TO BRANCH OR NOT TO BRANCH
Branching and non-branching in the Medvedev lattice of Π^0_1 classes

By

Christopher P. Alfeld

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

(MATHEMATICS)

at the

UNIVERSITY OF WISCONSIN – MADISON

2007

Abstract

This thesis analyzes the structure of the Medvedev lattice of non-empty Π^0_1 classes in 2^ω from the viewpoint of branching and non-branching degrees. This lattice is a countable distributive lattice with least and greatest element, which describes the relative information content of certain subsets of 2^ω.

Chapter 1 is an introduction, providing background history, notation, and an overview of necessary concepts.

Chapter 2 is essentially my paper "Non-Branching Degrees in the Medvedev Lattice of Π^0_1 classes."[1]. The chapter adds an additional theorem which strengthens the theorem on inseparable and not hyperinseparable classes. The chapter is also slightly more verbose.

We begin by taking an existing condition, homogeneous, which implies non-branching and define two successively weaker conditions, hyperinseparable and inseparable. We then demonstrate that inseparable is equivalent to non-branching and is invariant under Medvedev equivalence. Finally, we prove separation theorems, namely the existence of an inseparable and not hyperinseparable degree and the existence of a hyperinseparable and not homogeneous degree.

Chapter 3 defines a combinatorial method for constructing Π^0_1 classes by priority arguments. This section does not contain any difficult proofs but abstracts many of the common elements of such constructions. The definitions and results are used in Chapter 4.

Chapter 4 follows a program similar to Chapter 2. Specifically, we take an existing condition which we call totally separable, show that this implies branching, and then define two successively weaker conditions, hyperseparable and separable. Separable is the converse of inseparable and thus equivalent to branching and also invariant under Medvedev equivalence. We then prove the separation of separable and hyperseparable and some structural results on separable and not hyperseparable degrees. In the last portion, we prove the separation of hyperseparable and totally separable, i.e., the existence of a hyperseparable and not totally separable degree.

Acknowledgments

I thank my advisor, Steffen Lempp, for his guidance, unwavering support, and enthusiasm. Without him, this thesis would not exist. I also thank the other members of the logic group at UW-Madison for creating a dynamic and supportive environment. In particular, Asher Kach has been a close friend and colleague and has put forth tremendous amounts of time and effort in support of the logic program.

I thank Douglas Cenzer at University of Florida and Stephen Simpson at Penn State for their conversations and encouragement. Douglas Cenzer is a fantastic source for all things relating to Π^0_1 classes. Stephen Simpson is a major source of results and information relating to the Medvedev and Muchnik lattices.

I thank Paul Barford and the WAIL group for supporting me through a research assistantship and providing a fascinating and invigorating environment.

I thank all of my many friends who have made Madison feel like home. I thank the local and regional international folk dance community who have become a highly significant and valued part of my life. And I thank all the people at Planned Parenthood who I have volunteered with for providing an enjoyable and rewarding experience and for their friendship.

List of Figures

Contents

Chapter 1

Introduction

1.1 Introduction

A Π^0_1 class has many equivalent definitions. For the purposes of this thesis we will view a Π^0_1 class as the set of infinite paths through a computable tree. A good introduction and background on Π^0_1 classes can be found in Cenzer and Jockusch [4]. These classes appear frequently in computable mathematics. A survey of their uses can be found in Cenzer and Remmel [5]. See also Jockusch and Soare [6].

We will be concerned primarily with the Medvedev lattice of non-empty Π^0_1 subsets of 2^ω. In this context we say that a Π^0_1 class P is Medvedev below Q, written $P \leq_M Q$, if there is a computably continuous functional from Q into P, i.e., if it is possible to uniformly compute an element of P given an element of Q. A common intuition is to view a class as the solution set to some mathematical problem and $P \leq_M Q$ as stating that solving Q is sufficient to solve P. The Medvedev reduction forms a distributive lattice.[1] We will denote this lattice by $\mathcal{L}_M$. This lattice has recently been studied by Simpson [9], Binns [2], and Cenzer and Hinman [3]. In particular, Binns [2] demonstrates a dense splitting theorem, thus showing that there are no non-trivial non-splitting degrees. The main purpose of this thesis is to investigate the dual, branching and non-branching

[1] This lattice is also referred to in the literature as the *strong* lattice of Π^0_1 classes.

degrees.

It is an elementary exercise to show that the bottom degree of $\mathcal{L}_M$ does not branch. In [3], Cenzer and Hinman further demonstrated that the *homogeneous* degrees—those of splitting classes of computably enumerable (c.e.) sets—do not branch.

As a convention we say that a degree has some property if a member has that property. A clopen set C is *good* for a Π^0_1 class P if $P \cap C$ and $P \cap C^c$ are non-empty, i.e., if C splits P into two proper clopen subclasses.

We call a class P *inseparable* if, for every C good for P, $P \cap C \leq_M P \cap C^c$ or $P \cap C^c \leq_M P \cap C$. Thus inseparability states that there is no splitting of P into incomparable clopen subclasses. We will show that inseparability is an invariant of an $\mathcal{L}_M$-degree, and that inseparability is equivalent to being non-branching.

A natural strengthening of inseparability is to change the "or" to an "and". We call a class P *hyperinseparable* if, for every C good for P, $P \cap C \equiv_M P \cap C^c$. As $P \cap C \geq_M P$ for all C, it is equivalent to require $P \cap C \equiv_M P$. Thus hyperinseparability states that all nonempty clopen subclasses of P are equivalent to P. It is clear that hyperinseparability implies inseparability and thus being non-branching.

A result of Cenzer and Hinman [3] shows that homogeneity implies hyperinseparability. We will show:

$$\text{Homogeneous} \Rightarrow \text{Hyperinseparable} \Rightarrow \text{Inseparable} \Leftrightarrow \text{Non-Branching}.$$

We will also show that no additional implications hold, i.e., that these are distinct classes of non-branching degrees. Along the way we will show downward density: below any degree and above 0_M there is a non-zero degree which is inseparable and not hyperinseparable.

Conversely, a class P is *separable* if, there exists a clopen set C good for P, such that $P \cap C \perp_M P \cap C^c$. Separability is likewise an invariant of a $\mathcal{L}_M$-degree, and equivalent to being branching.

We then strengthen separability. We call a class P *hyperseparable* if, for every C good for P, $P \cap C \perp_M P \cap C^c$. It is clear that hyperseparability implies separability and thus being branching.

Similar to homogeneous and hyperinseparable, there is a preexisting notion stronger than hyperseparable. It does not have a name in the literature. Call a class P *totally separable* if, for all $X, Y \in P$, $X \perp_T Y$. In [7], Jockusch and Soare show the existence of a totally separable class. It is immediate that totally separable implies hyperseparable.

We will show:

$$\text{Totally separable} \Rightarrow \text{Hyperseparable} \Rightarrow \text{Separable} \Leftrightarrow \text{Branching.}$$

We will also show that no additional implications hold, i.e, that these are distinct class of branching degrees. We will also show some results about the occurrence of separable and not hyperseparable degrees.

The remainder of this chapter will provide definitions, conventions, and some basic results. Chapter 2 will describe and provide results on non-branching degrees. Chapter 3 will present some basic combinatorial definitions and results that are generally useful for constructing Π^0_1 class with priority arguments. Chapter 4 describe and provide results on branching degrees.

1.2 Definitions, Conventions, and Basic Theory

The notation used in this thesis generally conforms to that found in Cenzer and Hinman [3]. For an overview of the concepts and the theory of computability theory, see Rogers [8] or Soare [12].

Given a string $\sigma \in 2^{<\omega}$, we denote the length of σ by $|\sigma|$. The initial substring relation is denoted by $\prec$. Concatenation of two strings is written as $\sigma^\frown\tau$. The empty string is denoted by $\emptyset$, the string of a single 1 by 1, and of a single 0 by 0. Truncation to the first n coordinates is denoted by $\sigma \upharpoonright n$. For $X \in 2^\omega$ we say $\sigma \prec X$ if σ is an initial segment of X.

A tree, $\mathbb{T}$, is a subset of $2^{<\omega}$ closed downward under $\prec$. The set of infinite paths through $\mathbb{T}$ is denoted by $[\mathbb{T}]$ and the set of extendible members, those which are initial substrings of members of $[\mathbb{T}]$, is denoted by $\mathrm{Ext}(\mathbb{T})$. For $\sigma \in 2^{<\omega}$, define $\sigma^\frown\mathbb{T} = \{\sigma^\frown\tau : \tau \in \mathbb{T}\}$.

A Π^0_1 class P is a non-empty subset of 2^ω such that there exists a computable tree $\mathbb{P}$ with $P = [\mathbb{P}]$.[2] We denote the tree of initial substrings of members by $\mathcal{T}_P$. Note that $[\mathcal{T}_P] = P$, $\mathcal{T}_P$ is uniquely determined by P, and $\mathcal{T}_P$ is the set of extendible members of any computable tree generating P. Similar to the above, define $\sigma^\frown P = [\sigma^\frown\mathcal{T}_P]$.

Define $I(\sigma) = \{X \in 2^\omega : \sigma \prec X\}$. In 2^ω, any clopen subset will be a finite union of such cones. A clopen set C will be *good* for a class P if $P \cap C \neq \emptyset \neq P \cap C^c$. By abuse of notation we say that $\sigma \in C$ if it is an initial substring of some $X \in C$. Similarly, if $\mathbb{T}$ is a tree, then $\mathbb{T} \cap C = \{\sigma \in \mathbb{T} : \sigma \in C\}$.

A central concept in the study of the Medvedev lattice is that of a computably

[2]This definition is for the purposes of this thesis. In full generality, a Π^0_1 class is a (possibly empty) subset of ω^ω. There are also several equivalent alternatives to the computable tree definition. See [4].

continuous functional. Consider the following definition.[3]

Definition 1.2.1. *A partial computable function* $\phi\colon 2^{<\omega} \to 2^{<\omega}$ *is a* tree map *if it satisfies the following two properties:*

$$\forall \sigma, \tau \in \mathrm{dom}(\phi)\big(\sigma \preceq \tau \Rightarrow \phi(\sigma) \preceq \phi(\tau)\big), \tag{1.1}$$

$$\forall X \in [\mathrm{dom}(\phi)] \forall n \exists m \big(|\phi(X \restriction m)| > n\big). \tag{1.2}$$

A computably continuous functional *is a function* $\Phi\colon 2^{\omega} \to 2^{\omega}$ *such that there exists a tree map* ϕ *with* $\Phi(X) = \bigcup_n \phi(X \restriction n)$.

The following lemma states that we can assume Φ, and thus ϕ, to be total. For a proof, see [3].

Lemma 1.2.2. *Let* P *and* Q *be* Π^0_1 *classes such that* $P \leq_M Q$*. Then there exists a total computable functional* $\Phi : 2^{\omega} \to 2^{\omega}$ *such that* $\Phi(Q) \subseteq P$*.*

Thus when we say $\Phi\colon Q \to P$ we mean that there is a total tree map ϕ with $\phi(\mathcal{T}_Q) \subseteq \mathcal{T}_P$.

For Π^0_1 classes P and Q say that Q is Medvedev above P, denoted $P \leq_M Q$, if there exists a computably continuous functional $\Phi\colon Q \to P$.

By applying $\leq_M$ to the non-empty Π^0_1 classes in 2^{ω} we obtain a degree structure which we denote by $\mathcal{L}_M$. We denote the bottom degree by $\mathbf{0}$ and the top degree by $\mathbf{1}$. It is useful to observe that $\mathbf{0}$ is the degree containing exactly all classes with a computable member. An example of the top degree is the separating class of $\{e : \phi_e(e) \downarrow = 0\}$ and $\{e : \phi_e(e) \downarrow = 1\}$. The symbols $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ will denote degrees in $\mathcal{L}_M$. For a Π^0_1 class P,

[3] There are alternative, equivalent, definitions. A common one is to define strings as partial computable functions and $\Phi(X)(n) = \phi^X(n)$.

the degree of P (under $\leq_M$) will be denoted by $\deg(P)$. Note that 2^ω is a Π^0_1 class in $\mathbf{0}$. For clarity we will denote 2^ω by 0_M when using it as a canonical member of $\mathbf{0}$.

For Π^0_1 classes P and Q say that Q is Muchnik above P, denoted $P \leq_w Q$, if for every $X \in Q$ there exists a computable Φ such that $\Phi(X) \in P$. In a similar manner to the above, we obtain a degree structure of Muchnik degrees, which we denote by $\mathcal{L}_m$. The Muchnik lattice is only briefly mentioned in this thesis, so will will not develop a full notation for it. For more information on the Muchnik lattice of Π^0_1 classes see [11].

An immediate but much used lemma is the following.

Lemma 1.2.3. *Let Q and P be Π^0_1 classes with $Q \subseteq P$. Then $Q \geq_M P$.*

Proof. The identity function serves as a witness. □

The following result from the theory of Π^0_1 classes will be important.

Lemma 1.2.4. *If $\mathbb{P}$ is a co-c.e. tree, then there exists a computable tree $\mathbb{Q}$, such that $[\mathbb{P}] = [\mathbb{Q}]$. Furthermore, we can effectively find $\mathbb{Q}$ from $\mathbb{P}$.*

Proof Sketch. Let $\{A_s\}_{s\in\omega}$ be an enumeration of $2^{<\omega} \setminus \mathbb{P}$ and $\mathbb{Q} = \{\sigma \colon \forall \tau \preceq \sigma (\tau \notin A_{|\sigma|})\}$. □

Definition 1.2.5. *Define P_e to be $[\mathbb{T}_e]$ where $\mathbb{T}_e$ is the eth co-c.e. tree.*

Chapter 2

Non-Branching Degrees

Most of the content of this chapter appears in [1]. This chapter is slightly more verbose and adds Theorem 2.4.4.

2.1 Inseparable degrees

Before we define inseparability consider the following tree characterization of meets.

Definition 2.1.1. *Given trees* $\mathbb{S}$ *and* $\mathbb{T}$ *define the* tree meet *by*

$$\mathbb{S} \wedge \mathbb{T} = (0^\frown \mathbb{S}) \cup (1^\frown \mathbb{T}).$$

Lemma 2.1.2. *If* $P \in \boldsymbol{a}$ *and* $Q \in \boldsymbol{b}$*, then* $[\mathcal{T}_P \wedge \mathcal{T}_Q] \in \boldsymbol{a} \wedge \boldsymbol{b}$.

See [3] for a proof and related results. For Π^0_1 classes P and Q we define $P \wedge Q = [\mathcal{T}_P \wedge \mathcal{T}_Q]$.

With this in hand we see that meets result in incomparable trees connected together into a single tree. Under Medvedev equivalence the actual connection might drift but we can hope it is preserved in some form. Thus to avoid being a proper meet we avoid having incomparable subtrees.

Definition 2.1.3. *A* Π^0_1 *class* P *is* inseparable *if for all clopen sets* C *good for* P*, either* $P \cap C \leq_M P \cap C^c$ *or* $P \cap C \geq_M P \cap C^c$.

