Decidability and Definability in the Σ_2^0-Enumeration Degrees

By

Thomas F. Kent

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

(Mathematics)

at the

UNIVERSITY OF WISCONSIN – MADISON

2005

Abstract

Enumeration reducibility was introduced by Friedberg and Rogers in 1959 as a positive reducibility between sets. The enumeration degrees provide a wider context in which to view the Turing degrees by allowing us to use any set as an oracle instead of just total functions. However, in spite of the fact that there are several applications of enumeration reducibility in computable mathematics, until recently relatively little research had been done in this area.

In Chapter 2 of my thesis, I show that the $\forall\exists\forall$-fragment of the first order theory of the Σ^0_2-enumeration degrees is undecidable. I then show how this result actually demonstrates that the $\forall\exists\forall$-theory of any substructure of the enumeration degrees which contains the Δ^0_2-degrees is undecidable. In Chapter 3, I present current research that Andrea Sorbi and I are engaged in, in regards to classifying properties of non-splitting Σ^0_2-degrees. In particular I give proofs that there is a properly Σ^0_2-enumeration degree and that every Δ^0_2-enumeration degree bounds a non-splitting Δ^0_2-degree.

Advisor: Prof. Steffen Lempp

Acknowledgements

I am grateful to Steffen Lempp, my thesis advisor, for all the time, effort, and patience that he put in on my behalf. His insight and suggestions have been of great worth to me, both in and out of my research. I am especially grateful for his help in getting me back in school after my two-year leave of absence and for offering me a research assistantship so I could study for a year with him in Germany.

I am also grateful to Andrea Sorbi for funding a visit to Siena, Italy that allowed me to do research with him, and for the friendship that has grown from our research contact. Hopefully we will be able to go running together in the mountains again.

I would like to thank Todd Hammond for introducing me to mathematical logic, to Mirna Dzamonja for getting me excited about Computability Theory, and to Jerome Keisler, Ken Kunen, Arnie Miller, and Patrick Speissegger for teaching interesting logic classes. I would like to thank all of the wonderful teachers over the years who have encouraged my interest in mathematics, especially Patty Av3ery and Slade Skipper. Thanks also go to Eric Bach, Joel Robbin, and Mary Ellen Rudin for help they have given and for serving on my defense committee.

I am very appreciative for my parents and sister, for the support and love they have given me over the past 31 years.

The most appreciation, however, goes to my wonderful wife, Joy, for always being there for me. I could not have made it without her encouragement and unconditional love.

Contents

Chapter 1

Introduction

1.1 Background

Enumeration reducibility was first introduced by Friedberg and Rogers [FR59] in 1959. Since this time, there has been a steady increase in the interest and study of enumeration degrees. Informally, a set A is enumeration reducible to a set B, written $A \leq_e B$, if there is an effective procedure to enumerate A given any enumeration of B. At first, this definition seems to be somewhat weaker than that of Turing reducibility, where a set A is Turing reducible to a set B ($A \leq_T B$) if there is an effective procedure to decide the characteristic function of A given the characteristic function of B. On the other hand, enumeration reducibility can be viewed as an extension of Turing reducibility in the following manner. First, for any function φ, define

$$\text{graph}(\varphi) = \{\langle x, y\rangle : \varphi(x) = y\}.$$

Then, for total functions f and g, it is immediate that $f \leq_T g$ if and only if $\text{graph}(f) \leq_e \text{graph}(g)$. While Turing reducibility is restricted to total functions, there is no such constraint on enumeration reducibility as the functions f and g are allowed to range over partial functions as well. From this, we may view enumeration reducibility as providing a wider context than Turing reducibility. In fact, the

Turing degrees are isomorphic to a substructure of the enumeration degrees called the total degrees. A total degree is an enumeration degree that contains a total function.

The restriction of enumeration reducibility to partial functions coincides with Kleene's [Kle52] definition of reducibility between partial functions, and they both give rise to what is called the partial degrees. By allowing enumeration reducibility to range over all subsets of the natural numbers (and not just partial functions), the induced degree structure does not change since all sets A are enumeration equivalent to a partial function, namely $\{\langle x, 1\rangle : x \in A\}$. Thus, the partial degrees are isomorphic to the enumeration degrees, even though we allow the oracle of our computation to be any set instead of restricting it to partial functions.

Aside from providing a wider context in which to view the Turing degrees and an alternate formulation for the partial degrees, several other natural uses of the enumeration reducibility have been found in other areas of computable mathematics. For example, Ash, Knight, Manasse, and Slaman [AKMS89] used it for analysis of types in effective model theory, and Ziegler [Zie80] (cf. [HS88]) used used a variant of enumeration reducibility in the study of existentially closed groups. More recently, in computable analysis, while examining reducibilities between continuous functions, Miller [Mil04] introduced the continuous degrees and showed that these degrees may be viewed as a proper substructure of the enumeration degrees which properly contains the total degrees. Another application comes from Feferman's Theorem [Fef57] which states that every truth table degree contains a first order theory. Case [Cas71] has pointed out that since the truth table reduction used in the proof is essentially enumeration reducibility, and that a theory is axiomatizable if and only if it is effectively enumerable (Craig [Cra53]), the enumeration degrees may be thought of as degrees of nonaxiomatizability. Lastly, Scott [Sco75] and Cooper [Coo90] have shown how enumeration operators can be used to provide a countable version of the graph model for λ-calculus.

Comprehensive summaries of additional results in, and uses of, the enumeration degrees can be found in [Coo90] and [Sor97]. In the following sections of this chapter, we highlight the main definitions and theorems that are pertinent to the main results of this thesis.

1.2 Enumeration Reducibility

Intuitively, we say that a set A is enumeration reducible to a set B if there is an effective procedure to enumerate A given any enumeration of B. More formally, given a computably enumerable functional Φ, we define

$$\Phi^B = \{x : \exists \langle x, F\rangle \in \Phi \ \&\ F \subseteq B \ \&\ F \text{ is finite}\}$$

where we identify the finite set F with a natural number (its canonical index) and $\langle \cdot, \cdot \rangle$ is a computable bijection from pairs of natural numbers to natural numbers.

We say that A is *enumeration reducible* to B, $A \leq_e B$, if there is a computably enumerable functional Φ such that $A = \Phi^B$. The relation $\leq_e$ is a pre-order on the powerset of natural numbers and, as such, generates an equivalence relation, denoted $\equiv_e$, on the powerset of the natural numbers. By $\deg_e(A)$, we denote the equivalence class, or degree, of the set A. The least enumeration degree, $\mathbf{0}_e$, is the set of c.e. sets since trivially, $A \leq_e \emptyset$ for every c.e. set A. The enumeration degrees form an upper semi-lattice where we define $\mathbf{a} \vee \mathbf{b} = \deg_e(A \oplus B)$ with $A \in \mathbf{a}$ and

$B \in \mathbf{b}$. Case [Cas71] proved that there are pairs of enumeration degrees that do not have a meet by showing that every countable non-principal ideal has an exact pair. An exact pair for an ideal $\mathcal{I}$ is a pair of incomparable degrees $\mathbf{a}$ and $\mathbf{b}$ such that $\mathbf{x} \in \mathcal{I}$ if and only if $\mathbf{x} \leq \mathbf{a}$ and $\mathbf{x} \leq \mathbf{b}$. Thus, if $\mathbf{a}$ and $\mathbf{b}$ have a meet, say $\mathbf{m}$, then $\mathbf{m}$ must be the greatest element of $\mathcal{I}$, making $\mathcal{I}$ principal.

Rogers [Rog67] defined a computable embedding of the Turing degrees into the enumeration degrees via the function $\iota : \iota(\deg_T(A)) \mapsto \deg_e(\chi(A))$, where $\chi(A)$ is the characteristic function of A. If $A \leq_T B$, it follows that $\chi(A) \leq_e \chi(B)$. This is because, by definition, a Turing operator has access to the characteristic function of a set, while an enumeration operator only has access to the members of the set. By replacing a set with its characteristic function, all of the negative information (e.g. $x \notin A$) that the Turing functional has access to has been replaced by positive information (e.g. $A(x) = 0$). We call the image of the Turing degrees under this embedding the total degrees since every degree in the range of ι contains a total function, namely $\chi(A)$ for some A, and all total degrees are in the range of ι.

At this point, it is useful to note that $\chi(A) \equiv_e A \oplus \overline{A}$, and so from now on we will use the convention that $\iota(A) = A \oplus \overline{A}$ (and so $\deg_e(\iota(A))$ is a total degree). One way in which this fact is useful is that we can use it to show that the c.e. Turing degrees are isomorphic to the Π^0_1-enumeration degrees. Let A be a c.e. set. Then $\iota(A) = A \oplus \overline{A} \equiv_e \overline{A}$, a Π^0_1-set. This also shows us that $\deg_e(\overline{K})$ is the greatest Π^0_1-degree.

McEvoy [McE85] defined a jump operation on the enumeration degrees that was later expanded by Cooper [Coo84]. For every set A, define $K_A = \left\{x : x \in \Phi_x^A\right\}$ where Φ_x is the x^{th} enumeration operator in some fixed computable ordering. We then define $A' = K_A \oplus \overline{K_A}$. (The reason that we do not define $A' = K_A$ as in the Turing degrees is that $K_A \equiv_e A$. Also, we do not define $A' = \overline{K_A}$ since it is not always the case that $\overline{K_A} \geq_e A$.) The enumeration jump has the same properties as the Turing Jump: $A \leq_e B \Rightarrow A' \leq_e B'$ and $A <_e J(A)$. Another useful property of the enumeration jump is that it commutes with ι, i.e. $\iota(\mathbf{a}') = \iota(\mathbf{a})'$. A corollary of this is that the jump of every enumeration degree is a total degree.

In 1955, Medvedev [Med55] showed the existence of a quasi-minimal degree, a non-total degree with dze as the only total degree less than it, proving that the total degrees are a proper substructure of the enumeration degrees. In 1971, Gutteridge [Gut71] extended this result by proving that there are no minimal enumeration degrees, thus proving that the Turing degrees and enumeration degrees have distinct elementary theories.

Gutteridge's result, while showing the enumeration degrees are downwards dense, left open the question of whether the entire structure is dense. Cooper [Coo84] (see also [LS92]) proved that the degrees below $\mathbf{0}_e'$ are dense and later proved that the degrees below $\mathbf{0}_e^{(6)}$ are not dense [Coo90]. Finally, Slaman and Woodin [SW97] proved that the degrees below $\mathbf{0}_e''$ are not dense by constructing a pair of properly Π^0_2-degrees $\mathbf{a}$ and $\mathbf{b}$ such that $\mathbf{b}$ is a minimal cover over $\mathbf{a}$. This result is the best possible since the degrees below $\mathbf{0}_e'$ coincide with the Σ^0_2-enumeration degrees, and every Σ^0_2-enumeration degree contains only Σ^0_2-sets [Coo84].

1.3 The Σ^0_2-Enumeration Degrees

After proving that the Σ^0_2-enumeration degrees are dense and form an ideal below $\mathbf{0}'_e$, Cooper noted that these properties are similar to those of the c.e. Turing degrees (which form a dense ideal below $\mathbf{0}'$) and asked if these two degree structures were elementarily equivalent. In her thesis, Ahmad [Ahm89] showed that this is not the case by proving that the diamond lattice embeds into the Σ^0_1-enumeration degrees preserving 0 and 1 (cf. [Ahm91]) and that there are non-splitting Σ^0_2-enumeration degrees (cf. [AL98]). These results stand in sharp contrast to Lachlan's [Lac66] Non-Diamond Theorem, and Sacks' [Sac63] Splitting Theorem for the c.e. Turing degrees.

Lachlan's Non-Diamond Theorem states that the diamond lattice cannot be embedded into the computably enumerable (c.e.) Turing degrees preserving 0 and 1, i.e. if two c.e. degrees nontrivially join to $\mathbf{0}'$ then there is a non-zero degree that lies below both of them. When a lattice is embedded into another partial order preserving 0 and 1, the lattice is embedded so that all meets and joins are preserved, and the least and greatest elements of the lattice are mapped respectively to the least and greatest elements of the partial order. To date, there is no complete classification of what finite lattices can be embedded into the c.e. Turing degrees preserving 0 and 1. However, by extending Ahmad's Diamond Theorem, Lempp and Sorbi [LS02] proved that every finite lattice is embeddable into the Σ^0_2-enumeration degrees preserving 0 and 1.

A non-splitting degree is a degree that is not the non-trivial join of two lesser degrees. Sacks' Splitting Theorem states that any non-trivial c.e. Turing degree is the join of two incomparable c.e. degrees. In the Δ_2^0-Turing degrees, any minimal degree is trivially non-splitting. However, Ahmad's non-splitting result is interesting since she constructed a non-splitting degree in a dense partial order. What is even more interesting is that using non-splitting degrees, we can construct what is known as an Ahmad pair [Ahm89] (cf. [AL98]). An Ahmad pair consists of two incomparable Σ_2^0-enumeration degrees $\mathbf{a}$ and $\mathbf{b}$ such that if $\mathbf{x} < \mathbf{a}$ then $\mathbf{x} < \mathbf{b}$. This condition implies that $\mathbf{a}$ must be non-splitting.

The density of the Σ_2^0-enumeration degrees, the classification of which finite lattices embed into the Σ_2^0-enumeration degrees preserving 0 and 1, and the existence of Ahmad pairs are very important results in determining the decidability of the $\forall\exists$-theory of the Σ_2^0-enumeration degrees, which is still an open question.

1.4 Decidability

Questions dealing with the decidability of theories have been of primary interest to computability theorists. The main goal of these questions is to determine if the theory for some fixed algebraic structure is decidable and, if not, at what level of quantifier alternations does undecidability occur. For example, Lerman and Shore [LS88] demonstrated that the $\forall\exists$-theory in the language of reducibility of the Δ_2^0-Turing degrees is decidable, and Schmerl (cf. [Ler83]) has shown that the $\forall\exists\forall$-theory is undecidable. In the c.e. Turing degrees, Lempp, Nies, and Slaman [LNS98] have shown that the $\forall\exists\forall$-theory in the language of reducibility is undecidable, while it is still an open question as to whether the $\forall\exists$-theory is decidable or not.

Similar questions are being investigated regarding the enumeration degrees and several of its substructures. The usual technique to show that a theory of a particular algebraic structure is undecidable is to embed, in a uniform manner, another class of algebraic structures, which is known to have a hereditarily undecidable theory, into the structure in question. By using this technique, Slaman and Woodin [SW97] were able to embed all finite graphs into the Σ^0_2-enumeration degrees in such a way that the first order theory of finite graphs was then interpretable in the theory of the Σ^0_2-enumeration degrees. Since the theory of finite graphs is hereditarily undecidable (cf. [Nie96]), this implies that the first order theory of the Σ^0_2-enumeration degrees is undecidable. (Actually, as we will show in Chapter 2, a little more work shows that they proved that the $\forall\exists\forall\exists\forall$-fragment of the first order theory is undecidable.)

The usual technique to show that a theory fragment is decidable is to reduce sentences in the theory fragment to questions that are algebraic in nature. For example, since the Π^0_1-enumeration degrees are computably isomorphic to the c.e. Turing degrees, any theorem in the c.e. Turing degrees is true in the Π^0_1-enumeration degrees. Furthermore, since any finite partial order can be embedded into the c.e. Turing degrees, it follows that any finite partial order can be embedded into the Σ^0_2-enumeration degrees. (The same result could be obtained by applying the lattice embedding theorem of Lempp and Sorbi [LS02].) A $\exists$-sentence describes a finite partial order. Thus, a $\exists$-sentence is true if and only if it describes a consistent partial order.

In Chapter 2, we improve on the result of Slaman and Woodin by showing that the $\forall\exists\forall$-theory of the Σ^0_2-enumeration degrees is undecidable by showing that every finite bi-partite graph can be effectively embedded into this structure. This leaves open the question as to whether the $\forall\exists$-theory is decidable. The construction is performed in such a way that it is also shown that the $\forall\exists\forall$-theory of any substructure of the enumeration degrees which contains the Δ^0_2-enumeration degrees is undecidable.

The decidability of the $\forall\exists$-theory of a partial order can be rephrased in purely algebraic terms as follows.

1.4.1 Question. Is it possible to effectively decide if, given finite posets $\mathcal{P} \subseteq \mathcal{Q}_0, \ldots, \mathcal{Q}_n$ for some $n \geq 0$, any embedding of $\mathcal{P}$ into the Σ^0_2-enumeration degrees can be extended to the embedding of some $\mathcal{Q}_i$? (The choice of i may depend on the embedding of $\mathcal{P}$.)

Lempp, Slaman, and Sorbi [LSS] solved a major subproblem of this question known as the Extension of Embeddings problem. The Extension of Embeddings problem is the same as above, only setting $n = 0$. Their proof relies heavily on the facts that we outlined in 1.3 (i.e. density, Ahmad pairs, and lattice embeddings).

In chapter 3, we present an overview of the current research that the author is engaged in with Andrea Sorbi in gaining a better understanding of the algebraic properties of non-splitting degrees. It is hoped that a better understanding of these properties will help us in our efforts to determine if the $\forall\exists$-theory fragment is decidable. In chapter 3, a direct construction of a non-splitting degree on a tree of strategies is presented, and then the construction is modified in several ways to show the existence of non-splitting properly Σ^0_2-degrees, low non-splitting degrees, and that every non-trivial Δ^0_2degree bounds a non-splitting degree.

The last result is interesting since it shows that the embedding ι maps every non-trivial principal generated by a c.e. Turing degree an ideal in the enumeration degrees whose elementary theory is different.

1.5 Technical Details and Definitions

We recall that a Σ^0_2-approximation $\langle B_s \rangle_{s\in\omega}$ to a set B is a computable sequence of computable sets such that $x \in B \Rightarrow \lim_s B_s(x) = 1$.

1.5.1 Definition. Given a Σ^0_2-approximation $\langle B_s \rangle_{s\in\omega}$, we say that the stage s is *thin* if $B_s \subseteq B$, and we say that the approximation is *good* if it contains infinitely many thin stages.

1.5.2 Lemma. Given a computable Σ^0_2-approximation $\langle B_s \rangle_{s\in\omega}$ to a set B, there is a good Σ^0_2-approximation $\langle B'_s \rangle_{s\in\omega}$, uniform in the index of $\langle B_s \rangle_{s\in\omega}$, such that B'_s is finite for all s.

Proof. See Lachlan and Shore [LS92]. $\square$

Throughout this paper we always assume that all Σ^0_2-approximations are good, as guaranteed by the lemma.

1.5.3 Notation. When we refer to the *least* finite set with a certain property we are referring to the finite set with least canonical index that has the specified property.

1.5.4 Definition. Given a Σ^0_2-approximation $\langle X_s \rangle_{s\in\omega}$ to a set X, an element x, and a stage s, we define $a(X; x, s)$, the *age* of x in X at stage s, to be the least stage $s_x \leq s+1$ such that for all stages t, if $s_x \leq t \leq s$ then $x \in X_t$. If Z is a finite set, then we define $a(X; Z, s)$, the *age* of Z in X at the stage s, to be $\max\{a(X; z, s) : z \in Z\}$. Given Σ^0_2-approximations $\langle X_s \rangle_{s\in\omega}$ and $\langle Y_s \rangle_{s\in\omega}$ to sets X and Y respectively, the *least oldest* element in $X - Y$ at the stage s is the least element $x \in X_s - Y_s$ such that for all $y \in X_s - Y_s$, $a(X, x, s) \leq a(X, y, s)$, and the *least oldest* subset of $X - Y$ at the stage s is the least $F \subseteq X_s - Y_s$ such that for all $G \subset X_s - Y_s$, $a(X, F, s) \leq a(X, G, s)$.

1.5.5 Definition. Fix an enumeration operator Ψ and a Σ^0_2-set B. For any $x \in \Psi_s^{B_s}$ and stage s, we define the *use of x at stage s*, $u(x, s)$, to be the least oldest $F \subseteq B_s$ such that $x \in \Psi_s^F$. If $x \notin \Psi_s^{B_s}$, then $u(x, s)$ is undefined.

