Moduli of Computation

by

Peter Michael Gerdes

B.S. (California Institute of Technology) 2001

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Logic and the Methodology of Science

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:
Professor Leo Harrington, Chair
Professor Theodore Slaman
Professor Jon Wilkening

Fall 2008

The dissertation of Peter Michael Gerdes is approved:

Chair | Date

Date

Date

University of California, Berkeley

Fall 2008

Moduli of Computation

Abstract

Moduli of Computation

by

Peter Michael Gerdes
Doctor of Philosophy in Logic and the Methodology of Science

University of California, Berkeley

Professor Leo Harrington, Chair

The relation between a functions rate of growth and it's computational properties is a traditional, and well studied, problem in computability theory. However, this relationship has been almost exclusively studied in a somewhat piecemeal fashion by fixing some notion of a fast growing function and classifying the degrees of those functions. Making use of the notion of a modulus of computation, a measure of the rate of growth sufficent to compute a given set, as introduced by Groszek and Slaman we explore the connection between rate of growth and Turing degree in a more general setting. In particular we do this by focusing on two particular types of moduli: the self-moduli (those functions computable from any faster growing function) and the uniform moduli (functions witnessing a rate of growth sufficent to guarantee uniform computation by larger functions). After exploring the behavior of these objects we characterize the uniform self-moduli and extend this characterization to sets with uniform moduli and in so doing answer a question of Groszek and Slaman. Finally we demonstrate that there are examples of self-moduli that are very non-uniform.

Professor Leo Harrington
Dissertation Committee Chair

Chapter 1

Background

Before we present the results of our investigations into the computational properties of fast growin functions – or even define the notion of a moduli of computation – we first must explain our notation and review some necessery background. Limitations of space and time prevent us from offering a comprehensive introduction to computability theory so we refer the reader to [19] or [17] for a primer on classical computability theory and will assume they are familiar with the basic results on arithmetic sets and functions. Some passing familiarity with hyperarithmetic theory (for instance the results in the first chapter or two of [18]) and the use of arithmetic forcing in computability theory (such as is covered in [16] will also be helpful but we will endeavor to give a quick review of the relevant results in these areas to allow someone with only a background in the classical computability theory to follow our arguments. First, however, we document the notation we will use for standard concepts in mathematics and computability theory.

1.1 Notation

1.1.1 Strings, Sets & Functions

We use the following standard notation for dealing with functions and sets.

Conventions 1.1.1 (Functions, Sets & Variables)**.**

1. $\mathcal{P}(S)$ is the power set of S, X^Y is the set of total functions from Y to X and $(X \cup \uparrow)^Y$ is the set of partial functions from Y to X.

2. ω is the natural numbers, ω^ω is the baire space and 2^ω the cantor space. We identify integers with the corresponding ordinals, e.g. $n = \{m|m < n\}$ and identify members of 2^ω with sets of integers via their characteristic functions. Functions are identified with sets of ordered pairs.

3. We write $f(x)\downarrow$ to denote that the function f is defined at x and $f(x)\uparrow$ or $f(x)\not\downarrow$ to indicate that $f(x)$ is undefined.

4. Unless stated otherwise σ, τ, ν range over $2^{<\omega}$ or $\omega^{<\omega}$, f, g, h range over ω^ω and ϵ, δ range over $\omega^{<\omega} \cup (\omega\cup\uparrow)^\omega$ or $(2\cup\uparrow)^{<\omega} \cup (2\cup\uparrow)^\omega$. $e, i, j, k, n, m, x, y, z$ range over ω and capital roman letters range over sets of integers.

We adopt the following standard definitions and conventions for strings.

Conventions 1.1.2 (Strings & Coding)**.**

1. A string is a member of $\omega^{<\omega}$ identified with the integers via a canonical bijection. A binary string is a string whose range is contained in $2 = \{0, 1\}$. An infinite string is a member of ω^ω.

2. $\emptyset$ is the empty string. $\langle x_0, x_1, x_2, \ldots, x_k\rangle$ denotes the string mapping 0 to x_0, 1 to x_1 and so forth.