To show a degree is non-branching we strive to show that all of its members are inseparable, i.e., do not resemble tree meets. Throughout this thesis we will repeatedly be attempting to show properties of every member of a degree. The general technique will be to show that some property of a single member ensures a (possibly weaker) property of every member. In the case of inseparability we have the strongest possible result, namely:

Theorem 2.1.4. *Let Q be an inseparable Π^0_1 class and $P \equiv_M Q$. Then P is inseparable.*

Proof. Fix $\Phi : P \to Q$ and $\Psi : Q \to P$ as computable functionals witnessing $P \equiv_M Q$. Fix C, a clopen set good for P, and let $D = \Psi^{-1}(C)$. As Ψ is continuous, D is clopen. Let i be the identity functional. If $Q \cap D = \emptyset$, then $\Psi(Q) \subseteq P \cap C^c$ and $\mathbb{P} \cap C \xrightarrow{i} \mathbb{P} \xrightarrow{\Phi} \mathbb{Q} \xrightarrow{\Psi} \mathbb{P} \cap C^c$ witnesses $P \cap C^c \leq_M P \cap C$. Symmetrically, if $Q \cap D^c = \emptyset$, then $P \cap C \leq_M P \cap C^c$. If D is good for Q, then, without loss of generality, $Q \cap D \leq_M Q \cap D^c$. Let $\Omega\colon Q \cap D^c \to Q \cap D$ witness this reduction. Note that the predicate $X \in D$ is computable. We can define the computable functional as illustrated in Figure 1 and Figure 2, namely

$$\Theta(X) = \begin{cases} \Psi\big(\Omega\big(\Phi(X)\big)\big) & \text{if } \Phi(X) \in D^c, \\ \Psi\big(\Phi(X)\big) & \text{if } \Phi(X) \in D. \end{cases} \tag{2.1}$$

witnessing $P \cap C \leq_M P \cap C^c$. Thus P is inseparable. □

Corollary 2.1.5. *A degree $\boldsymbol{a} \in \mathcal{L}_M$ is inseparable iff $\boldsymbol{a}$ is non-branching.*

Proof. We prove the contrapositive for both directions. Assume $\mathbf{a}$ is branching and let $\mathbf{b}, \mathbf{c} \in \mathcal{L}_M$ be such that $\mathbf{b} \perp \mathbf{c}$ and $\mathbf{a} = \mathbf{b} \wedge \mathbf{c}$. Fix $Q \in \mathbf{b}$ and $R \in \mathbf{c}$. Then $Q \wedge R \in \mathbf{a}$ and $I(0)$ witnesses that $Q \wedge R$ is not inseparable. By Theorem 2.1.4, no member of $\mathbf{a}$ is

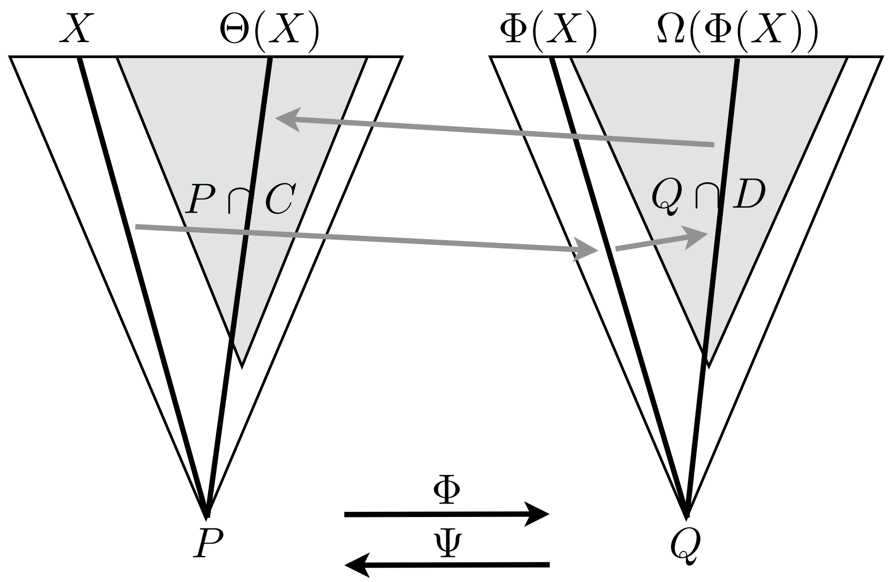

Figure 1: Theorem 2.1.4: $\Phi(X) \in D^c$

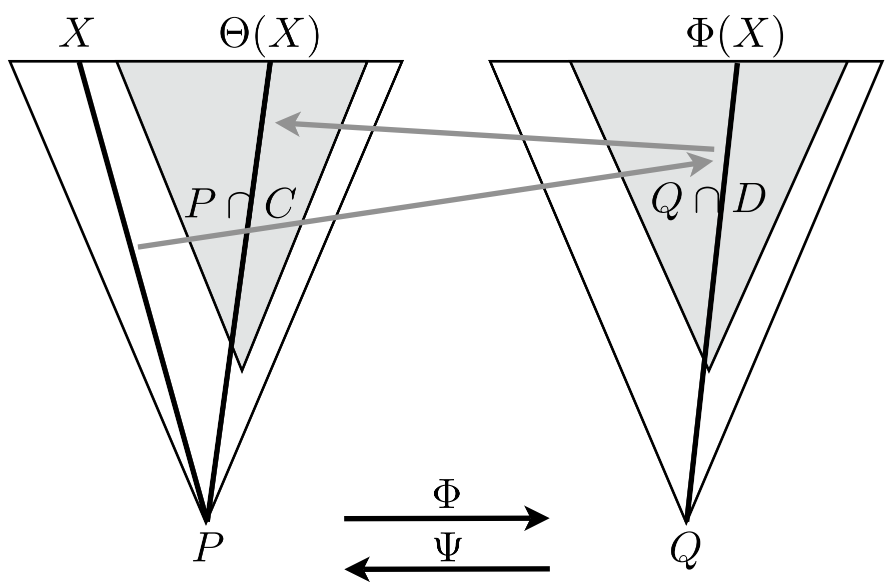

Figure 2: Theorem 2.1.4: $\Phi(X) \in D$

inseparable, so $\mathbf{a}$ is not inseparable. For the converse assume $\mathbf{a} \in \mathcal{L}_M$ is not inseparable and fix $P \in \mathbf{a}$ and a clopen set C good for P such that $P \cap C \perp_M P \cap C^c$. Let $R = P \cap C$ and $Q = P \cap C^c$ and note that R and Q are Π^0_1 classes. By Lemma 1.2.3, $R \geq_M P$ and $Q \geq_M P$. If $R \leq_M P$, say by Φ, then $\Phi \upharpoonright Q$ witnesses $R \leq_M Q$, a contradiction. Thus $R >_M P$ and similarly $Q >_M P$. Let $S = R \wedge Q$ and observe $S \geq_M P$. Define the computable functional

$$\Psi(X) = \begin{cases} 0^\frown X & \text{if } X \in C, \\ 1^\frown X & \text{if } X \in C^c. \end{cases} \tag{2.2}$$

Observe Ψ witnesses $S \leq_M P$ and thus $S \equiv_M P$. It follows that $\mathbf{a} = \deg(R) \wedge \deg(Q)$ and $\deg(R) \perp \deg(Q)$. Thus $\mathbf{a}$ is branching. □

2.2 Hyperinseparable degrees

We strengthen inseparability by requiring reductions in both directions.

Definition 2.2.1. *A Π^0_1 class P is* hyperinseparable *if for any clopen set C good for P, $P \cap C \equiv_M P \cap C^c$.*

Observe that replacing $P \cap C \equiv_M P \cap C^c$ with $P \cap C \equiv_M P$ results in an equivalent definition. Hyperinseparability claims that any clopen subclass "looks" the same as the whole class.

We might hope that, like inseparability, hyperinseparability is an invariant of a degree. This is not the case.

Lemma 2.2.2. *Let $\boldsymbol{a} \in \mathcal{L}_M$. If $\boldsymbol{a} < \boldsymbol{1}$, then $\boldsymbol{a}$ has a non-hyperinseparable member.*

Proof. Fix $P \in \mathbf{a}$ and $Q >_M P$. Then $P \wedge Q \in \mathbf{a}$ and $I(0)$ witnesses that $P \wedge Q$ is not hyperinseparable as $(P \wedge Q) \cap I(0) \equiv_M P <_M Q \equiv_M (P \wedge Q) \cap I(1)$. □

There is, however, a weaker property of all members of a hyperinseparable degree.

Theorem 2.2.3. *Let Q be a hyperinseparable Π^0_1 class and $P \equiv_M Q$. Then there exists a Π^0_1 class $R \subseteq P$ such that $R \equiv_M P$ and R is hyperinseparable.*

Proof. Let $\Phi\colon P \to Q$ and $\Psi\colon Q \to P$ witness $P \equiv_M Q$. Let $R = \Psi(Q)$ and note that R is a Π^0_1 subclass of P. By Lemma 1.2.3, $R \geq_M P$, and the map $\Psi(\Phi(\cdot))$ witnesses $R \leq_M P$, thus $R \equiv_M P$. Fix a clopen set C good for R and let $D = \Psi^{-1}(C)$. As Ψ is onto R, D is good for Q. By hyperinseparability there exists $\Omega\colon Q \cap D \to Q \cap D^c$. Define

$$\Theta(X) = \begin{cases} \Psi\big(\Phi(X)\big) & \text{if } \Phi(X) \in Q \cap D^c, \\ \Psi\big(\Omega(\Phi(X))\big) & \text{if } \Phi(X) \in Q \cap D. \end{cases} \tag{2.3}$$

Thus $R \cap C^c \leq_M R \cap C$. A symmetric argument replacing Ω with $\Omega'\colon Q \cap D^c \to Q \cap D$ shows $R \cap C^c \geq_M R \cap C$. As C was arbitrary, R is hyperinseparable. □

This result is strengthened by the following lemma which states that such a class, one with an equivalent hyperinseparable subclass, essentially behaves as a hyperinseparable class.

Lemma 2.2.4. *Let P be a Π^0_1 class and $R \subseteq P$ with $R \equiv_M P$. If C is a clopen set which is good for both P and R such that $R \cap C \equiv_M R \cap C^c$, then $P \cap C \equiv_M P \cap C^c$.*

Proof. Let $\Omega\colon P \to R$ witness $R \leq_M P$. Let $\Phi\colon R \cap C \to R \cap C^c$ and $\Psi\colon R \cap C^c \to R \cap C$

witness $R \cap C \equiv_M R \cap C^c$. Define

$$\Theta(X) = \begin{cases} \Phi\big(\Omega(X)\big) & \text{if } \Omega(X) \in R \cap C, \\ \Omega(X) & \text{if } \Omega(X) \in R \cap C^c. \end{cases} \tag{2.4}$$

Observe that Θ witnesses $P \cap C^c \leq_M P \cap C$. A symmetric argument with Ψ in place of Φ demonstrates that $P \cap C \leq_M P \cap C^c$. Thus $P \cap C \equiv_M P \cap C^c$. □

A way to think of this Lemma is as follows. If P is a member of a hyperinseparable degree, then it has some hyperinseparable core R. Any clopen set good for P which does not ignore R (either by avoiding it or absorbing it) splits P into two equivalent pieces.

2.3 Homogeneous degrees

So far we have not proved the existence of any non-trivial non-branching degrees. It is an easy exercise to show that the bottom degree, **0**, is non-branching. The top degree, **1**, is trivially non-branching. Both **0** and **1** are examples of homogeneous classes. In [3] Cenzer and Hinman show that all homogeneous degrees are non-branching.

Definition 2.3.1. [3, Def. 8] *A tree* $\mathbb{P}$ *is* homogeneous *if*

$$\forall \sigma, \tau \in \mathbb{P}\ \forall i \in 2 \big[|\sigma| = |\tau| \Rightarrow (\sigma^\frown i \Leftrightarrow \tau^\frown i)\big].$$

A class P *is* homogeneous *if* $\mathcal{T}_P$ *is.*

This definition is very closely tied to separating classes. If A and B are subsets of ω, then we define the *separating class* $\mathcal{S}(A, B) = \{C : A \subseteq C \subseteq \omega \setminus B\}$. In the case that A and B are c.e., $\mathcal{S}(A, B)$ is a Π^0_1 class [3].

Lemma 2.3.2. [3, Prop. 9] *For any* Π^0_1 *class* P*,*

$$P \text{ is homogeneous} \Leftrightarrow P \text{ is a c.e. separating class.}$$

Note that $\mathbf{0}$ is the degree of the separating class of $\emptyset$ and ω and $\mathbf{1}$ is the degree of the separating class of $\{n\colon \{n\}(n)\downarrow= 0\}$ and $\{n\colon \{n\}(n)\downarrow= 1\}$ [10].

Homogeneous classes have the property of being closed under string splice operations.

Definition 2.3.3. *Given* $\sigma, \tau \in 2^{<\omega}$ *define* τ splice σ*, denoted by* $\tau \,/\, \sigma$*, as*

$$(\tau \,/\, \sigma)(i) = \begin{cases} \sigma(i) & \text{if } i < |\sigma|, \\ \tau(i) & \text{if } |\sigma| \le i < |\tau|, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Lemma 2.3.4. *For* $\sigma \in 2^{<\omega}$ *the function* $\tau \mapsto \tau \,/\, \sigma$ *is a computable tree map.*

Proof. Let $\phi(\tau) = \tau \,/\, \sigma$. That ϕ is computable is immediate. Fix $\tau, \tau' \in 2^{<\omega}$ with $\tau \preceq \tau'$. By definition $\phi(\tau)\restriction|\sigma| = \phi(\tau')\restriction|\sigma|$. For $|\sigma| \le i < |\tau'|$, $\phi(\tau')(i) = \tau'(i) = \tau(i) = \phi(\tau)(i)$. Thus $\phi(\tau) \preceq \phi(\tau')$. Now fix $X \in 2^\omega$ and n. Let $m > n$, then $|\phi(X\restriction m)| = |X\restriction m| = m > n$. □

Lemma 2.3.5. *If* P *is a homogeneous* Π^0_1 *class, then for all* $\sigma, \tau \in \mathcal{T}_P$*,* $\tau \,/\, \sigma \in \mathcal{T}_P$*.*

Proof. Fix σ and induct on $|\tau|$. By assumption $\sigma \in \mathcal{T}_P$, thus for $|\tau| \le |\sigma|$ the conclusion holds. Now fix τ with $n = |\tau| > |\sigma|$ and assume the result holds for τ' with $|\tau'| < n$. Then $\tau\restriction(n-1) \,/\, \sigma \in \mathcal{T}_P$. If $\tau = \tau\restriction(n-1)^\frown 0$, then, by homogeneity, $(\tau\restriction(n-1) \,/\, \sigma)^\frown 0$ is in $\mathcal{T}_P$ but this is just $\tau \,/\, \sigma$. Similarly if $\tau = \tau\restriction(n-1)^\frown 1$, then $\tau \,/\, \sigma = (\tau\restriction(n-1) \,/\, \sigma)^\frown 1$ is in $\mathcal{T}_P$. □

Homogeneous classes fit into our context by the following lemma.

Lemma 2.3.6. [3, Lemma 6, rephrased] *If P is a homogeneous Π^0_1 class, then P is hyperinseparable.*

The following proof is essentially equivalent to that of [3] but phrased in terms of Lemma 2.3.5.

