligible to act at the next substage.

3. Otherwise, then,
   - let $x$ be the least $y$ such that either $\Theta_\alpha(X_\alpha;y)$ or $\Xi_\alpha(B;y)$ is undefined,
• if $\Theta_\alpha(X_\alpha;x) \uparrow$, define $\Theta_\alpha(X_\alpha;x) = D_\alpha(x)$ with $\theta(x) = x$,

• if $\Xi_\alpha(B;x) \uparrow$, then define $\Xi_\alpha(B;x)[s] := D_\alpha(x)[s]$ with use $\xi(x) := \phi(x)$, where $\phi$ is the use of $\Phi(A,B)$, and

• let $\alpha^\prime\langle0\rangle$ be eligible to act next.

Case 3. $\xi = \beta$ is an $S_{e,i}$-strategy for some $e, i$. Let $\alpha = \text{top}(\beta)$. We perform the following,

Program $\beta$:

1. If $\beta$ has already been satisfied, as defined in 2a below, then $\beta^\prime\langle d \rangle$ is eligible to act.

2. (Travel a link $(\alpha, \beta)$) If a link $(\alpha, \beta)$ was created and it has never been cancelled or travelled since it was created, then travel the link $(\alpha, \beta)$ by cases.

Case 2a. (Successful closure) $(\exists x) \ g_\beta(x) \not\downarrow X_\alpha(x)$. (Notice that $g_\beta$ was correct at the stage we created the current link $(\alpha, \beta)$, so this error must occur during the $A$-gap of the $S$-strategy $\beta$.) Then:
   • enumerate $d(\beta)$, the largest confirmed candidate of $\beta$, into $D_\alpha$,
   • we say that $\beta$ is satisfied at stage $s$,
   • initialize all nodes $> \beta$, and go to stage $s + 1$.

Case 2b. (Unsuccessful closure) Otherwise. Then:
   • define $u = \max\{\phi(y) : g(y) \downarrow\} = \phi(d(\beta))$,
   • set $f_\beta \uparrow (\phi(d(\beta)) + 1) = B \uparrow (\phi(d(\beta)) + 1)$,
   • set $r^A(\beta) = u + 1$, and
   • initialize all nodes to the right of $\beta^\prime\langle g \rangle$, and go to stage $s + 1$.

In either case, the link $(\alpha, \beta)$ is removed.

[Remark. We have that if step 2 of program $\beta$ occurs at stage $s$, then $\alpha$ is visited at stage $s$.]

3. (Building $f_\beta$) If $c(\beta) \downarrow c$, $\lambda_\beta(c) \downarrow D_\alpha(c) = 0$, $\Theta_\alpha(X_\alpha;c) \downarrow$, and $\Xi_\alpha(B;c) \downarrow$ then:

Case 3a. $(\exists \beta') \beta'^\prime\langle g \rangle \subset \beta$ and $\text{dom}(f_{\beta'}) \subseteq [0, \xi_\alpha(c)]$. Then:
   • initialize all nodes $> \beta^\prime\langle w \rangle$, and go to stage $s + 1$.

Case 3b. Otherwise, then:
• for any \( x \leq \xi_\alpha(c) \), if \( f_\beta(x) \uparrow \), then define \( f_\beta(x) = B(x) \),
• set \( d(\beta) = c(\beta) \); \( d \) is said to be confirmed,
• cancel \( c(\beta) \), so that \( c(\beta) \uparrow \), and
• let \( \beta^{*}\langle g \rangle \) be eligible to act next.

4. If \( c(\beta) \uparrow \), then define \( c(\beta) \) as fresh, initialize nodes \( \geq \beta^{*}\langle w \rangle \), and go to stage \( s + 1 \).
5. Otherwise, let \( \beta^{*}\langle w \rangle \) be eligible to act at the next substage.

This completes the description of the construction.

5.6 The Verification

In this section, we verify the satisfaction of the requirements. First we investigate some global properties that hold at the end of an arbitrary stage. These properties ensure that the construction is implemented properly.

**Proposition 5.6.1.** Let \( s \) be a stage.

(i) There is at most one link that is travelled during stage \( s \).

(ii) If a link \( (\alpha, \beta) \) is travelled at stage \( s \), then before we travel the link, \( \alpha \) is visited and step 2 of program \( \alpha \) occurs at stage \( s \).

(iii) There are no \( \alpha_1, \alpha_2, \beta_1, \) and \( \beta_2 \) such that \( \alpha_1 \subset \alpha_2 \subset \beta_1 \subset \beta_2 \) and both links \( (\alpha_1, \beta_1) \) and \( (\alpha_2, \beta_2) \) exist at the end of stage \( s \).