3. When we need to explicitly indicate that we want to understand a string σ (or any other coded object) as an integer we write $\ulcorner\sigma\urcorner$.

4. Conversely given an integer s we denote the string with code s by $s^\star$ and use $(s^\star)_k$ to denote $s^\star$ applied to k.

5. We select a canonical bijection between $\omega \times \omega$ and ω and use the notation $\langle\langle x, y\rangle\rangle$ to denote the image of the ordered pair (x, y) under this bijection.

6. The length of σ is denoted $|\sigma|$.

7. The concatenation of σ and τ is denoted $\sigma\widehat{\ }\tau$.

8. We write $\sigma \mid \tau$ if σ and τ lack a common extension and $\sigma \nmid \tau$ if they have a common extension.

9. If $X, Y \subset \omega$ then $X \oplus Y = \{z|(z = 2x \land x \in X) \lor (z = 2y + 1 \land y \in Y)\}$.

10. If $X_i \subset \omega, i \in \omega$ then $\bigoplus_{i\in\omega} X_i = \{\langle\langle i, x\rangle\rangle | x \in X_i\}$.

11. If $A \subset \omega$ $A^{[n]} = A_n$ where $A = \bigoplus_{n\in\omega} A_n$.

12. If $f, g \in \omega^\omega$ then $f \oplus g = h \in \omega^\omega$ where $h(2x) = f(x) \wedge h(2x+1) = g(x)$.

13. If $f_i \in \omega^\omega, i \in \omega$ then $\bigoplus_{i\in\omega} f_i = f \in \omega^\omega$ where $f(\langle\langle i, x\rangle\rangle) = f_i(x)$.

14. If $f \in \omega^\omega$ then $f^{[n]} = f_n$ where $f = \bigoplus_{i\in\omega} f_i$.

15. If $f \in \omega^\omega$ and $X \subset \omega$ then $f \oplus X = G_f \oplus X$ where G_f is the graph of f.

We also extend the standard definition of majorization to include partial functions and introduce some nonstandard notation that we will be making extensive use of in the remainder of this work.

Definition 1.1.3. Given $\epsilon, \delta \in (\omega\cup\uparrow)^\omega$ we say ϵ **majorizes** (**dominates**) δ, denoted $\epsilon \gg \delta$ ($\epsilon \gg^* \delta$), if for every x (for all but finitely many x):

$$x \in \operatorname{dom} \epsilon \cap \operatorname{dom} \delta \implies \epsilon(x) \geq \delta(x)$$

Also it will be convenient to take our coding functions to have a few extra properties. In particular we assume that $\langle\langle 0, 0\rangle\rangle = 0$ and that our coding function for $\omega^{<\omega}$ satisfies the conditions given in the following lemma.

Lemma 1.1.4. *There is a primitive recursive bijection taking $\sigma \in \omega^{<\omega}$ to $\ulcorner\sigma\urcorner \in \omega$ that satisfies:*

1. $\sigma \subset \sigma' \implies \ulcorner\sigma\urcorner < \ulcorner\sigma'\urcorner$

2. $\operatorname{dom} \sigma \supseteq \operatorname{dom} \tau \wedge \sigma \gg \tau \implies \ulcorner\sigma\urcorner > \ulcorner\tau\urcorner$

Proof. If we were coding finite partial functions we could satisfy these conditions by encoding the finite partial function $\sigma^* \in (\omega\cup\uparrow)^{<\omega}$ in the following manner.

$$\ulcorner\sigma^*\urcorner = 2^{y_0+1} \cdot 3^{y_1+1} \cdot \ldots \cdot p_k^{y_k+1}$$
$$y_k = \begin{cases} \sigma^*(k) & \text{if } \sigma^*(k)\downarrow \\ -1 & \text{if } \sigma^*(k)\uparrow \end{cases}$$

To build a bijective coding function for $\omega^{<\omega}$ we merely list off finite partial functions in order of their code and map $\sigma \in \omega^{<\omega}$ to k if it is the k-th member of $\omega^{<\omega}$ to appear in our list of finite partial functions. □

1.1.2 Computability

Conventions 1.1.5.