We define $\Psi^B[0] = \emptyset$ and for $s > 0$, we define

$$\Psi^B[s] = \left\{x \in \Psi_s^{B_s} : u(x, s) = u(x, s-1)\right\}.$$

1.5.6 Lemma. The sequence $\langle \Psi^B[s] \rangle_{s\in\omega}$ is a Σ^0_2-approximation to Ψ^B.

Proof. $x \in \Psi^B$ if and only if $\lim_s u(x, s)$ exists. □

We use the standard notation and terminology of strings which can be found in [Soa87]. In particular, given strings α and β, we use $\alpha \subseteq \beta$ ($\alpha \subset \beta$) to denote that β extends (properly extends) α. We say α is to the left of β ($\alpha <_L \beta$) if α is lexicographically less than β but $\alpha \not\subseteq \beta$. Furthermore, by $\alpha \leq \beta$ we denote non-strict lexicographical ordering ($\alpha <_L \beta$ or $\alpha \subseteq \beta$), and by $\alpha < \beta$ we denote strict lexicographical ordering ($\alpha \leq \beta$ and $\alpha \neq \beta$).

Chapter 2

The $\forall\exists\forall$-Theory of the Σ^0_2-Enumeration Degrees is Undecidable

2.1 The Theorems and the Algebraic Component of the Proof

This section and the next closely follow Lempp, Nies and Slaman [LNS98]. Our main result is

2.1.1 Theorem. The $\forall\exists\forall$-theory of the Σ^0_2-enumeration degrees in the language of partial orderings is undecidable.

We recall that a set of first order sentences S is *hereditarily undecidable* if there is no computable set of sentences separating $\overline{S}$ and $S \cap V$ where V is the set of all valid sentences in the language of S. The proof of Theorem 2.1.1 uses the following theorem:

2.1.2 Theorem. [Nie96] The $\exists\forall$- (and hence the $\forall\exists\forall$-) theory of the finite bipartite graphs with nonempty left and right domains in the language of one binary relation, but without equality, is hereditarily undecidable.

We will use Theorem 2.1.2 to prove Theorem 2.1.1 via the Nies Transfer Lemma. Before we state this lemma, we need to define what it means for one class of structures to be elementarily definable in another class of structures.

2.1.3 Definition. A Σ_k-formula is a prenex formula that begins with an $\exists$-quantifier and contains $k-1$ quantifier alternations. A Π_k-formula is a prenex formula that begins with a $\forall$-quantifier and contains $k-1$ quantifier alternations.

2.1.4 Definition. Let $\mathcal{L}_\mathcal{C}$ and $\mathcal{L}_\mathcal{D}$ be finite relational languages not necessarily containing equality.

1. A Σ_k-scheme s for $\mathcal{L}_\mathcal{C}$ and $\mathcal{L}_\mathcal{D}$ consists of a Σ_k-formula $\varphi_U(\overline{x};\overline{y})$ (in the language $\mathcal{L}_\mathcal{D}$), and for each m-ary relation symbol $R \in \mathcal{L}_\mathcal{C}$, two Σ_k-formulas $\varphi_R(\overline{x}_0,\ldots,\overline{x}_{m-1};\overline{y})$ and $\varphi_{\neg R}(\overline{x}_0,\ldots,\overline{x}_{m-1};\overline{y})$ (again in $\mathcal{L}_\mathcal{D}$).

2. For a Σ_k-scheme s, we define a Π_{k+1}-formula $\alpha(\overline{p})$, called a *correctness condition*, for a list of parameters $\overline{p}$, as the conjunction of the following formulas:

   (a) (coding the universe) $\{\overline{x} : \varphi_U(\overline{x},\overline{p})\} \neq \emptyset$, and

   (b) (coding the relations) for each m-ary relation symbol R in the language $\mathcal{L}_\mathcal{C}$, the set
   $\{(\overline{x}_0,\ldots,\overline{x}_{m-1}) : \forall i < m(\varphi_U(\overline{x}_i,\overline{p}))\}$ is the disjoint union of the two sets
   $\{(\overline{x}_0,\ldots,\overline{x}_{m-1}) : \varphi_R(\overline{x}_0,\ldots,\overline{x}_{m-1},\overline{p})\}$ and
   $\{(\overline{x}_0,\ldots,\overline{x}_{m-1}) : \varphi_{\neg R}(\overline{x}_0,\ldots,\overline{x}_{m-1},\overline{p})\}$.

3. Define a formula $\varphi_{eq(\mathcal{C})}(x,y)$ as the conjunction of all formulas $\forall \overline{z}(R(x,\overline{z}) \leftrightarrow R(y,\overline{z}))$ where R ranges over all relations $R \in \mathcal{L}_\mathcal{C}$ and over all permutations of the arguments of R. (The purpose of this formula is to redefine equality if the language contains equality.) For an $\mathcal{L}_\mathcal{C}$-structure $\mathcal{C}$, define the induced quotient structure $\mathcal{C}/eq(\mathcal{C})$ in the obvious way. Similarly define a formula $\varphi_{eq(\mathcal{D})}(x,y)$ and a quotient structure $D/eq(D)$, using the relations $R \in \mathcal{L}_\mathcal{D}$.

4. A class $\mathcal{C}$ of relational structures, in the language $\mathcal{L}_\mathcal{C}$, is Σ_k-*elementarily definable with parameters* in a class of relational structures $\mathcal{D}$, in the language $\mathcal{L}_\mathcal{D}$, if there is a Σ_k-scheme s such that for each structure $C \in \mathcal{C}$, there is a structure $D \in \mathcal{D}$ and a finite set of parameters $\overline{p} \in D$ satisfying the following:

   (a) (correctness condition) $D \vDash \alpha(\overline{p})$, and

   (b) (coding the structure) $C/eq(C) \cong \tilde{C}/eq(\tilde{C})$, where $\tilde{C}$ is the substructure of D defined by $\tilde{C} = \{\overline{x} : \varphi_U(\overline{x};\overline{p})\}$, and for each m-ary relation symbol $R \in \mathcal{L}_\mathcal{C}$, the relation $\tilde{R}$ on $\tilde{C}$ is defined by

$$\tilde{R} = \{(\overline{x}_0, \ldots, \overline{x}_{m-1}) : \varphi_R(\overline{x}_0, \ldots, \overline{x}_{m-1}; \overline{p})\}\,.$$

We state two more theorems that are needed to prove our main result.

2.1.5 Theorem (Nies Transfer Lemma [Nie96]). Fix $k \geq 1$ and $r \geq 2$. Suppose a class of structures $\mathcal{C}$ is Σ_k-elementarily definable with parameters in a class of structures $\mathcal{D}$ (in finite relational languages $\mathcal{L}_\mathcal{C}$ and $\mathcal{L}_\mathcal{D}$, respectively). Then the hereditary undecidability of the Π_{r+1}-theory of $\mathcal{C}$ implies the hereditary undecidability of the Π_{r+k}-theory of $\mathcal{D}$.

2.1.6 Theorem. The class of finite bipartite graphs with nonempty left and right domains in the language of one binary relation, but without equality, is ∃-elementarily definable, with parameters, in the partial ordering $\mathfrak{S}$ of the Σ_2^0-enumeration degrees (i.e. in the class $\{\mathfrak{S}\}$).

The balance of this chapter after the current section is dedicated to proving Theorem 2.1.6. The presented construction considers only the cases when the sizes of the left and right domains are both greater than or equal to two. This is done to simplify the construction but in no way affects the result since the ∃∀-theory of this subclass of structures is also undecidable. The construction is easily modified to accommodate all finite bipartite graphs with non-empty left and right domains; however, the extra technical details that come with this addition obfuscate the finer points of what is happening.

As a side note, we mention that the method of coding finite bipartite graphs was used in [LN95] to establish the undecidability of the ∀∃∀∃-theory of the enumerable wtt-degrees and in [LNS98] to establish the undecidability of the ∀∃∀-theory of the computably enumerable Turing degrees.

Proof of Theorem 2.1.1. Apply the Nies Transfer Lemma (setting $k = 1$ by Theorem 2.1.6 and $r = 2$ by Theorem 2.1.2) in order to obtain the hereditary undecidability of the ∀∃∀-theory of the Σ_2^0-enumeration degrees. □

We will perform the construction in such a way as to show the following corollary:

2.1.7 Corollary. The $\forall\exists\forall$-theory of the Δ^0_2-enumeration degrees in the language of partial orderings is undecidable.

Proof. The proof of Theorem 2.1.6 actually shows that the class of finite bipartite graphs with nonempty left and right domains in the language without equality is $\exists$-elementarily definable in the partial ordering of the Σ^0_2-enumeration degrees using Δ^0_2-degrees as parameters. Since the Δ^0_2-degrees are a proper subclass of the Σ^0_2-degrees, we are able to restrict all quantifiers in the defined $\exists$-scheme to Δ^0_2-degrees. □

Once we have shown that both the Δ^0_2- and Σ^0_2-enumeration degrees are undecidable, by an extension of the above argument, we get the following corollary:

2.1.8 Corollary. If $\mathfrak{M}$ is a substructure of the enumeration degrees which contains the Δ^0_2-degrees then the $\forall\exists\forall$-theory of $\mathfrak{M}$ is undecidable.

2.2 The Requirements

In this section we will introduce the requirements that need to be satisfied to prove the main theorem and justify how their satisfaction implies the desired result. We begin with a definition.

2.2.1 Definition. Let I be a computable subset of ω. We say that the set of degrees $\{\mathbf{a}_i : i \in \omega\}$ is independent if for every $j \in I$, $\mathbf{a}_j \not\leq \bigvee_{i \in I - \{j\}} a_i$.

Fix a finite bipartite graph with nonempty left domain $L = \{0, 1, \ldots, n\}$, nonempty right domain $R = \left\{\tilde{0}, \tilde{1}, \ldots, \tilde{n}\right\}$ and edge relation $E \subseteq L \times R$.

We code the left domain using a $\exists$-formula $\psi(x;a,b,c)$. We will represent each vertex $i \in L$ by a difference of two intervals $[\mathbf{a}_i, \mathbf{a}) - [\mathbf{c}, \mathbf{0}'_e]$ of Σ^0_2-degrees ($\mathbf{0}'_e$ is defined later) where the following properties hold:

(2.1) $\mathbf{a} = \bigvee_{i \in L} \mathbf{a}_i$;

(2.2) for all $i, j \in L$, if $i \neq j$ then $\mathbf{c} \leq \mathbf{a}_i \vee \mathbf{a}_j$;

(2.3) for all $i \in L$, $\mathbf{c} \not\leq \mathbf{a}_i$;

(2.4) the degrees $\mathbf{a}_0, \ldots, \mathbf{a}_n$ are independent; and

(2.5) there exists a Σ^0_2-enumeration degree $\mathbf{b}$ incomparable with each $\mathbf{a}_i$ and $\mathbf{a}$, such that $\forall \mathbf{x} \leq \mathbf{a}(\mathbf{x} \not\leq \mathbf{b} \Leftrightarrow \exists i \leq n(\mathbf{a}_i \leq \mathbf{x}))$.

The $\exists$- (in fact quantifier free) formula $\psi(x;a,b,c)$ used to code the left domain is now chosen to be

$$x \leq a \;\&\; x \not\leq b \;\&\; x \not\geq c.$$

In the course of the construction, we build Σ^0_2-sets $A_0, \ldots, A_n$, A, B, and C, and set $\mathbf{a}_i = \deg_e(A_i)$ for all $i \in L$, $\mathbf{a} = \deg_e(A)$, $\mathbf{b} = \deg_e(B)$ and $\mathbf{c} = \deg_e(C)$. (Even though we build these sets as Σ^0_2-sets, we will actually construct them using Δ^0_2-approximations.) We now outline the requirements that these sets need to meet in order to satisfy the above properties.

To ensure (2.1) and (2.2), for all $i, j \in L$ with $i \neq j$, we construct enumeration operators $\Theta_{i,j}$ to meet the global requirements:

$$\begin{aligned} &\mathcal{J} &&: A = \textstyle\bigoplus_{i \in L} A_i =_{\text{def}} \{\langle x, i\rangle : x \in A_i\}, \\ &\mathcal{P}_{i,j} &&: C = \Theta_{i,j}^{A_i \oplus A_j} \text{ if } i \neq j. \end{aligned}$$

We ensure (2.3) and (2.4) by requiring for all enumeration operators Ξ and Ψ, and all $i \in L$:

$$\begin{aligned} \mathcal{N}_{\Xi,i} &: C \neq \Xi^{A_i} \text{ and} \\ \mathcal{I}_{\Psi,i} &: A_i \neq \Psi^{\bigoplus_{j\neq i} A_j}. \end{aligned}$$

Finally, in order to ensure (2.5), we require that for all enumeration operators Φ and Ω and for all $j \in L$:

$$\begin{aligned} \mathcal{S}_{\Omega} &: \exists\Gamma(\Omega^A = \Gamma^B) \text{ or } \exists\Delta, i \in L(A_i = \Delta^{\Omega^A}), \\ \mathcal{T}_{\Phi,j} &: A_j \neq \Phi^B, \end{aligned}$$

where Γ and Δ are enumeration operators built by us that depend on Ω and j. We mention here that the requirements $\mathcal{S}_{\Omega}$ and $\mathcal{T}_{\Phi,j}$ generalize a theorem of Ahmad. In [AL98] she constructs what is known as an Ahmad pair: two Σ^0_2-enumeration degrees $\mathbf{a}$ and $\mathbf{b}$ such that $\mathbf{a} \not\leq \mathbf{b}$ but for all degrees $\mathbf{c} < \mathbf{a}$, $\mathbf{c} \leq \mathbf{b}$.

The right domain is coded in a similar manner using Σ^0_2-sets $\tilde{A}_{\tilde{0}}, \ldots, \tilde{A}_{\tilde{n}}$, $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$, and requirements $\tilde{\mathcal{J}}$, $\tilde{\mathcal{P}}_{\tilde{\imath},\tilde{\jmath}}$, $\tilde{\mathcal{N}}_{\Xi,\tilde{\imath}}$, $\tilde{\mathcal{I}}_{\Psi,\tilde{\imath}}$, $\tilde{\mathcal{S}}_{\Omega}$ and $\tilde{\mathcal{T}}_{\Phi,\tilde{\jmath}}$. The $\exists$-formula $\varphi_U(x;\overline{y})$ required by Definition 2.1.4 can now be chosen as $\psi(x;\mathbf{a},\mathbf{b},\mathbf{c}) \vee \psi(x;\tilde{\mathbf{a}},\tilde{\mathbf{b}},\tilde{\mathbf{c}})$.

The reason that we use an ambiguous representation of the vertices is that we need a $\exists$-formula to define the universe. We could represent the left domain by the minimal degrees satisfying $\psi(\mathbf{x};\mathbf{a},\mathbf{b},\mathbf{c})$, i.e. $\{\mathbf{a}_0,\ldots,\mathbf{a}_n\}$, but this would be a $\forall$-formula and hence only imply that the $\forall\exists\forall\exists$-theory of $\mathfrak{S}$ is undecidable. Given a degree $\mathbf{x}$ that satisfies $\psi(\mathbf{x};\mathbf{a},\mathbf{b},\mathbf{c})$, properties (2.1) - (2.4) allow us to unambiguously recover the vertex that this degree represents.

In defining a copy of the edge relation $E(\cdot,\cdot)$, we need to make sure that the formulas $\varphi_E(x,\tilde{x},\overline{y})$ and $\varphi_{\neg E}(x,\tilde{x},\overline{y})$ do not depend on the particular pair of degrees that are chosen to represent the vertices. To accomplish this, we build two more Σ^0_2-enumeration degrees $\mathbf{e}_0$ and $\mathbf{e}_1$ satisfying for all $i \in L$ and $\tilde{\imath} \in R$:

(2.6) $E(i,\tilde{\imath})$ iff $\mathbf{e}_0 \leq \mathbf{a}_i \vee \tilde{\mathbf{a}}_{\tilde{\imath}}$ iff $\mathbf{e}_1 \not\leq \mathbf{a}_i \vee \tilde{\mathbf{a}}_{\tilde{\imath}}$, and

(2.7) $\neg E(i,\tilde{\imath})$ iff $\mathbf{e}_0 \not\leq \mathbf{a}_i \vee \tilde{\mathbf{a}}_{\tilde{\imath}}$ iff $\mathbf{e}_1 \leq \mathbf{a}_i \vee \tilde{\mathbf{a}}_{\tilde{\imath}}$.

The $\exists$-formula $\varphi_E(x,\tilde{x},e_0,e_1)$ required by Definition 2.1.4 can now be chosen as:

$$(\exists x_1 \leq x)(\exists \tilde{x}_1 \leq \tilde{x})(\exists z)(\psi(x_1,a,b,c) \;\&\; \psi(\tilde{x}_1,\tilde{a},\tilde{b},\tilde{c}) \;\&\; z \geq x_1 \;\&\; z \geq \tilde{x}_1 \;\&\; e_1 \not\leq z)$$

The choice of the $\exists$-formula for $\varphi_{\neg E}(x,\tilde{x},e_0,e_1)$ is similar (the only difference is that e_1 has been replaced by e_0):

$$(\exists x_1 \leq x)(\exists \tilde{x}_1 \leq \tilde{x})(\exists z)(\psi(x_1,a,b,c) \;\&\; \psi(\tilde{x}_1,\tilde{a},\tilde{b},\tilde{c}) \;\&\; z \geq x_1 \;\&\; z \geq \tilde{x}_1 \;\&\; e_0 \not\leq z)$$

To ensure the equivalences dictated by (2.6) and (2.7), we build Σ^0_2-sets E_0 and E_1, setting $\mathbf{e}_0 = \deg_e(E_0)$ and $\mathbf{e}_1 = \deg_e(E_1)$, and meeting for each enumeration operator Υ, each $i \in L$, and each $\tilde{\imath} \in R$, the following requirements:

$$\begin{aligned}
\mathcal{E}^0_{i,\tilde{\imath}} &: E(i,\tilde{\imath}) \Rightarrow E_0 = \Lambda^{A_i \oplus \tilde{A}_{\tilde{\imath}}}_{0,i,\tilde{\imath}},\\
\mathcal{F}^1_{\Upsilon,i,\tilde{\imath}} &: E(i,\tilde{\imath}) \Rightarrow E_1 \neq \Upsilon^{A_i \oplus \tilde{A}_{\tilde{\imath}}},\\
\mathcal{E}^1_{i,\tilde{\imath}} &: \text{not } E(i,\tilde{\imath}) \Rightarrow E_1 = \Lambda^{A_i \oplus \tilde{A}_{\tilde{\imath}}}_{1,i,\tilde{\imath}},\\
\mathcal{F}^0_{\Upsilon,i,\tilde{\imath}} &: \text{not } E(i,\tilde{\imath}) \Rightarrow E_0 \neq \Upsilon^{A_i \oplus \tilde{A}_{\tilde{\imath}}}.
\end{aligned}$$

where $\Lambda_{0,i,\tilde{\imath}}$ and $\Lambda_{1,i,\tilde{\imath}}$ are enumeration operators built by us.

It is clear that the above requirements establish conditions (2.1) - (2.7) and that the formulas φ_U, φ_E, and $\varphi_{\neg E}$ establish Theorem 2.1.6.

Finally, in order to show Corollary 2.1.7, we add an additional requirement which when met ensures that A and $\tilde{A}$ are low. The following definitions, theorems, and lemma motivate the requirement that we use:

2.2.2 Definition. [Coo87], [McE85] Given a set $A \subset \omega$, we define

- $K_A = \left\{x : x \in \Phi_x^A\right\}$, where Φ_x is the x^{th} enumeration operator under some fixed computable listing;
- the jump of A to be $A' =_{\text{def}} K_A \oplus \overline{K_A}$; and
- $\mathbf{a}' = \deg_e(A')$ where $\mathbf{a} = \deg_e(A)$.

Cooper and McEvoy show that the jump operator in the enumeration degrees has the same properties as the jump operator in the Turing degrees. Namely, $A \leq_e B \Rightarrow A' \leq eB'$ and $A <_e A'$.

2.2.3 Theorem. [McE85] $\mathbf{0}_e' = \deg_e(\emptyset') = \deg_e(\overline{K})$ where $\overline{K}$ denotes the compliment to the halting problem.

2.2.4 Theorem. [Coo87] $\mathbf{0}_e'$ is the maximal Σ_2^0-enumeration degree. i.e. $A \leq_e \overline{K}$ if and only if A is Σ_2^0.

2.2.5 Definition. An enumeration degree $\mathbf{a}$ is low if $\mathbf{a}' = \mathbf{0}_e'$. A set is low if its enumeration degree is low.

2.2.6 Lemma. [MC85] A set A is low if and only if there exists a Σ_2^0- or Δ_2^0-approximation to A such that for all enumeration operators Φ and all x, $\lim_s \Phi^A[s](x)$ exists.