**Proof.** It is easy to see that both (i) and (ii) hold by observing the construction.

For (iii), suppose to the contrary that \( s \) is the least stage such that there are \( \alpha_1, \alpha_2, \beta_1, \) and \( \beta_2 \) with \( \alpha_1 \subset \alpha_2 \subset \beta_1 \subset \beta_2 \), and such that both links \( (\alpha_1, \beta_1) \) and \( (\alpha_2, \beta_2) \) exist at the end of stage \( s \). By the minimality of \( s \), exactly one of the two links \( (\alpha_1, \beta_1) \), and \( (\alpha_2, \beta_2) \) is created during stage \( s \). We consider two cases.

**Case 1.** The link \( (\alpha_1, \beta_1) \) is created at stage \( s \).

By the construction, \( s = 2n + 1 \) for some \( n \), and \( f_{\beta_1} \) is set to be totally undefined at stage \( s \). We analyze the location of \( \beta_2 \). If \( \beta_1^{*}\langle w \rangle \subseteq \beta_2 \), then by the construction at stage \( s \), \( \beta_2 \) is initialized during stage \( s \), so the link \( (\alpha_2, \beta_2) \) is removed during stage \( s \), contradicting the choice of \( \beta_2 \). If \( \beta_1^{*}\langle g \rangle \subseteq \beta_2 \), then by the definition of the priority tree \( T \), \( \text{top}(\beta_2) \neq \alpha_2 \), so that there is no link \( (\alpha_2, \beta_2) \) which can be created in the construction. If \( \beta_1^{*}\langle d \rangle \subseteq \beta_2 \),
then the current $d(\beta_2)$ must be defined after $\beta_1$ created its inequality at argument $d(\beta_1)$, after which no link $(\alpha_1, \beta_1)$ can be created, since $\beta_1$ has satisfied its requirement through $\lambda_{\beta_1}(d(\beta_1)) = 0 \neq 1 = D(d(\beta_1))$. So case 1 does not happen.

**Case 2.** The link $(\alpha_2, \beta_2)$ is created at stage $s$.

Let $s_1 < s$ be the stage at which the current link $(\alpha_1, \beta_1)$ was created. By the proof in case 1, we only need to consider the case of $\beta_1^\hat{\langle} w \rangle \subseteq \beta_2$. By the construction at stage $s_1$, $\beta_2$ was initialized during stage $s_1$. So $f_{\beta_2}$ is totally undefined at the end of stage $s_1$. And in fact, all nodes $\geq \beta_1^\hat{\langle} w \rangle$ were initialized at stage $s_1$. Therefore $\beta_2$ cannot be visited at any stage $> s_1$ unless the link $(\alpha_1, \beta_1)[s_1]$ has been removed. This contradicts the choice of $s$.

(iii) holds.

The Proposition follows.

**Proposition 5.6.2.** (i) Let $\beta$ be an $S$-strategy. Then for any $s, t$, if $f_{\beta}$ is totally undefined at substage $t$ of stage $s$, then $r^A(\beta) = -1$ holds at the end of substage $t$ of stage $s$.

(ii) Let $\beta$ be an $S$-strategy, and $s$ be a stage. Let $s^-$ be the greatest stage $t < s$ such that $\beta$ was initialized. Then for any $s^- < s_1 < s_2 < s$, if both $f_{\beta}[s_1]$ and $f_{\beta}[s_2]$ are not empty, then

$$\text{dom}(f_{\beta}[s_1]) \subseteq \text{dom}(f_{\beta}[s_2])$$

(iii) Let $\beta, \beta'$ be $S$-strategies with $\beta^\hat{\langle} g \rangle \subset \beta$. For any $s$, if both $f_{\beta}$ and $f_{\beta'}$ are non-empty at the end of stage $s$, then $\text{dom}(f_{\beta}[s]) \subseteq \text{dom}(f_{\beta'}[s])$.

**Proof.** Both (i) and (ii) are easy facts by observing the construction.

For (iii), we prove the proposition by induction on the stages. Suppose that (iii) holds for all $s' < s$. Consider a stage $s$ at which $f_{\beta}$ is built. By program $\beta$ in the construction, there are two cases.

**Case 1.** A link $(\alpha, \beta)$ for $\alpha = \text{top}(\beta)$ is travelled unsuccessfully at stage $s$. Let $s_1$ be the stage at which the current link $(\alpha, \beta)$ was created. Then $\text{dom}(f_{\beta}[s]) = \text{dom}(f_{\beta}[s_1 - 1])$. 