Proof. Fix a clopen set C good for P. Fix $\sigma \in \mathcal{T}_P$ such that $I(\sigma) \subseteq C$. Then, by Lemma 2.3.5, $\tau \mapsto \tau \,/\, \sigma$ witnesses $P \cap I(\sigma) \leq_M P$ and thus $P \cap C \leq_M P$. By Lemma 1.2.3 $P \cap C \geq_M P$, thus $P \cap C \equiv_M P$ and P is hyperinseparable. □

2.4 An inseparable and not hyperinseparable degree

In this section we will demonstrate the existence of degrees which are inseparable (and thus non-branching) and not hyperinseparable. We need to construct a degree which has an inseparable member and every member is not hyperinseparable. This construction can be done with a single Π^0_1 class.

Lemma 2.4.1. *Let P be a Π^0_1 class. If for any clopen set C good for P, $P \cap C <_M P \cap C^c$ or $P \cap C >_M P \cap C^c$, then $\deg(P)$ is inseparable and not hyperinseparable.*

Proof. That $\deg(P)$ is inseparable is immediate. Assume $\deg(P)$ is hyperinseparable. By Theorem 2.2.3 there exists $Q \subseteq P$ such that $Q \equiv_M P$ and Q is hyperinseparable. Fix a clopen set C good for Q and observe that C is also good for P. As Q is hyperinseparable, $Q \cap C \equiv_M Q \cap C^c$ and thus, by Lemma 2.2.4, $P \cap C \equiv_M P \cap C^c$, a contradiction. Thus $\deg(P)$ cannot be hyperinseparable. □

Our goal is to construct a class for which every clopen splitting reduces in exactly one direction. In essence we will embed a uniform descending chain into the class and show that if a clopen subclass contains the tail, then it reduces from but not to the remainder of the class.

Using a density theorem such as that of Binns [2] and paying close attention to effectiveness we arrive at the following result.

Lemma 2.4.2. *Given indices for Π^0_1 classes Q and P with $Q >_M P$ and an index for $\Phi : Q \to P$ witnessing $Q \geq_M P$, we can effectively find an index for a total computable function f such that*

$$Q >_M P_{f(0)} >_M P_{f(1)} > \ldots >_M P. \tag{2.5}$$

Proof. Observe that the proof in [2] is effective. □

We now turn to the construction of our class.

Theorem 2.4.3. *Given $\boldsymbol{b} \in \mathcal{L}_M$ there exists $\boldsymbol{a} \in \mathcal{L}_M$ such that $\boldsymbol{0} <_M \boldsymbol{a} <_M \boldsymbol{b}$ and $\boldsymbol{a}$ is inseparable and not hyperinseparable.*

Proof. Fix $Q \in \mathbf{b}$. By Lemma 2.4.2 there exists a computable function f such that $Q >_M P_{f(0)} >_M P_{f(1)} >_M \ldots >_M 0_M$. Let $\mathbb{P}_{f(i)}$ be a tree such that $P_{f(i)} = [\mathbb{P}_{f(i)}]$ and similarly for $\mathbb{Q}$. Fix X, a c.e. set $>_T \emptyset$. We will build a tree $\mathbb{R}$, a set Y, and strings $\{\beta_i\}_{i\in\omega}$,$\{\delta_i\}_{i\in\omega}$ with the following structure:

$$\lim_i \beta_i = Y, \tag{2.6a}$$

$$\delta_i \succeq \beta_{i-1} \text{ for } i > 0, \tag{2.6b}$$

$$X \equiv_T Y, \tag{2.6c}$$

$$R = [\mathbb{R}] = \{Y\} \cup \Big(\bigcup_{i\in\omega} \beta_i {}^\frown P_{f(i)}\Big) \cup \Big(\bigcup_{i\in\omega} \delta_i {}^\frown P_{f(i)}\Big). \tag{2.6d}$$

Furthermore, (2.6d) will be a disjoint union. We will show that $\mathbf{a} = \deg(R)$ satisfies our theorem.

For each i we have a strategy with two possible states, *wait* and *stop*. Our construction proceeds in stages. At each stage some finite number of strategies will be active. Strategies start in state *wait* and may move to state *stop* at some later stage. Once in state *stop* a strategy will remain in that state unless injured. When a strategy i moves from state *wait* to state *stop* it will injure all strategies j with $j > i$. Our argument thus proceeds in typical finite injury fashion; for any given strategy there will be a finite stage after which the strategy is never again injured.

Let $\mathbb{R}_s$, $\beta_{i,s}$, and $\delta_{i,s}$ denote $\mathbb{R}$, β_i, and δ_i at the end of stage s, respectively. To ensure that $\mathbb{R}$ is computable we construct an increasing total computable function, $l(s)$, such that $\sigma \in \mathbb{R}_s \setminus \mathbb{R}_{s-1}$ iff $l(s-1) < |\sigma| \leq l(s)$. We will define $m(s)$ such that strategy i is active at stage s iff $i \leq m(s)$. Fix an enumeration $\{X_s\}_{s\in\omega}$ of X such that $|X_s \setminus X_{s-1}| = 1$.

Each strategy i will have two strings β_i and δ_i and above each it will build a copy of $\mathbb{P}_{f(i)}$. So long as $i \notin X$ the construction for strategies $j > i$ continues above β_i. If i enters X, then the work above β_i will be abandoned, β_i and δ_i will swap roles, and construction continues above the new β_i. In this manner we encode X in $Y = \lim_i \beta_i$. Strategy i is in state *wait* while it is waiting for i to enter X. Once i enters X it will change to state *stop*.

To begin our construction let $l(-1) = 0$, $m(-1) = -1$, and $\mathbb{R}_{-1} = \emptyset$.

Assume we have run our construction to stage s. Thus $\beta_{i,t}$, $\delta_{i,t}$, $l(t)$, $m(t)$, and $\mathbb{R}_t$ are defined for all $t < s$. Let $x \in X_s \setminus X_{s-1}$ (recall that such exists and is unique). If $x \leq m(s-1)$, then strategy x needs to change state and injure all higher strategies. In such a case let $\beta_{i,s} = \delta_{i,s-1}$, $\delta_{i,s} = \beta_{i,s-1}$, and $j = x + 1$. Here j denotes next strategy to activate. If $x > m(s-1)$, then simply let $j = m(s-1)+1$.

Having dealt with any possible injury we are ready to expand our tree. We want to activate strategy j and then let all active strategies grow. Fix σ and τ minimal such that $l(s-1) < |\beta_{j-1,s-1}| + |\sigma| < |\beta_{j-1,s-1}| + |\tau|$ and σ and τ are leaves of $\beta_{j-1,s-1} {}^\frown \mathbb{P}_{f(j)}$. Such σ,τ exist as $[\mathbb{P}_{f(j)}] >_M 0_M$ and, furthermore, can be found computably from j and $\beta_{j-1,s-1}$. If $j \notin X$, then let $\beta_{j,s} = \sigma$ and $\delta_{j,s} = \tau$. If $j \in X$, then let $\beta_{j,s} = \tau$ and $\delta_{j,s} = \sigma$. Let $m(s) = j$ and $l(s) = \max\{|\delta_{j,s}|, |\beta_{j,s}|\}$. For all $i < j$, let $\beta_{i,s} = \beta_{i,s-1}$, $\delta_{i,s} = \delta_{i,s-1}$. Let

$$
\begin{aligned}
\mathbb{R}_s = \mathbb{R}_{s-1} \cup & \\
& \{\sigma \colon \sigma \preceq \beta_{j,s} \text{ and } l(s-1) < |\sigma|\} \cup && \text{(2.7a)} \\
& \{\sigma \colon \sigma \preceq \delta_{j,s} \text{ and } l(s-1) < |\sigma|\} \cup && \text{(2.7b)} \\
& \{\beta_{i,s} {}^\frown \sigma \colon i \leq j,\ \sigma \in \mathbb{P}_{f(i)},\ \text{and } l(s-1) < |\sigma| + |\beta_{i,s}| \leq l(s)\} \cup && \text{(2.7c)} \\
& \{\delta_{i,s} {}^\frown \sigma \colon i \leq j,\ \sigma \in \mathbb{P}_{f(i)},\ \text{and } l(s-1) < |\sigma| + |\delta_{i,s}| \leq l(s)\}. && \text{(2.7d)}
\end{aligned}
$$

Claim. *For each i there exists s such that strategy i is not injured after stage s.*

Proof. Each strategy i only injures strategies j with $j > i$. Thus strategy 0 is never injured. Assume the claim holds for all strategies $h < i$ and fix r such that no strategy $h < i$ is injured after stage r. If any strategy $g < h$ changed state it would injure strategy h, thus no strategy $g < h$ can change state after stage r. Strategy h will change state at most once after stage r. Fix stage $s > r$ such that h does not change state after stage s. So no strategy $h < i$ will change state after stage s and thus strategy i will not be injured after stage s. The results follows by induction. □

For any strategy i we can fix s as above and fix $s' \geq s$ such that $X_{s'}(i) = X(i)$.

Observe that strategy i will not be injured or change state after stage s'.

Claim. $\lim_s m(s) = \infty$ *and* $\lim_s l(s) = \infty$.

Proof. Observe that $l(s)$ is strictly increasing. Thus $\lim_s l(s) = \infty$. The function $m(s)$ only decreases when a strategy i changes state at which point it decreases to $i+1$. If no strategy changes state, then $m(s)$ increases by 1. For any n, fix s such that strategy n is not injured after stage s. Then no strategy $j < n$ will need to change state after stage s. If $m(s) < n$, then no strategy changes state and $m(s+1) = m(s)+1$. This continues until a stage t with $m(t) > n$. No strategy below n will need to change state at a later stage so for all $r > t$, $m(r) \geq n$. □

As a corollary, $\mathbb{R}$ is infinite and thus $R = [\mathbb{R}]$ is a non-empty Π^0_1 class.

Claim. *For all* i, $\beta_i = \lim_s \beta_{i,s}$ *and* $\delta_i = \lim_s \delta_{i,s}$ *exist.*

Proof. Fix s such that strategy i is not injured and does not change state after stage s. Then $\beta_{i,t} = \beta_{i,s}$ for all $t > s$ and similarly for $\delta_{i,t}$. □

Claim. $\beta_{i-1} \prec \beta_i$ *and* $\beta_{i-1} \prec \delta_i$.

Proof. Fix i and observe that β_{i-1} will only change value when strategy $i-1$ is injured or changes state. In both cases strategy i will be injured and any value of β_i destroyed. Thus β_i will settle down only after β_{i-1} does. As β_i is chosen to extend β_{i-1} the first result holds. A symmetric argument shows the second result. □

As a corollary, $Y = \lim_i \beta_i$ exists.

Claim. $X \equiv_T Y$.

Proof. Assume we have an oracle X and wish to compute whether $x \in Y$. Run the computation until a stage s such that there exists i with $\beta_{i,s}$ defined, $|\beta_{i,s}| > x$, and $X \restriction i = X$. We can compute such a stage knowing X. As no $h \leq i$ will enter X, strategy i will not be injured or change state after stage s. Thus $Y(x) = \beta_{i,s}(x)$ and $Y \leq_T X$. For the converse assume we have an oracle for Y and wish to compute $x \in X$. Run the computation until a stage s such that $\beta_{x,s} \prec Y$. Then $X(x) = X_s(x)$ and $X \leq_T Y$. So $X \equiv_T Y$. □

Claim. *$R = \{Y\} \cup \left(\bigcup_{i\in\omega} \beta_i {}^\frown P_i\right) \cup \left(\bigcup_{i\in\omega} \delta_i {}^\frown P_i\right)$ and this union is disjoint.*

Proof. As notation define $\mathbb{T}{\restriction}n = \{\sigma \in \mathbb{T} : |\sigma| \leq n\}$. Define

$$\mathbb{T}_s = \{\sigma : \sigma \preceq \beta_{m(s),s}\} \cup \bigcup_{i \leq m(s)} \beta_{i,s} {}^\frown \mathbb{P}_{f(i)} {\restriction} l(s) \cup \bigcup_{i \leq m(s)} \delta_{i,s} {}^\frown \mathbb{P}_{f(i)} {\restriction} l(s). \tag{2.8}$$

We will show that for all s, $\mathbb{T}_s \subseteq \mathbb{R}_s$, and that for all $\sigma \in \mathbb{R}_{s+1} \setminus \mathbb{R}_s$, σ extends some path in $\mathbb{T}_s$. Thus $\mathrm{Ext}(\mathbb{R}) = \lim_s \mathbb{T}_s$ and the claim follows.

Equation (2.8) is trivially true for $\mathbb{T}_{-1}$. Assume (2.8) holds for all $\mathbb{T}_t$ with $t < s$. If there is a strategy i, $i \leq m(s-1)$, which needs to change state at stage s, then we drop $m(s)$ down to $i+1$, and swap β_i and δ_i. As both $\beta_{i,s-1}$ and $\delta_{i,s-1}$ are in $\mathbb{T}_{s-1}$ so are $\beta_{i,s}$ and $\delta_{i,s}$. If no strategy needs to change states, then $m(s) = m(s-1)+1$. In either case, $m(s)$ is one more than the largest active (uninjured) strategy. Strategy $m(s)$ chooses σ and τ which are leaves of $\mathbb{R}_{m(s)}$ and are longer than $l(s-1)$. By assumption all initial substrings of σ and τ up to length $l(s-1)$ are in $\mathbb{T}_{s-1}$ and by (2.7a) and (2.7b) we will have all initial substrings of σ and τ in $\mathbb{R}_s$. The strings $\beta_{m(s),s}$ and $\delta_{m(s),s}$ are chosen from σ and τ. Finally, (2.7c) and (2.7d) fill in our trees up to length $l(s)$. Strategies only add strings longer than $l(s-1)$ at stage s. Thus anytime a string in $\mathbb{T}_{s-1}$ is not extended at stage s it will never be extended and thus is not in $\mathrm{Ext}(\mathbb{R})$. Conversely, only strings in $\mathbb{T}_{s-1}$ are extended at stage s, so any member of $\mathrm{Ext}(\mathbb{R})$ must be in every $\mathbb{T}_s$.

To see that the union is disjoint observe that β_i and δ_i are chosen from the leaves of $\mathbb{P}_{f(i-1)}$. Thus $\mathbb{P}_{f(i-1)}$ will not provide any ancestors of β_i or δ_i. The result follows by induction. $\square$

We now prove that R and $\mathbf{a}$ have the desired properties.

Recall that $Q \subseteq R \Rightarrow Q \geq_M R$ and $\sigma^\frown Q \equiv_M Q$.

Claim. *For any clopen set C good for R, $R \cap C <_M R \cap C^c$ or $R \cap C >_M R \cap C^c$.*

Proof. Fix a clopen set C good for R. Without loss of generality we can assume $Y \in P \cap C^c$. Let $G = \bigcup_{i\in\omega}\{\beta_i, \delta_i\}$. Let $A = \{i : (\gamma^\frown P_{f(i)}) \cap (R \cap C) \neq \emptyset$ for some $\gamma \in G\}$ and $B = \{\gamma \in G : \exists i(\gamma^\frown P_{f(i)}) \cap (R \cap C) \neq \emptyset\}$. As $Y \in R \cap C^c$, both A and B are finite and co-infinite. In addition, every string in $\mathbb{R} \cap C$ is either a prefix of some $\gamma \in B$ or of the form $\gamma^\frown \varepsilon$ for some $\gamma \in B$, $\varepsilon \in \mathbb{P}_{f(i)}$, and $i \in A$. Fix $n > \sup A$, and, for all $i \in A$, let Θ_i be a computably continuous functional with $\Theta_i \colon P_{f(i)} \to P_{f(n)}$. Fix $\sigma \in G$ such that $\sigma^\frown P_{f(n)} \subseteq R$.