Thus, satisfying the requirement

$$\mathcal{L}_{\Pi,x} \quad : \exists^{\infty} s \left(x \in \Pi^{A \oplus \tilde{A}}[s] \right) \Rightarrow x \in \Pi^{A \oplus \tilde{A}}$$

for all enumeration operators Π and all $x \in \omega$ will guarantee that A and $\tilde{A}$ are both low, as well as A_i and $\tilde{A}_{\tilde{\imath}}$ for all $i \in L$ and $\tilde{\imath} \in R$. We construct B and $\tilde{B}$ in such a way that the lowness of A and $\tilde{A}$ guarantees that B, $\tilde{B}$, C, $\tilde{C}$, E_0, and E_1 are Δ^0_2.

2.3 The Intuition for the Strategies

We briefly outline the strategies used to meet the above requirements. The key part of the construction, and the part that makes it a $\mathbf{0}'''$ construction, is the interplay between the $\mathcal{S}$- and the $\mathcal{T}$-strategies. We will first explain this interaction and then add the $\mathcal{J}$-, $\mathcal{P}$-, $\mathcal{N}$- and $\mathcal{I}$-strategies. In describing the interaction between the $\mathcal{S}$- and $\mathcal{T}$- strategies, their action in the actual construction, and in their verification, we closely follow the construction of Lempp, Slaman, and Sorbi [LSS]. Since the strategies for the left and the right domains are the same, in what follows, we initially only describe the strategies that are necessary for building the left domain. We then add the $\mathcal{E}^j$-, $\mathcal{F}^j$-, and $\mathcal{L}$-strategies, which are the strategies that define the relationship between the left and right domains.

The $\mathcal{S}$-requirement

This strategy will try to build Γ while lower priority $\mathcal{T}$-requirements try to destroy Γ and build Δ. The strategy is as follows:

1. Pick the least element $x \in \Omega^A$ that has no coding number b_x.

2. Pick a coding number b_x for x larger than any number seen so far in the construction.

3. Enumerate b_x into B and the axiom $\langle x, F_x \rangle$ into Γ where F_x is the finite set composed of b_x and all current killing points for this strategy (picked by lower priority $\mathcal{T}$-requirements).

4. For all $x \in \Omega^A - \Gamma^B$, with b_x defined, enumerate b_x into B.

5. For all $x \in \Gamma^B - \Omega^A$, extract b_x from B.

Without interference from lower priority $\mathcal{T}$-requirements, it is clear that $\mathcal{S}$ successfully builds Γ.

The $\mathcal{T}$-requirement

The strategy of the $\mathcal{T}$-requirement varies markedly depending on whether or not there is an active $\mathcal{S}$-requirement above it on the tree of strategies. Hence, we will slowly work up to the full strategy (action below several active higher priority $\mathcal{S}$-requirements) in three stages.

One $\mathcal{T}$-requirement in isolation

In isolation, the $\mathcal{T}$-requirement follows the basic Friedberg-Muchnik strategy as follows: A witness a is chosen from a stream (defined below) of available witnesses and enumerated into the set A_j. When, if ever, the element a enters Φ^B, the strategy will extract a from A_j while restraining B. Since there is no active higher priority $\mathcal{S}$-requirement, we do not have to worry about a B-correction, in response to this extraction, that may injure our computation.

One $\mathcal{T}$-requirement below one $\mathcal{S}$-requirement

The case where one $\mathcal{T}$-strategy is below one $\mathcal{S}$-strategy is somewhat more complicated. The strategy proceeds as above and tries to find a number $a \in \Phi^B$ which can be extracted from A_j while still maintaining $\Gamma^B \subseteq \Omega^A$ and $a \in \Phi^B$. If such a number is ever found, the strategy will diagonalize and stop. If no such number is found, a stream of elements will be generated such that the removal of any one of these elements from A_j causes numbers to leave B. We then restrict all future changes of A_j to elements from this stream. This will put us in a position to meet the $\mathcal{S}$-requirement via the second alternative, at the expense of failing to achieve the $\mathcal{T}$-requirement, by destroying Γ and building a Δ which allows us to calculate Φ^A from B.

More precisely, the $\mathcal{T}$-strategy proceeds as follows:

1. Pick a fresh killing point q for Γ. Put q into B and require all future Γ-axioms $\langle x, F_x \rangle$ to include q in the oracle set F_x.

2. Pick a fresh witness z and put z into A_j.

3. Wait for $z \in \Phi^B$ via some axiom $\langle z, F \rangle$ at some future stage s.

4. Extract z from A_j and allow the $\mathcal{S}$-strategy to correct B (possibly injuring $\Phi^B(z)$).

5. From now on, if ever $\Gamma^B[s] \subseteq \Phi^A$ (while $z \notin A_j$), then cancel all action between stage s and now, restrain $F \subseteq B$, and stop. (In this case, we call the computation $\Phi^B(z)$ Γ-*cleared*.)

6. While waiting for Step 5 to apply, put z into the stream S; restrict all future changes in $A_j \upharpoonright s$ to numbers in S; extract q from B; add the axiom $\langle z, \Gamma^B[s]\rangle$ in Δ; add axioms $\langle z', \emptyset\rangle$ to Δ for all $z' < s$ with $z' \in A_j[s] - S$; and restart at Step 1 with a fresh killing point q.

The possible outcomes of the above $\mathcal{T}$-strategy are as follows:

(A) Wait forever at Step 3: Then $z \in A_j - \Phi^B$, and Γ is not affected since $q \in B$.

(B) Stop eventually at Step 5: Then $z \in \Phi^B - A_j$, and Γ is not affected since $z \in B$.

(C) Looping between Step 1 and Step 6 infinitely often: Then the $\mathcal{T}$-requirement may not be satisfied by the action of this strategy. Additionally, Γ^B will be finite since all killing points are eventually extracted from B, and all but finitely many Γ-axioms $\langle x, F\rangle$ contain one of these killing points in F. However, $A_j = \Delta^{\Omega^A}$ can be seen to hold as follows: For all $z \notin S$, the A_j-restraint from Step 6 guarantees that $z \in A_j$ if and only if $z \in \Delta^{\Phi^A}$, so we restrict our attention to elements $z \in S$. If $z \in A_j$ (and was enumerated into S at stage s_z) then $\Omega^A[s_z] \subseteq \Omega^A$ (assuming that no other strategies remove numbers in $A[s_z]$ from A, and so no number from $\Omega^A[s_z]$, the set Ω^A measured immediately before the extraction of z from A_j, can leave Ω^A), implying that $z \in \Delta^{\Omega^A}$. Conversely, if $z \in \Delta^{\Omega^A}$, then $\Gamma^B[s_z] \subseteq \Omega^A$, and since

Step 5 never applies, we must have that $z \in A_j$.

One $\mathcal{T}$-requirement below several $\mathcal{S}$-requirements

In this case, the $\mathcal{T}$-strategy is basically a nested version of the previous strategy: If we generate only a finite number of witnesses, or if we find a witness which is Γ-cleared for all Γ's above, then we diagonalize finitarily. Otherwise, we find the lowest priority $\mathcal{S}$-requirement such that infinitely many witnesses are not Γ-cleared for its Γ. We will then use these witnesses to form a stream from which lower priority strategies will have to work.

In addition to the finitary outcomes, we now have i_0 many infinitary outcomes (where i_0 is the number of Γ's that our strategy has to deal with). More details on this interaction will be given in the formal construction.

One $\mathcal{T}$-requirement below another $\mathcal{T}$-requirement

Assume that we have one $\mathcal{T}$-requirement $\mathcal{G}$ below another $\mathcal{T}$-requirement $\hat{\mathcal{G}}$. If $\mathcal{G}$ assumes finite outcome (A) or (B) for $\hat{\mathcal{G}}$, then $\mathcal{G}$ will act as described above. Otherwise, $\mathcal{G}$ assumes the infinite outcome (C) of $\hat{\mathcal{G}}$. In this case, $\mathcal{G}$ assumes that Γ^B is finite and, in fact, will only be able to act at stages in which $\hat{\mathcal{G}}$ has extracted the latest killing point q from B. Thus, $\mathcal{G}$ can now act as if in isolation, the restriction being that it can only use witnesses in the stream defined by $\hat{\mathcal{G}}$, so as to keep Δ correct. Note that when $\mathcal{G}$ puts a number z into, or extracts a number z from, the set A_j at a stage s, all numbers greater than z are removed from the stream and dumped into A_i since their assumption about Ω^A may now be incorrect. When a number z is dumped into a set Z, z is permanently enumerated into Z, and for any functional Δ that is being built, with $Z = \Delta^X$, the axiom $\langle z, \emptyset \rangle$ is enumerated into Δ.

The $\mathcal{J}$-requirement

The $\mathcal{J}$-strategy is a global strategy and operates by defining $\langle x, i \rangle \in A$ if and only if $x \in A_i$.

The $\mathcal{N}$- and $\mathcal{P}$-requirements

The $\mathcal{N}$-strategy is a standard Friedberg-Muchnik strategy and acts like the $\mathcal{T}$-strategy in isolation. The strategy will choose a new coding number c from the stream and enumerate c into C. When, if ever, the element c enters Ξ^{A_i}, the strategy will extract c from C while restraining A_i.

The $\mathcal{P}$-strategy is a global strategy and works in conjunction with the $\mathcal{N}$-strategies. Whenever some $\mathcal{N}_{\Xi,i}$-strategy enumerates an element c into C, the $\mathcal{P}$-strategy chooses coding numbers a_j (for $j \in L$) from the stream, enumerates a_j into A_j, and enumerates the axiom $\langle c, \{a_j\} \oplus \{a_k\} \rangle$ into $\Theta_{j,k}$ (for $j \neq k$). If ever c is extracted from C, then a_j is extracted from A_j for all $j \neq i$.

The $\mathcal{I}$-requirements

The $\mathcal{I}$-requirement is a standard Friedberg-Muchnik strategy and acts like the $\mathcal{T}$-strategy in isolation. The strategy will enumerate a new coding number a, chosen from the stream of available witnesses, into the set A_i. When, if ever, the element a enters $\Psi^{\bigoplus_{j \neq i} A_j}$, the strategy will extract a from A_i while restraining $\bigoplus_{j \neq i} A_j$.

The $\mathcal{E}^0$-, $\mathcal{F}^0$-, $\mathcal{F}^1$-, and $\mathcal{E}^1$-requirements

In this section, we describe the action of the strategies that define the edge relationship between the vertices in L and those in R. For the same reason that we only described the strategies that deal with the left domain, here we will only describe the action of the $\mathcal{F}^0$- and $\mathcal{E}^0$-strategies in that the $\mathcal{F}^1$- and $\mathcal{E}^1$-strategies have the same behavior.

The $\mathcal{F}^0$-strategy

We assume that $i \in L$, $\tilde{\imath} \in R$, and $\neg E(i, \tilde{\imath})$. Like the $\mathcal{T}$-requirement, this is a standard Friedberg-Muchnik strategy and acts just like a $\mathcal{T}$-strategy in isolation. The strategy will enumerate a new coding number e, larger than any number seen so far in the construction, and not from the stream, into the set E_0. When, if ever, the element e enters $\Upsilon^{A_i \oplus \tilde{A}_{\tilde{\imath}}}$, the strategy extracts e from E_0 while restraining $A_i \oplus \tilde{A}_{\tilde{\imath}}$.

The $\mathcal{E}^0$-strategy

This is a global strategy which works in conjunction with the $\mathcal{F}^0$-strategies, and builds an enumeration functional $\Xi_{0,i,\tilde{\imath}}$ for every $i \in L$ and $\tilde{\imath} \in R$ with $E(i, \tilde{\imath})$. Whenever an element e is enumerated into E_0, for every $i \in L$ and $\tilde{\imath} \in R$, new coding numbers a_i and $\tilde{a}_{\tilde{\imath}}$ are chosen from the stream, and the axiom $\langle e, \{a_i\} \oplus \{\tilde{a}_{\tilde{\imath}}\} \rangle$ is enumerated into $\Xi_{0,i,\tilde{\imath}}$. If e is ever extracted from E_0 by some $\mathcal{F}^0_{\Upsilon,j,\tilde{\jmath}}$-strategy, the $\mathcal{E}^0$-strategy extracts those a_i from A_i and $\tilde{a}_{\tilde{\imath}}$ from $\tilde{A}_{\tilde{\imath}}$ with $i \neq j$ and $\tilde{\imath} \neq \tilde{\jmath}$.

The $\mathcal{L}$-strategy

The action of the lowness strategy is similar to the Friedberg-Muchnik strategy, however it picks no coding numbers and only restrains $A \oplus \tilde{A}$. Specifically, the strategy waits for x to enter $\Phi^{A\oplus\tilde{A}}$ and when, if ever, this happens, restrains $A \oplus \tilde{A}$ by restraining all set A_i and $\tilde{A}_{\tilde{\imath}}$ for $i \in L$ and $\tilde{\imath} \in R$.

2.4 The Tree of Strategies

For the sake of simplifying notation, in what follows we will refer to the $\mathcal{F}^0$- and $\mathcal{F}^1$-requirements as $\mathcal{F}$-requirements, the $\mathcal{I}$- and $\tilde{\mathcal{I}}$-requirements as $\mathcal{I}$-requirements, the $\mathcal{S}$- and $\tilde{\mathcal{S}}$-requirements as $\mathcal{S}$-requirements, etc. Fix an arbitrary effective priority ordering $\{R_e\}_{e\in\omega}$ of all $\mathcal{N}$-, $\mathcal{I}$-, $\mathcal{S}$-, $\mathcal{T}$-, $\mathcal{F}$-, and $\mathcal{L}$-requirements. The $\mathcal{J}$-, $\mathcal{P}$- and $\mathcal{E}$-requirements will not be put on the tree of strategies since they are handled globally. Furthermore, we only put an $\mathcal{F}$-requirement into the priority ordering if its assumption about the edge relationship is true.

We define $\Sigma = \{\text{stop} < \infty_0 < \infty_1 < \infty_2 < \cdots < \text{wait} < \text{so}\}$ as our set of outcomes. ("so" stands for "$\mathcal{S}$'s outcome".) We define $\mathbf{T} \subset \Sigma^{<\omega}$ and refer to it as our *tree of strategies*. Each node of $\mathbf{T}$ will be associated with, and thus identified with, a strategy.

We assign requirements to nodes on $\mathbf{T}$ by induction as follows: The empty node is assigned to requirement R_0, and no requirement is *active* or *satisfied along* the empty node. Given an assignment to a node $\alpha \in \mathbf{T}$, we distinguish cases depending on the requirement R assigned to α:

Case 1: R is an $\mathcal{S}$-requirement: Then call R *active along* $\alpha^\frown\langle\text{so}\rangle$ *via* α. For all other requirements R', call R' *active* or *satisfied along* $\alpha^\frown\langle\text{so}\rangle$ *via* $\beta \subset \alpha$ if and only if it is so along α. Assign to $\alpha^\frown\langle\text{so}\rangle$ the highest priority requirement that is neither active nor satisfied along $\alpha^\frown\langle\text{so}\rangle$.

Case 2: R is an $\mathcal{N}$-, $\mathcal{I}$-, $\mathcal{F}$-, or $\mathcal{L}$-requirement. Then for $o \in \{\text{stop}, \text{wait}\}$, call R *satisfied along* $\alpha^\frown\langle o\rangle$ *via* α; and for all other requirements R', call R' *active* or *satisfied along* $\alpha^\frown\langle o\rangle$ *via* $\beta \subset \alpha$ if and only if it is so along α. Assign to $\alpha^\frown\langle o\rangle$ (for $o \in \{\text{stop}, \text{wait}\}$) the highest priority requirement that is neither active nor satisfied along $\alpha^\frown\langle\text{wait}\rangle$.

Case 3: R is a $\mathcal{T}$-requirement. Let $\beta_0 \subset \cdots \subset \beta_{i_0-1}$ be all the strategies $\beta \subset \alpha$ such that some $\mathcal{S}$-requirement is active along α via β_i. We denote by $\mathcal{S}_i$ the $\mathcal{S}$-requirement for β_i. (Here we allow $i_0 = 0$, in which circumstance this case is handled the same way as Case 2.) Then, for $o \in \{\text{stop}, \text{wait}\}$, call R *satisfied along* $\alpha^\frown\langle o\rangle$ *via* α; and for all other requirements R', call R' *active* or *satisfied along* $\alpha^\frown\langle o\rangle$ *via* $\beta \subset \alpha$ if and only if it is so along α. If $i_0 > 0$, fix $i \in [0, i_0)$. Call $\mathcal{S}_i$ *satisfied along* $\alpha^\frown\langle\infty_i\rangle$ *via* β_i and call any $\mathcal{S}_j$ requirement, for $j \in (i, i_0)$, neither active nor satisfied along $\alpha^\frown\langle\infty_i\rangle$; any other requirement is *active* or *satisfied along* $\alpha^\frown\langle\infty_i\rangle$ *via* $\beta \subset \alpha$ if and only if it is so along α. For any outcome $o \in \{\text{stop}, \text{wait}\} \cup \{\infty_i : i \in [0, i_0)\}$, assign to $\alpha^\frown\langle o\rangle$ the highest priority requirement neither active nor satisfied along $\alpha^\frown\langle o\rangle$. (The intuition is that under the finitary outcomes $\langle\text{stop}\rangle$ and $\langle\text{wait}\rangle$, the $\mathcal{T}$-requirement is assumed to be satisfied finitarily by diagonalization; whereas under outcome $\langle\infty_i\rangle$, the $\mathcal{S}_i$-requirement, while previously satisfied via an enumeration operator Γ_i, is now assumed to be satisfied by α constructing an enumeration operator Δ_i, while all $\mathcal{S}_j$-requirements active via some strategy between β_i and α are assumed to be injured.)

The *tree of strategies* $\mathbf{T}$ is now the set of all nodes $\alpha \in \Sigma^{<\omega}$ to which requirements have been assigned.

2.5 The Construction

The construction proceeds in stages $s \in \omega$. Before beginning, we give some conventions and definitions.

When we *initialize* a strategy, we make all the parameters undefined and make the stream $S(\alpha)$ of α empty.

The stream $S(\emptyset)$ of the root node $\emptyset$ of our tree of strategies at any stage s is $[0, s)$. The streams $S(\alpha)$ for $\alpha \neq \emptyset$ are defined during the construction.

A strategy will be *eligible* to act if it is along the current approximation $f_s \in \mathbf{T}$ to the true path $f \in [\mathbf{T}]$ of the construction. At a stage s, if $\alpha \subseteq f_s$, s is called an α-stage.

At an α-stage s, a number z in the stream $S(\alpha)$ is *suitable for* α if, for every set X in $\{A, A_0, \ldots, A_n\} \cup \left\{\tilde{A}, \tilde{A}_{\tilde{0}}, \ldots, \tilde{A}_{\tilde{n}}\right\}$,

1. z is not currently *in use for* X by any strategy (i.e., z is not the current witness or coding number targeted for X by any strategy that has not been initialized since z has been picked).

2. z has not been dumped into X.

3. z is greater than $|\alpha|$ or any stage at which any $\beta \supseteq \alpha$ has changed any set, picked any number, or extended any enumeration operator.

4. z is greater than any stage which any $\beta \subset \alpha$ with finitary outcome $\langle \text{wait} \rangle$ or $\langle \text{stop} \rangle$ along α has first taken on this outcome since its last initialization.

5. z is greater than z' many numbers in S which are not in use for X by any $\beta \subseteq \alpha$ where z' is the greater of the last number in use by α and the most recent stage at which α was initialized.

During the course of the construction, all parameters are assumed to remain unchanged unless specified otherwise.

At the end of each stage s, we will dump certain elements into their respective target sets and initialize certain strategies as described below under *Ending the stage* s.

We now proceed with the construction.

Stage 0*:* Initialize all $\alpha \in \mathbf{T}$.

Stage $s > 0$*:* Each stage s is composed of substages $t \leq s$ such that some strategy $\alpha \in \mathbf{T}$, with $|\alpha| = t$, acts at substage t of stage s and decides which strategy will act at substage $t+1$ or whether to end the stage. If during a substage, there are no suitable numbers in the stream for that strategy, we end the current stage and continue with stage $s + 1$. The longest strategy eligible to act during a stage s is called the current approximation to the true path at stage s and is denoted f_s.

Substage t *of stage* s*:* Suppose a strategy α of length t is eligible to act at this substage. We distinguish cases depending on the requirement R assigned to α.