1. Given some predicate $\varphi(x)$ we denote the operation returning the least integer y satisfying $\varphi(x)$ by $\mu y\,(\varphi(y))$. If no solution to $\varphi(x)$ exists then $\mu y\,(\varphi(y))\uparrow$.

2. Variables of the form $\underset{\sim}{d}$ range over Turing degrees and $\underset{\sim}{X}$ denotes the Turing degree of the set X.

3. $\underset{\sim}{0}$ denotes the degree of computable sets.

4. X' denotes the jump of a set X and $\underset{\sim}{d}'$ denotes the degree of X' for $X \in \underset{\sim}{d}$.

5. If f is a partial computable function then $f(x)\downarrow_s$ denotes that $f(x)$ converges in at most s steps and $f(x)\not\downarrow_s$ denotes that it does not.

6. Φ_e is the e-th partial computable functional where $\Phi_e(X;x)$ is the result of running the e-th algorithm with input x on the oracle X.

7. We denote the primitive recursive function computing s stages of $\Phi_e(X;x)$ by $\Phi_{e,s}(X;x)$ and define
$$\Phi_{e,s}(X;x) = \begin{cases} \Phi_e(X;x) & \text{if } \Phi_e(X;x')\downarrow_s \\ \uparrow & \text{otherwise} \end{cases}$$

8. We define Φ^2_e to be the e-th $0,1$ valued partial computable functional defined so:
$$\Phi^2_e(X;x) = \begin{cases} 0 & \text{if } \Phi_e(X;x)\downarrow = 0 \\ 1 & \text{if } \Phi_e(X;x)\downarrow \neq 0 \\ \uparrow & \text{if } \Phi_e(X;x)\uparrow \end{cases}$$

9. We use $\Phi_e(g)$ where $g \in \omega^\omega$ to denote $\Phi_e(G_g)$ where G_g is the graph of g. We adopt the same convention for $\Phi^2_e(g)$.

10. $\mathcal{W}^X_e = \{x | \Phi_e(X;x)\downarrow\}$ is the e-th set computably enumerable in X. We abbreviate $\mathcal{W}^\emptyset_e$ by $\mathcal{W}_e$.

11. Two computations $\Phi_e(X;x), \Phi_i(Y;y)$ are strongly equal (alternatively Kleene equal), denoted $\Phi_e(X;x) \simeq \Phi_i(Y;y)$ if $(\Phi_e(X;x)\uparrow \wedge \Phi_i(Y;y)\uparrow) \vee \Phi_e(X;x)\downarrow = \Phi_i(Y;y)\downarrow$.

12. $\Phi_e(X) \simeq \Phi_i(Y) \iff (\forall z)\,(\Phi_e(X;z) \simeq \Phi_i(Y;z))$

13. The use of $\Phi_{e,s}(Y;x)$ is denoted $u(\Phi_e(Y;x),s)$ and refers to the largest location in Y examined during s steps of the computation. We stipulate $u(\Phi_e(Y;x),s) \leq s$ and define the use of $\Phi_e(Y;x)$ by

$$u(\Phi_e(Y;x)) = \max_s u(\Phi_e(Y;x),s)$$

14. An index for a function f (a set S) relative to X is an integer e such that $f \simeq \Phi_e(X)$ ($S = \mathcal{W}_e^X$). An index simplicitor is an index relative to $\emptyset$.

15. A sequence $\langle z_k \rangle_{k\in\omega}$ is computable if the function taking k to z_k is computable.

1.2 Trees

We adopt the following fairly standard conventions to describe trees.

Conventions 1.2.1.

- A **tree** is a subset of $\omega^{<\omega}$ viewed as a poset under $\subseteq$ possessing a unique minimal element.
- A tree T is **downward closed** or closed under initial segments if $\sigma \in T$ and $\tau \subseteq \sigma$ then $\tau \in T$. All trees we use in this work are downward closed unless we explicitly indicate otherwise.
- A **branch** through T is a maximal linearly ordered subset of T. When the tree ordering is given by $\subseteq$ we identify branches with the union of their members.
- An **infinite branch** is a branch with height at least ω. We denote the set of infinite branches through T by $[T]$.
- We denote the set of $\sigma \in T$ that extend to an infinite branch by T^∞.
- A **terminal node** is a maximal element of $(T, <_T)$.
- The subtree of T extending $\sigma \in T$ is the denoted ${}^{T}\!/_{(\sigma)} = \{\tau \in T | \sigma \leq \tau\}$ and forms a tree under the inherited order.
- A tree is perfect if every $\sigma \in T$ is extended by some τ, τ' with $\tau \mid \tau'$.