We can now define $\Phi \colon R \cap C \to R \cap C^c$ as follows. Fix $U \in R \cap C$. Let $\gamma \in G$, i, and $V \in P_{f(i)}$ be such that $U = \gamma^\frown V$. Let $\Phi(U) = \sigma^\frown \Theta_i(V)$. Observe that Φ only used A, B, Θ_i for $i \in A$, and σ. This is a finite amount of information, so Φ is a computable functional. Thus $R \cap C \geq_M R \cap C^c$.

Let $m = \sup A$ and observe that $R \cap C \equiv_M \bigwedge_{i\in A}(P_{f(i)} \cap C) \geq_M \bigwedge_{i\in A} P_{f(i)} \equiv_M P_{f(m)}$. If $R \cap C^c \geq_M R \cap C$, then $P_{f(n)} \geq_M R \cap C^c \geq_M R \cap C \equiv_M P_{f(m)}$. But $n > m$ so $P_{f(n)} <_M P_{f(m)}$, a contradiction. Thus $R \cap C^c \not\geq_M R \cap C$. So $R \cap C >_M R \cap C^c$. $\square$

By Lemma 2.4.1, **a** is inseparable and not hyperinseparable.

Claim. $\mathbf{0} < \mathbf{a} < \mathbf{b}$.

Proof. That $\mathbf{0} < \mathbf{a}$ is immediate as 0 is homogeneous and thus hyperinseparable and **a** is not. Let $C = I(\delta_0)$ and note $Y \in R \cap C^c$ so $R \leq_M R \cap C^c <_M P \cap C \equiv_M P_{f(0)} <_M Q$ and $\mathbf{a} < \mathbf{b}$. □

This concludes the proof of the theorem. □

The following is a strengthening of the Theorem.

Theorem 2.4.4. *Given Q and R with $Q >_M R$ there exists P such that* $\deg(P)$ *is inseparable and not hyperinseparable and $Q >_M P \geq_w R$.*

Proof. We modify the construction used in Theorem 2.4.3. Using Lemma 2.4.2 we can have $P_{f(i)} >_M R$ for all i. The rest of the construction proceeds unmodified. Then, any path in $Y \in P$ will be of the form $Y = \sigma ^\frown X$ where $X \in P_{f(i)}$ for some i. Unfortunately, we cannot, given $Y \in P$, uniformly compute $\sigma ^\frown X = Y$. But, as $P_{f(i)} >_M R$, for any particular Y, the map $Y \mapsto X \mapsto R$ is computable. Thus $P \geq_w R$. □

2.5 A hyperinseparable and not homogeneous degree

In this section we will construct a degree which is hyperinseparable and not homogeneous. We first explore a condition which guarantees hyperinseparability and is easy to work into a construction. With that in hand we explore certain properties of homogeneous degrees and find a specific, if technical, property which can be diagonalized against. We then build such a class via a finite injury construction.

Consider the following generalization of homogeneity.

Definition 2.5.1. *For a tree $\mathbb{T}$, n is a* duplication level *if*

$$\forall \sigma, \tau \in \mathbb{T}\big[(|\sigma| = n \text{ and } |\tau| \geq n) \Rightarrow (\tau \,/\, \sigma \in \mathbb{T})\big].$$

Homogeneity is equivalent to every level being a duplication level.

Lemma 2.5.2. *If P is a Π^0_1 class and $\mathcal{T}_P$ has an infinite number of duplication levels, then P is hyperinseparable.*

Proof. Fix a clopen set C good for P and fix m, n, and α_i such that $C = \bigcup_{i<m} I(\alpha_i)$ with $|\alpha_i| = n$. Let $n' \geq n$ be a duplication level of P and choose a σ such that $\sigma \succeq \alpha_i$ for some i and $|\sigma| = n'$. Then $\tau \mapsto \tau \,/\, \sigma$ witnesses $P \cap I(\sigma) \leq_M P \cap C$ and thus $P \cap C^c \leq_M P \cap C$. A symmetric argument shows $P \cap C \leq_M P \cap C^c$. □

We now turn to properties of homogeneous degrees relating to effectiveness. Consider the following example, illustrated in Figure 3. Fix a homogeneous class Q and $P \equiv_M Q$ with $\Phi\colon P \to Q$ and $\Psi\colon Q \to P$. For any $\sigma \in \mathcal{T}_P$ we define

$$\theta_\sigma(\tau) = \psi\big(\phi(\tau) \,/\, \phi(\sigma)\big).$$

As ϕ and ψ are computable and by Lemma 2.3.4, θ_σ is a computable tree map. Note that an index for θ_σ can be computed from σ. For $C = I(\psi(\phi(\sigma)))$ we have $\Theta_\sigma\colon P \cap C^c \to P \cap C$. In this manner, in homogeneous degrees, we can effectively convert members of $\mathcal{T}_P$ into witnesses of inseparability. In this simple form we have very little control over

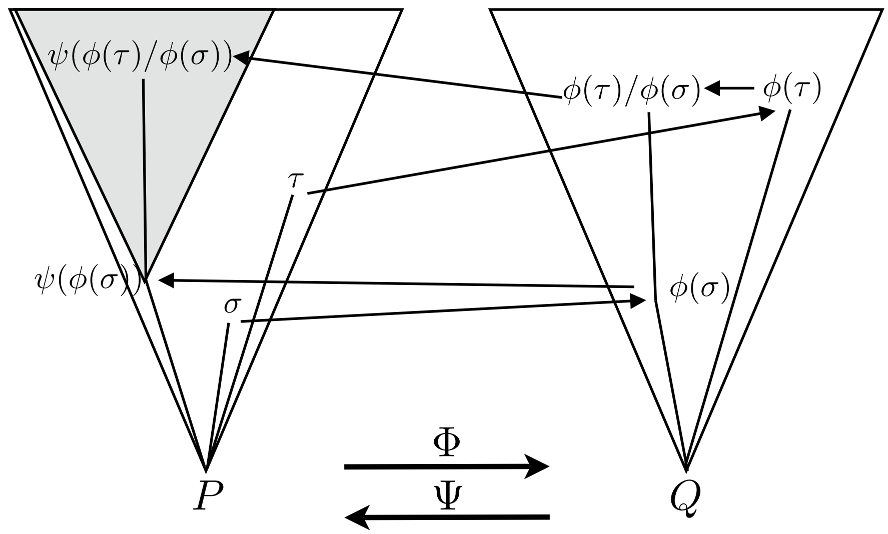

Figure 3: Effectiveness in homogeneous degrees

what clopen subclasses we are reducing between but it serves as an initial example of the technique.

The purpose behind the following definitions and lemmas is to do something similar to the above except that we constrain σ and C. Specifically, for a given level n, we want $C \subseteq I(\sigma \restriction n)$, i.e., C and σ in a single cone generated at level n.

Rather than diagonalize against possible classes Q, we will diagonalize against pairs of functions witnessing the equivalence. In all that follows, ϕ and ψ should be thought of as possible witnesses $\Phi\colon P \to Q$ and $\Psi\colon Q \to P$.

Definition 2.5.3. *For a Π^0_1 class P, and functions $\phi, \psi \colon 2^{<\omega} \to 2^{<\omega}$ define*

$$f_{\phi,\psi}(n) = \mu\sigma[\sigma \in \mathcal{T}_P \text{ and } |\sigma| > n \text{ and } \exists i(|\sigma| \geq i > 0 \text{ and } |(\psi\phi)^i(\sigma)| > n \text{ and } (\psi\phi)^i(\sigma)\upharpoonright n = \sigma\upharpoonright n)], \tag{2.9}$$

$$\mathrm{ind}_{\phi,\psi}(n) = \mu i[(\psi\phi)^i(f_{\phi,\psi}(n))\upharpoonright n = f_{\phi,\psi}(n)\upharpoonright n], \tag{2.10}$$

$$\hat{f}_{\phi,\psi}(n) = (\psi\phi)^{\mathrm{ind}_{\phi,\psi}(n)}(f_{\phi,\psi}(n)), \tag{2.11}$$

$$\theta_{\phi,\psi}(n,\sigma,\tau) = \psi\big(\phi(\tau) \,/\, \phi((\psi\phi)^{\mathrm{ind}_{\phi,\psi}(n)-1}(\sigma))\big). \tag{2.12}$$

We write $f_{\phi,\psi}(n)\downarrow$ when a valid σ exists and $f_{\phi,\psi}(n)\uparrow$ otherwise, and similarly for the other functions.

In our context, $\sigma = f_{\phi,\psi}(n)$ is the element of $\mathcal{T}_P$ which we use to generate a function, $\theta_{\phi,\psi}(n,\sigma,\cdot)$, mapping into $I(\hat{f}_{\phi,\psi}(n))$. Note that when ϕ and ψ are total computable functions $f_{\phi,\psi}$, $\mathrm{ind}_{\phi,\psi}$, and $\hat{f}_{\phi,\psi}$ are all computable, and $\Theta_{\phi,\psi}$ is computably continuous.

Lemma 2.5.4. *If P and Q are Π^0_1 classes with $\Phi \colon P \to Q$ and $\Psi \colon Q \to P$ computable functionals, then $f_{\phi,\psi}(n)\downarrow$ for all n. Furthermore, if Q is homogeneous, then*

$$\forall n\big[\theta_{\phi,\psi}(n, f_{\phi,\psi}(n), \emptyset)\upharpoonright n = f_{\phi,\psi}(n)\upharpoonright n \text{ and } \Theta_{\phi,\psi}(n, f_{\phi,\psi}(n), \cdot) \colon P \to P\big]. \tag{2.13}$$

Proof. Let condition $(\star)$ be

$$\sigma \in \mathcal{T}_P \text{ and } |\sigma| > n \text{ and } |\sigma| \geq i > 0 \text{ and } |(\psi\phi)^i(\sigma)| > n \text{ and } (\psi\phi)^i(\sigma)\upharpoonright n = \sigma\upharpoonright n. \tag{2.14}$$

To prove the first claim it suffices to show that for all n there exists σ and i satisfying $(\star)$. Fix n and any $X \in P$. Observe that $(\psi\phi)^a$ is a tree map for all a. Let $A =$

$\{(\Psi\Phi)^a(X) : a \in \omega\}$ and note $A \subseteq P$. If A is finite, then there exist a, b with $a < b$ and $(\Psi\Phi)^a(X) = (\Psi\Phi)^b(X)$. If A is infinite, then, by compactness and the Pigeonhole Principle, there exist a, b with $a < b$ and $(\Psi\Phi)^a(X) \restriction n = (\Psi\Phi)^b(X) \restriction n$. In either case let s be large enough so that $|(\psi\phi)^a(X \restriction s)| > n$ and $|(\psi\phi)^b(X \restriction s)| > n$. Then $\sigma = (\psi\phi)^a(X \restriction s)$ and $i = b - a$ satisfies $(\star)$.

If Q is homogeneous, then, by Lemma 2.3.5, θ is a map from $\mathcal{T}_P$ to $\mathcal{T}_P$. As

$$\begin{aligned} \theta_{\phi,\psi}(n, f_{\phi,\psi}(n), \emptyset) = \psi(\phi(\emptyset) \,/\, \phi((\psi\phi)^{i-1}(f_{\phi,\psi}(n)))) = \\ \psi(\phi((\psi\phi)^{i-1}(f_{\phi,\psi}(n)))) = (\psi\phi)^i(f_{\phi,\psi}(n)), \end{aligned} \tag{2.15}$$

we find that $\theta_{\phi,\psi}(n, f_{\phi,\psi}(n), \emptyset) \restriction n = f_{\phi,\psi}(n) \restriction n$. □

We can now diagonalize against pairs of partial computable functions $\langle\phi, \psi\rangle$ while ensuring that our witnesses share a cone above level n. To ensure that n is a duplication level we copy that cone across level n.

Theorem 2.5.5. *There exists a degree $\boldsymbol{a} \in \mathcal{L}_M$ which is hyperinseparable and not homogeneous.*

Proof. We will construct a computable tree, $\mathbb{P}$, via a finite injury construction. The description begins with a listing of structures and related variables used in the construction. This is followed by a list of invariants which will be preserved at every stage. The class $P = [\mathbb{P}]$ will be such that $\deg(P)$ is hyperinseparable and not homogeneous. We identify the pair $\langle\phi, \psi\rangle$ with a code e and, for simplicity, write f_e and θ_e for $f_{\phi,\psi}$ and $\theta_{\phi,\psi}$ respectively.

We work to satisfy the following requirements:

$$\mathcal{R}_e\colon \exists n\Big[f_e(n)\downarrow\Rightarrow \big(\exists \tau\in \mathcal{T}_P\ \theta_e(n,f_e(n),\tau)\notin \mathcal{T}_P \tag{2.16}$$
$$\text{or } \exists X\in P\ \exists m\ \forall\sigma\prec X|\theta_e(n,f_e(n),\sigma)|<m\big)\Big],$$

$$\mathcal{P}\colon \exists^\infty n\ [n \text{ is a duplication level for } P]. \tag{2.17}$$

Requirement $\mathcal{R}_e$ should understood to say that if θ_e appears to induce a Medvedev equivalence, then it is with a non-homogeneous class, i.e., if $f_e(n)$ converges, and θ_e behaves like a functional, then its range is not P.

Requirement $\mathcal{R}_e$ is of higher priority than $\mathcal{R}_d$ iff $e<d$.

The actual objects constructed are as follows.

- $\mathbb{P}$ is the tree defining the Π^0_1 class. Elements are enumerated into $\mathbb{P}$. $\mathbb{P}_s$ denotes the elements in $\mathbb{P}$ at the end of stage s.

- $l(s)$ is a total computable function defined such that all elements of length $l(s)$ or less have been added to $\mathbb{P}$ by the end of stage s.

The function $l(s)$ will ensure that $\mathbb{P}$ is a computable tree.

The following objects are used internally to construct the above.

- L is the set of live strings. L_s denotes the elements of L at the end of stage s. At stage s, the only paths eligible for extension are those in L_{s-1} of length $l(s-1)$. We will prove that $\text{Ext}(\mathbb{P})=L$.

In addition, for each requirement $\mathcal{R}_e$ we have the following.

- n_e corresponds to the n in the definition of $\mathcal{R}_e$, (2.16). $n_{e,s}$ denotes the value of n_e at the end of stage s. In the absence of injury n_e is constant once defined.

- t_e is the "protection level" of requirement $\mathcal{R}_e$. It will also be a duplication level. $t_{e,s}$ denotes the value of t_e at the end of stage s.

The following object is not required by the construction but will be tracked for use in the proofs of correctness.

- τ_e corresponds to the τ in the definition of $\mathcal{R}_e$, (2.16). $\tau_{e,s}$ will denote the value of τ_e at the end of stage s. In the absence of injury τ_e is constant once defined.

Finally, the following are not objects but rather subsets or notation for various values. These are not modified directly but rather reflect any changes of the objects they are based on.

- $L_s^l = \{\rho : \rho \in L_s \text{ and } |\rho| = l(s)\}$, i.e., the set of live leaves of $\mathbb{P}_s$.
- $L_s^{t_e} = \{\rho : \rho \in L_s \text{ and } |\rho| = t_{e,s}\}$, i.e., the set of live nodes at level $t_{e,s}$,
- $L^{t_e} = \lim_s L_s^{t_e}$.