Case 1: R is an $\mathcal{S}_\Omega$-requirement: For the least oldest $z \in \Omega^A - \Gamma^B$ choose a new coding number b_z, if it is not already defined, larger than any number seen so far in the construction. Enumerate b_z into B and the axiom $\langle z, F\rangle$ into Γ, where the oracle set F contains b_z and all the current killing points q for Γ defined by $\mathcal{T}$-strategies $\beta \supseteq \alpha^\frown\langle\text{so}\rangle$. For any $z' \in \Gamma^B - \Omega^A$, remove $b_{z'}$ from B. End the substage by letting $\alpha^\frown\langle\text{so}\rangle$ be eligible to act next and set the stream $S(\alpha^\frown\langle\text{so}\rangle) = S(\alpha) \cap [s_0, s)$, where s_0 is the most recent stage less than or equal to s at which α was initialized.

Case 2: R is an $\mathcal{L}_{\Pi,x}$-requirement: Pick the first subcase which applies:

Case 2.1: $x \notin \Pi^{A\oplus\tilde{A}}$: Let $\alpha^\frown\langle\text{wait}\rangle$ be eligible to act next and set the stream $S(\alpha^\frown\langle\text{wait}\rangle) = S(\alpha) \cap [s_0, s)$, where s_0 is the most recent stage $\leq s$ at which α was initialized.

Case 2.2: $x \in \Pi^{A\oplus\tilde{A}}$: Let $\alpha^\frown\langle\text{stop}\rangle$ be eligible to act next and set the stream $S(\alpha^\frown\langle\text{wait}\rangle) = S(\alpha) \cap [s_0, s)$, where s_0 is the greater of the most recent stage $\leq s$ when α was initialized and the least stage $s' \leq s$ such that α was active and $x \in \Pi^{A\oplus\tilde{A}}[s_1]$ for all $s' \leq s_1 \leq s$.

Case 3: R is an $\mathcal{N}_{\Xi,i}$-requirement: Pick the first subcase which applies.

Case 3.1: α has not been eligible to act since its most recent initialization or some coding number a_j, for $j \in L$, is not defined: For each $j \in L$ with a_j undefined, choose a new distinct coding number a_j that is suitable for α and end the current stage.

Case 3.2: All coding numbers a_j, for $j \in L$, are defined but the coding number c is not defined: Choose c larger than any number seen so far in the construction. Enumerate c into C, a_j into A_j for all $j \in L$, the axioms $\langle c, \{a_j\} \oplus \{a_k\}\rangle$ into $\Theta_{j,k}$ for all j, k with $j \neq k$, and end the current stage.

Case 3.3: The coding number c is defined and $c \notin \Xi^{A_i}$: Let $\alpha^\frown\langle\text{wait}\rangle$ be eligible to act next and set the stream $S(\alpha^\frown\langle\text{wait}\rangle) = [s_0, s)$ where s_0 is the stage at which c was chosen.

Case 3.4: The coding number c is defined and $c \in C \cap \Xi^{A_i}$: Then α *stops* the strategy by extracting c from C, all a_j from A_j with $(a_j \neq a_i)$, and ending the current stage.

Case 3.5: The coding number c is defined and $c \in \Xi^{A_i} - C$: Let $\alpha^\frown\langle\text{stop}\rangle$ be eligible to act next and set the stream $S(\alpha^\frown\langle\text{stop}\rangle) = [s_0, s)$, where s_0 is the stage at which α stopped.

Case 4: R is an $\mathcal{I}_{\Psi,i}$- requirement: Pick the first subcase which applies.

Case 4.1: α has not been eligible to act since its most recent initialization or the coding number a_i is undefined: Choose a coding number a_i suitable for α, enumerate a_i into A_i, and end the current stage.

Case 4.2: a_i is defined and $a_i \notin \Psi^{\bigoplus_{j\neq i} A_j}$: Let $\alpha^\frown\langle\text{wait}\rangle$ be eligible to act next and set the stream $S(\alpha^\frown\langle\text{wait}\rangle) = [s_0, s)$, where s_0 is the stage at which a_i was chosen.

Case 4.3: a_i is defined and $a_i \in A_i \cap \Psi^{\bigoplus_{j\neq i} A_j}$: Then α *stops* the strategy by extracting a_i from A_i, and ending the current stage.

Case 4.4: a_i is defined and $a_i \in \Psi^{\bigoplus_{j\neq i} A_j} - A_i$: Let $\alpha^\frown\langle\text{stop}\rangle$ be eligible to act next and set the stream $S(\alpha^\frown\langle\text{stop}\rangle) = [s_0, s)$, where s_0 is the stage at which α stopped.

Case 5: R is an $\mathcal{F}^j_{\Upsilon,i,\tilde{i}}$-requirement: Pick the first subcase which applies.

Case 5.1: α has not been eligible to act since its most recent initialization or some coding number a_k, for $k \in L$, or $\tilde{a}_{\tilde{k}}$, for $\tilde{k} \in R$, is not defined: For each $k \in L$ with a_k undefined, and $\tilde{k} \in R$ with $\tilde{a}_{\tilde{k}}$ undefined, choose new distinct coding numbers a_k and $\tilde{a}_{\tilde{k}}$ that are suitable for α and end the current stage.

Case 5.2: All coding numbers a_k, for $k \in L$ and $\tilde{a}_{\tilde{k}}$ for $\tilde{k} \in R$ are defined but the coding number e_j is not defined: Choose e_j larger than any number seen so far in the construction. For all $k \in L$ and $\tilde{k} \in R$, enumerate a_k into A_k, e_j into E_j, and the axioms $\langle e_j, \{a_k\} \oplus \{\tilde{a}_{\tilde{k}}\}\rangle$ into $\Lambda_{j,k,\tilde{k}}$. End the current stage.

Case 5.3: The coding number e_j is defined and $e_j \notin \Upsilon^{A_i \oplus \tilde{A}_{\tilde{\imath}}}$: Let $\alpha^\frown\langle\text{wait}\rangle$ be eligible to act next and set the stream $S(\alpha^\frown\langle\text{wait}\rangle) = [s_0, s)$ where s_0 is the stage at which e_j was chosen.

Case 5.4: The coding number e_j is defined and $e_j \in E_j \cap \Upsilon^{A_i \oplus \tilde{A}_{\tilde{\imath}}}$: Then α *stops* the strategy by extracting e_j from E_j, every a_k from A_k for $k \neq i$, every $\tilde{a}_{\tilde{k}}$ from $\tilde{A}_{\tilde{k}}$ for $\tilde{k} \neq \tilde{\imath}$, and ending the current stage.

Case 5.5: The coding number e_j is defined and $e_j \in \Upsilon^{A_i \oplus \tilde{A}_{\tilde{\imath}}} - E_j$: Let $\alpha^\frown\langle\text{stop}\rangle$ be eligible to act next and set the stream $S(\alpha^\frown\langle\text{stop}\rangle) = [s_0, s)$, where s_0 is the stage at which α stopped.

Case 6: R is a $\mathcal{T}_{\Phi,j}$-requirement: Let $\beta_0 \subset \beta_1 \subset \cdots \subset \beta_{i_0-1} \subset \alpha$ be all the strategies such that some $\mathcal{S}_i$ is active along α via β_i (allowing $i_0 = 0$). For every $i < i_0$, and for every x that has been dumped into A_j, enumerate the axiom $\langle x, \emptyset\rangle$ into Δ_i. (In the following subcases, the enumeration operators Ω_i and Γ_i are those of β_i, for $i \in [0, i_0)$.)

Pick the first case which applies.

Case 6.1: α has not been eligible to act since its most recent initialization: For each $i \in [0, i_0)$, pick killing points q_i larger that any number seen so far in the construction and enumerate q_i into B. End the current stage s.

Case 6.2: α has current killing points but the witness z_{i_0} is undefined: Choose

z_{i_0} suitable for α, add z_{i_0} to A_j, initialize all strategies $\beta \supseteq \alpha^\frown\langle\text{wait}\rangle$, and end the current stage.

Case 6.3: z_{i_0} is defined and $z_{i_0} \notin \Phi^B$: End the substage by letting $\alpha^\frown\langle\text{wait}\rangle$ be eligible to act next and setting the stream $S(\alpha^\frown\langle\text{wait}\rangle) = S(\alpha) \cap [s_0, s)$ where s_0 is the stage at which z_{i_0} was chosen.

Case 6.4: α has stopped (as defined below) and has not been initialized since then: End the substage by letting $\alpha^\frown\langle\text{stop}\rangle$ be eligible to act next and setting the stream $S(\alpha^\frown\langle\text{stop}\rangle) = S(\alpha) \cap [s_0, s)$, where s_0 is the stage at α stopped.

Case 6.5: Otherwise $z_{i_0} \in A_j \cap \Phi^B$: We call z_{i_0} a *realized* witness. We now distinguish two subcases.

Case 6.5.1 $i_0 = 0$: Then α *stops* by extracting z_{i_0} from A_j and ending the current stage s.

Case 6.5.2 Otherwise $i_0 > 0$: Then α *stops* as follows:

2.5.1 Definition. For $i \in [0, i_0)$, call z Γ_i*-cleared* if

$$\Gamma_i^B[s_z] \subseteq \Omega_i^{A-\{\langle z,j\rangle\}},$$

where s_z is the stage at which z became a realized witness of α.

α first extracts z_{i_0} from A_j. We then have further subcases depending on whether we have a witness which is "fully Γ-cleared."

Case 6.5.2.1 Some witness $z \in S(\alpha)$ (current or former uncancelled, picked since α's most recent initialization) is Γ_i-cleared for all $i \in [0, i_0)$: Then α *stops* by removing z from A_j (if necessary), adding $B[s_z]$ into B, setting $z_{i_0} = z$ as its current witness and ending the current stage s.

Case 6.5.2.2 Otherwise: We will now define the streams associated with α's infinitary outcomes. We will use z_i to denote the least element of the stream $S(\alpha^\frown\langle\infty_i\rangle)$.

α acts as follows: Fix the least $i < i_0$ for which there is a current or former uncancelled witness z (minimal for this i, picked since α's most recent initialization) such that:

$$z \notin S(\alpha^\frown\langle\infty_i\rangle)$$
$$z \text{ is } \Gamma_k\text{-cleared for all } k \in (i, i_0), \text{ and}$$
$$z > \max\{z_k : k \leq i \text{ and } z_k \text{ currently defined}\}.$$

(Here we set $\max(\emptyset) = -1$. Note that the above condition holds trivially for $z = z_{i_0}$ and $i = i_0 - 1$, so z as defined above must exist.)

Then α

1. extracts q_k (for each $k \in [i, i_0)$) from B;

2. picks new q_k (for each $k \in [i, i_0)$) larger than any number seen so far in the construction and enumerates them into B;

3. cancels Δ_k for all $k \in (i, i_0)$;

4. cancels all (former or current) witnesses $z' \neq z$ of α with $z' \notin S(\alpha^\frown\langle\infty_k\rangle)$, for all $k \in (i, i_0]$ makes z_k undefined, and sets $S(\alpha^\frown\langle\infty_k\rangle) = \emptyset$;

5. adds z to $S(\alpha^\frown\langle\infty_i\rangle)$ and sets $z_i = z$ if z_i is currently undefined;

6. adds the axiom $\langle z, \Gamma_i^B[s_z]\rangle$ into Δ_i;

7. adds axioms $\langle z', \emptyset\rangle$ into Δ_i for all $z_i < z' < \max(S(\alpha^\frown\langle\infty_i\rangle))$ with $z' \in A_j - S(\alpha^\frown\langle\infty_i\rangle)$; and

8. ends the substage by letting $\alpha^\frown\langle\infty_i\rangle$ be eligible to act next.

Ending the stage s: If the stage s ended at Case 3.4, 4.3, 5.4, 6.5.1 or 6.5.2.1, let z_i be the number extracted by the strategy f_s from A_i. For every $\beta \subseteq f_s$, if β is a $\mathcal{T}_{\Phi,i}$-requirement then for every $x \in S(\beta)$, if $x > z_i$, dump x into A_i.

For every $\alpha >_L f_s$, if α is an $\mathcal{E}^j_{i,\tilde{i}}$-strategy, and α's diagonalization witness e_j is defined, enumerate e_j into E_j and the axiom $\langle e_j, \emptyset \rangle$ into $\Lambda_{j,k,\tilde{k}}$, for all $k \in L$ and $\tilde{k} \in R$. If α is a $\mathcal{P}_{\Xi,i}$-requirement and α's diagonalization witness c is defined, enumerate c into C and the axiom $\langle c, \emptyset \rangle$ into $\Theta_{j,k}$ for all $j, k \in L$. Initialize every strategy $\alpha >_L f_s$.

2.6 The Verification

Let $f = \liminf_s f_s$ be the *true path* of the construction, defined more precisely by induction by

$$f(n) = \liminf_{\{s : f\restriction n \subset f_s\}} f_s(n).$$

2.6.1 Lemma. (Tree Lemma)

(i) Each $\alpha \subset f$ is initialized at most finitely often.

(ii) For each strategy $\alpha \subset f$, the stream $S(\alpha)$ is an infinite set. No number can leave $S(\alpha)$ unless α is initialized. For every $X \in \{A, A_0, \dots, A_n\} \cup \left\{\tilde{A}, \tilde{A}_{\tilde{0}}, \dots, \tilde{A}_{\tilde{n}}\right\}$ and every stage s, there are an α-stage $t > s$ and a number $z > s$ such that z is suitable for α to enumerate into X at stage t.

(iii) The true path f is an infinite path through $\mathbf{T}$.

(iv) For any requirement $R_e = \mathcal{S}_\Omega$ or $\mathcal{T}_{\Phi,j}$, there is a strategy $\alpha \subset f$ such that the requirement is active via α along all sufficiently long $\beta \subset f$, or is satisfied via α along all β with $\alpha \subset \beta \subset f$. (In particular, for any requirement R_e, there is a longest strategy assigned to R_e along f.)

Proof. (i) Proceed by induction on α and note that the only time a strategy is initialized is when it is to the left of the true path or in Case 6.2, which can only happen finitely often.

(ii) Proceed by induction on $|\alpha|$ and note for the last part of (ii) that any number just entering $S(\alpha)$ is suitable for α at that stage.

(iii) A stage s is ended before substage s only under Cases 3.1, 3.2, 3.4, 4.1, 4.3, 5.1, 5.2, 5.4, 6.1, 6.2, or 6.5.1. By (ii), we cannot stop cofinitely often at 3.1, 4.1, 5.1 or 6.2 due to lack of suitable numbers.

(iv) By an easy induction argument on e. □

We now verify the satisfaction of the requirements.

2.6.2 Lemma. ($\mathcal{J}$-Lemma) The $\mathcal{J}$-requirement is satisfied.

Proof. Immediate from the definition of A. □

2.6.3 Lemma. ($\mathcal{I}$- Lemma) All $\mathcal{I}$-requirements are satisfied.

Proof. Fix a requirement $\mathcal{I}_{\Psi,i}$. By the Tree Lemma (Lemma 2.6.1(iv)), there is an $\mathcal{I}$-strategy $\alpha \subset f$ such that $\mathcal{I}_{\Psi,i}$ is satisfied along all β with $\alpha \subset \beta \subset f$. Then $\alpha^\frown\langle o\rangle \subset f$ for $o \in \{\text{stop}, \text{wait}\}$.

By the construction, the fact that α is eventually no longer initialized, and the Tree Lemma (Lemma 2.6.1(ii)), α eventually has a fixed diagonalization witness. Call this witness z.

If $\alpha^\frown\langle\text{wait}\rangle \subset f$ then $z \in A_i - \Psi^{\bigoplus_{j\neq i} A_j}$ by the construction, thus the requirement $\mathcal{I}_{\Psi,i}$ is clearly satisfied.

Otherwise $\alpha^\frown\langle\text{stop}\rangle \subset f$, so α stops at some stage s, and $z \in \Psi^{\bigoplus_{j\neq i} A_j}[s] - A_i$. We will show that *no* set changes at any number $< s_z$ (where s_z is the stage $\leq s$ at which z became a realized witness) by considering all possible strategies β.

Case A: $\beta <_L \alpha$: Then β is no longer eligible to act after stage s (or else α would be initialized and lose its witness).

Case B: $\beta > \alpha^\frown\langle\text{stop}\rangle$: The first time β is eligible to act after α stops is the first time β is eligible to act after being initialized: Thus β cannot change $\bigoplus_{j\neq i} A_j$ at any number that would injure $\Psi^{\bigoplus_{j\neq i} A_j}$.

Case C: $\beta^\frown\langle o\rangle \subseteq \alpha^\frown\langle\text{stop}\rangle$ for some $o \in \{\text{stop}, \text{wait}\}$: Then β cannot change $\bigoplus_{j\neq i} A_j$ without initializing α.

Case D: $\beta^\frown\langle\infty_i\rangle \subseteq \alpha$ for some $i \in \omega$: Then z was put by β into the stream of $\beta^\frown\langle\infty_i\rangle$, and at stage s, β adds a number $> z$ into the stream of $\beta^\frown\langle\infty_i\rangle$. At the first β-stage $s' > s$, β picks a coding number z' which is too large to injure $\Psi^{\bigoplus_{j\neq i} A_j}[z]$, and after stage s, β does not change $\bigoplus_{j\neq i} A_j$ at a number less than z'. So β cannot injure $\Psi^{\bigoplus_{j\neq i} A_j}(z)$ after stage s.

Case E: $\beta^\frown\langle\text{so}\rangle \subseteq \alpha$ is an $\mathcal{S}$-requirement: Then β never extracts any elements from $\bigoplus_{j\neq i} A_j$. □

2.6.4 Lemma. ($\mathcal{L}$-Lemma) All $\mathcal{L}$-requirements are satisfied.

Proof. The proof that $\mathcal{L}_{\Pi,x}$-requirements are satisfied is similar to the proof of Lemma 2.6.3. □

2.6.5 Lemma. ($\mathcal{N}$- and $\mathcal{P}$-Lemma) All $\mathcal{N}$- and $\mathcal{P}$-requirements are satisfied.

Proof. Fix a requirement $\mathcal{N}_{\Xi,i}$. The proof that $\mathcal{N}_{\Xi,i}$ is satisfied is similar to the proof of Lemma 2.6.3 with the additional case that when, if ever, $\mathcal{N}_{\Xi,i}$ extracts c from C, the $\mathcal{P}$-requirements will extract a_j from A_j for all $j \in L - \{i\}$. However, it is immediate that this action does not injure the $\Xi^{A_i}(c)$ computation.

Fix a requirement $\mathcal{P}_{i,j}$ and fix some element c that was targeted to enter C by some $\mathcal{N}_{\Xi,k}$-strategy α at, say, stage s. If α was ever initialized at some stage $s_0 > s$, then by the action at the end of stage s_0, c is enumerated into C and the axiom $\langle c, \emptyset \rangle$ into $\Theta_{i,j}$. In addition, c will never be chosen again as a diagonalization number by any other $\mathcal{N}$-strategy.

Assume that α was never initialized after stage s. We have two cases to consider.

Case 1: $c \in C$: At some stage $s_1 \geq s$ we enumerated the axiom $\langle c, \{a_i\} \oplus \{a_j\} \rangle$ into $\Theta_{i,j}$, the elements a_i into A_i, a_j into A_j, and c into C. Since α was not initialized after stage s_1, no other strategy could extract either a_i from A_i or a_j from A_j without initializing α, and hence $c \in \Theta_{i,j}^{A_i \oplus A_j}$.

Case 2: $c \notin C$: We have two subcases to consider.

Case 2a: c was never enumerated into C by α: By Lemma 2.6.1(ii), we must have $\alpha <_L f_{s_1}$ for all stages $s_1 > s$, and hence no axiom of the form $\langle c, \{a_i\} \oplus \{a_j\} \rangle$ was enumerated into $\Theta_{i,j}$. Therefore $c \notin \Theta_{i,j}^{A_i \oplus A_j}$.

Case 2b: Otherwise: This case is similar to Case 1. At some stage $s_1 \geq s$ we enumerate the axiom $\langle c, \{a_i\} \oplus \{a_j\} \rangle$ into $\Theta_{i,j}$, the elements a_i into A_i, a_j into A_j and c into C. Then, at some later stage $s_2 > s_1$, c is extracted from C by α, and so a_i is extracted from A_i or a_j from A_j. Since α was not initialized after stage s_2, and by Lemma 2.6.1(ii), no other strategy could enumerate either a_i back into A_i or a_j back into A_j, and hence $c \notin \Theta_{i,j}^{A_i \oplus A_j}$. □

2.6.6 Lemma. ($\mathcal{E}$- and $\mathcal{F}$-Lemma) All $\mathcal{E}$- and $\mathcal{F}$-requirements are satisfied.

Proof. The proof that all $\mathcal{E}^j_{i,\tilde{\imath}}$- and $\mathcal{F}^j_{\Upsilon,i,\tilde{\imath}}$-requirements are satisfied is similar to the proof of Lemma 2.6.5. □

2.6.7 Lemma. ($\mathcal{T}$-Lemma) All $\mathcal{T}$-requirements are satisfied.

Proof. Fix a requirement $\mathcal{T}_{\Phi,j}$. The proof that $\mathcal{T}_{\Phi,j}$ is satisfied is similar to the proof of Lemma 2.6.3. The difference is in how we handle *Case E*.