133
Let $b = \text{dom}(f_\beta[s])$, and let $s_0$ be the first stage at which $f_\beta$ was defined on $[0, b]$.

By program $\beta$, case 3b of program $\beta$ occurred at stage $s_0$. By the construction, there was no link that was travelled by the substage at which $\beta$ was visited during stage $s_0$. Therefore, $\beta'$ was visited at stage $s_0$, and case 3b of program $\beta'$ occurred at stage $s_0$. By the assumption in case 3b of program $\beta$, we have that both $f_{\beta'}$ and $f_\beta$ are non-empty at the end of stage $s_0$, and $\text{dom}(f_{\beta'}[s_0]) \supseteq \text{dom}(f_\beta[s_0])$. By the choice of $s$, $\beta'$ has not been initialized during stages $[s_0, s]$, by (ii) if $f_{\beta'}[s]$ is not empty, then $\text{dom}(f_{\beta'}[s]) \supseteq \text{dom}(f_{\beta'}[s_0])$, (iii) follows in case 1.

**Case 2.** Case 3b of program $\beta$ occurs at stage $s$. As the same as that in case 1, by observing the construction, we have that for any $\xi \subset \beta$, $\xi$ is visited at stage $s$, so that $\beta'$ is visited at stage $s$, and furthermore, case 3b of program $\beta'$ occurs at stage $s$. By the assumption of case 3b of program $\beta$, the domain of $f_{\beta'}$ is larger than that of $f_\beta$ at the end of stage $s$. (iii) follows in case 2.

The proposition follows.

**Definition 5.6.3.** Suppose that $K \not\subseteq_{bT} B$ and $B \not\subseteq_T \emptyset$.

(i) Let $\delta_s$ be the last node which is eligible to act at stage $s$.

(ii) Define the true path $TP \in [T]$ of the construction by $TP = \liminf_s \delta_s$.

Hereafter whenever we consider a node $\xi$ on the true path, we will use the notation $\xi \in TP$ rather than $\xi \subset TP$.

**Proposition 5.6.4.** (Existence of the true path) Suppose $\xi \in TP$. Then there is some $a$ such that $\xi^\langle a \rangle$ is visited infinitely often and initialized only finitely many times. Hence $\xi^\langle a \rangle \in TP$.

**Proof.** We prove by induction on the length of $\xi$. Suppose by induction that the proposition holds for all $\xi' \subset \xi$ and $\xi \in TP$. Let $s_0$ be minimal after which $\xi$ will never be initialized. By the inductive hypothesis, $\xi$ will be visited infinitely often.

We prove the proposition for $\xi$ by cases.

**Case 1.** $\xi = \delta$ is a $P_e$-strategy for some $e$.

By program $\delta$, there are two subcases to consider.

134
Subcase 1a. There are infinitely many $\delta$-expansionary stages.

By the construction, step 2 of program $\delta$ occurs infinitely many times, so that $\Delta_\delta$ is built infinitely many times, and that $\Delta_\delta(B)$ is built as a total function. Since $K \not\leq_T B$, there is some $m$ such that $\Delta_\delta(B;m) \downarrow \neq K(m)$ holds permanently. Let $n$ be the least such $m$, and let $\Delta_\delta(B;n)$ be created at stage $v > s_0$. By the choice of $n$, $\Psi_\delta(B;a_\delta(n))[v] = 0$ and it will hold permanently. Let $u > v$ be the stage at which $n$ enters $K$.

Suppose that $\beta_1 \subset \beta_2 \subset \cdots \subset \beta_l$ are all $S$-strategies $\beta$ with $\beta^*(g) \subseteq \delta$.

By the choice of $s_0$, $f_{\beta_j}$ will never be set to be totally undefined after stage $s_0$ by initialization for any $j$. Therefore, for every $j \in \{1, 2, \cdots, l\}$, $f_{\beta_j}$ is set to be totally undefined after stage $s_0$ only if an error occurs between $f_{\beta_j}$ and $B$.

By inductive hypothesis, case 3b of program $\beta_j$ occurs infinitely many times, so that $f_{\beta_j}$ will be built infinitely many times for all $j \in \{1, 2, \cdots, l\}$.

In particular, $f_{\beta_l}$ will be built infinitely many times. By the assumption of $B \not\leq_T \emptyset$, we can choose a stage $s_1 > u$ at which there is a number $b$ such that $f_{\beta_l}(b) \downarrow 0 \neq 1 = B(b)$ occurs.

By Proposition 5.6.2 (iii), for any $j \in \{1, 2, \cdots, l\}$, if $f_{\beta_j}$ is not empty at the beginning of stage $s_1$, then there is an error between $f_{\beta_j}$ and $B$ that occurs exactly at stage $s_1$. We have that for every $j \in \{1, 2, \cdots, l\}$, if $f_{\beta_j} \neq \emptyset$ at the beginning of stage $s_1$, then $\beta_j$ opens its $A$-gap during stage $s_1$.