The following invariants are maintained at all stages s and are used in the proof of correctness. We need $\mathbb{P}$ to be computable and L co-c.e. For every s,

$$l(s-1) \leq l(s), \tag{2.18}$$

$$\sigma \in \mathbb{P}_s \setminus \mathbb{P}_{s-1} \Rightarrow l(s-1) < |\sigma| \leq l(s), \tag{2.19}$$

$$\sigma \in L_s \setminus L_{s-1} \Rightarrow l(s-1) < |\sigma| \leq l(s). \tag{2.20}$$

We want L to have no dead ends and describe live nodes. For every s,

$$\sigma \in L_s \text{ a leaf} \Rightarrow |\sigma| = l(s), \tag{2.21}$$

$$\sigma \in \mathbb{P}_s \setminus \mathbb{P}_{s-1} \Rightarrow \exists \rho \in L_{s-1}[\sigma \succ \rho]. \tag{2.22}$$

The interaction of requirements is controlled by the t_e's. The t_e's form an increasing sequence and will be the duplication levels. In addition, each t_e defines the scope of protection of $\mathcal{R}_e$. Lower priority requirements are not allowed to kill any nodes below t_e. Our witnesses will be chosen above the previous protection level and below ours. For every e and s,

$$d < e \text{ and } t_{d,s} \text{ defined and } t_{e,s} \text{ defined} \Rightarrow t_{d,s} < t_{e,s}, \tag{2.23}$$

$$\mathcal{R}_e \text{ not active at stage } s \Rightarrow L_s^{t_e} = L_{s-1}^{t_e}, \tag{2.24}$$

$$\forall \sigma \in L \forall \rho \in L_s^{t_e} [\sigma \,/\, \rho \in L], \tag{2.25}$$

$$t_{e,s} \geq n_{e,s} > t_{e-1,s} \text{ if } n_{e,s} \text{ defined}, \textit{and} \tag{2.26}$$

$$t_{e,s} \geq |\tau_{e,s}| > n_{e,s} \text{ if } \tau_{e,s} \text{ defined.} \tag{2.27}$$

We treat all objects from Definition 2.5.3 as using L for $\mathcal{T}_P$. This treatment is justified by our proof that $\mathcal{T}_P = \text{Ext}(\mathbb{P}) = L$. For example, $f_e(n_{e,s})[s]$ uses L_s for $\mathcal{T}_P$. Note that this guarantees that $f_e(n_{e,s})[s] \in L_s$.

Each strategy acts in three states, *uninitialized*, *wait1*, *wait2*, and then stops in the state *stop*. Whenever a strategy acts it will injure all lower priority strategies. In *uninitialized* the strategy splits all paths at the previous protection level and chooses an n_e. This technique of splitting is used repeatedly. The strategy will guarantee that at most one of the two splits are killed, thus preserving the common ancestors in L. In *wait1* the strategy waits for $f_e(n_e)$ to converge. If it does, then the strategy ensures a split above $f_e(n_e)$ and raises the protection level to $|f_e(n_e)|$. In *wait2* the strategy waits for a τ to appear. If τ appears the strategy kills the corresponding θ and raises the protection level to $|\tau|$. See Figure 4 for a possible execution of a strategy.

At the beginning $\mathbb{P} = L = \emptyset$, all other variables are undefined, and all strategies

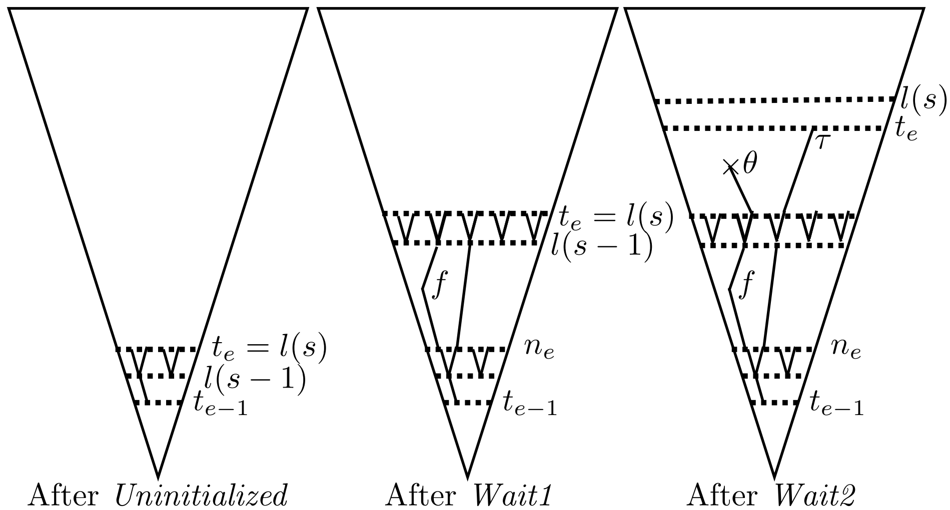

Figure 4: Theorem 2.5.5: Strategy for $\mathcal{R}_e$

are in the state *uninitialized*. At each stage s the highest priority requirement $\mathcal{R}_e$ with $e \leq s$ that requires attention acts. For all other requirements d that are not in the state *uninitialized*, $n_{d,s} = n_{d,s-1}$, $\tau_{d,s} = \tau_{d,s-1}$, and $t_{d,s} = t_{d,s-1}$. The strategy for $\mathcal{R}_e$ is injured when a higher priority strategy for $\mathcal{R}_d$ $(d < e)$ changes t_d. When $\mathcal{R}_e$ is injured it is *reset*: it returns to the state *uninitialized* and $n_{d,s}$, $\tau_{d,s}$, and $t_{d,s}$ become undefined. The strategy is described by state below.

Uninitialized: $\mathcal{R}_e$ always requires attention. Our goal in this state is to choose n_e and split all paths of $L_s^{t_{e-1}}$ to keep them alive. Let

$$L_s = L_{s-1} \cup \{\rho^\frown 0, \rho^\frown 1 : \rho \in L_{s-1}^l\}, \tag{2.28}$$

$$\mathbb{P}_s = \mathbb{P}_{s-1} \cup L_s, \tag{2.29}$$

$$t_{e,s} = n_{e,s} = l(s) = l(s-1) + 1. \tag{2.30}$$

In defining t_e we reset all lower priority requirements. Note that we preserve all invariants. We enter the state *wait1*.

Wait1: $\mathcal{R}_e$ requires attention if $f_e(n_{e,s-1})[s-1]\downarrow$. We need to preserve f_e. To do so we need to raise our protection level and split f_e so that we can preserve its life if it is later necessary to kill a child of it. We do this in a manner identical to the above. Let

$$L_s = L_{s-1} \cup \{\sigma^\frown 0, \sigma^\frown 1 : \sigma \in L^l_{s-1}\}, \tag{2.31}$$

$$\mathbb{P}_s = \mathbb{P}_{s-1} \cup L_s, \tag{2.32}$$

$$t_{e,s} = l(s) = l(s-1) + 1, \tag{2.33}$$

$$n_{e,s} = n_{e,s-1}. \tag{2.34}$$

As before, the definition of t_e causes all lower priority strategies to reset. Note that we preserve all invariants. We enter the state *wait2*.

Wait2: $\mathcal{R}_e$ requires attention if

$$\exists \tau \in L_{s-1}\big[|\tau| > n_{e,s-1} \text{ and} \tag{2.35}$$

$$\theta_e(n_{e,s-1}, f_e(n_{e,s-1})[s], \tau)[s-1] \in L_{s-1} \text{ and} \tag{2.36}$$

$$\tau \restriction t_{e-1,s-1} = f_e(n_{e,s-1})[s-1] \restriction t_{e-1,s-1} \text{ and} \tag{2.37}$$

$$\tau \restriction n_{e,s-1} \neq f_e(n_{e,s-1})[s-1] \restriction n_{e,s-1} \text{ and} \tag{2.38}$$

$$|\theta_e(n_{e,s-1}, f_e(n_{e,s-1})[s-1], \tau)[s-1]| > t_{e,s-1}\big]. \tag{2.39}$$

We want τ to witness $\mathcal{R}_e$ [(2.35) and (2.36)]. Also, τ should be in the same cone above t_{e-1} [(2.37)] but in a different subcone than $\theta_e(n_e, f_e(n_e), \tau)$ [(2.38)] so that τ can stay alive when we kill $\theta_e(n_e, f_e(n_e), \tau) \succeq f_e(n_e)$. Finally, we want $\theta_e(n_e, f_e(n_e), \tau)$ to be above our split over $f_e(n_e)$ so that killing it does not kill $f_e(n_e)$ [(2.39)]. If such a τ exists, then we kill the resulting $\theta_e(n_e, f_e(n_e), \tau)$ and duplicate the kill across all members of

$L^{t_{e-1}}_{s-1}$. To ensure that L_s has no dead ends we need to remove a several strings from it, namely, to kill a path σ we need to remove σ and all ancestors of σ up to the nearest split. Define the *kill set* of σ by

$$ks(\sigma) = \{\epsilon \in L_{s-1} : \forall\delta \in L_{s-1}[\delta \succeq \epsilon \Rightarrow \delta \preceq \sigma]\}.$$

For simplicity of notation let $\sigma = \theta_e\big(n_{e,s-1}, f_e(n_{e,s-1})[s], \tau\big)[s]$. Let

$$\tau_{e,s} = \tau, \tag{2.40}$$

$$L_s = L_{s-1} \setminus \bigcup_{\gamma \in L^{t_{e-1}}_{s-1}} ks(\sigma \,/\, \gamma), \tag{2.41}$$

$$\mathbb{P}_s = \mathbb{P}_{s-1}, \tag{2.42}$$

$$l(s) = l(s-1), \tag{2.43}$$

$$t_{e,s} = \max\{t_{e,s-1}, |\tau|\}, \tag{2.44}$$

$$n_{e,s} = n_{e,s-1}. \tag{2.45}$$

We reset all lower priority requirements and enter state *stop*. Observe that all invariants are preserved.

Stop: $\mathcal{R}_e$ never requires attention.

Claim. *Every requirement acts a finite number of times.*

Proof. In the absence of injury each requirement acts at most three times, once each in the states *uninitialized*, *wait1*, and *wait2*. Thus, in typical finite injury fashion, every requirement acts a finite number of times. □

Claim. *For all s, $L_s \neq \emptyset$.*

Proof. At stage 0, $\mathcal{R}_0$ is in state *uninitialized* and requires attention. It sets $L_0 = \{0, 1\}$ and $t_{0,0} = 1$. As there are no higher priority strategies $\mathcal{R}_0$ will never be injured and thus $t_{0,0}$ will never be undefined. At all later stages 0 and 1 will be protected by $t_{0,s}$ and thus $\{0, 1\} \subseteq L_s$. □

Claim. $\lim_s l(s) = \infty$ *and* $\lim_s L_s = \mathrm{Ext}(\mathbb{P})$.

Proof. By the previous Lemma, for all s, $T_s \supseteq L_s \neq \emptyset$. As each requirement acts a finite number of times every requirement acts at least once in the state *uninitialized*. During this action $l(s)$ is increased by 1. Thus $\lim_s l(s) = \infty$. As paths leave L at most once, $\lim_s L_s$ is well defined. If $\tau \in L_s$ for all s, then as L_s has no leaves shorter than $l(s)$, for any n we can find s such that $l(s) > n$ and thus τ has a child of length n. As any elements added to L are added to $\mathbb{P}$ we have $\tau \in \mathrm{Ext}(\mathbb{P})$. Conversely, if $\tau \in \mathrm{Ext}(\mathbb{P})$, then it must have children of all lengths. Let s be such that $\tau \in L_s \setminus L_{s-1}$. If τ ever left L after stage s, then by the invariants it could have no more children enter $\mathbb{P}$. But τ has infinitely many children, a contradiction. Thus $\mathrm{Ext}(\mathbb{P}) \subseteq \lim_s L_s$. So $\mathrm{Ext}(\mathbb{P}) = \lim_s L_s$. □

Claim. *Every requirement* $\mathcal{R}_e$ *is satisfied.*

Proof. Assume otherwise. As $\mathcal{R}_e$ acts a finite number of times, let s be such that $\mathcal{R}_e$ acts last at stage s. We now have three cases, depending on which state $\mathcal{R}_e$ acted in at stage s.

Case 1: If $\mathcal{R}_e$ acted in state *uninitialized*, then $f_e(n_{e,s-1}) \uparrow$, otherwise there would be some t such that $f_e(n_{e,s-1})[t] \downarrow$ and $\mathcal{R}_e$ would have acted later. So n_e is defined and $f_e(n_e) \uparrow$ so $\mathcal{R}_e$ is satisfied.

Case 2: If $\mathcal{R}_e$ acted in state *wait1*, then $f_e(n_{e,s})[s] \downarrow$. Observe that $n_e = n_{e,s}$, $f_e(n_e) = f_e(n_{e,s})[s]$, and, by definition, $|f_e(n_e)| > n_{e,s}$ and thus, by invariants, $|f_e(n_e)| >$

$t_{e-1,s} = t_{e-1}$. Let $\gamma = f_e(n_e) \upharpoonright t_{e-1}$. Let s' be the last stage that $\mathcal{R}_e$ acted in state *uninitialized*, and note that in state we put $\gamma^\frown 0$ and $\gamma^\frown 1$ into $L_{s'}$ and raised the protection level, $t_{e,s'}$, to $|\gamma^\frown 0|$. As $\mathcal{R}_e$ was not injured after stage s' both $\gamma^\frown 0$ and $\gamma^\frown 1$ are in L and, as a result, have children of arbitrary length. Without loss of generality we can assume that $f_e(n_e) \succeq \gamma^\frown 0$. Fix $X \in P \succeq \gamma^\frown 1$, such exists as $f_e(n_e) \in L^{t_e}$. If $|\theta_e(n_e, f_e(n-e), \sigma)| < m$ for some m and all $\sigma \in P$, then $\mathcal{R}_e$ is satisfied. Assume otherwise and assume that $\mathcal{R}_e$ is not satisfied. Then $\forall \tau \in \mathcal{T}_P[\theta_e(n_e, f_e(n_e), \tau) \in \mathcal{T}_P]$. There is some child τ of $\gamma^\frown 1$ which is long enough to satisfy the attention requirements of state *wait2*. But then $\mathcal{R}_e$ would have acted in *wait2*, a contradiction. Thus $\mathcal{R}_e$ must be satisfied.