- An tree is computable just if T, viewed as a set of integers, is computable.

During our constructions we will often need to ensure something happens along every infinite branch of T that extends the node σ. With this in mind we make the following definition.

Definition 1.2.2. Given an ω-tree T say a set of nodes C is a **cover** of σ (on T) if every infinite branch of T extending σ extends an element of C.

1.2.1 Forcing

We refer the interested reader to Odifreddi [16, 15] for complete coverage whose relation $\Vdash_T$ is our relation $\Vdash_T^*$ and relation $\Vdash_T^w$ is our relation $\Vdash_T$. While Odifreddi defines (strong) Cohen forcing (forcing using arbitrary conditions in $2^{<\omega}$) and presents (strong) local forcing (forcing using conditions from some tree) as a minor modification to this definition we will be interested only in local forcing so define strong local forcing as the primary notion. Additionally we deviate from Odifreddi by defining forcing in $\omega^{<\omega}$ instead of $2^{<\omega}$.

When working with forcing we assume that that universal quantifier $\forall x$ is merely shorthand for $\neg\exists x\neg$ and that our formulas are in prenex normal form. Extending the definition to other formulas is straightforward.

Definition 1.2.3. Given $T \subset \omega^{<\omega}$, a perfect tree, we define the **strong forcing** relation $\sigma \Vdash_T^* \psi$ for $\sigma \in T$ and a formula ψ containing the free function variable g by induction on the complexity of ψ.

$$\psi \in \Delta_1^0 \Rightarrow \sigma \Vdash_T^* \psi \leftrightarrow \sigma \models \psi$$
$$\psi = \exists x \varphi(x) \Rightarrow \sigma \Vdash_T^* \psi \leftrightarrow (\exists x)\left[\sigma \Vdash_T^* \varphi(x)\right]$$
$$\psi = \neg\varphi \Rightarrow \sigma \Vdash_T^* \psi \leftrightarrow (\forall \tau \supset \sigma, \tau \in T)\left(\neg\tau \Vdash_T^* \varphi\right)$$

Where we understand $\sigma \models \psi$ to hold if ψ can be seen to hold when g is replaced by σ without reference any location at which σ is undefined.

When the tree T is just $\omega^{<\omega}$ we drop the subscript. We now briefly remind the reader of standard definitions used to describe forcing.

Definitions 1.2.4.

1. For $g \in [T]$, $g \Vdash_T^* \varphi \iff (\exists n)\,(g\restriction_n \Vdash_T^* \varphi)$.

2. $g \in [T]$ is κ-generic if for every $\varphi \in \Sigma_n^0$ either $g \Vdash_T^* \varphi$ or $g \Vdash_T^* \varphi$.

3. σ **forces** $\varphi \in \Sigma_n^0$ denoted $\sigma \Vdash_T \varphi$ if for every κ-generic $g \supset \sigma \implies g \Vdash_T^* \varphi$. Note that $\sigma \Vdash^* \varphi$ entails $\sigma \Vdash \varphi$.

We present the fundamental properties of the forcing relation without proof.

Lemma 1.2.5. *g is n-generic if and only if g forces the same Σ_n^0 formulas that it makes true.*

Lemma 1.2.6. *For every formula φ and string $\sigma \in T$ there is some $\sigma' \in T, \sigma' \supset \sigma$ with either $\sigma' \Vdash^* \varphi$ or $\sigma' \Vdash^* \neg\varphi$.*

1.3 Hyperarithmetic Theory

1.3.1 Ordinal Notations

The theory of computable notations for ordinals is too complex to allow a full presentation here so we present a system of ordinal notations similar to the one defined by Soare in [18] and remind the reader of several important results but refer them to the literature [18, 17, 3, 10] for a complete treatment.