By Proposition 5.6.2 (i), for any $j \in \{1, 2, \cdots, l\}$, if $f_{\beta_j}$ is empty at the beginning of stage $s_1$, $r^A(\beta_j) = -1$ holds at both the beginning and the end of stage $s_1$.

By program $\delta$, we have that $\delta$ requires attention at stage $s_1$, and we let $\delta$ receive attention by enumerating its witness $a_\delta(n)$ into $A$.

By the choice of $n$, for any $s \geq s_1$, we have that $\Psi_\delta(B;a_\delta(n)) = 0 \neq 1 = A(a_\delta(n))$ holds during stage $s$, contrary to there being infinitely many $S$-expansionary stages. This case is impossible.

Subcase 1b. Otherwise.
In this case $\Delta_\delta$ is built only finitely many times. Let $s_1 > s_0$ be minimal after which $\Delta_\delta$ will never be built.

By the choice of $s_1$, $\delta^\wedge(1)$ will never be initialized after stage $s_1$, and by program $\delta$, for any $s > s_1$, if $\delta$ is visited at stage $s$, so is $\delta^\wedge(1)$.

The proposition follows in case 1.

Case 2. $\xi = \alpha$ is an $\mathcal{R}$-strategy.

Observing program $\alpha$, we consider two subcases.

Subcase 2a. Step 3 of program $\alpha$ occurs infinitely many times.

Then $\alpha^\wedge(0) \in TP$. By choice of $s_0$, $\alpha^\wedge(0)$ will never be initialized after stage $s_0$.

Furthermore $\alpha^\wedge(0)$ is visited infinitely often. Therefore $\alpha^\wedge(0) \in TP$ and the proposition holds in this case.

Subcase 2b. Otherwise.

Suppose that Step 3 of program $\alpha$ occurs at most finitely many times so that $\alpha^\wedge(0)$ is visited at most a finite number of times. We will show that $\alpha^\wedge(1) \in TP$. To show that $\alpha^\wedge(1)$ is initialized finitely often, first note that by the choice of $s_0$, only nodes $\xi \supset \alpha^\wedge(0)$ can initialize $\alpha^\wedge(1)$.

Let $s_1 > s_0$ be minimal after which step 3 of program $\alpha$ will never occur. Then for any $s > s_1$, if a node $\beta' \supset \alpha^\wedge(0)$ is visited at stage $s$, then there is an $\mathcal{R}$-strategy $\alpha' \subseteq \alpha$, and a link $(\alpha', \beta')$ which is travelled at stage $s$.

Suppose that $\alpha_1 \subset \alpha_2 \subset \cdots \subset \alpha_{n-1}$ are all $\mathcal{R}$-strategies $\alpha'$ with $\alpha'^\wedge(0) \subseteq \alpha$. Let $\alpha_n = \alpha$.

We prove by induction that for each $i \leq n$, there is a stage $v_i$ after which there will be no links $(\alpha_j, \beta_j)$ that can be either created or travelled for all $j \geq i$ and for all $\beta_j \supseteq \alpha^\wedge(0)$.

For $i = n$. Define

$$b_n = \max\{\xi_{\alpha_n}(x)[t] \mid t \leq s_1, \Xi_{\alpha_n}(x)[t] \downarrow\}$$

For every $s > s_1$, define
\[ p_n[s] = \max \{ y \mid f_\beta(y)[s] \downarrow, \top(\beta) = \alpha_n \} \]

By the construction, we have that for every \( s > s_1 \),

- if a link \((\alpha_i, \beta')\) is travelled for some \( i < n \) and some \( \beta' \supseteq \alpha_n \langle 0 \rangle \), then there is no \( \beta \) such that \( f_\beta \) is built during stage \( s \) for any \( \beta \) with \( \top(\beta) = \alpha_n \). In this case, we have \( p_n[s] \leq p_n[s-1] \), and
- if a link \((\alpha_n, \beta_n)\) is travelled at stage \( s \), then \( p_n[s] \leq b_n \).

Therefore in any case we have that for all \( s > s_1 \), \( p_n[s] \leq b_n \). By the construction at odd stages, a link \((\alpha_n, \beta_n)\) can be created at a stage \( s > s_1 \) only if there is an element \( b \leq b_n \) that enters \( B \) at stage \( s \). Since \( b_n \) is a fixed number, \( B \) changes below \( b_n \) only finitely many times. Therefore there are only finitely many stages at which we create links \((\alpha_n, \beta_n)\). Since once a link is travelled, it is removed immediately, there are only finitely many stages at which a link \((\alpha_n, \beta)\) is either created or travelled.