Case 3: If $\mathcal{R}_e$ acted in state *wait2*, then it has killed τ_e. Thus n_e and τ_e witness the satisfaction of $\mathcal{R}_e$. All we need to do is show that $f_e(n_e)$ and τ_e are alive. By the invariants, no requirements of lower priority can kill them, and as $\mathcal{R}_e$ is uninjured after stage s no higher priority requirements killed $f_e(n_{e,s})$ or $\tau_{e,s}$. Both are in the same cone above $\gamma \in L^{t_{e-1}}$ so we only need to worry that the killing of $\gamma = \theta_e(n_e, f_e(n_e), \tau_e)$ killed either. Killing γ cannot kill τ_e, as by the conditions of state *wait2* they are incomparable. Those conditions also require σ to be above the split above $f_e(n_e)$ so killing γ will kill at most one of $f_e(n_e)^\frown 0$ and $f_e(n_e)^\frown 1$. Thus all witnesses are preserved and $\mathcal{R}_e$ is satisfied. □

Claim. *The requirement $\mathcal{P}$ is satisfied.*

Proof. For any n, by the invariants, there exists e such that $t_e > n$. Fix such an e and let s be large enough that $\mathcal{R}_e$ never acts after stage s. Then $t_{e,s}$ is a duplication level as L^{t_e} will not change and all actions after stage s will preserve the invariant that L^{t_e} is a duplication level. □

So $P = [\mathbb{P}]$ satisfies the requirements. By Lemma 2.5.2, P is hyperinseparable. If P is in a homogeneous degree, then there exist ϕ, ψ as in Lemma 2.5.4. But then $\mathcal{R}_{\phi,\psi}$ guarantees that $n_{f,g}$ and $\tau_{f,g}$ witness a contradiction. Thus P is not a homogeneous degree so $\deg(P)$ is a hyperinseparable and not homogeneous degree. □

Chapter 3

Tree Lifes

There are many ways to construct Π^0_1 classes via a priority argument. This chapter formalizes a method in which a tree is enumerated along with a total computable function which tightly bounds the length of paths added at each stage. The combination of enumeration and length function ensures that the tree is in fact computable and thus produces a Π^0_1 class. This technique is seen in Chapter 2 and in the literature, e.g., in [3]. The formalization described below will be used in the next chapter to prove Theorem 4.5.2.

3.1 Basic Definitions and Results

Definition 3.1.1. *A finite tree $L \subseteq 2^{<\omega}$ is a* strict tree *if all dead ends are of the same length (necessarily maximal). The length of L, denoted $l(L)$, is the length of the dead ends. The set of dead ends is denoted $D(L)$.*

Definition 3.1.2. *For strict trees L and M, M is a* growth *of L if $l(M) \geq l(L)$ and*

$$\forall \sigma \in M \setminus L \exists \tau \in D(L)[\sigma \succ \tau]. \tag{3.1}$$

Call a leaf of maximal length a living leaf. Then a growth can be characterized by two conditions: (1) the length can not decrease, i.e., at least one living leaf must survive, and (2) any additional nodes must extend living leaves. Thus, a valid growth may consist of extending living leaves, pruning part of the tree, or a combination of both.

Definition 3.1.3. *A* tree life *is a sequence of strict trees* $\{L_s : s \in \omega\}$ *such that for all* $s > 0$, L_s *is a growth of* L_{s-1} *and* $\lim_s l(L_s) = \infty$. *A tree life is* computable *if there exists a total computable function* f *such that* $f(s) = L_s$.

To simplify notation we will hereafter omit ": $s \in \omega$", i.e., we will simply write $\{L_s\}$.

Lemma 3.1.4. *For any tree life* $\{L_s\}$*, any* s*, and any* $\sigma \in L_{s+1} \setminus L_s$, $l(L_s) < |\sigma| \leq l(L_{s+1})$.

Proof. Fix $\sigma \in L_{s+1} \setminus L_s$. Then, as L_{s+1} is a growth of L_s, there must be some $\tau \in D(L_s)$ with $\tau \prec \sigma$. Thus $|\sigma| > |\tau| = l(L_s)$. That $|\sigma| \leq l(L_{s+1})$ is immediate. □

Corollary 3.1.5. *For any tree life* $\{L_s\}$*, any* s*, and any* $\sigma \in L_s$*, if* $\sigma \notin L_{s+1}$*, then for all* $t > s$, $\sigma \notin L_t$.

Observe that this corollary implies that $\lim_s L_s$ is well defined; it is a d.c.e. set.

Lemma 3.1.6. *For a tree life* $\{L_s\}$, $\lim_s L_s = [\bigcup_s L_s]$.

Proof. The inclusion $\subseteq$ is immediate. For $\supseteq$, fix $X \in \lim_s L_s$. Fix n and let $\sigma = X \upharpoonright n$ and s be such that $\sigma \in L_s$. As σ has descendants of arbitrary length, σ must be in every L_t for $t > s$. As n was arbitrary, $X \in [\bigcup_s L_s]$. □

Lemma 3.1.7. *For a computable tree life* $\{L_s\}$, $\bigcup_s L_s$ *is computable and* $\lim_s L_s$ *is co-c.e.*

Proof. Observe that $l(s)$ is a computable function.

Fix $\sigma \in \bigcup_s L_s$ and t such that $l(t) \geq |\sigma|$. Then $\sigma \in \bigcup_s L_s$ if and only if $\sigma \in L_t$. Thus $\bigcup_s L_s$ is computable. As $\lim_s L_s$ is the difference of a computable set ($\bigcup_s L_s$) and a c.e. set (the nodes that leave), it is co-c.e. □

Corollary 3.1.8. *For a computable tree life* $\{L_s\}$*,* $\lim_s L_s$ *is a* Π^0_1 *class.*

Proof. By Lemma 3.1.7, $\bigcup_s L_s$ is a computable tree. By Lemma 3.1.6, $\lim_s L_s = [\bigcup_s L_s]$ and thus, is a Π^0_1 class. □

3.2 Growths

Having defined the basic construction and shown that it results in computable trees we now define some growth operations which are effective.

Definition 3.2.1. *For a a non-empty strict tree* L*, the* single extension *of* L*, denoted* $\hat{L}$*, is defined by*

$$\hat{L} = L \cup \{\sigma^\frown i : \sigma \in D(L), i \in 2\}. \tag{3.2}$$

Define $\hat{\emptyset} = \{\emptyset, 0, 1\}$.

Note that $l(\hat{L}) = l(L) + 1$.

Definition 3.2.2. *For a strict tree* L *and* $\sigma \in L$*, the* trim *of* L *by* σ*, denoted* $\operatorname{trim}(L, \sigma)$ *is defined by*

$$\operatorname{trim}(L, \sigma) = \{\tau \in L : \exists \nu \succeq \tau [\nu \perp \sigma]\}. \tag{3.3}$$

Note that $\operatorname{trim}(L, \sigma)$ is L with σ and all descendants removed. We also remove ancestors of σ which do not lead to other, non-σ, descendants to ensure that $\operatorname{trim}(L, \sigma)$ is a strict tree. Note that $l(\operatorname{trim}(L, \sigma)) \leq l(L)$.

Lemma 3.2.3. *For a strict tree L, $\hat{L}$ is a strict tree and a growth of L.*

Proof. Fix $\tau \in D(\hat{L})$. By definition, $\tau = \sigma^\frown i$ for $\sigma \in D(L)$ and $i \in 2$. Then $|\sigma| = l(L)$, $|\tau| = l(L) + 1$. As τ was arbitrary, $\hat{L}$ is a strict tree.

Fix $\tau \in \hat{L} \setminus L$. Then $\tau = \sigma^\frown i$ for $\sigma \in D(L)$ and $i \in 2$, and σ witnesses that $\hat{L}$ is a growth. □

Lemma 3.2.4. *For a strict tree L and $\sigma \in L$, $\mathrm{trim}(L, \sigma)$ is a strict tree and either empty or a growth of L.*

Proof. Let $M = \mathrm{trim}(L, \sigma)$ and assume M has a dead end α with $|\alpha| < l(L)$. As $|\alpha| < l(L)$ there is an immediate successor of α, $\beta \in L$. As $\beta \notin M$, β is comparable with σ but α is not, a contradiction. Thus M is a strict tree.

As trim only removes paths, if M is not empty, then it contains a path of length $l(L)$. Thus M is a growth of L. □

Chapter 4

Branching Degrees

We now turn to branching degrees and a parallel hierarchy to that developed in Chapter 2.

4.1 Separable Degrees

We define separability as the inverse of inseparability. Separable degrees are those whose members can be split into incomparable clopen subclasses.

Definition 4.1.1. *A Π^0_1 class P is* separable *if there exists a clopen set C good for P such that $P \cap C \perp_M P \cap C^c$.*

The primary results move over directly.

Theorem 4.1.2. *Separability is an invariant of a Medvedev degree, i.e., if $P \equiv_M Q$ and P is separable, then Q is separable.*

Proof. This theorem is the contrapositive of Theorem 2.1.4. □

Corollary 4.1.3. *A degree $\boldsymbol{a}$ is separable iff $\boldsymbol{a}$ is branching, i.e., there exists $\boldsymbol{b} > \boldsymbol{a}$, $\boldsymbol{c} > \boldsymbol{a}$ with $\boldsymbol{a} = \boldsymbol{b} \wedge \boldsymbol{c}$.*

Proof. This is the contrapositive of Corollary 2.1.5. □

4.2 Hyperseparable Degrees

Definition 4.2.1. *A Π^0_1 class P is* hyperseparable *if for all clopen sets C good for P, $P \cap C \perp_M P \cap C^c$.*

Observe that hyperseparable implies separable.

As in the case of to hyperinseparability, it is too much to hope that this would be invariant.

Theorem 4.2.2. *For any Π^0_1 class P, there exists a class Q, with $Q \equiv_M P$ and Q not hyperseparable.*

Proof. Take $Q = P \wedge P$ and observe that $C = I(0)$ contradicts hyperseparability. □

Additional results about hyperseparable degrees, in the context of non-separability, can be found in Section 4.4.

4.3 Totally Separable Degrees

As with homogeneous in the non-branching case, there is a preexisting condition which is stronger than hyperseparable. I was unable to find a name for it in the literature, so I refer to it as totally separable.

Definition 4.3.1. *A Π^0_1 class P is* totally separable *if for all $X, Y \in P$, $X \perp_T Y$.*

Note that totally separable implies hyperseparable.

Jockusch and Soare proved the existence of a totally separable class.

Theorem 4.3.2. *[7] There exists a totally separable class.*

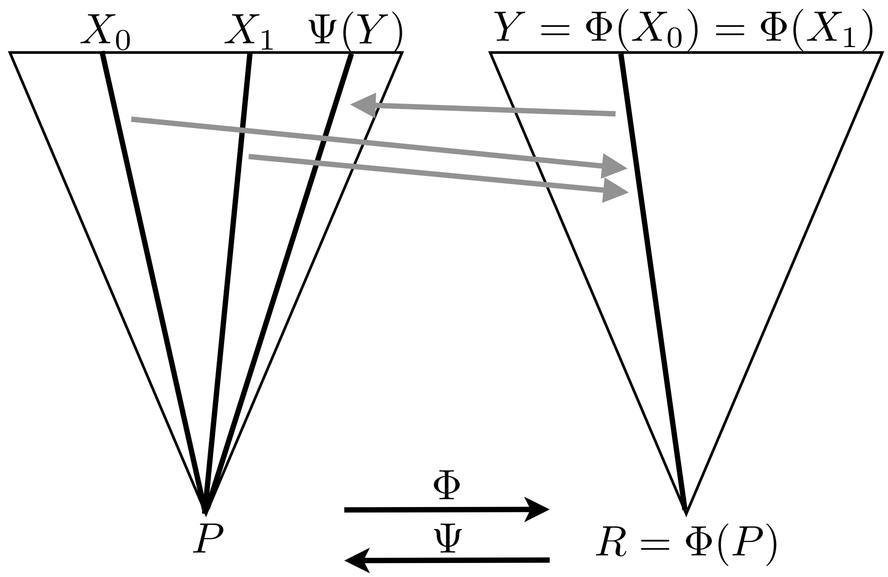

Figure 5: Theorem 4.3.3: $\Phi : P \to R$ injective.

Totally separable is a very strong condition which enforces a great deal of structure on the other members of the degree. The following is very similar to Theorem 2.2.3, i.e., it shows that members of a totally separable degree contain a totally separable core.

Theorem 4.3.3. *Let P be a totally separable Π^0_1 class and Q a Π^0_1 class with $Q \equiv_M P$. Then there exists a Π^0_1 class $R \subseteq Q$ such that $R \equiv_M Q$ and R is totally separable. Furthermore, if $\Phi : P \to Q$ and $\Psi : Q \to P$ witness $Q \equiv_M P$, then $\Phi : P \to R$ and $\Psi : R \to P$ are bijections.*

Proof. Let $R = \Phi(P)$. By Lemma 1.2.3, $R \geq_M Q$. The function $X \mapsto \Psi(X) \mapsto \Phi(\Psi(X))$ witnesses $Q \geq_M R$. Thus $R \equiv_M Q$.

By definition $\Phi : P \to R$ is surjective. Assume Φ is not injective and fix X_0, X_1 in P with $\Phi(X_0) = \Phi(X_1) = Y$. Assume $\Psi(Y) \neq X_0$ (it must differ from one of X_0 and X_1). But $\Psi(Y) \leq_T X_0$ as $X_0 \mapsto Y \mapsto \Psi(Y)$, a contradiction of P being totally separable.