Case E: $\beta^\frown\langle \text{so}\rangle \subseteq \alpha$ and β's $\mathcal{S}$-requirement is active along α via β: Then α stops via Case 6.5.2.1 of the construction where $\beta = \beta_i$ for some β_i mentioned in Case 6.5.2.2. Thus z is Γ_i-cleared, i.e.,

$$\Gamma_i^B[s_z] \subseteq \Omega_i^{A-\{\langle z,j\rangle\}},$$

where s_z is the stage at which z became a realized witness of α. By the action at stage s (the stage when $\mathcal{T}_{\Phi,j}$ stops),

$$\Gamma_i^B[s] \subseteq \Omega_i^A[s],$$

so any later Γ_i-correction by β will only involve Γ_i-axioms defined after stage s_z, and thus will change any set only on numbers $> s_z$.

To complete this lemma, we add an additional case.

Case F: $\beta^\frown\langle \text{so}\rangle \subseteq \alpha$ and β's $\mathcal{S}$-requirement is not active along α via β: Then some α' with $\beta \subset \alpha' \subset \alpha$ kills β's enumeration operator Γ. Therefore

$$\Gamma_i^B[s_z] \subseteq \Omega_i^A[s_z]$$

by the action of β at stage s_z. Any later Γ-correction performed by β will only involve Γ-axioms defined after stage s_z, and hence will change any set only on numbers $> s_z$. $\square$

2.6.8 Lemma. ($\mathcal{S}$-Lemma) All $\mathcal{S}$-requirements are satisfied.

Proof. Fix a requirement $\mathcal{S}_\Omega$. By the Tree Lemma (Lemma 2.6.1(iv)), there is a longest $\mathcal{S}_\Omega$-strategy $\beta \subset f$. Again by the Tree Lemma (Lemma 2.6.1(iv)), we may now distinguish two cases:

Case 1: $\mathcal{S}_\Omega$ is active via β along all α with $\beta \subset \alpha \subset f$: Suppose that β is no longer initialized after, say, stage s_0.

For the sake of a contradiction, assume first that there is some $z \in \Omega^A - \Gamma^B$. Choose z_0 to be the least oldest such z with age s_z. Fix $s_1 \geq s_0, s_z$ such that no $\mathcal{T}$-strategy with killing point $\leq z$ (for this Γ) executes Step (i) of Case 6.5.2.2 of the construction. Then by the first β-expansionary stage $\geq s_1$, β will permanently put z into Γ^B by Case 1 of the construction.

If $z \in \Gamma^B$, then by Γ-correction of β under Case 1 of the construction, $z \in \Omega^A$.

Case 2: There is a $\mathcal{T}_{\Phi,j}$-strategy $\alpha \subset f$ such that $\mathcal{S}_\Omega$ is satisfied via α along all ξ with $\beta \subset \xi \subset f$: Then β is α's strategy β_i, $\alpha^\frown\langle\infty_i\rangle \subset f$, and we need to show that $\Delta_i^{\Omega^A} = A_j$ (for the enumeration operator Δ_i built by α after α's last initialization and after α cancels Δ_i for the last time).

We show that $A_j =^* \Delta_i^{\Omega^A}$ by distinguishing two cases for arguments $z \geq z_i$ of $\Delta_i^{\Omega^A}$:

Case 2a: $z \notin S(\alpha^\frown\langle\infty_i\rangle)$: Then, once $z < \max(S(\alpha^\frown\langle\infty_i\rangle[s])$, no strategy can remove z from A_j (and so by (7) of Case 6.5.2.2 of the construction, $z \in A_j$ if and only if $z \in \Delta_i^{\Omega^A}$). To see this, note that only strategies $\xi \subset \alpha$ with infinitary outcome along α can possibly change $A_j(z)$ (by the usual initialization argument). But, after stage s, any such ξ cannot put z into the stream of any strategy $\zeta \supset \xi$. If ξ is a $\mathcal{T}$- or $\mathcal{I}$-strategy, it will no longer remove z as a realized witness, and it will not remove z for Γ-correction (as in Case 4.3 or Case 6.5.1 of the construction) since ξ does not stop (as Case 4.3 or Case 6.5.1 does not apply). If ξ is an $\mathcal{S}$-strategy, then ξ does not remove numbers from A_j.

Case 2b: $x \in S(\alpha^\frown\langle\infty_i\rangle)$: We first observe that

$$z \in A_j \Leftrightarrow \langle z, j\rangle \in A, \tag{2.1}$$

$$z \in \Delta_i^{\Omega^A} \Leftrightarrow \Gamma_i^B \subseteq \Omega^A, \text{ and} \tag{2.2}$$

$$\Gamma_i^B[s_z] \not\subseteq \Omega^{A-\{\langle z,j\rangle\}} \tag{2.3}$$

by meeting the $\mathcal{J}$-requirement, the definition of Δ_i, and the fact that α does not stop, respectively.

Thus, if $z \notin A_j$, by (1.1) and (1.3) we have $\Gamma_i^B[s_z] \not\subseteq \Omega^A$, which by (1.2), gives us $z \notin \Delta^{\Omega^A}$.

On the other hand, if $z \in A_j$, then by (1), $\langle z, j\rangle \in A$ so it follows that

$$\Gamma_i^B[s_z] \subseteq \Omega^{A\cup\{\langle z,j\rangle\}}[s_z] \subseteq \Omega^A$$

and we have $z \in \Delta_i^{\Omega^A}$. □

2.6.9 Lemma. The sets B, $\tilde{B}$, C, $\tilde{C}$, E_0, and E_1 are Δ_2^0.

Proof. We prove that B is Δ_2^0. The proof for $\tilde{B}$ is similar. In the construction, only under Case 1 and Case 6.5.2.2 do we enumerate elements into or extract elements from B.

Fix an element z and an enumeration operator Π. By Lemma 2.6.4, the limit $\lim_s \Pi^A(z)[s]$ converges. Hence, any $\mathcal{S}_\Pi$-strategy that chooses a coding number c_z for z under Case 1 will enumerate c_z into and extract c_z from B a finite number of times. Furthermore, we choose our coding numbers c_z in such a way that if ever $\mathcal{S}_\Pi$ is reset, no other strategy will enumerate c_z into B.

Under Case 6.5.2.2, a killing point can be enumerated into and extracted from B at most once. Like in the previous case, we choose new killing points in such a way that no killing point, once cancelled, will ever be used again by another strategy. Therefore B is Δ^0_2.

By Lemma 2.6.4, $A \oplus \tilde{A}$ is low and by Lemma 2.6.5, C, $\tilde{C} \leq_e \tilde{A} \oplus A$. Therefore both C and $\tilde{C}$ are Δ^0_2. An element x may be enumerated into E_j at most once by Case 5.2 of the construction and extracted at most once by Case 5.4. The only other time that x may be enumerated into E_j is when it is dumped in due to initialization. This may happen at most once and after this, x will never be extracted from E_j. Therefore both E_0 and E_1 are Δ^0_2.

□

This completes the proof of the theorem.

Chapter 3

Non-Splitting Enumeration Degrees

3.1 Introduction

A non-zero degree $\mathbf{a}$ is non-splitting if whenever $\mathbf{a} = \mathbf{b} \vee \mathbf{c}$ then $\mathbf{a} = \mathbf{b}$ or $\mathbf{a} = \mathbf{c}$. It has been shown that every c.e. Turing degree splits as the non-trivial join of two smaller c.e. Turing degrees [Sac63]. On the other hand a minimal Δ^0_2-Turing degree is trivially non-splitting.

In her thesis, Ahmad [Ahm89] (cf. [AL98]) constructed a non-splitting Σ^0_2-enumeration degree. This result is interesting since, unlike the Δ^0_2-Turing degrees, the Σ^0_2-enumeration degrees are dense, and furthermore, the existence of non-splitting degrees allows us to construct Ahmad pairs, as discussed in Chapter 2.

As of yet, no direct construction of a non-splitting degree using a tree of strategies has been published in the literature. In this section we present such a construction. We then present current research that Andrea Sorbi and the author are engaged in regarding non-splitting degrees by showing how to modify this proof in order to construct a low non-splitting degree, a properly Σ^0_2-non-splitting degree, and finally to demonstrate that every non-trivial Δ^0_2-degree bounds a non-splitting degree.

3.2 Non-splitting Degrees

3.2.1 Theorem ([Ahm89] (cf. [AL98])). There exists a non-zero non-splitting enumeration degree.

In order to prove this theorem, we build a Σ^0_2-set A in stages, meeting for all enumeration functionals Φ, Ψ, Ω_0, and Ω_1 the following requirements:

$$\mathcal{N}_\Phi \quad : \; A \neq \Phi,$$
$$\mathcal{S}_{\Psi,\Omega_0,\Omega_1} \quad : \; A = \Psi^{\Omega_0^A \oplus \Omega_1^A} \Rightarrow \exists \Gamma_0, \Gamma_1 \left[A =^* \Gamma_0^{\Omega_0^A} \text{ or } A =^* \Gamma_1^{\Omega_1^A} \right].$$

Here Γ_0 and Γ_1 are enumeration operators built by us and local only to the strategy by which they are built. Upon satisfaction of the requirements, it follows that $\deg_e(A)$ is non-splitting.

During the course of the construction, for each strategy on the tree, a stream of elements will be enumerated from which the strategy will be required to pick its witnesses. These streams will be enumerated in such a way as to guarantee that every strategy which is on the true path will have an infinite number of witnesses from which to choose coding locations.

We now present the strategies.

The Strategy for $\mathcal{N}_\Phi$

The $\mathcal{N}$-requirement follows the basic Friedberg-Muchnik strategy as follows. The least witness a that is not dumped into A is chosen from a stream (defined below) of available witnesses and enumerated into the set A. When, if ever, the element a enters Φ, the strategy will extract a from A, dump all elements $y > a$ from the stream into A, and stop.

The Strategy for $\mathcal{S}_{\Psi,\Omega_0,\Omega_1}$

Our description of this strategy operates under the assumption that $A = \Psi^{\Omega_0^A \oplus \Omega_1^A}$, since if otherwise, the requirement is trivially met. The strategy attempts to build two enumeration functionals Γ_0 and Γ_1 such that, if the above assumption is true, either $A =^* \Gamma_0^{\Omega_0^A}$ or $A =^* \Gamma_1^{\Omega_1^A}$.

At each stage of the construction, the stream Q of available witnesses for the $\mathcal{S}$-strategy will be partitioned into four sets Q_w, Q_0, Q_1, and $Q_{\neq}$. The set Q_w will be the set of elements x for which we are currently waiting to see if an extraction x from A will cause x to leave $\Psi^{\Omega_0^A \oplus \Omega_1^A}$. The sets Q_i will be the elements x for which we have $x \in A$ if and only if $x \in \Gamma_i^{\Omega_i^A}$. Finally, $Q_{\neq}$ will consist of the members of the stream Q that entered after a successful diagonalization of A with $\Psi^{\Omega_0^A \oplus \Omega_1^A}$. A particular witness may be in only one set at a time and may move from Q_w to Q_0 to Q_1 to $Q_{\neq}$, possibly skipping a set in the sequence, but will never be allowed to move backwards through the sequence. We explain how this process is accomplished.

As potential witnesses are enumerated into the stream of the $\mathcal{S}$-strategy, they will first be placed in Q_w. The streams of lower priority strategies which assume that the length of agreement between A and $\Psi^{\Omega_0^A \oplus \Omega_1^A}$ is finite will be restricted to elements of Q_w. If we ever see an element $x \in Q_w$ with $x \in A \cap \Psi^{\Omega_0^A \oplus \Omega_1^A}$, we will dump all $Q_w - \{x\}$ into A and extract x from A. We now have several cases to consider in determining what action to take.

If, when we extract x from A, x leaves $\Psi^{\Omega_0^A \oplus \Omega_1^A}$ then we enumerate the axioms $\langle x, \Omega_i^{A \cup \{x\}} \rangle$ into Γ_i for $i \leq 1$ and the element x into Q_0. Assume that at some later stage, x is re-enumerated into A. After this, any element that is moved from Q_w into Q_0 will have enumerated Γ_0 and Γ_1 axioms under the assumption $x \in A$. Thus, if x is ever extracted from A, these latter axioms may give incorrect computations. Hence, whenever x is extracted from A, we will dump almost all of the elements of the stream that are larger than x into A. This gives us the following strategy: From now on, each time we see $x \notin A$ and $x \notin \Gamma_0^{\Omega_0^A}$ we dump $Q_w \cup \{y \in Q_0 : y > x\}$ into A. If we ever see $x \notin A$ and $x \in \Gamma_0^{\Omega_0^A}$, then we dump all elements of $(Q_w \cup Q_0) - \{x\}$ into A, and enumerate x into Q_1. From this point on, we monitor x and whenever we see x leave A, we dump all of $Q_w \cup Q_0 \cup \{y \in Q_1 : y > x\}$ into A.

If during this process, we ever see an element $x \in Q_w \cup Q_0 \cup Q_1$ with $x \in \Psi^{\Omega_0^A \oplus \Omega_1^A} - A$, then we have successfully diagonalized and may stop the strategy by dumping all elements of $(Q_w \cup Q_0 \cup Q_1) - \{x\}$ into A. At subsequent stages when this strategy is active, all new witnesses that enter the stream Q are enumerated into $Q_{\neq}$ and we only allow lower priority strategies that assume $A \neq \Psi^{\Omega_0^A \oplus \Omega_1^A}$ to act. The streams of these strategies are restricted to elements of $Q_{\neq}$. Strictly speaking, handling of this case is not needed in order to successfully meet the requirement, but it helps to simplify the bookkeeping.

Whenever an element x is dumped into A, we enumerate x into A and enumerate the axiom $\langle x, \emptyset \rangle$ into Γ_0 and Γ_1.

Justification of the strategy. Assume that $A = \Psi^{\Omega_0^A \oplus \Omega_1^A}$, and in addition, assume that the $\mathcal{N}$-requirements are met and, as such, A is not c.e. During the construction, we construct A in such a way as to guarantee that it is Δ_2^0.

If only a finite number of elements have been enumerated into Q_1, then the above two assumptions give us that Q_0 must contain an infinite number of elements that are not dumped into A. Choose $x \in Q_0$ such that x was not dumped into A. When x was enumerated into Q_0, say at stage s, an axiom of the form $\langle x, \Omega_0^{A \cup \{x\}}[s] \rangle$ was enumerated into Γ_0. Since x was not dumped into A, we know that after the stage at which x entered the stream of the $\mathcal{N}$-strategy, no element less than x was extracted from A since otherwise, x would have been dumped into A. In addition, when x was initially enumerated into Q_0, all elements of $Q_w - \{x\}$, at that stage, were dumped into A. Therefore $A[s] \subset A$, and so if $x \in A \Rightarrow x \in \Gamma_0^{\Omega_0^A}$. Similarly, since $x \notin Q_1$ we know that $x \notin A \Rightarrow x \notin \Gamma_0^{\Omega_0^A}$. Thus $A =^* \Gamma_0^{\Omega_0^A}$, the only possible disagreement occurring on the finite set Q_1.

Otherwise, Q_1 must contain an infinite number of elements that are not dumped into A. Similar reasoning as above gives us that $x \in A \Rightarrow x \in \Gamma_1^{\Omega_1^A}$. At the stage x was enumerated into Q_1, x was extracted from A, but we still had $x \in \Gamma_0^{\Omega_0^A}$. Thus, by dumping all of the elements of $(Q_1 \cup Q_w) - \{x\}$ into A we force $x \in \Gamma_0^{\Omega_0^A}$ permanently. If we ever saw $x \notin A$ and $x \in \Gamma_1^{\Omega_1^A}$, then by dumping $(Q_1 \cup Q_0 \cup Q_w) - \{x\}$ into A we force $x \in \Gamma_1^{\Omega_1^A}$ permanently. However, this implies that $x \in \Phi^{\Omega_0^A \oplus \Omega_1^A}$ permanently, and so by restraining $x \notin A$, we force a permanent disagreement, contradicting our assumption. Hence, we conclude that $A = \Gamma_1^{\Omega_1^A}$.

The Tree of Strategies

Fix an arbitrary effective priority ordering $\{R_e\}_{e\in\omega}$ of all $\mathcal{N}$- and $\mathcal{S}$-requirements. We define $\Sigma = \{\text{stop} < \gamma_1 < \gamma_0 < \text{wait}\}$ to be our set of outcomes. We define $\mathbf{T} \subset \Sigma^{<\omega}$ and refer to it as our *tree of strategies*. Each node $\alpha \in \mathbf{T}$ will be associated with, and thus identified with, the requirement $R_{|\alpha|}$.

We assign requirements to nodes on $\mathbf{T}$ by induction as follows: The empty node is defined to be in $\mathbf{T}$ and assigned to requirement R_0. Given an assignment to a node $\alpha \in \mathbf{T}$, we distinguish cases depending on the requirement R assigned to α:

Case 1: R is an $\mathcal{S}$-requirement: Define $\alpha^\frown\langle o\rangle \in \mathbf{T}$ for all $o \in \Sigma$.

Case 2: R is an $\mathcal{N}$-requirement: Define $\alpha^\frown\langle\text{stop}\rangle$ and $\alpha^\frown\langle\text{wait}\rangle \in \mathbf{T}$.

The Construction

The construction proceeds in stages $s \in \omega$. First, we give some conventions and definitions.

When we *initialize* a strategy, we undefine all parameters, redefine all local enumeration operators to be empty, dump $S(\alpha) - F$ into A (where F is a finite set that we do not want to be dumped into A), and set $S(\alpha) = \emptyset$. The stream $S(\emptyset)$ of the root node $\emptyset$ of our tree of strategies at any stage s is $[0, s)$. The streams $S(\alpha)$ for $\alpha \neq \emptyset$ are defined during the construction. A strategy will be *eligible* to act if it is along the current approximation $f_s \in \mathbf{T}$ to the true path $f \in [\mathbf{T}]$ of the construction. At a stage s, if $\alpha \subseteq f_s$, we will call s an α-stage.

At an α-stage s, we call a number z in the stream $S(\alpha)$ *suitable for* α if

1. z is not currently *in use for* A by any strategy (i.e., z is not the current

witness or coding number targeted for A by any strategy that has not been initialized since z has been picked).

2. z has not been dumped into A.

3. z is greater than $|\alpha|$ or any stage at which any $\beta \supseteq \alpha$ has changed any set, picked any number, or extended any enumeration operator.

4. z is greater than any stage s_β since which any $\beta \subset \alpha$ with finitary outcome $\langle\text{wait}\rangle$ or $\langle\text{stop}\rangle$ along α has first taken on this outcome.

During the course of the construction, all parameters are assumed to remain unchanged unless specified otherwise. At the end of each stage s, we will dump certain elements into A and initialize certain strategies as described below under *Ending the stage s.*

We now proceed with the construction.

Stage 0*:* Initialize all $\alpha \in \mathbf{T}$.

Stage $s > 0$: Each stage s is composed of substages $t \leq s$ such that some strategy $\alpha \in \mathbf{T}$, with $|\alpha| = t$, acts at substage t of stage s and decides which strategy will act at substage $t + 1$ or whether to end the stage. The longest strategy eligible to act during a stage s is called the current approximation to the true path at stage s and is denoted f_s.

Substage t of stage s: Suppose a strategy α of length t is eligible to act at this substage. We distinguish cases depending on the requirement R assigned to α. Choose the first case which applies.

Case 1: α is an $\mathcal{N}_\Phi$-requirement: Pick the first subcase which applies.

Case 1.1: α has not been eligible to act since its most recent initialization or has no coding number z defined: Pick z to be the least suitable witness from $S(\alpha)$. If no such z is available, end the current stage. Otherwise, enumerate z into A and end the current substage and let $\alpha^\frown\langle\text{wait}\rangle$ be eligible to act next.