Let \( v_n > s_1 \) be minimal after which there will be no link \((\alpha_n, \beta_n)\) that can be either created or travelled.

Suppose by induction that \( v_{i+1} \) is a minimal stage after which there will be no link \((\alpha_j, \beta_j)\) which is either created or travelled for all \( j \in \{i + 1, i + 2, \ldots, n\} \), and all \( \beta_j \supseteq \alpha^\langle 0 \rangle \).

Define

\[ b_i = \max \{ \xi_{\alpha_i}(x)[t] \mid \Xi_{\alpha_i}(x) \downarrow, \ t \leq v_{i+1} \} \]

For any \( s > v_{i+1} \), define

\[ p_i[s] = \max \{ y \mid f_\beta(y)[s] \downarrow, \beta \supseteq \alpha^\langle 0 \rangle, \top(\beta) = \alpha_i \} \]

By the construction, it is easy to see from an inductive argument that for all \( s > v_{i+1} \),

- if a link \((\alpha_j, \beta_j)\) is travelled for some \( j < i \), and some \( \beta_j \supseteq \alpha^\langle 0 \rangle \), then \( p_i[s] \leq p_i[s-1] \); and
• If a link \((\alpha_i, \beta_i)\) for some \(\beta_i \supseteq \alpha^\langle 0 \rangle\) is travelled at stage \(s\), then \(p_i[s] \leq b_i\).

This shows that for all \(s > v_{i+1}\), \(p_i[s] \leq b_i\). By the construction, if a link \((\alpha_i, \beta_i)\) for some \(\beta_i \supseteq \alpha^\langle 0 \rangle\) is created at a stage \(s > v_{i+1}\), then there is a number \(b \leq b_i\) which enters \(B\) at stage \(s\). Since \(b_i\) is a fixed number, the creation of links \((\alpha_i, \beta_i)\) for \(\beta_i \supseteq \alpha^\langle 0 \rangle\) occurs only finitely many times, so that there is a stage \(v_i > v_{i+1}\) say, after which there will be no link of the form \((\alpha_i, \beta_i)\) for any \(\beta_i \supseteq \alpha^\langle 0 \rangle\) which can be either created or travelled.

Therefore there is a stage \(v_1\) say after which no link \((\alpha_i, \beta)\) can be created or travelled for any \(i \leq n\) and any \(\beta \supseteq \alpha^\langle 0 \rangle\). So there are only finitely many stages at which some node \(\xi \supseteq \alpha^\langle 0 \rangle\) is visited.

Thus \(\alpha^\langle 1 \rangle\) is initialized only finitely many times.

By the proof above, there are only finitely many stages at which either Step 2 or Step 3 of program \(\alpha\) occurs. \(\alpha^\langle 1 \rangle\) is visited at almost every stage at which \(\alpha\) is visited.

Therefore in Subcase 2b, we have that \(\alpha^\langle 1 \rangle\) is initialized only finitely many times and visited infinitely often.

**Case 3.** \(\xi = \beta\) is an \(S\)-strategy.

Let \(\alpha = \top(\beta)\).

**Subcase 3a.** A link \((\alpha, \beta)\) is travelled once and successfully closed.

Then \(\beta^\langle d \rangle\) is visited infinitely often and only initialized finitely many times. So \(\beta^\langle d \rangle \in TP\).

**Subcase 3b.** \((\alpha, \beta)\) is travelled infinitely often and unsuccessfully closed \(\beta\)'s \(A\)-gap.

As in Subcase 3a, \(\beta^\langle g \rangle \in TP\).

**Subcase 3c.** Otherwise.

By the assumption of this case, \(f_\beta\) is built only finitely many times, since if this is not true, then \(f_\beta\) is built as a computable function, and \(f_\beta = B\), contradicting the hypothesis \(B \not\subseteq T \emptyset\).

By program \(\beta\), \(\lim_s c(\beta)[s] \downarrow c(\beta) < \omega\). By the choice of \(c(\beta)\), \(c(\beta) \not\subseteq D_\alpha\). Let \(s_1\) be the stage after which neither of the step 1, 2, 3, or 4 of program \(\beta\) occurs, therefore, for any \(s > s_1\), if \(\beta\) is visited at stage \(s\), so is \(\beta^\langle w \rangle\).
Therefore $\beta^*(w)$ is initialized only finitely many times, and visited infinitely often, $\beta^*(w) \in TP$.

The proposition follows in Case 3. □

Since the true path exists only if both $K \not\subseteq_{bT} B$, and $B \not\subseteq T \emptyset$ hold as proved in Proposition 5.6.4, we always assume the two conditions from now on.