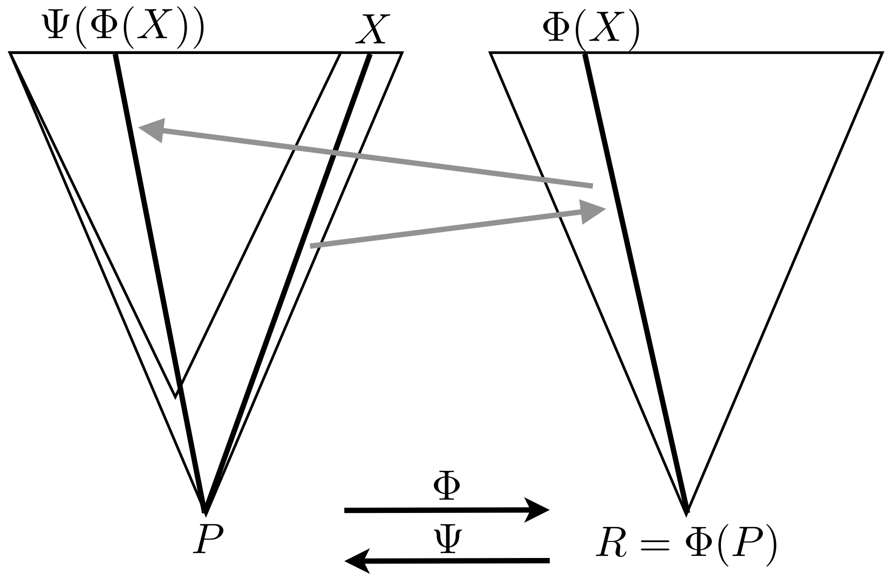

Figure 6: Theorem 4.3.3: $\Psi : R \to P$ surjective

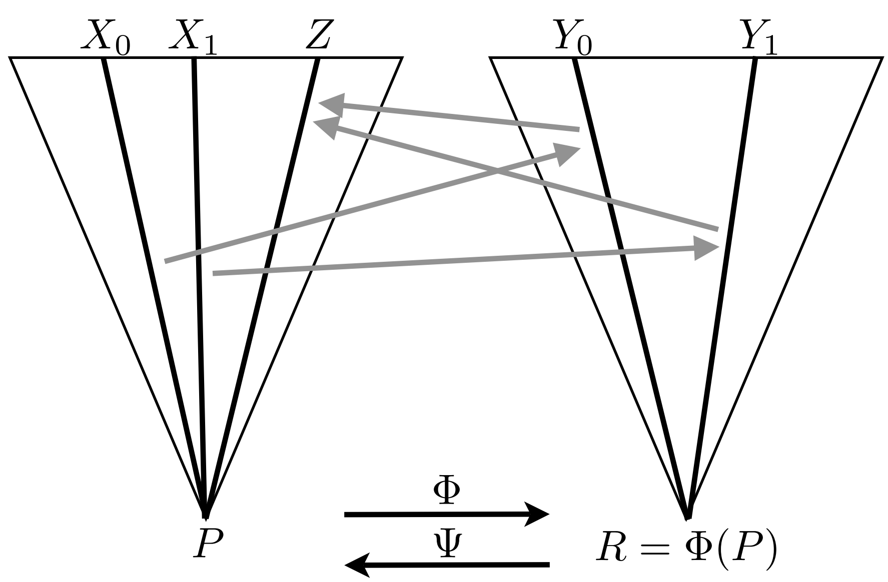

Figure 7: Theorem 4.3.3: $\Psi : R \to P$ injective

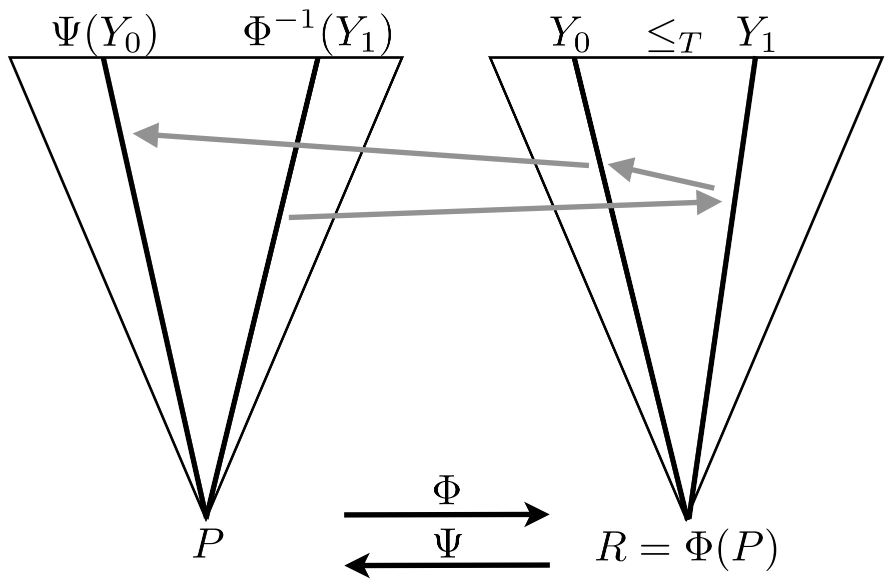

Figure 8: Theorem 4.3.3: R totally separable.

Thus $\Phi : P \to R$ is injective and thus bijective. See Figure 5.

Assume $\Psi : R \to P$ is not surjective. Fix $X \in P \setminus \Psi(R)$. Then $\Psi(\Phi(X)) \in \Psi(R)$ and thus not equal to X, but is Turing reducible from X, a contradiction. Thus $\Psi : R \to P$ is surjective. See Figure 6.

Assume $\Psi : R \to P$ is not injective. Fix Y_0 and Y_1 in R such that $\Psi(Y_0) = \Psi(Y_1) = Z$. As Φ is bijective there exist X_0, X_1 in P with $X_0 \neq X_1$, $Y_0 = \Phi(X_0)$, and $Y_1 = \Phi(X_1)$. Assume $Z \neq X_0$ (the other case is symmetric). Then $Z \leq_T X_0$ as $X_0 \mapsto Y_0 \mapsto Z$, a contradiction. Thus Ψ is injective and thus bijective. See Figure 7.

Assume R is not totally separable and fix Y_0 and Y_1 in R with $Y_0 \neq Y_1$ and $Y_0 \leq_T Y_1$. Let $X_0 = \Psi(X_0)$ and $X_1 = \Phi^{-1}(Y_1)$. Then $X_0 \leq_T X_1$ via $X_1 \mapsto Y_1 \mapsto Y_0 \mapsto X_0$, a contradiction. See Figure 8. Thus R is totally separable. □

Corollary 4.3.4. *If P and Q are totally separable with $P \equiv_M Q$, then P is computably*

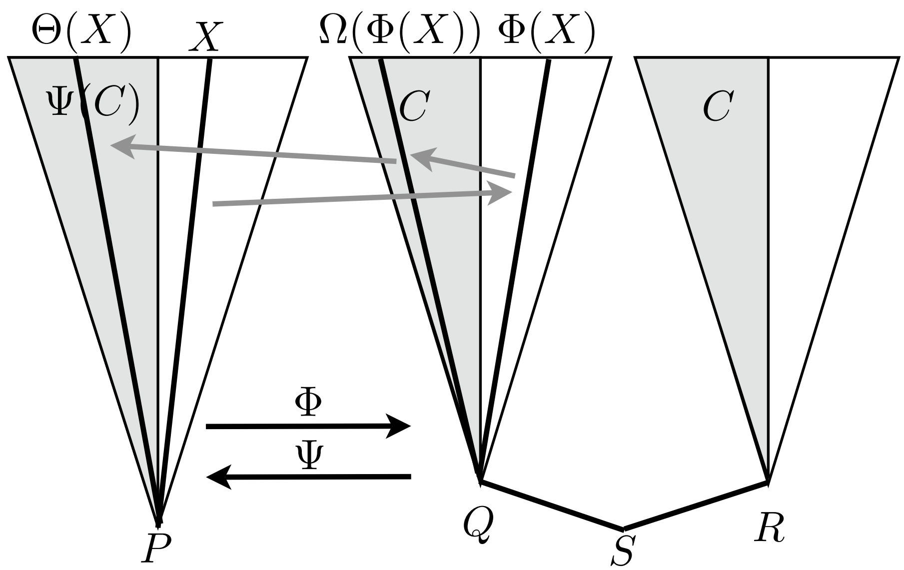

Figure 9: Theorem 4.4.1: $\Phi(X) \in C^c$

isomorphic to Q.

Proof. Repeat the proof of Theorem 4.3.3 with P in place of R. □

The following Lemma shows that in the situation of Theorem 4.3.3, P retracts onto its totally separable core.

Lemma 4.3.5. *If* Q *and* R *are such that* $R \subseteq Q$*,* R *is totally separable, and* $\Phi : Q \to R$ *is a computably continuous functional, then* $\Phi(X) = X$ *for all* $X \in R$*, i.e.,* Φ *is a retraction.*

Proof. If $\Phi(X) = Y \neq X$ for some X in R, then $Y \leq_T X$, a contradiction. □

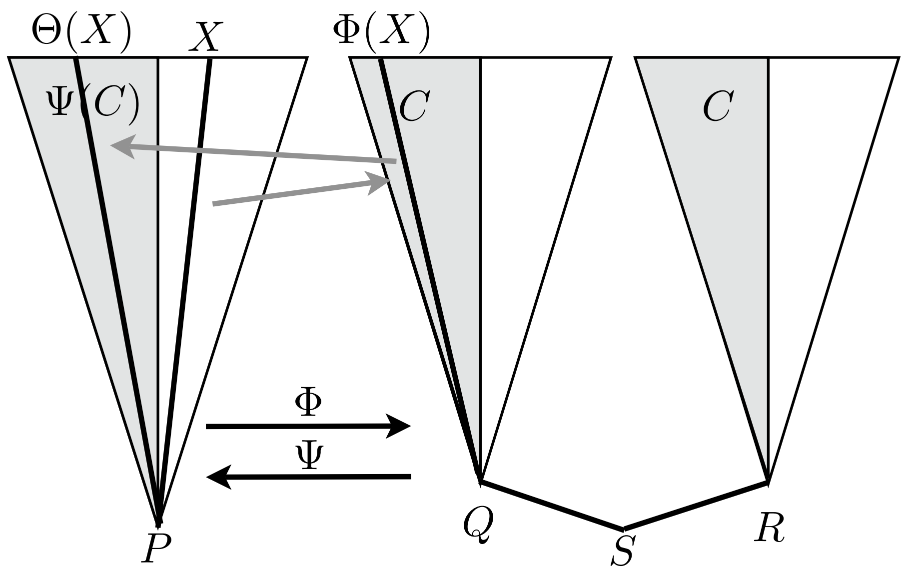

Figure 10: Theorem 4.4.1: $\Phi(X) \in C$

4.4 Separable and Not Hyperseparable

Theorem 4.4.1. *If Q and R are hyperinseparable with $Q \perp_M R$, then* $\deg(Q \wedge R)$ *is separable and not hyperseparable.*

Proof. Let $S = Q \wedge R$ and $\mathbf{a} = \deg(S)$. Then S is separable. Consider any $C_Q \subset I(0)$ good for $0^\frown Q$ and $C_R \subset I(1)$ good for $1^\frown R$. By hyperinseparability there is a reduction from $0^\frown Q \cap C_Q$ to $0^\frown Q \cap C_Q^c$ and similarly for C_R. Thus there is a reduction to $C = C_Q \cup C_R$ from C^c witnessing that S is not hyperseparable.

Let P be any class with $P \equiv_M S$. Let $\Phi : P \to S$ and $\Psi : S \to P$ witness $P \equiv_M S$. Let C_Q and C_R be clopen sets satisfying: $C_Q \subset I(0)$, $C_R \subset I(1)^\frown R$, C_Q good for $0^\frown Q$, C_R good for $1^\frown R$, and $\Psi(C_R \cup C_Q)$ good for P. The last requirement can be achieved by choosing C_R and C_Q small enough: fix $\sigma \in \mathcal{T}_R$ long enough such that there exists $\tau \in \mathcal{T}_S$ with $\phi(\sigma) \perp \tau$ and let $C_R = I(\sigma)$; similarly for C_Q.

Let $C_S = \Psi(C)$. Fix $\Omega : S \cap C^c \to S \cap C$. Define Θ by

$$\Theta(X) = \begin{cases} \Psi(\Phi(X)) & \text{if } \Phi(X) \in C, \\ \Psi(\Omega(\Phi(X))) & \text{if } \Phi(X) \in C^c. \end{cases} \tag{4.1}$$

See Figures 9 and 10. Then Θ witnesses $S \cap C^c \geq_M S \cap C$. So P is not hyperseparable. As P was arbitrary, $\deg(Q \wedge R)$ is not hyperseparable. □

The previous theorem provides a method for constructing separable and not hyperseparable degrees from hyperinseparable degrees. The following theorem of Binns can be used to construct homogeneous (and thus hyperinseparable degrees and thus separable and not hyperseparable) with various structure.

Lemma 4.4.2. *[2] Let A be a c.e. set and P a Π^0_1 class with $P > \mathbf{0}$. Then there exist c.e. sets A^0, A^1 such that*

$$A^0 \cap A^1 = \emptyset, \tag{4.2}$$

$$A^0 \cup A^1 = A, \tag{4.3}$$

$$\forall i \in \{0,1\} \forall f \in P[A^i \not\geq_T f]. \tag{4.4}$$

The idea is to construct a pair of hyperinseparable (actually homogeneous) degrees whose meet, by Theorem 4.4.1, is separable and not hyperseparable, but whose join is as high as we want it. We are also able to avoid a cone.

Corollary 4.4.3. *For any $\boldsymbol{b}, \boldsymbol{c} > 0$ with $\boldsymbol{b}$ homogeneous there exists $\boldsymbol{b}^0$, $\boldsymbol{b}^1$, and $\boldsymbol{a}$ such*

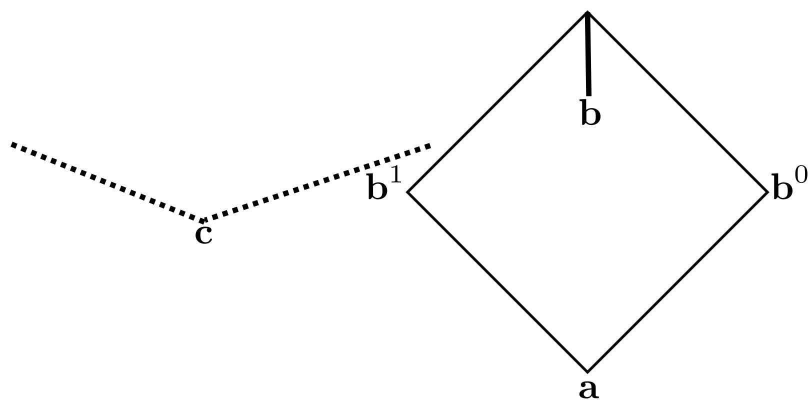

Figure 11: Theorem 4.4.3

that

$$\boldsymbol{a} = \boldsymbol{b}^0 \wedge \boldsymbol{b}^1, \tag{4.5}$$

$$\boldsymbol{a} \textit{ is separable and not hyperseparable}, \tag{4.6}$$

$$\boldsymbol{b}^0, \boldsymbol{b}^1, \boldsymbol{a} \not\geq \boldsymbol{c}, \tag{4.7}$$

$$\boldsymbol{b}^0 \vee \boldsymbol{b}^1 \geq \boldsymbol{b}. \tag{4.8}$$

Proof. Fix $S = \mathcal{S}(A, B) \in \mathbf{b}$ with A and B c.e. Fix $P \in \mathbf{c}$. Let A^0 and A^1 be as in Lemma 4.4.2. Let $S^0 = S(A^0, B)$ and $S^1 = S(A^1, B)$. Let $Q = S^0 \wedge S^1$. Note that homogeneity implies hyperinseparability. By Theorem 4.4.1, Q is separable and not hyperseparable. By the conditions on A^0 and A^1, $S^0 \not\geq_M P$ and $S^1 \not\geq P$, thus $Q \not\geq_M P$.

Finally, for $X \oplus Y \in S^0 \vee S^1$ define

$$Z(n) = \begin{cases} 0 & \text{if } X(n) = 0 \text{ or } Y(n) = 0, \\ 1 & \text{else.} \end{cases} \tag{4.9}$$

Then $Z \in S$, thus $S^0 \vee S^1 \geq_M S$. Letting $\mathbf{b}^0 = \deg(S^0)$, $\mathbf{b}^1 = \deg(S^1)$, and $\mathbf{a} = \deg(Q)$ we arrive at the result. This result is partially illustrated by Figure 11; note that $\mathbf{b}^0 \vee \mathbf{b}^1$ may or may not be in the cone above $\mathbf{c}$. $\square$

Corollary 4.4.4. *There exists a degree $\boldsymbol{a}$ such that $\boldsymbol{a}$ is separable and not hyperseparable.*

4.5 Hyperseparable and Not Totally Separable

Finally, we work to separate the notions of hyperseparable and not totally separable. The task is twofold: we must build a hyperseparable degree and avoid being totally separable.

The first task is complicated by the fact that, previously, the only known construction of a hyperseparable degree was to build a totally separable degree. We will use Theorem 4.3.3 to show that it is sufficient to build a class which is hyperseparable and not totally separable. We then use the methods of Chapter 3 to build such a class.

Theorem 4.5.1. *If P is hyperseparable and not totally separable, then* $\deg(P)$ *is hyperseparable and not totally separable.*

Proof. That $\deg(P)$ is hyperseparable is immediate. Assume $\deg(P)$ is totally separable. Then, by Theorem 4.3.3, there exists $R \subseteq P$, $R \equiv_M P$, and R totally separable. As P is not totally separable, $R \neq P$. Let C be a clopen set such that $R \subseteq P \cap C \subset P$.