Definition 1.3.1. Using transfinite induction define the integer $a = \langle\langle u, v\rangle\rangle$ to be a notation for the ordinal κ if

1. $\kappa = 0$, $u = 0$ and $v = 0$.

2. $\kappa = \alpha +_{\mathcal{O}} 1$, $u = 1$ and v is a notation for α.

3. $\kappa = \lim\limits_{n\to\infty} \lambda_n$ for a strictly increasing sequence λ_n, $u = 2$ and for each n $\Phi_v(\emptyset; n)$ is a notation for λ_n.

A slightly more involved inductive argument is required to demonstrate that there is a computable binary operation $+_{\mathcal{O}}$ that implements ordinal addition in a computable fashion on notations but we leave the proof to the reader. To ease our use of ordinal notations we adopt the following abbreviations.

Notation 1.3.2.

- If κ is as in part 3 of definition 1.3.1 we write $\vec{\lambda}^{\kappa}_{n}$ to denote λ_n.
- We denote the set of all ordinal notations by $\bar{\mathcal{O}}$.

From this point on we abuse notation and, when there is no danger of confusion, cease to distinguish between the notations and the ordinals they represent. To facilitate this identification we use the variables α, β, γ and λ for ordinal notations when we wish to manipulate them as if they were the ordinals themselves. Thus we will simple write $\alpha +_{\mathcal{O}} \beta$ instead of $\alpha +_{\mathcal{O}} \beta$ when we add notations. Similarly we will often simply write $\kappa = \lim_{n\to\infty} \lambda_n$ and omit saying that κ is the effective limit of λ_n when there is no danger of confusion. When clarity requires we distinguish between the notation and the ordinal we adopt the notation:

Notation 1.3.3.

- If α is an ordinal notation then $|\alpha|_{\mathcal{O}}$ is the ordinal it represents.
- If α is an ordinal notation we write $\ulcorner \alpha \urcorner$ to emphasize that we understand α as an integer in that context.
- We allow numerals to represent both the integer and the corresponding unique ordinal notation depending on the context $n \in \omega$ from the integer we denote the notation

Unfortunately we can't entirely forget about the distinction between ordinals and their notations since there are many notations for every ordinal past ω. We cannot, therefore, induce a well-ordering on the set of notations. However, we can introduce an ordering $<_{\mathcal{O}}$ on notations which is a well-ordering below any particular notation. Intuitively a notation α is below a notation a' when a appears somewhere along the sequence constructing a'. We formalize this as follows.

Definition 1.3.4. The relation $<_{\mathcal{O}}$ is the smallest transitive anti-reflexive relation on notations satisfying the conditions:

1. For every ordinal notation κ, $\kappa <_{\mathcal{O}} \kappa +_{\mathcal{O}} 1$.
2. If the notation κ is defined by the effective proper limit $\lim\limits_{n\to\infty} \lambda_n$ as in definition 1.3.1 part 3 then for every n $\lambda_n <_{\mathcal{O}} \kappa$.

This system provides a (non-unique) notation for every ordinal below ω_1^{ck}, the first non-computable ordinal. Standard results [18] assure us that this is the best we can hope to achieve in an effective system. We remark without proof that given a notation κ the set of notations $\bar{\mathcal{O}}_{<\beta} = \{\beta | \beta <_{\mathcal{O}} \kappa\}$ is a computably enumerable set well-ordered by $<_{\mathcal{O}}$. Though we could not give an effective system of unique notations there is a canonical Π^1_1 set $\mathcal{O}$ of unique notations where a Π^1_1 set is defined in the following manner.