**Proposition 5.6.5** (Possible outcomes along $TP$). Given $\xi \in TP$:

(i) If $\xi = \delta$ is a $P_e$-strategy for some $e$, then $\delta^*(1) \in TP$ and $A \neq \Psi_e(B)$.

(ii) If $\xi = \alpha$ is a $R$-strategy, then
   
   (a) if $\alpha^*(0) \in TP$, then $D_\alpha = \Theta_\alpha(X) = \Xi_\alpha(B)$;
   
   (b) if $\alpha^*(1) \in TP$, then $\Phi_\alpha$ is partial or $\Phi_\alpha(A,B) \neq X_\alpha$.

(iii) If $\xi = \beta$ is an $S$-strategy, then for $\alpha = \text{top}(\beta)$, we have:
   
   (a) if $\beta^*(d) \in TP$, then $\lim_s d(\beta)[s] \downarrow d(\beta) < \omega$ and $\lambda_\beta(d(\beta)) = 0 \neq 1 = D_\alpha(d(\beta))$.
   
   (b) if $\beta^*(g) \in TP$, then $g_\beta$ is a computable function and $g_\beta = X_\alpha$.
   
   (c) if $\beta^*(w) \in TP$, then $\lim_s c(\beta)[s] \downarrow c(\beta) < \omega$ and $\lambda_\beta(c(\beta)) \neq 0 = D_\alpha(c(\beta))$.

**Proof.** By Proposition 5.6.4, we can choose $s_0$ minimal after which $\xi \in TP$ will never be initialized.

For (i). By Proposition 5.6.4, $\delta^*(1) \in TP$. Suppose to the contrary that $A = \Psi_e(B)$.

By program $\delta$, step 2 of program $\delta$ occurs infinitely many times. Therefore, $\Delta_\delta(B)$ is total. Since $K \not\subseteq_{bT} B$, we can choose the least $n$ such that a permanent inequality $\Delta_\delta(B;n) = 0 \neq 1 = K(n)$ appears. Let $v > s_0$ be the stage at which the computation $\Delta_\delta(B;n)$ was created. Notice that $\Psi_e(B;a_\delta(n))[v] = 0$ and $B$ will never change below $\psi_e(a_\delta(n))$ after stage $v$.

By the proof in case 1 of Proposition 5.6.4, there is a stage $s_1$ at which we can enumerate $a_\delta(n)$ into $A$.

By the choice of $n$ and $s_1$, $\Psi_e(B;a_\delta(n)) = 0 \neq 1 = A(a_\delta(n))$ will be preserved forever. A contradiction.

Therefore we have that $A \neq \Psi_e(B)$.

For (ii). Let $\alpha^*(0) \in TP$. 

139
By Proposition 5.6.3, both $\Theta_\alpha$ and $\Xi_\alpha$ are built infinitely many times.

By Step 3 of program $\alpha$, it suffices to prove that the following constraints of $\alpha$ are satisfied during the course of the construction:

For any $d$, and any $s > s_0$, if $d$ is enumerated into $D_\alpha$ at stage $s$, then both $\Theta_\alpha(X; d)$ and $\Xi_\alpha(B; d)$ are undefined during stage $s$.

By Proposition 5.6.1 (i), there is at most one link which is travelled during stage $s$.

Let $d$ be enumerated into $D_\alpha$ at stage $s$. Then there is an $S$-strategy $\beta$ such that $\top(\beta) = \alpha$ and case 2a of program $\beta$ occurs at stage $s$, and $d = d(\beta)[s]$.

By Proposition 5.6.1 (ii), $\alpha$ is visited at stage $s$, and there is a link $(\alpha, \beta)$ which was created at a stage $s^-(> s_0) < s$ and travelled at stage $s$ successfully. By the construction at stage $s^-$, there was an error between $f_\beta$ and $B$ occurred at stage $s^-$, since $d = d(\beta)[s] = d(\beta)[s^-] = \dom(f_\beta[s^-])$, we have that $\Xi_\alpha(B; d(\beta))$ becomes undefined during stage $s^-$. By the link $(\alpha, \beta)[s^-]$, step 3 of program $\alpha$ has never occurred during stages $[s^-, s]$. Therefore, $\Xi_\alpha(B; d)$ is undefined during and after stage $s$.

By the assumption of case 2a of program $\beta$ at stage $s$, there has been an error between $g_\beta$ and $X_\alpha$ during stages $[s^-, s]$, therefore, $\Theta_\alpha(X_\alpha; d)$ has become undefined during stage $s$.

Therefore, for any number $d$ chosen after stage $s_0$, the enumeration of $d$ into $D_\alpha$ does always respect the constraints imposed by $\alpha$. Both $\Theta_\alpha(X_\alpha) = D_\alpha$ and $\Xi_\alpha(B) = D_\alpha$ are satisfied.