Case 1.2: The coding number z is defined and $z \in A - \Phi$: Set the stream $S(\alpha^\frown\langle\text{wait}\rangle) = [s_0, s) \cap S(\alpha)$ where s_0 is the stage at which z was chosen by α. End the current substage and let $\alpha^\frown\langle\text{wait}\rangle$ be eligible to act next.

Case 1.3: The coding number z is defined and $z \in A \cap \Phi$: Extract z from A, dump $S(\alpha) - \{z\}$ into A, and enumerate z into F. End the current substage and let $S(\alpha^\frown\langle\text{stop}\rangle)$ be eligible to act next.

Case 1.4: The coding number z is defined and $z \in \Phi - A$: Set the stream $S(\alpha^\frown\langle\text{stop}\rangle) = [s_0, s) \cap S(\alpha)$ where s_0 is the stage at which z was extracted from A by α. End the current substage and let $\alpha^\frown\langle\text{stop}\rangle$ be eligible to act next.

Case 2: α is an $\mathcal{S}_{\Psi,\Omega_0,\Omega_1}$-requirement: Let s_0 be the most recent stage at which α was eligible to act. If α is not stopped, enumerate $S(\alpha)\cap[s_0, s)$ into $S(\alpha^\frown\langle\text{wait}\rangle)$. Otherwise, enumerate $S(\alpha)\cap[s_0, s)$ into $S(\alpha^\frown\langle\text{stop}\rangle)$. Pick the first subcase which applies.

Case 2.1: α has stopped since its most recent initialization: End the current substage and let $\alpha^\frown\langle\text{stop}\rangle$ be eligible to act next.

Case 2.2: There is an element $z \in S(\alpha)$, which has not been dumped into A, such that $z \in \Phi^{\Omega_0^{A-\{z\}} \oplus \Omega_1^{A-\{z\}}}$: Let z_0 be the least such z. Stop the strategy by extracting z_0 from A, if necessary, dumping $S(\alpha) - \{z_0\}$ into A, and enumerate z_0 into F. End the current substage and let $\alpha^\frown\langle\text{stop}\rangle$ be eligible to act next.

Case 2.3: There is an element $z \in S(\alpha^\frown\langle\gamma_1\rangle)$ such that $z \notin A$ but $z \in A[s_1]$

where s_1 is the last stage at which α was active: Let z_0 be the least such z, dump $S(\alpha) \cap (z_0, s)$ into A, and enumerate z_0 into F. End the current stage and continue with stage $s+1$.

Case 2.4: There is an element $z \in S(\alpha^\frown\langle\gamma_0\rangle)$ such that $z \notin A$ but $z \in \Gamma_0^{\Omega_0^A}$: Let z_0 be the least such z, dump $(S(\alpha^\frown\langle\gamma_0\rangle) \cup S(\alpha^\frown\langle\text{wait}\rangle)) - \{z_0\}$ into A, enumerate z_0 into $S(\alpha^\frown\langle\gamma_1\rangle)$, and enumerate z_0 into F. End the current substage and let $\alpha^\frown\langle\gamma_1\rangle$ be eligible to act next.

Case 2.5: There is an element $z \in S(\alpha^\frown\langle\gamma_0\rangle)$ such that $z \notin A$ but $z \in A[s_0]$ where s_0 is the last stage at which α was active: Let z_0 be the least such z, dump $S(\alpha) \cap (z_0, s)$ into A, and enumerate z_0 into F. End the current stage and continue with stage $s+1$.

Case 2.6: There is a $z \in S(\alpha^\frown\langle\text{wait}\rangle)$ that has not been dumped into A, with $z \in \Psi^{\Omega_0^{A\cup\{z\}} \oplus \Omega_1^{A\cup\{z\}}}$ and $z \notin \Psi^{\Omega_0^{A-\{z\}} \oplus \Omega_1^{A-\{z\}}}$: Let z_0 be the least such z. Enumerate the axioms $\langle z_0, \Omega_0^{A\cup\{z_0\}}\rangle$ into Γ_0 and $\langle z_0, \Omega_1^{A\cup\{z_0\}}\rangle$ into Γ_1, extract z_0 from A if necessary, dump $S(\alpha^\frown\langle\text{wait}\rangle) - \{z_0\}$ into A, and enumerate z_0 into F. End the current substage and let $\alpha^\frown\langle\gamma_0\rangle$ be eligible to act next.

Case 2.7: Otherwise: End the current substage and let $\alpha^\frown\langle\text{wait}\rangle$ be eligible to act next.

Ending the stage s: Initialize every $\beta >_L f_s$. If $\delta > f_s$ is an $\mathcal{N}$-strategy and has a defined coding number which has been dumped into A, initialize every $\beta \geq \delta$. Set $F = \emptyset$ (Where F is a set of elements that were not dumped into A).

The Verification

Let $f = \liminf_s f_s$ be the *true path* of the construction, defined more precisely by induction by

$$f(n) = \liminf_{\{s: f\upharpoonright n \subset f_s\}} f_s(n).$$

3.2.2 Lemma. i. Once an element is dumped into A, it is never removed from A.

ii. $\{A_s\}$ is a Δ^0_2-approximation to A.

Proof. i. By the definition of a suitable witness, no $\mathcal{N}$-strategy may use a dumped element as a coding location. A witness of an $\mathcal{N}$-strategy is dumped into A only when the strategy is initialized, so the next time it is active, it will choose a new, suitable witness. By the restriction on Case 2.2, an $\mathcal{S}$-requirement can never extract a dumped element.

ii. Let z be an element that was not dumped into A. By the definition of suitable, if z was picked as a coding location by an $\mathcal{N}$-requirement β, then $z \geq |\beta|$. In addition, during the construction a particular element may be picked as a coding location by a particular $\mathcal{N}$-strategy at most once. This implies that each $\mathcal{N}$-strategy β with $|\beta| \leq z$, may enumerate z into and extract z from A at most once.

If β is an $\mathcal{S}$-requirement, with $|\beta| \leq z$ then β may never enumerate z into A, and may extract z from A only in Cases 2.2 and 2.6. Due to the way elements are moved through the streams of β's possible outcomes, it is clear that β may extract z from A at most once for Case 2.2 and at most once for Case

2.6.

Since **T** is a finitely branching tree, there are only finitely many nodes of level $\leq z$, and so z may be enumerated into A at most finitely many times.

□

3.2.3 Lemma. (Tree Lemma)

i. For every $\alpha \subset f$, $S(\alpha)$ is infinite, and there are infinitely many elements of $S(\alpha)$ which are not dumped into A.

ii. f is infinite.

Proof. i. Elements are dumped into A only when an element of lesser is extracted from A under Cases 1.3, 2.2, 2.3, 2.4, 2.5, and 2.6. Furthermore, when elements are dumped into A, there is always at least one element that is not dumped into A which is less than the dumped elements.

Let $z_0 \in S(\alpha)$ be the least element that is not dumped into A. Since A is Δ^0_2, let s_0 be the least stage such that $A(z_0; s) = A(z_0)$ for all $s \geq s_0$. After stage s_0, no elements will be dumped into A due to z_0 being extracted from A. By induction, we can define the infinite sequence $\langle z_i : i \in \omega \rangle$ where z_{i+1} is the least element $\geq s_i$ not dumped into A (after stage s_i).

ii. A stage s ends prematurely during the construction only in Cases 1.1, 1.3, and 2.5 of the construction. Let s_0 be the least stage after which α is never initialized. If α is an $\mathcal{N}$-strategy, by part i, after stage s_0, α can end a stage prematurely only a finite number of times under Case 1.1, and under

Case 1.3 only once. If α is an $\mathcal{S}$-requirement, since A is Δ^0_2, after stage s_0, a particular element may cause α to end a stage prematurely only a finite number of times under Case 2.5 after which one of the other cases will act. Hence, α cannot end cofinitely many stages.

□

3.2.4 Lemma. Every strategy $\alpha \in f$ meets its requirement.

Proof. Let s_0 be the least stage after which α is never initialized.

Case 1: α is an $\mathcal{N}$-requirement: Since $S(\alpha)$ is infinite, and contains an infinite number of elements that are not dumped into A, at some stage $s_1 \geq s_0$, α chooses a suitable diagonalization witness z and enumerates z into A. If ever z enters Φ, α will extract z from Φ.

We show that no other strategy can destroy α's A-computation. Once α chooses z as a diagonalization witness, no strategy $\beta <_L \alpha$ is active after stage s_0, and z is not in the stream of any strategy $\beta > \alpha$ after stage s_1. No $\mathcal{N}$-strategy $\beta \subset \alpha$ can use z as a witness by our definition of suitable, so z cannot be enumerated into A by any other strategy after stage s_1. If some $\beta \subset \alpha$ extracts z from A after stage s_1, then α will be initialized by this action, contradicting our assumption about s_0.

Case 2: α is an $\mathcal{S}$-requirement: Let $\alpha^\frown\langle o\rangle \subset f$, and let s_0 be least such that $\alpha^\frown\langle o\rangle$ is never initialized after stage s_0.

If $\langle o\rangle = \langle \text{stop}\rangle$ then the α executed Case 2.2 at some stage $s_1 \geq s_0$ on behalf of some diagonalization witness z. No $\beta <_L \alpha$ can remove elements from the Ψ-use of z since they are not active after stage s_0. After stage s_1, all the elements of

$S(\beta)$ for $\beta >_L \alpha$ or $\beta \supset \alpha$ are greater than s_1 and, hence, no such β can remove an element from the Ψ-use of z. If after stage s_1, a strategy $\beta \subset \alpha$ removes an element $\leq s_1$ from A, then it did so under Cases 1.3, 2.2, or 2.4 which would cause α to be initialized and thus $A \neq \Psi^{\Omega_0^A \oplus \Omega_1^A}$.

Assume that $\langle o \rangle = \langle \text{wait} \rangle$. In this case, α takes on the outcomes $\langle \gamma_0 \rangle$ and $\langle \gamma_1 \rangle$ only finitely often, so say that after stage $s_1 \geq s_0$, whenever α is active, then α takes on outcome $\langle \text{wait} \rangle$. By Lemma 3.2.3.i, after stage s_1, $S(\alpha^\frown\langle \text{wait} \rangle)$ contains infinitely many elements that are not dumped into A. Since after stage s_0, α never takes on the outcomes $\langle \text{stop} \rangle$, $\langle \gamma_1 \rangle$ or $\langle \gamma_0 \rangle$, we must have that for all $z \in S(\alpha^\frown\langle \text{wait} \rangle)$, if z is not dumped into A then $z \in A \Rightarrow z \notin \Psi^{\Omega_0^A \oplus \Omega_1^A}$. This implies that $\forall z \in S(\alpha^\frown\langle \text{wait} \rangle)$, $z \notin \Psi^{\Omega_1^A \oplus \Omega_2^A}$. This then implies that $\Psi^{\Omega_0^A \oplus \Omega_1^A}$ is c.e., but by Case 1, A is not c.e, so $A \neq \Psi^{\Omega_0^A \oplus \Omega_1^A}$.

Thus, if $A = \Psi^{\Omega_0^A \oplus \Omega_1^A}$ then $\alpha^\frown\langle \gamma_i \rangle \subset f$ for some $i \leq 1$. Let $s_1 \geq s_0$ be least such that after stage s_1, $\alpha^\frown\langle \gamma_i \rangle$ is never initialized. Choose $z \in S(\alpha)$ to be an element which is not dumped into A. Let $s_2 \geq s_1$ be the stage at which Case 2.6 of the construction is applied on behalf of z. The strategy first tries to ensure that $A(z) = \Gamma_0^{\Omega_0^A}(z)$ by enumerating the axioms $\langle z, \Omega_0^{A \cup \{z\}}[s_1] \rangle$ into Γ_0 and $\langle z, \Omega_1^{A \cup \{z\}}[s_1] \rangle$ into Γ_1, and dumping all elements of $S(\alpha^\frown\langle \text{wait} \rangle) - \{z\}$ into A.

At no stage greater than s_2 may any strategy to the left of α act since this would initialize α. In addition, the elements of the stream of any strategy to the right of α are larger than z. So, if at a later stage $s_3 > s_2$, we have $A[s_3] \restriction z \nsubseteq A[s_2] \restriction z$, then some strategy $\beta \supseteq \alpha$ extracted an element $y < z$ from A via Case 1.3, 2.2, or 2.6. However, this action would cause α to be initialized. Therefore, for all $t \geq s_0$, $A[s_0] \restriction z \subseteq A[t] \restriction z \subseteq A \restriction z$, and it follows that $\Omega_0^A[s_0] \subseteq \Omega_0^{A \cup \{z\}}$ and

$\Omega_1^A[s_1] \subseteq \Omega_1^{A\cup\{z\}}$. Hence, $z \in A$ implies $z \in \Gamma_0^{\Omega_0^A}$, $z \in \Gamma_1^{\Omega_1^A}$, and $z \in \Psi^{\Omega_0^A \oplus \Omega_1^A}$.

It remains to show that if $z \notin A$ then $z \notin \Gamma_i^{\Omega_i^A}$. Assume otherwise. Let $s_4 > s_2$ be the least stage at which α is active and $z \notin A[s_4]$ and $z \in \Gamma_i^{\Omega_i^A}[s_4]$. If $i = 0$, then at stage s_4 Case 2.4 applies, and z would be enumerated into $S(\alpha^\frown\langle\gamma_1\rangle)$ causing $\alpha^\frown\langle\gamma_0\rangle$ to be initialized. If $i = 1$, then the dumping action that occurred by Case 2.4 on behalf of z ensures that $z \in \Gamma_0^{\Omega_0^A}$. Since $z \in \Gamma_0^{\Omega_0^A}$ and $z \in \Gamma_1^{\Omega_1^A}$ then $z \in \Psi^{\Omega_0^A \oplus \Omega_1^A}$, but this would imply that α would execute Case 2.2 on behalf of z and take on the outcome of $\langle\text{stop}\rangle$, thus initializing $\alpha^\frown\langle\gamma_1\rangle$. □

3.3 A Low Non-Splitting Degree

3.3.1 Theorem. ([Ahm89] (cf. [AL98])) There exists a low non-splitting enumeration degree.

We modify Theorem 3.2.1 by adding the following lowness requirement:

$$\mathcal{L}_{x,\Phi} \quad : \exists^\infty s(x \in \Phi^A[s]) \Rightarrow x \in \Phi^A.$$

Naive Strategy for $\mathcal{L}_{x,\Phi}$

This procedure guarantees lowness of A. Denote the stream associated with this strategy on the tree as S.

1. Wait for $x \in \Phi^{A\cup S}$.

2. Dump S into A and stop.

The behavior of this strategy is similar to that of the $\mathcal{N}$-requirements. The two possible outcomes on the tree are $\langle\text{wait}\rangle$ if the strategy waits at Step 1 forever and $\langle\text{stop}\rangle$ if the strategy finally stops at Step 2.

Verification (sketch) Let $\alpha \subset f$ be an $\mathcal{L}$-strategy, and s_0 be the least stage such that α is never initialized after s_0. If x never enters $\Phi^{A \cup S}$ then the requirement is met. Assume that at some stage $s_1 > s_0$, x enters $\Phi^{A \cup S}$. Then α dumps S into A and takes on the outcome $\langle \text{stop} \rangle$. No strategy to the left of α can destroy this computation since none is active after stage s_0. No strategy to the right of, or below, α can destroy the computation since all coding locations chosen before stage s_1 are dumped into A, and all chosen after are larger than the use. If a strategy above α extracts an element from the use, then this would cause α to be initialized, contradicting our assumption.

3.4 A properly Σ_2^0-Non-splitting Degree

3.4.1 Theorem. There exists a properly Σ_2^0-non-splitting enumeration degree.

Proof. In the proof of Theorem 3.2.1, replace the requirement $\mathcal{N}_\Phi$ by the following requirement:

$$\mathcal{N}_{B,\Phi,\Psi} \quad : B = \Phi^A \text{ and } A = \Psi^B \Rightarrow \exists x \in B(\lim_s B_s(x) \uparrow).$$

Strategy for $\mathcal{N}_{B,\Phi,\Psi}$ [CC88]

1. Choose a suitable witness x from the stream $S(\alpha)$ and enumerate x into A.

2. Wait for $x \in \Psi^B$ via some minimal finite set $D \subseteq B$ such that $x \in \Psi^D$ and $D \subseteq \Phi^A$. Once this happens, dump $S(\alpha) - \{x\}$ into A.

3. Remove x from A (possibly allowing $D \not\subseteq \Phi^A$).

4. Wait for $x \notin \Psi^B$.

5. Enumerate x into A (forcing $D \subseteq \Phi^A$).

6. Wait for $x \in \Psi^B$ and $D \subseteq B$.

7. Go to Step 3.

Verification (sketch) We will have two outcomes for this strategy: $x \in A$, which will correspond to Steps 2 and 6 of the strategy, and $x \notin A$, which corresponds to Step 4. The first time we move to Step 3, we dump $S(\alpha) - \{x\}$ into A, forcing $D \subseteq \Phi^A$ whenever $x \in A$. Thus, if $B = \Phi^A$ and $A = \Psi^B$, then we loop through Step 7 infinitely often. Due to fact that D is finite, each time we loop through Step 7, some element of D has left B at Step 4 and re-entered B at Step 6. Thus we know that for some $d \in D$, $\lim_s B_s(d) \uparrow$, and so $\langle B_s \rangle$ is a Σ^0_2-approximation to B. Since this is true for every $B \equiv_e A$, A must be properly Σ^0_2.

The $\mathcal{S}$-strategy is modified by dumping only those elements into A which had axioms enumerated into the Γ_i while x was an element of A. Since the membership of x in A is changed only when α takes on one of the $\langle \gamma_i \rangle$ outcomes, we guarantee that infinitely elements of $S(\alpha^\frown\langle \gamma_i \rangle)$ are not dumped into A by this action.

With these modifications we immediately see that the $\mathcal{S}$-strategies still meet their requirements, and the set constructed is non-splitting. □

3.5 Bounding Non-splitting Degrees

3.5.1 Theorem. The non-splitting degrees are downwards dense in the Δ^0_2-enumeration degrees. i.e. every Δ^0_2-enumeration degree bounds a non-splitting Δ^0_2-enumeration degree.

Given a Δ_2^0-approximation $\langle A_s \rangle$ to a set A, we construct in stages an enumeration operator Θ meeting the following requirements:

$\mathcal{R}$: $B = \Theta^A$

$\mathcal{S}_{\Psi,\Omega_0,\Omega_1}$: $B = \Psi^{\Omega_0^B \oplus \Omega_1^B} \Rightarrow \exists \Gamma_0, \Gamma_1 [B =^* \Gamma_0^{\Omega_0^B} \text{ or } B =^* \Gamma_1^{\Omega_1^B}] \text{ or } \exists \Lambda [A = \Lambda]$,

$\mathcal{N}_\Phi$: $B = \Phi \Rightarrow \exists \Delta (A = \Delta)$.

Here Δ, Γ_0, Γ_1, and Λ are enumeration operators built by us and local to only the strategy by which they are built. The set B is Δ_2^0 and thus by setting $\mathbf{b} = \deg_e(B)$, we prove the theorem.

Naive Strategy for $\mathcal{N}_\Phi$

This is a modified Friedberg permitting strategy.

1. Set $n = 0$.

2. Choose a number c_n larger than any number seen so far in the construction.

3. While $c_n \notin \Phi$, enumerate $\langle c_n, A \upharpoonright c_n \rangle$ into Θ.

4. When c_n enters Φ, stop enumerating axioms, enumerate $\bigcap \{D : \langle c_n, D \rangle \in \Theta\}$ into Δ, return to Step 2 and start cycle $n + 1$.

5. From now on, while $c_n \notin B$, halt all processing for all $m > n$.

6. When c_n re-enters B, resume processing for all $m > n$.

Analysis of the $\mathcal{N}_\Phi$-strategy:

During the course of the construction, the axioms $\langle c_n, A \upharpoonright c_n \rangle$ will be enumerated in such a way so as to guarantee that as each cycle n of the strategy passes through Step 4, we have $\bigcap\{D : \langle c_m, D\rangle \in \Theta\} \subseteq \bigcap\{D : \langle c_n, D\rangle \in \Theta\}$ for all $m < n$. Since A is Δ^0_2 and the c_n are strictly increasing, this will allow us to conclude that if we choose infinitely many c_n, and all of them are eventually in B, then $A = \Delta$. By assumption, however, this cannot happen since A is not c.e. Therefore, there is a least n for which we either wait forever at Step 3, which yields $c_n \in B - \Phi$, or we return to Step 5 infinitely often, which yields $c_n \in \Phi - B$. Since the use of each each axiom defined for each c_n contains only elements less than c_n, it follows that B is Δ^0_2. Therefore, in the latter case we will eventually wait forever at Step 5.