Using Lemma 1.2.3 twice, $P \cap C \leq_M R \equiv_M P \leq_M P \cap C^c$, contradicting that P is hyperseparable. Thus, $\deg(P)$ is not totally separable. □

Theorem 4.5.2. *There exists a degree which is hyperseparable and not totally separable.*

Proof. We will build a computable tree life $\{L_s\}$. By Corollary 3.1.8, $P = \lim_s L_s$ will be a Π^0_1 class. We will build P to be hyperseparable and not totally separable. By Theorem 4.5.1, $\deg(P)$ will be hyperseparable and not totally separable.

Let $\langle C, \phi \rangle$ be an enumeration of all pairs of clopen subclasses of 2^ω and partial computable functions. For convenience we often refer to such pairs by their index, e, in the enumeration. We also start the enumeration at $e = 1$. We will blur the distinction between e and $\langle C, \phi \rangle$. For each $\langle C, \phi \rangle$ we work to satisfy the requirement:

$$\mathcal{R}_{C,\phi} : C \text{ good for } P \Rightarrow \exists X \in P \cap C[\Phi(X) \notin P \cap C^c]. \tag{4.10}$$

To ensure that P is not totally separable we will use a very simple reduction and ensure that paths Turing equivalent through that reduction exist. Namely,

$$\mathcal{S} : \exists X, Y \in P \exists Z \in 2^\omega[X = 0^\frown Z \text{ and } Y = 1^\frown Z]. \tag{4.11}$$

We have a strategy acting on behalf of each $\mathcal{R}_e$ which will be careful to ensure that $\mathcal{S}$ is satisfied. Strategies are ordered in priority in the order of the enumeration with earlier strategies having higher priority. Each node has a protection level. The function $r_s : 2^{<\omega} \to \omega \cup \{\omega\}$ indicates the protection level, i.e., the protection level of σ at stage s is $r_s(\sigma)$. Lower numbers indicate higher protection levels. A strategy may protect a node with its own priority. Each strategy has two states: *wait* and *stop*. Strategies begin in state *wait* and may at some point act and enter state *stop*. Once in state *stop*, a strategy will not act unless injured. When a state acts, it injures all lower priority strategies resetting them to state *wait*. The construction thus progresses in typical finite injury fashion. Denote the state of strategy e at stage s by $\text{state}_s(e)$.

In order for all strategies to be able to find witnesses to kill, they must obey a simple rule regarding protection levels. For a node σ protected at level d, strategy e ($e > d$) may only kill $\tau \succeq \sigma$ if $|\tau| \geq |\sigma| + 2(e - d)$. As we shall see in the claims below, this will ensure that every strategy is able to kill a witness if needed.

Let

$$S_s = \{\sigma : 0^\frown\sigma \in D(L_s) \text{ and } 1^\frown\sigma \in D(L_s)\}. \tag{4.12}$$

To ensure $\mathcal{S}$ is satisfied we require all strategies to preserve $S_s \neq \emptyset$ and only trim the tree at even stages, growing it with single extensions an odd stages. This will ensure that S_s is never empty and every requirement can act if necessary.

Begin with $L_0 = \{\emptyset, 0, 1\}$, $r_0(\sigma) = \infty$ for all σ, and $\text{state}_0(e) = \textit{wait}$ for all e.

Assume we have run the construction up to stage s. Thus L_{s-1}, r_{s-1}, and state_{s-1} are all defined. A strategy $e = \langle C, \phi \rangle$ is eligible to act if $\text{state}_{s-1}(e) = \textit{wait}$ and

$$\begin{aligned}\exists\sigma \in L_{s-1} \cap C[\phi(\sigma) \in L_{s-1} \cap C^c \\ \text{and } \forall\tau \preceq \phi(\sigma)[r_{s-1}(\tau) > e \text{ or } |\phi(\sigma)| \geq |\tau| + 2(e - r_{s-1}(\tau))] \\ \text{and } \exists\nu \in S_{s-1}[0^\frown\nu \not\preceq \phi(\sigma) \text{ and } 1^\frown\nu \not\preceq \phi(\sigma)]].\end{aligned} \tag{4.13}$$

If s is odd or if no such e exists, then let $L_s = \hat{L}_{s-1}$, $r_s = r_{s-1}$ and $\text{state}_s = \text{state}_{s-1}$. Otherwise let e be the highest priority (least index) strategy eligible to act. Let $\sigma' \in$

$D(L_s)$ be a child of σ (possibly equal to σ). Let

$$L_s = \text{trim}(\widehat{L_{s-1}, \phi}(\sigma)), \tag{4.14}$$

$$r_s(\tau) = \begin{cases} e & \tau = \sigma', \\ r_{s-1}(\tau) & r_{s-1}(\tau) < e, \\ \omega & \text{else}, \end{cases} \tag{4.15}$$

$$\text{state}_s(n) = \begin{cases} \textit{stop} & n = e, \\ \textit{wait} & n > e, \\ \text{state}_{s-1}(n) & \text{else}. \end{cases} \tag{4.16}$$

Equation (4.14) describes the evolution of the tree life. Equations (4.15) and (4.16) serve to protect σ, stop strategy e, and injure (reset) all lower priority strategies. Observe that $S_s \neq \emptyset$ as $0^\frown\nu$ and $1^\frown\nu$ were not killed for ν as in (4.13).

This completes the construction. We now prove that the result has the desired properties.

Claim. *Fix any d and e with $d < e$. If strategy d is not injured after stage t, then strategy e will be injured less than or equal to 2^{e-d-1} times after stage t.*

Proof. Fix d and t and let $I(e)$ denote the maximum number of times e could be injured after stage t. We will show by induction that $I(e) \leq 2^{e-d-1}$.

Consider $e = d+1$. Then d will be injured if d acts after stage t. As d is not injured after stage t it will act at most once and thus $I(e) = 1 \leq 2^{e-d-1} = 2^{d+1-d-1} = 2^0 = 1$.

Assume $I(e') \leq 2^{e-d-1}$ for all $d < e' < e$. Any time a strategy below $e-1$ is injured, $e-1$ is also injured. Thus $I(e-1)$ is an accurate count of the number of times e might be injured by strategies $< e-1$. Each time $e-1$ is injured, e is injured. In addition, $e-1$ may act once before being injured again, injuring e as well. Thus $I(e) = 2I(e-1) = 2(2^{e-1-d-1}) = 2^{e-1-d-1+1} = 2^{e-d-1}$. □

Claim. *Fix any d and e with $d < e$. If strategy d is not injured after stage t, then strategy e will act less than 2^{e-d} times after stage t.*

Proof. Strategy e can only act once before being injured again. Thus the total number of times it can act is equal to the number of times it is injured plus one. By the previous claim this is less than or equal to $2^{e-d-1}+1$ which is less than 2^{e-d}. □

Claim. *For all σ and s such that $r_s(\sigma) = e < \omega$ and strategy e is not injured at or after stage s, $\sigma \in \lim_s L_s$.*

Proof. As strategy e is not injured, no strategy of higher priority will kill any ancestor of σ, so our only worry is that lower priority strategies will kill all the children of σ. When σ was protected, it was a leaf node. Thus any strategy which kills ancestors of σ must obey the protection. Namely, for $d > e$, d can only kill $\tau \succeq \sigma$ if $|\tau| \geq |\sigma| + 2(d-e)$.

Let μ be the standard measure on 2^ω, i.e., $\mu(I(\tau)) = 2^{-|\tau|}$. For a finite L_s, define $\mu(I(\sigma) \cap L_s)$ to be $\mu(I(\sigma) \cap \bigcup_{\tau \in D(L_s)} I(\tau))$, that is, we presume that L_s will have all possible children. We will show that, for all $t > s$, $\mu(I(\sigma) \cap L_t) > 0$ and, thus, $\sigma \in L_t$.

Fix a stage t and let d be the lowest priority (highest index) strategy to act so far. For each $e < f \leq d$ let N_f be the number of children of σ strategy f has killed and $\{\tau_{i,f}\}$ be the set of these children. Strategy f can kill only a single child when it acts, so by the previous claim, $N_f < 2^{f-e}$. The requirement on the length of τ requires that

$I(\tau_{i,f}) \leq 2^{-(|\sigma|+2(d-e))}$.

$$\mu(I(\sigma) \cap L_t) = \mu(I(\sigma)) - \sum_{e<f\leq d}\sum_{i<N_f} \mu(I(\tau_{i,f})) \tag{4.17}$$

$$\geq 2^{-|\sigma|} - \sum_{e<f\leq d} N_f 2^{-(|\sigma|+2(f-e))} \tag{4.18}$$

$$> 2^{-|\sigma|} - \sum_{e<f\leq d} 2^{f-e}2^{-(|\sigma|+2(f-e))} \tag{4.19}$$

$$= 2^{-|\sigma|} - \sum_{i=1}^{d-e} 2^i 2^{-(|\sigma|+2i)} \tag{4.20}$$

$$= 2^{-|\sigma|} - \sum_{i=1}^{d-e} 2^{-|\sigma|-i} \tag{4.21}$$

$$= 2^{-|\sigma|}\Big(1 - \sum_{i=1}^{d-e} 2^{-i}\Big) \tag{4.22}$$

$$> 0. \tag{4.23}$$

□

Claim. $\{L_s\}$ *is a computable tree life.*

Proof. The previous claim shows that, at all stages, L_s is nonempty. By Lemmas 3.2.3 and 3.2.4, each L_s is a growth of L_{s-1}. As single extensions and trims are computable, it is a computable tree life. □

Thus, by Corollary 3.1.8, $P = [\lim_s L_s]$ is a non-empty Π^0_1 class.

Claim. $\forall n \exists s \forall t > s[|S_t| > n]$.

Proof. Define a *clump* in S_t to be a proper subset $U \subset S_t$ maximal with respect to $U = \{\sigma^\frown\tau : \tau \in 2^i\}$ for some σ and i. Let c_t be the size of the smallest clump in S_t.

First we show that $c_{t+1} \geq c_t$ for all t. At each stage something may be killed and then every living leaf is extended, i.e., each stage is a composition of (possibly) a trim and a single extension. If $c_t = 0$, then the result follows immediately so assume $c_t > 0$. There are four possibilities:

1. Nothing in S_t is killed. Then the clump doubles and $c_{t+1} = 2c_t$.

2. The smallest clump is killed. As it was not everything there is another clump of at least equal size. That clump will double, but it could now be everything in which case it is not a clump but rather two clumps of size equal to the original. So $c_{t+1} \geq c_t$.

3. Everything but the smallest clump is killed. Then the smallest clump will double but as it is now everything it is now two clumps rather than one. So $c_{t+1} = c_t$.

4. Part of a clump is killed. At worst it will kill half the smallest clump. The other half will then double and $c_{t+1} = c_t$.

At odd stages, case (1) occurs, so $c_{t+1} > c_t$ for t odd. Thus c_t is unbounded in t and S_t is unbounded in t. □

Observe that while clumps of arbitrary finite size exist during the construction they may move around. The final Π^0_1 class may be very non-clumpy.

Claim. *For all e, $\mathcal{R}_e$ is satisfied.*

Proof. By a previous claim, let s be sufficiently large such that strategy e is not injured at or after stage s. Let $e = \langle C, \phi \rangle$. If C is not good for P, then we are done. Assume C is good for P.

Assume there exists $X \in P \cap C$ such that $\Phi(X) \in P \cap C^c$. For stages $t > s$, the set of nodes protected by strategies $d < e$ will stay fixed. Thus there is a stage t and an n such that $\sigma = X \upharpoonright n \in L_t$, $|\phi(\sigma)| \geq |\tau| + 2(e - r_{t-1}(\tau))$ for all $\tau \preceq \phi(\sigma)$ with $r_{t-1}(\tau) < e$. Strategy e may still not be able to act because of the requirement to preserve S_t. As higher priority strategies will not act again the strategy will continue to be otherwise eligible to act at later stages. Let Y be such that $X = i \frown Y$ for some $i \in 2$. If we never act that means that at each stage t, $S_t = \{Y \upharpoonright l(t)\}$, contradicting the previous claim that $|S_t|$ is unbounded. □

Claim. $\mathcal{S}$ *is satisfied.*

Proof. By the definition of S_s and that L_s is a tree life, if a string σ leaves S_s, i.e., $\sigma \in S_s \setminus S_{s+1}$, then no child of it can ever enter S_s later. Thus, using Claim 4.5, $\lim_s S_s$ exists and is non-empty. Then, for any Z in $\lim_s S_s$, $X = 0 \frown Z$ and $Y = 1 \frown Z$ serve as witnesses that $\mathcal{S}$ is satisfied. □

Thus P is hyperseparable as for any C good for P and any Φ, $\mathcal{R}_{\langle \phi, C \rangle}$ shows that Φ is not a witness of $P \cap C \geq_M P \cap C^c$. As $\mathcal{S}$ is satisfied, P contains a pair of comparable paths, namely $0 \frown X$ and $1 \frown X$ for some X. □

Bibliography

[1] Christopher P. Alfeld, *Non-branching degrees in the Medvedev lattice of Π^0_1 classes*, The Journal of Symbolic Logic **72** (2007), 81–97.

[2] Stephen Binns, *A splitting theorem for the Medvedev and Muchnik lattices*, Mathematical Logic Quarterly **49** (2003), no. 4, 327–335. MR MR1987431 (2004f:03079)

[3] Douglas Cenzer and Peter G. Hinman, *Density of the Medvedev lattice of Π^0_1 classes*, Archive for Mathematical Logic **42** (2003), no. 6, 583–600. MR MR2001061 (2004k:03081)

[4] Douglas Cenzer and Carl G. Jockusch, Jr., *Π^0_1 classes—structure and applications*, Computability theory and its applications (Boulder, CO, 1999), Contempory Mathematics, vol. 257, American Mathematics Society, Providence, RI, 2000, pp. 39–59. MR MR1770733 (2001h:03074)

[5] Douglas Cenzer and Jeffrey B. Remmel, *Π^0_1 classes in mathematics*, Handbook of recursive mathematics, Vol. 2, Studies in Logic and the Foundations of Mathematics, vol. 139, North-Holland, Amsterdam, 1998, pp. 623–821. MR MR1673586 (2001d:03108)

[6] Carl G. Jockusch, Jr. and Robert I. Soare, *Degrees of members of Π^0_1 classes*, Pacific J. Math. **40** (1972), 605–616. MR MR0309722 (46 #8827)

[7] ———, *Π^0_1 classes and degrees of theories*, Trans. Amer. Math. Soc. **173** (1972), 33–56. MR MR0316227 (47 #4775)

[8] Hartley Rogers, Jr., *Theory of recursive functions and effective computability*, MIT Press, Cambridge, MA, 1987. MR MR886890 (88b:03059)

[9] Stephen G. Simpson, *Mass problems and randomness*, The Bulletin of Symbolic Logic **11** (2005), no. 1, 1–27. MR MR2125147

[10] ______, Π^0_1 *sets and models of* WKL_0, Reverse Mathematics 2001 (To Appear), Lecture Notes in Logic, AK Peters, Ltd., 2005.

[11] ______, *An extension of the recursively enumerable Turing degrees*, Journal of London Mathematical Society (2006), 14, Accepted for publication 4 July 2006.

[12] Robert I. Soare, *Recursively enumerable sets and degrees*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin, 1987. MR MR882921 (88m:03003)