Let $\alpha^*(1) \in TP$. We prove that $X_e \neq \Phi_e(A, B)$. Suppose to the contrary that $X_e = \Phi_e(A, B)$. By the assumption of this case, step 3 of program $\alpha$ occurs only finitely many times, and that there are infinitely many $\alpha$-expansionary stages. Therefore, there is a stage $s_1 > s_0$ after which step 3 of program $\alpha$ will never occur. However there are infinitely many stages at which we travel a link $(\alpha, \beta)$ for some $\beta$.

By the choice of $s_1$, $\Xi_\alpha$ is a finite set, let $s_2$ be the stage $> s_1$ after which $B$ will never change below $\max\{\xi_\alpha(x)[s_1] \mid \Xi_\alpha(B; x)[s_1] \downarrow\}$. 

140
By the $\mathcal{S}$-strategies, for any $\beta$ with $\text{top}(\beta) = \alpha$, if $f_\beta$ is created after stage $s_2$, then there will be no link $(\alpha, \beta)$ which can be created by using the difference between $B$ and $f_\beta$.

Therefore we can choose a stage $s_3 > s_2$ after which there is no link $(\alpha, \beta)$ which can be created for any $\beta$. By the construction, once we travel a link $(\alpha, \beta)$, it is removed. There is a stage $s_4 > s_3$ after which there is no link from $\alpha$ to any $\mathcal{S}$-strategy $\beta$ which can be either created or travelled. This contradicts the assumption that step 2 of program $\alpha$ occurs infinitely many times.

We have that $X_e \neq \Phi_e(A, B)$.

For (iii)(a). If $\beta^\prec(d) \in TP$ then there is some stage $s$ where Case 2a of program $\beta$ enumerates $d(\beta) = \lim_s d(\beta)[s]$ into $D_\alpha$. Then at all stages $t \geq s$, program $\beta$ enacts Case 1. Furthermore since $d(\beta)$ is only defined when $\lambda(d(\beta)) = 0$, we have that $\lambda_\beta(d(\beta)) = 0 \neq 1 = D_\alpha(d(\beta))$.

For (iii)(b). By the choice of $s_0$, and by the assumption of this case, $\beta^\prec(g)$ is never initialized after stage $s_0$.

By program $\beta$, case 3b of program $\beta$ occurs infinitely many times. By the choice of $s_0$, $f_\beta$ becomes totally undefined at any stage $s > s_0$ only if a link $(\alpha, \beta)$ is created at stage $s$, and this link will certainly be travelled unsuccessfully, instead of being initialized.

For a fixed number $d$, $\phi_\alpha(d)$ is a fined number, so that $B$ changes below $\phi_\alpha(d)$ only finitely many times. Therefore $d(\beta)[s]$ will be unbounded in the construction. By the construction at odd stages, if a link $(\alpha, \beta)$ is created at stage $s$, then $d(\beta)[s]$ is defined, and $g_\beta$ is built on the initial segment $d(\beta)[s]$. Therefore $g_\beta$ is built as a computable function.

Notice that $g_\beta$ will never be set totally undefined after stage $s_0$.

We prove that for any $x$, if $g_\beta(x)$ is created at a stage $v > s_0$, then for any $s \geq v$,

$$g_\beta(x) = X_\alpha[s](x).$$

Given an $x$, let $v > s_0$ be the stage at which $g_\beta(x)$ is created and defined as 0 (if it is 1, then $g_\beta(x) = X_\alpha(x)$ takes already the permanent value).

Suppose that $v_0 < v_1 < v_2 < \cdots$ are all stages $\geq v$ at which a link $(\alpha, \beta)$ is created, and let $t_i$ be the stage at which the link $(\alpha, \beta)[v_i]$ is travelled. Then $v_0 = v$. 

141
By the choice of $t_0$, the link $(\alpha, \beta)[v_0]$ is unsuccessfully travelled at stage $t_0$, this means that for any $t \in [v_0, t_0]$, $g_{\beta}(x) = X_{\alpha}[t](x)$. Since $t_0$ is $\alpha$-expansionary, we have $g_{\beta}(x) = X_{\alpha}[t_0](x) = \Phi_{\alpha}(A, B; x)[t_0]$. By the $A$-restraint $r_A(\beta)[t_0]$, the definition of $f_{\beta}[t_0]$, and by the convention of $\Phi_{\alpha}$, we have that for any $s \in [t_0, v_1)$, $g_{\beta}(x) = \Phi_{\alpha}(A, B; x)[s] = X_{\alpha}[s](x)$ is preserved.