Naive Strategy for $\mathcal{S}_{\Psi,\Omega_0,\Omega_1}$

This strategy will build two enumeration operators Γ_0 and Γ_1 such that if $B = \Psi^{\Omega_0^B \oplus \Omega_1^B}$, then either $B =^* \Gamma_0^{\Omega_0^B}$ or $B =^* \Gamma_1^{\Omega_1^B}$. For this strategy, we partition the stream Q into streams Q_W, Q_0, Q_1, and Q_S, and an additional set $\mathcal{D}$. The stream Q_W is the set of elements x on which the strategy is waiting to see if ever $B(x) = \Psi^{\Omega_0^B \oplus \Omega_1^B}(x) = 1$. The streams Q_i, for $i \leq 1$, are the sets of elements that witness $B = \Gamma_i^{\Omega_i^B}$ (assuming that $B = \Psi^{\Omega_0^B \oplus \Omega_1^B}$). The set $\mathcal{D}$ is a set of elements that may be used to diagonalize B against $\Psi^{\Omega_0^B \oplus \Omega_1^B}$. Finally, the stream Q_S is the set of elements which believe that we have successfully diagonalized B against $\Psi^{\Omega_0^B \oplus \Omega_1^B}$ using an element of $\mathcal{D}$.

As the strategy proceeds, elements may move from Q_W to Q_0 to Q_1 to $\mathcal{D}$, or directly into Q_S, but never in any other order. Each time an element moves between streams, we will dump all elements from the streams through which it has already moved into B so as to preserve any Γ_i-computation that we may see.

When we dump an element y into B, we enumerate the axiom $\langle y, \emptyset\rangle$ into Θ.

The basic strategy is as follows:

1. Set $n = 0$.

2. Wait for an $x \in Q_W$ such that $B(x) = \Psi^{\Omega_0^B \oplus \Omega_1^B}(x) = 1$ and x is not dumped into B. Let x_n be the least such x. While we are waiting at this step, enumerate any new elements of the stream Q into Q_W.

3. Extract x_n from Q_W, dump $Q_W - \{x_n\}$ into B, and enumerate x_n into Q_0.

4. Enumerate the axiom $\langle x_n, \Omega_i^B\rangle$ into Γ_i for $i \leq 1$.

5. Begin cycle $n + 1$ starting at Step 2.

6. If $x_n \notin B$ and $x_n \notin \Gamma_0^{\Omega_0^B}$, cancel all cycles $m > n$, dump $\{y \in Q_0 : y > x_n\} \cup Q_W$ into B, and begin cycle $n+1$ starting at Step 2. Remain with this cycle at Step 6.

7. Otherwise, if $x_n \notin B$ and $x_n \in \Gamma_0^{\Omega_0^B}$, cancel all cycles $m > n$, extract x_n from Q_0, dump $(Q_W \cup Q_0) - \{x_n\}$ into B, enumerate x_n into Q_1, and begin cycle $n + 1$ starting at Step 2. Go on to Step 8 with this cycle.

8. If $x_n \notin B$ and $x_n \notin \Gamma_1^{\Omega_1^B}$, cancel all cycles $m > n$, dump $\{y \in Q_1 : y > x_n\} \cup Q_0 \cup Q_W$ into B, and begin cycle $n+1$ starting at Step 2. Remain with this cycle at Step 8.

9. Otherwise, if $x_n \notin B$ and $x_n \in \Gamma_1^{\Omega_1^B}$, cancel all cycles $m > n$, extract x_n from Q_1, dump $(Q_W \cup Q_0 \cup Q_1) - \{x_n\}$ into B. Enumerate x_n into $\mathcal{D}$ and $\bigcap\{D : \langle x_n, D\rangle \in \Theta\}$ into Λ. Go on to Step 10 with this cycle.

10. From now on, if $x_n \notin B$, do the following: Let m_0 be least such that for all $m \geq m_0$, if x_m is defined then $x_m \notin \mathcal{D}$. Cancel all cycles $m \geq m_0$, dump $Q_W \cup Q_0 \cup Q_1$ into B, and enumerate any new elements of Q into Q_S. When x_n re-enters B, begin cycle m_0 starting at Step 2.

Analysis and outcomes of the $\mathcal{S}_{\Psi,\Omega_0,\Omega_1}$-strategy:

The dumping action that occurs when an element moves from stream to stream ensures that, via the Γ_i-axioms that are enumerated in Step 4, we have $x \in B \Rightarrow x \in \Gamma_i^{\Omega_i^B}$ for $i \leq 1$. This can bee seen to hold as follows. If x is dumped into B then we just enumerate the axiom $\langle x, \emptyset \rangle$ into Γ. If x is not dumped into B, assume that Γ_i-axioms for x are enumerated at stage s. After stage s, no element less than x leaves B since otherwise, x would have been dumped into B. Furthermore, at stage s, we dump all elements which are currently in some stream and greater than x into B, giving $B[s] - \{x\} \subseteq B$.

If at some stage $s' > s$, we see $x \in \Gamma_0^{\Omega_0^B}[s'] - B[s']$, then the dumping action that occurs at stage s ensures that $B[s'] - \{x\} \subseteq B$ and thus $x \in \Gamma_0^{\Omega_0^B}$ permanently. Therefore, assuming that x is not enumerated into $\mathcal{D}$, we have $x \in B \Leftrightarrow x \in \Gamma_1^{\Omega_1^B}$. If, however, there is a stage $s'' > s'$ at which we see $x \in \Gamma_1^{\Omega_1^B}[s''] - B[s'']$, then a similar argument as above gives us that $B[s''] - \{x\} \subseteq B$ and hence $x \in \Gamma_1^{\Omega_1^B}$.

Therefore, for all $x \in \mathcal{D}$, we know that $x \in \Gamma_0^{\Omega_0^B} \cap \Gamma_1^{\Omega_1^B}$. By the Γ_i-axioms that were enumerated at stage s, this implies that $x \in \Psi^{\Omega_0^B \oplus \Omega_1^B}$. As in the analysis of the $\mathcal{N}$-strategy, for each $x_n \in \mathcal{D}$ the axioms $\langle x_n, A \upharpoonright x_n \rangle$ will be enumerated in such a way so as to guarantee that $\bigcap \{D : \langle x_m, D \rangle \in \Theta\} \subseteq \bigcap \{D : \langle x_n, D \rangle \in \Theta\}$ for all $m < n$. Since A is Δ_2^0 and the $x_n \in \mathcal{D}$ are strictly increasing, this will allow us to conclude that if $\mathcal{D}$ is infinite and $\mathcal{D} \subseteq B$, then $A = \Lambda$. By assumption, however, this cannot happen since A is not c.e. Therefore, as B is Δ^0_2, either there is a least n for which we wait forever at Step 10, which yields $x_n \in B - \Psi^{\Omega_0^B \oplus \Omega_1^B}$, or $\mathcal{D} \subseteq B$ and hence is finite.

This gives us the following possible outcomes for the strategy:

wait: Wait at Step 2 forever for some n. In this case we have that either B is c.e. or $B \neq \Psi^{\Omega_0^B \oplus \Omega_1^B}$. By the satisfaction of the $\mathcal{N}$-strategies, B cannot be c.e.

γ_0: Infinitely many cycles end up waiting at Step 6, but only finitely many at Steps 8 and 10. Then $B =^* \Gamma_0^{\Omega_0^B}$ if $B = \Psi^{\Omega_0^B \oplus \Omega_1^B}$.

γ_1: Infinitely many cycles end up waiting at Step 8 and only finitely many at Step 10. Then $B =^* \Gamma_1^{\Omega_1^B}$ if $B = \Psi^{\Omega_0^B \oplus \Omega_1^B}$.

stop: Either infinitely many cycles end up waiting at Step 10, which gives A is c.e., or a single cycle waiting at Step 10 halts all higher cycles forever, which yields $B \neq \Psi^{\Omega_0^B \oplus \Omega_1^B}$.

Interactions Between the Strategies

The Γ_i-axioms defined by an $\mathcal{S}$-strategy α are dependent on the assumption that no diagonalization witness of a higher priority strategy, nor any element of the stream of any strategy to the left of α, leaves B. To handle this dependency during the construction, if such an element does leave B, we will initialize α. Since B is Δ^0_2, each element of a stream can initialize lower-priority strategies only finitely often. Furthermore, if α is on the true path, there will be only finitely many elements in streams to the left of α, and hence α will be initialized only finitely often.

One $\mathcal{N}$-strategy Below One $\mathcal{S}$-strategy

Assume that there is a single $\mathcal{S}$-strategy α and a single $\mathcal{N}$-strategy β of lower priority. The dumping mechanism of α in Steps 6 and 8 can potentially injure β in the following manner. Assume that for $i \leq 1$, β has chosen diagonalization witnesses c_i and has enumerated enumerated the non-empty sets $D_i = \bigcup\{D : \langle x_i, D\rangle \in \Theta\}$ into Δ. Clearly, if an element of D_i leaves A then c_i will leave B. In addition, since α is above β on the tree, both c_0 and c_1 are elements of, say, stream Q_0 of α, and as such, α has defined Γ_0-axioms for each of them.

If at some stage s we see $c_0 \notin B[s]$ and $c_0 \notin \Gamma_0^{\Omega_0^B}[s]$, then, via Step 6 of the $\mathcal{S}$-strategy, c_1 will be dumped into B. If at a later stage c_0 re-enters B, elements of $D_1 - D_0$ are free to leave A without causing c_1 to leave B, thus destroying our Δ-computation.

To avoid this eventuality we change both the manner in which an $\mathcal{N}$-strategy chooses diagonalization witnesses and the dumping action of the $\mathcal{S}$-strategy. Assume that β has chosen $c_0, \ldots, c_n$ as its diagonalization witnesses and currently $\{c_0, \ldots, c_n\} \subseteq B$ and so is looking for a new witness. In this case, β will only choose a c_{n+1} whose previous Θ-axioms were enumerated at stages during which $\{c_0, \ldots, c_n\} \subseteq B$. Since there are infinitely many elements from which α can choose c_{n+1}, if truly $\{c_0, \ldots, c_n\} \subseteq B$ then a witness meeting this criterion will be found. The reason that we do this is to ensure that if $c_i \notin B$ then $c_j \notin B$ for all $j > i$.

This fact will be used by the modification to the $\mathcal{S}$-strategy. In Step 6, if α sees $x_i \notin B$, it will only dump those $x_j \in Q_0$ into B which have $x_j > x_i$ and $x_j \in B$ while $x_j \notin \Gamma_0^{\Omega_0^B}$. The change in Step 8 is similar. This will ensure that no current diagonalization witness of any lower priority $\mathcal{N}$-strategy will be dumped into B.

One $\mathcal{S}$-strategy Below One $\mathcal{S}$-strategy

This case is similar to that of one $\mathcal{N}$-strategy below one $\mathcal{S}$-strategy. The change here is that we will only enumerate into Q_W elements of the incoming stream that have had axioms defined while $\mathcal{D} \subseteq B$. This will then ensure that if any element of $\mathcal{D}$ leaves B then all larger elements will also leave, and thus we avoid having an incorrect Λ-computation.

The Tree of Strategies

Fix an arbitrary effective priority ordering $\{R_e\}_{e\in\omega}$ of all $\mathcal{N}$- and $\mathcal{S}$-requirements. We define $\Sigma = \{\text{stop} < \gamma_1 < \gamma_0 < \text{wait}\}$ to be our set of outcomes. We define $\mathbf{T} \subset \Sigma^{<\omega}$ and refer to it as our *tree of strategies*. Each node $\alpha \in \mathbf{T}$ will be associated with, and thus identified with, the requirement $R_{|\alpha|}$.

We assign requirements to nodes on $\mathbf{T}$ by induction as follows: The empty node is defined to be in $\mathbf{T}$ and assigned to requirement R_0. Given an assignment to a node $\alpha \in \mathbf{T}$, we distinguish cases depending on the requirement R assigned to α:

Case 1: R is an $\mathcal{S}$-requirement: Define $\alpha^\frown\langle o\rangle \in \mathbf{T}$ for all $o \in \Sigma$.

Case 2: R is an $\mathcal{N}$-requirement: Define $\alpha^\frown\langle \text{wait}\rangle \in \mathbf{T}$.

The Construction

The construction proceeds in stages $s \in \omega$. Before beginning, we give some conventions and definitions.

A strategy will be *eligible* to act if it is along the current approximation $f_s \in \mathbf{T}$ to the true path $f \in [\mathbf{T}]$ of the construction. At a stage s, if $\alpha \subseteq f_s$, we will call s an α-stage.

The stream $S(\emptyset)$ of the root node $\emptyset$ of our tree of strategies at any stage s is $[0, s)$. The streams $S(\alpha)$ for $\alpha \neq \emptyset$ are defined during the construction. When we *initialize* a strategy, we cancel all parameters and local enumeration operators, dump $S(\alpha) - F$ into B (where F is a finite set that we do not want to be dumped into B), and set $S(\alpha) = \emptyset$. When we dump an element x into B, we enumerate the axiom $\langle x, \emptyset \rangle$ into Θ.

During the course of the construction, all parameters are assumed to remain unchanged unless specified otherwise. We also assume that for all odd stages s, $A_s = A_{s+1}$. At the end of each even stage s, we will dump certain elements into A and initialize certain strategies as described below under *Ending the stage s*.

We now proceed with the construction.

Stage $s = 0$: Initialize all $\alpha \in \mathbf{T}$.

Stage $s + 1$ is odd: For every $\alpha \in \mathbf{T}$, do the following:

1. If there is an $x \in S(\alpha)$ such that $x \in B_s$ and $x \notin B_{s+1}$, initialize all $\beta >_L \alpha$.

2. If α is an $\mathcal{N}$-strategy and there is an $x \in \Delta$ such that $x \in A_s$ and $x \notin A_{s+1}$, then initialize all $\beta \geq \alpha^\frown\langle \text{wait} \rangle$.

3. If α is an $\mathcal{S}$-strategy and there is an $x \in \Lambda$ such that $x \in A_s$ and $x \notin A_{s+1}$, then initialize all $\beta \geq \alpha^\frown\langle \text{stop} \rangle$ and cancel Γ_0 and Γ_1.

Substage t of even stage $s+1$: Suppose a strategy α of length t is eligible to act at this substage. We distinguish cases depending on the requirement R assigned to α. Choose the first case which applies.

Case 1: α is an $\mathcal{N}$-strategy: If α has not been eligible to act since its last initialization, set $n = 0$. Choose the first subcase with applies.

Case 1.1: For some $m < n$, $c_m \notin B$: Enumerate $S(\alpha) - \{c_i : i \leq n\}$ into $S(\alpha^\frown\langle\text{wait}\rangle)$, end the current substage and let $\alpha^\frown\langle\text{wait}\rangle$ be eligible to act next.

Case 1.2: c_n is undefined: Choose $c_n \in S(\alpha)$ to be the least such that $c_n > a(B, \{c_i : i < n\})$ and is not dumped into B (where $a(B, \{c_i : i < n\})$ is the age of the set $\{c_i : i < n\}$ in the set B as defined in Definition 1.5.4). If c_n exists, enumerate $\langle c_n, A \restriction c_n\rangle$ into Θ and dump $S(\alpha) - \{c_i : i \leq n\}$ into Θ^A. If no such c_n exists, dump $S(\alpha) - \{c_i : i < n\}$ into Θ^A. In either case, initialize all $\beta \supseteq \alpha^\frown\langle\text{wait}\rangle$, and end the current stage.

Case 1.3: c_n is defined and $c_n \notin \Phi$: Enumerate $\langle c_n, A \restriction c_n\rangle$ into Θ. Enumerate $S(\alpha) - \{c_i : i \leq n\}$ into $S(\alpha^\frown\langle\text{wait}\rangle)$, end the current substage, and let $\alpha^\frown\langle\text{wait}\rangle$ be eligible to act next.

Case 1.4: Otherwise c_n is defined and $c_n \in \Phi$: Enumerate $\bigcap\{D : \langle c_n, D\rangle \in \Theta\}$ into Δ, set $n = n + 1$, end the current substage and let $\alpha^\frown\langle\text{wait}\rangle$ be eligible to act next.

Case 2: α is an $\mathcal{S}$-strategy: If this is the first stage at which α has been eligible to act since it was last initialized, set $n = 0$. Let s_0 be the last stage at which α was eligible to act since its last initialization, or, if no such stage exists, let s_0 be the stage of the most recent initialization.

Case 2.1: There is a stage s', with $s_0 \leq s' \leq s$, such that $\{d_i : i < n\} \nsubseteq B[s']$: Enumerate $S(\alpha) \cap [s_0, s)$ into $S(\alpha^\frown\langle\text{stop}\rangle)$, end the current substage, and let $\alpha^\frown\langle\text{stop}\rangle$ be eligible to act next.

Case 2.2: Otherwise $\{d_i : i < n\} \subseteq B[s']$ for $s_0 \leq s' \leq s$: Enumerate $S(\alpha) \cap$

$[s_0, s)$ into $S(\alpha^\frown\langle \text{wait}\rangle)$. Choose the first subcase which applies:

Case 2.2.1: There exists a $z \in S(\alpha^\frown\langle\gamma_1\rangle)$ such that $z \notin B$ but $z \in \Gamma_1^{\Omega_1^B}$: Let z_0 be the least such z. Extract z_0 from $S(\alpha^\frown\langle\gamma_1\rangle)$ and dump $(S(\alpha^\frown\langle\gamma_1\rangle) \cup S(\alpha^\frown\langle\gamma_0\rangle) \cup S(\alpha^\frown\langle \text{wait}\rangle)) - \{z_0\}$ into B. Enumerate z_0 into F, enumerate $S(\alpha) \cap [s_0, s)$ into $S(\alpha^\frown\langle \text{stop}\rangle)$, and enumerate $\bigcap\{G : \langle z_0, G\rangle \in \Theta\}$ into Λ. Set $d_n = z_0$, and let $n = n + 1$. Cancel Γ_0 and Γ_1. End the current substage and let $\alpha^\frown\langle \text{stop}\rangle$ be eligible to act next.

Case 2.2.2: There exists a $z \in S(\alpha^\frown\langle\gamma_1\rangle)$ such that $z \notin B$ but $z \in B[s_0]$: Let z_0 be the least such z. For all $z \in S(\alpha^\frown\langle\gamma_1\rangle)$ with $z > z_0$, if $z \in B$ and $z \notin \Gamma_1^{\Omega_1^B}$, dump z into B. End the current substage and let $\alpha^\frown\langle \text{wait}\rangle$ be eligible to act next.

Case 2.2.3: There exists a $z \in S(\alpha^\frown\langle\gamma_0\rangle)$ such that $z \notin B$ but $z \in \Gamma_0^{\Omega_0^B}$: Let z_0 be the least such z. Extract z_0 from $S(\alpha^\frown\langle\gamma_0\rangle)$ and dump $(S(\alpha^\frown\langle\gamma_0\rangle) \cup S(\alpha^\frown\langle \text{wait}\rangle)) - \{z_0\}$ into B. Enumerate z_0 into $S(\alpha^\frown\langle\gamma_1\rangle)$ and F. Cancel Γ_0. End the current substage and let $\alpha^\frown\langle\gamma_1\rangle$ be eligible to act next.

Case 2.2.4: There exists a $z \in S(\alpha^\frown\langle\gamma_0\rangle)$ such that $z \notin B$ but $z \in B[s_0]$: Let z_0 be the least such z. For all $z \in S(\alpha^\frown\langle\gamma_0\rangle)$ with $z > z_0$, if $z \in B$ and $z \notin \Gamma_0^{\Omega_1 0^B}$, dump z into B. End the current substage and let $\alpha^\frown\langle \text{wait}\rangle$ be eligible to act next.

Case 2.2.5: There exists a $z \in S(\alpha^\frown\langle \text{wait}\rangle)$, which has not been dumped into B, such that $z \in B \cap \Psi^{\Omega_0^B \oplus \Omega_1^B}$: Let z_0 be the least such z. For $i \leq 1$, enumerate $\langle z_0, \Omega_i^B\rangle$ into Γ_i. Extract z_0 from $S(\alpha^\frown\langle \text{wait}\rangle)$ and dump $S(\alpha^\frown\langle \text{wait}\rangle) - \{z_0\}$ into B, enumerate z_0 into $S(\alpha^\frown\langle\gamma_0\rangle)$ and into F. End the current substage and let $\alpha^\frown\langle\gamma_0\rangle$ be eligible to act next.