Suppose by inductive hypothesis we have:

1. for any $s \in [v_n, t_n]$, $g_{\beta}(x) = X_{\alpha}[s](x)$.
2. $g_{\beta}(x) = X_{\alpha}[t_n](x) = \Phi_{\alpha}(A, B; x)[t_n]$.
3. for any $s \in [t_n, v_{n+1})$, $g_{\beta}(x) = \Phi_{\alpha}(A, B; x)[s] = X_{\alpha}[s](x)$.
4. $g_{\beta}(y) = X_{\alpha}[v_{n+1}]$.

Since the link $(\alpha, \beta)[v_{n+1}]$ is unsuccessfully travelled at stage $t_{n+1}$, (1) holds for $n + 1$, and since $t_{n+1}$ is $\alpha$-expansionary, (2) for $(n + 1)$ holds, and furthermore, by the $A$-restraint at stage $t_{n+1}$, (3) for $(n + 1)$ holds, (4) holds since $\alpha$ will never be visited at odd stages, so there are no elements which enter $X_{\alpha}$ at odd stages.

This proves that $g_{\beta}(x) = X_{\alpha}(x)$.

Since $x$ is arbitrarily given, we have that for almost every $x$, $g_{\beta}(x) = X_{\alpha}(x)$, $X_{\alpha}$ is computable.

(iii)(b) follows.

For (iii)(c). By the assumption in this case, $g_{\beta}$ is built only finitely many times. If $f_{\beta}$ is built infinitely many times, then the final version of $f_{\beta}$, denoted by $f$, is built as a computable function, and $f = B$. A contradiction. Therefore $f_{\beta}$ is built only finitely many times. Let $s_1 > s_0$ be such that $f_{\beta}$ will never be built at any stage $s > s_1$. Furthermore, we can choose a stage $s_2 > s_1$ after which none of the steps 1, 2, 3, or 4 of program $\beta$ will occur. Therefore $\lim_{s} c(\beta)[s] = c(\beta)$ must be chosen before stage $s_2$. Clearly $c(\beta) \not\in D_{\alpha}$.

Since $\beta$ is visited infinitely many times, the only reason that case 3b will never occur after stage $s_2$ is that $\lambda_{\beta}(c(\beta)) \neq 0$. (iii)(c) follows.

Proposition 5.6.6 ($P$-satisfaction Proposition). For each $e$, $P_e$ is satisfied.
Proof. Given $e$, let $\delta$ be the $P_e$-strategy $\xi \in TP$. By Proposition 5.6.5 (i), necessarily $A \neq \Psi_e(B)$ so that $P_e$ is satisfied. The proposition follows.

Proposition 5.6.7. For each $e \in \omega$, if $X_e = \Phi_e(A, B)$ and $X_e$ and $B$ are not computable, then they do not form a minimal pair.

Proof. By Proposition 5.4.5, let $\alpha$ be the longest $R_e$-strategy on the true path $TP$. By Proposition 5.6.5 (ii), $\alpha^*(0) \in TP$ and $D_\alpha = \Theta_\alpha(X_\alpha) = \Xi_\alpha(B)$. By Proposition 5.6.5 (iii), for any $S$-strategy $\beta$, if $\text{top}(\beta) = \alpha$ and $\beta \in TP$ then either $\beta^*(w) \in TP$ or $\beta^*(d) \in TP$. In either case $\lambda_\beta \neq D_\alpha$ so that $X_\alpha$ and $B$ do not form a minimal pair. The proposition follows.
REFERENCES


Katie Brodhead was born Paul Katie Brodhead in Oak Park, IL in 1980. As a Chancellor’s scholar, Katie attended the University of Wisconsin-Madison from 1997 until 2000, when she earned a B.S. in mathematics. During the summer of 2000, Katie participated in the NSF-funded undergraduate mathematics research experience SIMU, at the University of Puerto Rico-Humacao. In 2003 Katie started graduate school at the University of Florida and earned a master’s degree in mathematics in 2005. During the summer of 2006, Katie was a graduate assistant for an NSF SEAGEP-funded undergraduate mathematics research experience at the University of the Virgin Islands. She went to the Chinese Academy of Sciences, Institute of Software during the summer of 2007 as an NSF fellow, participating in the East Asia and Pacific Summer Institutes. As an NSF SEAGEP fellow during the fall of 2007, Katie went to Victoria University of Wellington (Wellington, New Zealand). As an invited visitor and lecturer, Katie went to the University of Hawaii at Manoa in the spring of 2008. Katie earned her Ph.D. in mathematics from the University of Florida in 2008. In homage to heritage, her thesis and the immediate publications spawning therefrom are published under her first and last birth name.

*Corrigendum to Biographical Sketch, October 2015.*