Case 2.2.6: Otherwise: End the current substage and let $\alpha^\frown\langle\text{wait}\rangle$ be eligible to act next.

Ending the stage s: Initialize every $\beta >_L f_s$. Set $F = \emptyset$ (where F is a set of elements that were not dumped into B).

Verification

Let $f = \liminf_s f_s$ be the *true path* of the construction, defined more precisely by recursion as

$$f(n) = \liminf_{\{s: f\upharpoonright n \subset f_s\}} f_s(n).$$

3.5.2 Lemma. i. Once an element is dumped into B, it is never removed from B.

ii. $\langle B_s \rangle$ is a Δ^0_2-approximation to B.

Proof. i. Immediate since an element x is dumped into B by enumerating the axiom $\langle x, \emptyset \rangle$ into Θ.

ii. Consider an element $x \in \omega$. Only $\mathcal{N}$-strategies can enumerate x-axioms into Θ, and they are of the form $\langle x, D \rangle$, where $D = A[s] \upharpoonright x$ for some s. Therefore, there are only finitely many such axioms in Θ, and since A is Δ^0_2, $\lim_{s\to\infty} \Theta^A(x; s)$ exists.

□

3.5.3 Remark. Since every Δ^0_2-degree bounds a low degree, replacing A by a low set $\hat{A}$ in the statement of the theorem makes the above result trivial.

3.5.4 Lemma. If $\alpha \subseteq f$ is an $\mathcal{N}$-strategy and infinitely many elements of $S(\alpha)$ are not dumped into B by higher priority requirements, then

i. no diagonalization witness c_n is ever dumped into B after α's last initialization.

ii. α meets its requirement.

iii. $S(\alpha^\frown\langle\text{wait}\rangle)$ contains infinitely many elements which are not dumped into B by any higher priority requirement.

Proof. i. Let s be the least stage after which α is never initialized, and let c_n be the least diagonalization witness for α which is dumped into B at stage, say, $s_n > s$. Since α is not initialized after stage s, we may assume that for all $\beta <_L \alpha$, no element of $S(\beta)$ leaves B, and for all $\beta \subset \alpha$, no diagonalization witness leaves B. For $\beta > \alpha$, no element of $S(\beta)$ leaving B can cause any diagonalization witness of α to be dumped into B since any such element is larger than the use of any defined c_n-axiom, and will be dumped into B by stage s_n. Therefore, the only cases in which c_n could have been dumped into B are Cases 2.2.2 and 2.2.4.

This implies that there is a $\beta \in \mathbf{T}$ and an $i \leq 1$ such that $c_n \in S(\beta^\frown\langle\gamma_i\rangle)$ and $\beta^\frown\langle\gamma_i\rangle \subseteq \alpha$. In addition, there must be an $x_0 \in S(\beta)$, with $x_0 < c_n$, and a $j \leq 1$ such that

$$x_0 \notin B[s_n] \Rightarrow c_n \in B[s_n] \text{ and } c_n \notin \Gamma_j^{\Omega_j^B}[s_n].$$

Since, for all $\delta <_L \alpha$, no element of $S(\delta)$ left B, it follows that $x \in S(\beta^\frown\langle\gamma_i\rangle)$ and $j = i$.

In every case of the construction, whenever a strategy δ is active, δ either dumps all of the new elements of $S(\delta)$ into B, selects one as a new diagonalization witness and dumps the rest into B, or enumerates all of them into the stream of its current outcome. By induction it follows that at the stage an element is chosen by a strategy β as a coding location, all smaller elements of $S(\delta)$, for $\delta \subseteq \beta$, are either diagonalization witnesses of δ, dumped into B, or are in a stream $S(\epsilon)$ for some $\epsilon <_L \beta$. Therefore, x_0 is a diagonalization witness for some $\delta \subseteq \alpha$.

If x_0 is a diagonalization witness for some $\delta \subset \alpha$, then α would have been initialized at stage s_n, contrary to assumption. If x_0 is a diagonalization witness for α then $x_0 = c_m$ for some $m < n$. However, this would imply that $c_n \notin B[s_n]$, and c_n would not be dumped into B, also contrary to assumption.

ii. Assume that for some i, $c_i \notin \Phi$. Since A is Δ^0_2 and by Case 1.3, α will eventually enumerate enough axioms into Θ so that $c_i \in B$.

So, assume that α chooses infinitely many diagonalization witnesses c_i, and that for all i, $c_i \in \Phi$. Since B is Δ^0_2, we may assume that eventually each $c_i \in B$, since otherwise there would be some i such that $c_i \notin B$ and this would eventually be permanent. After this, we would choose no more diagonalization witnesses since we would have successfully diagonalized B against Φ. Since each c_i is not dumped into B, and $c_i \in B$, we must have

$$\bigcap \{D : \langle c_i, D\rangle \in \Theta[s]\} \subseteq \Lambda[s] \subseteq A[s] \subseteq A,$$

where s is the stage at which Case 1.4 was applied to c_i. Thus, $\Lambda \subseteq A$. Choose $x \in A$ and let s_x be the least stage after which x never leaves A.

Then for all $c_i > \max(x, s_x)$,

$$x \in \bigcap \{D : \langle c_i, D\rangle \in \Theta[s]\} \subseteq \Lambda.$$

Therefore $A \subseteq \Lambda$.

iii. By Lemmas 3.5.4.iii and 3.5.5.ii, $S(\alpha)$ contains infinitely many elements which are not dumped into B by higher priority strategies. The only time that α dumps elements into B is in Case 1.2, which by Lemma 3.5.4.ii can happen only finitely often. The only time that any $\beta \supset \alpha$ can dump an element of $S(\alpha)$ into B after stage s is in Case 2.2.2 or Case 2.2.4. Choose $x \in S(\alpha)$ which has not been dumped into B. Since B is Δ^0_2, x will cause only finitely many elements of $S(\alpha)$ to be dumped via Cases 2.2.2 and 2.2.4. Since $S(\alpha)$ is infinite, the conclusion follows.

□

3.5.5 Lemma. If $\alpha \subseteq f$ is an $\mathcal{S}$-strategy and infinitely many elements of $S(\alpha)$ are not dumped into B by higher priority requirements, then

i. α meets its requirement.

ii. If $\alpha^\frown\langle o\rangle \subset f$, then $S(\alpha^\frown\langle o\rangle)$ contains infinitely many elements which are not dumped into B by any $\beta \subseteq \alpha$.

Proof. i. Let s_0 be the least stage after which α is never initialized. Assume that $\alpha^\frown\langle\text{stop}\rangle \subset f$. A proof similar to the one found in Lemma 3.5.4.i shows that no defined diagonalization witness d_i is ever dumped into B after stage s_0. In addition, if α chooses infinitely many d_i, and for all i, $d_i \in B$, then by an argument similar to that in Lemma 3.5.4.ii, A is c.e. Therefore, for some i, $d_i \notin B$. Let n be the least such i and let t_n be the stage at which α enumerated the axioms $\langle d_n, \Omega_0^B[t_n]\rangle$ into Γ_0 and $\langle d_n, \Omega_1^B[t_n]\rangle$ into Γ_1. After stage s_0, no element in any $S(\beta)$ for $\beta <_L \alpha$ left B. Furthermore, due to the dumping action at even stages and Cases 2.2.2, 2.2.3, 2.2.4, and 2.2.5, every element of $S(\alpha)[t_n] - \{d_i : i \leq n\}$ has been dumped into B. As in Lemma 3.5.4.i, every element that is not eventually permanently picked as a diagonalization witness by some strategy is dumped into B, no diagonalization witness of any strategy $\beta \subset \alpha$ leaves B after stage t_n, and the extraction of any diagonalization witnesses of any $\beta >_L \alpha$ will not harm the Γ_i- or Θ-computations of any d_i. Therefore, by Case 2.2.3, $\Omega_0^B[t_n] \subseteq \Omega_0^{B-\{d_n\}}$ and by Case 2.2.1, $\Omega_1^B[t_n] \subseteq \Omega_1^{B-\{d_n\}}$. Hence $\Psi^{\Omega_0^B \oplus \Omega_1^B}(d_n) = 1 \neq 0 = B(d_n)$.

Assume that $B = \Psi^{\Omega_0^B \oplus \Omega_1^B}$. Furthermore, assume that $\alpha^\frown\langle\text{wait}\rangle \subset f$, and let $s_1 \geq s_0$ be least such that at no stage after s_1 is $\alpha^\frown\langle\text{wait}\rangle$ initialized. For every x that enters $S(\alpha^\frown\langle\text{wait}\rangle)$ after stage s_1, if x is not dumped into B then either $B(x) = \Psi^{\Omega_0^B \oplus \Omega_1^B}(x) = 0$ or $B(x) \neq \Psi^{\Omega_0^B \oplus \Omega_1^B}(x)$, since otherwise the strategy would execute Case 2.2.5 for some x and initialize $\alpha^\frown\langle\text{wait}\rangle$. If $B(x) = \Psi^{\Omega_0^B \oplus \Omega_1^B}(x) = 0$ for all such x, then B is c.e. since there are only finitely many elements in streams to the left of $\alpha^\frown\langle\text{wait}\rangle$, and after stage t_n, if an element is not dumped into B, then it eventually enters $S(\alpha^\frown\langle\text{wait}\rangle)$ and hence is not in B. However, by Lemma 3.5.6, f is infinite, and by Lemma 3.5.5, every $\mathcal{N}$-strategy meets its requirement, so B is not c.e. Therefore, there is some $x \in S(\alpha^\frown\langle\text{wait}\rangle)$ such that $\Psi^{\Omega_0^B \oplus \Omega_1^B}(x) \neq B(x)$.

Therefore $\alpha^\frown\langle\gamma_i\rangle \subset f$ for some $i \leq 1$. Let $s_2 \geq s_0$ be the least stage after which $\alpha^\frown\langle\gamma_i\rangle$ is not initialized. Choose $x \in S(\alpha^\frown\langle\gamma_i\rangle)$ which is not dumped into B. All numbers greater than x in the Γ_0- and Γ_1-uses of x have been dumped into B, and all numbers less than x have stabilized before the Γ_0- and Γ_1-axioms for x were defined, since otherwise x would have been dumped into B. Therefore, $x \in B$ implies $x \in \Gamma_0^B$ and $x \in \Gamma_1^B$.

If $i = 0$ and $x \in S(\alpha^\frown\langle\gamma_0\rangle)$, then $x \notin B$ implies $x \notin \Gamma_0^B$ since otherwise α would have executed Case 2.2.3 on behalf of x, causing $\alpha^\frown\langle\gamma_0\rangle$ to be initialized after stage s_2. Thus $B =^* \Gamma_0^B$.

If $i = 1$ and $x \in S(\alpha^\frown\langle\gamma_1\rangle)$, then $x \notin B$ implies $x \notin \Gamma_1^B$ since otherwise α would have executed Case 2.2.1 on behalf of x, causing $\alpha^\frown\langle\gamma_1\rangle$ to be initialized after stage s_2. Thus $B =^* \Gamma_1^B$.

ii. If $\alpha^\frown\langle o\rangle \subset f$, by the same argument used in Lemma 3.5.4.iii, it is not the case that cofinitely many elements of $S(\alpha)$ are dumped into B by higher priority strategies. Let $s_3 > s_0$ be least such that $\alpha^\frown\langle o\rangle$ is not initialized after stage s_3. If $\alpha^\frown\langle\text{wait}\rangle \subset f$, then after stage s_3, no higher priority strategy dumps any member of $S(\alpha^\frown\langle\text{wait}\rangle)$ into B.

Assume $\alpha^\frown\langle\gamma_i\rangle \subset f$ for some $i \leq 1$ and let b_0 be the least element of $S(\alpha^\frown\langle\gamma_i\rangle)$. By the construction, α cannot dump b_0 into B. Since B is Δ_2^0, let $t_0 \geq s_3$ be the least stage such that for all $s \geq t_0$, $B(b_0; s) = B(b_0)$. Define b_1 to be the least element that enters $S(\alpha^\frown\langle\gamma_i\rangle)$ after stage t_0. Since b_0 has reached its limit, b_1 cannot be dumped into B by α. Continuing in this manner, we construct an infinite sequence of elements $b_0 < b_1 < b_2 < \cdots \subseteq$

$S(\alpha^\frown\langle\gamma_i\rangle)$ that are not dumped into B by any strategy of priority higher than $\alpha^\frown\langle\gamma_i\rangle$.

Assume that $\alpha^\frown\langle\text{stop}\rangle \subset f$. Since there are only finitely many d_i chosen by α, say $d_0, \ldots, d_n$, and B is Δ^0_2, there is a least stage $s_4 \geq s_3$ such that $B(d_i)[s] = B(d_i)[s_1]$ for all $s \geq s_4$ and all $i \leq n$. (Clearly there is some i such that $b_i \notin B$.) After stage s_4, α will always take on outcome $\langle\text{stop}\rangle$ and will dump no more members of $S(\alpha)$ into B, and enumerate all of $S(\alpha) \cap (s_4, \infty)$ into $S(\alpha^\frown\langle\text{stop}\rangle)$.

□

3.5.6 Lemma. f is infinite.

Proof. Clearly the empty node is in f. Assume that $\alpha \subset f$, let $\langle o\rangle$ be the true outcome of α and assume that α is never initialized after stage s_0. Then there is a least stage $s_1 > s_0$ after which $f_s \geq \alpha^\frown\langle o\rangle$ for all $s \geq s_1$. By the construction, there are only finitely many elements in the streams $S(\beta)$ with $\beta <_L \alpha^\frown\langle o\rangle$. Since B is Δ^0_2, there is stage $s_2 > s_1$ after which no such element will cause $\alpha^\frown\langle o\rangle$ to be initialized.

If α is an $\mathcal{N}$-strategy, then by Lemma 3.5.4.ii, α chooses only finitely many diagonalization witnesses. If α is an $\mathcal{S}$-strategy, then by Lemma 3.5.5.ii, α also only chooses finitely many diagonalization witnesses. In either case, since B is Δ^0_2, there is a stage $s_3 > s_2$ after which α will never initialize $\alpha^\frown\langle o\rangle$. Furthermore, a stage can end prematurely only in Case 1.2, but by Lemmas 3.5.4.ii, 3.5.4.iii, and 3.5.5.iii, this can happen only finitely often for any given α. Therefore $\alpha^\frown\langle o\rangle \in f$. □

Bibliography

[Ahm89] S. Ahmad. *Some Results on Enumeration Reducibility.* PhD thesis, Simon Frasier University, 1989.

[Ahm91] S. Ahmad. Embedding the diamond in the Σ^0_2-enumeration degrees. *J. Symbolic Logic*, 56(1):195–212, 1991.

[AKMS89] C. Ash, J. Knight, M. Manasse, and T. Slaman. Generic copies of countable structures. *Ann. Pure Appl. Logic*, 42(3):195–205, 1989.

[AL98] S. Ahmad and A. H. Lachlan. Some special pairs of Σ^0_2 e-degrees. *Math. Log. Q.*, 44(4):431–449, 1998.

[Cas71] J. Case. Enumeration reducibility and partial degrees. *Arch. Math. Logic*, 2(4):419–439, 1970/1971.

[CC88] S. Cooper and C. S. Copestake. Properly Σ^0_2 enumeration degrees. *Z. Math. Logik Grundlag. Math.*, 34(6):491–522, 1988.

[Coo84] S. Cooper. Partial degrees and the density problem, part II: The enumeration degrees of the Σ^0_2-sets are dense. *J. Symbolic Logic*, 49(2):503–513, 1984.

[Coo87] S. Cooper. Enumeration reducibility using bounded information: counting minimal covers. *Z. Math. Logik Grundlag. Math.*, 33(6):537–560, 1987.

[Coo90] S. Cooper. Enumeration reducibility, nondeterministic computations and relative computability of partial functions. In *Recursion Theory Week (Oberwolfach, 1989)*, volume 1432 of *Lecture Notes in Math.*, pages 57–110. Springer, Berlin, 1990.

[Cra53] W. Craig. On axiomatizability within a system. *J. Symbolic Logic*, 18:30–32, 1953.

[Fef57] S. Feferman. Degrees of unsolvability associated with classes of formalized theories. *J. Symbolic Logic*, 22:161–175, 1957.

[FR59] R. Friedberg and H. Rogers. Reducibility and completeness for sets of integers. *Z. Math. Logik Grundlag. Math.*, 5:117–125, 1959.

[Gut71] L. Gutteridge. *Some Results on Enumeration Reducibility.* PhD thesis, Simon Frasier University, 1971.

[HS88] G. Higman and E. Scott. *Existentially closed groups*, volume 3 of *London Mathematical Society Monographs. New Series.* The Clarendon Press Oxford University Press, New York, 1988. Oxford Science Publications.

[Kle52] S. Kleene. *Introduction to Metamathematics.* D. an Nostrand Co., Inc., New York, 1952.

[Lac66] A. H. Lachlan. Lower bounds for pairs of recursively enumerable degrees. *Proc. London Math. Soc.*, 16(3):537–569, 1966.

[Ler83] M. Lerman. *Degrees of unsolvability.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1983. Local and Global Theory.

[LN95] S. Lempp and A. Nies. The undecidability of the Π_4-theory for the r.e. wtt- and Turing-degrees. *J. Symbolic Logic*, 60(4):1118–1136, 1995.

[LNS98] S. Lempp, A. Nies, and T. Slaman. The Π_3-theory of the computably enumerable Turing degrees is undecidable. *Trans. Amer. Math. Soc.*, 350(7):2719–2736, 1998.

[LS88] M. Lerman and R. Shore. Decidability and invariant classes for degree structures. *Trans. Amer. Math. Soc.*, 310(2):669–692, 1988.

[LS92] A. H. Lachlan and R. Shore. The n-rea enumeration degrees are dense. *Arch. Math. Logic*, 31(4):277–285, 1992.

[LS02] S. Lempp and A. Sorbi. Embedding finite lattices into the Σ^0_2-enumeration degrees. *J. Symbolic Logic*, 67(1):69–90, 2002.

[LSS] S. Lempp, T. Slaman, and S. Sorbi. Extensions of embeddings in the Σ^0_2-enumeration degrees. Accepted for publication in J. Math. Log.

[MC85] K. McEvoy and S. Cooper. On minimal pairs of enumeration degrees. *J. Symbolic Logic*, 50(4):983–1001, 1985.

[McE85] K. McEvoy. Jumps of quasi-minimal enumeration degrees. *J. Symbolic Logic*, 50(3):839–848, 1985.

[Med55] Y. Medvedev. Degrees of difficulty of the mass problem. *Dokl. Akad. Nauk SSSR (N.S.)*, 104:501–504, 1955.

[Mil04] J. Miller. Degrees of unsolvability of continuous functions. *J. Symbolic Logic*, 69(2):555–584, 2004.

[Nie96] A. Nies. Undecidable fragments of elementary theories. *Algebra Universalis*, 35(1):8–33, 1996.

[Rog67] H. Rogers. *Theory of recursive functions and effective computability.* McGraw-Hill Book Co., New York, 1967.

[Sac63] G. E. Sacks. On the degrees less than $0'$. *Ann. of Math. (2)*, 77:211–231, 1963.

[Sco75] D. Scott. Lambda calculus and recursion theory. In *Proceedings of the Third Scandinavian Logic Symposium (Univ. Uppsala, Uppsala, 1973)*, pages 154–193. Stud. Logic Found. Math., Vol. 82, Amsterdam, 1975. North-Holland.

[Soa87] R. I. Soare. *Recursively enumerable sets and degrees.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1987. A study of computable functions and computably generated sets.

[Sor97] A. Sorbi. The enumeration degrees of the Σ^0_2-sets. In A. Sorbi, editor, *Complexity, Logic, and Recursion Theory*, volume 187 of *Lecture Notes in Pure and Appl. Math.*, pages 303–330. Dekker, New York, 1997.

[SW97] T. Slaman and W. Woodin. Definability in the enumeration degrees. *Arch. Math. Logic*, 36(4-5):255–267, 1997. Sacks Symposium (Cambridge, MA, 1993).

[Zie80] M. Ziegler. Algebraisch abgeschlossene Gruppen. In S. I. Adian, W. W. Boone, and G. Higman, editors, *Word problems, II (Conf. on Decision Problems in Algebra, Oxford, 1976)*, volume 95 of *Stud. Logic Foundations Math.*, pages 449–576. North-Holland, Amsterdam, 1980.