Definition 1.3.5. A set $S \subset \omega$ is in Π^1_n if there is some formula ψ defining S so that for some arithmetic formula φ

$$\psi = \underbrace{\forall X_1 \exists X_2 \ldots}_{n} \varphi(X_1, X_2, \ldots, X_k)$$

S in in Σ^1_n if insted

$$\psi = \underbrace{\exists X_1 \forall X_2 \ldots}_{n} \varphi(X_1, X_2, \ldots, X_k)$$

Finally S is in Δ^1_n if if is in both Σ^1_n and Π^1_n

Having a computable means to represent ordinals beyond ω now gives us a way to extend familiar operations from the arithmetic hierarchy to the transfinite. For instance we can now define the α iterate of the Turing jump for any ordinal α below ω_1^{ck}.

Definition 1.3.6. Given $X \subseteq \omega$ and an ordinal notation κ define $X^{(\kappa)}$ inductively as follows.

$$X^{(\kappa)} = \begin{cases} X & \text{if } \kappa = 0 \\ (X^{(\alpha)})' & \text{if } \kappa = \alpha +_{\mathcal{O}} 1 \\ \bigoplus_{n \in \omega} X^{(\vec{\lambda}^{\kappa}_n)} & \text{if } \kappa = \lim_{n \to \infty} \lambda_n \end{cases}$$

As is frequently the case when dealing with ordinal notations $X^{(\kappa)} \equiv_T X^{(\kappa')}$ whenever $|\kappa|_{\mathcal{O}} = |\kappa'|_{\mathcal{O}}$ [17, 12, 18]. Thus we really have defined a notion of the α jump for every ordinal α below ω_1^{ck}.

Definition 1.3.7. A set S is hyperarithmetic if $S \leq_{\mathbf{T}} 0^{(\alpha)}$ for some notation α.

We are now able to state (without proof) the famous result characterizing the hyperarithmetic sets.

Theorem 1.3.8. *A set is hyperarithmetic if and only if it is* Δ^1_1.

1.3.2 REA sets

While the jump operator provides one means of iteratively building sets in the hyperarithmetic hierarchy this yields only a fairly limited class of sets. Jockusch and Shore [9, 7] defined a much richer class of sets, the REA sets, by generalizing the idea of the Turing jump to that of a pseudo-jump operator and then effectively iterating these operators transfinitely many times. An in depth treatment of REA sets and operators can be found in [9, 7] but we briefly review the concepts we will need.

Definition 1.3.9. A pseudo-jump operator $\mathcal{J}_e$ is functional acting on $\mathcal{P}(\omega)$ with an action given by $\mathcal{J}_e(X) = X \oplus \mathcal{W}_e^X$.

Definition 1.3.10. For every notation κ we define theκ-REA operator $\mathcal{J}_e^\kappa$ to satisfy.

1. $\mathcal{J}_e^0(X) = X$.
2. If $\kappa = \alpha +_{\mathcal{O}} 1$ then

$$\mathcal{J}_e^\kappa(X) = \begin{cases} \mathcal{J}_i(\mathcal{J}_e^\alpha(X)) & \text{if } \Phi_e(\emptyset;\alpha)\downarrow = i \\ \emptyset & \text{if } \Phi_e(\emptyset;\alpha)\uparrow \end{cases}$$

3. If κ a limit notation then $\mathcal{J}_e^\kappa(X) = \bigoplus_{n\in\omega} \mathcal{J}_c^{\beta_n}(X)$ where $\langle\beta_n\rangle_{n\in\omega}$ is a (uniformly chosen from κ canonical) computable enumeration of the notations below κ.

We define indexes for REA operators in the natural manner.

Definition 1.3.11. We define the REA operator $\mathcal{J}_r^*$ with REA index r to be $\mathcal{J}_e^\kappa$ where $r = \langle\langle\kappa, e\rangle\rangle$. We call an REA operator with index $\langle\langle\beta, j\rangle\rangle$ a β-REA operator.

Definition 1.3.12. S is an REA set with index c if $S = \mathcal{J}_c^*(\emptyset)$. If $c = \langle\langle\alpha, j\rangle\rangle$ we also say that S is an α-REA set.

1.3.3 Hyperarithmetic Formula

We adopt a slightly unusual, though still fairly standard, approach to the hyperarithmetic sets. Instead of defining the hyperarithmetic sets as the closure of the arithmetic sets under effective operations (e.g. jumps and effective unions) we define a computable fragment of $\mathcal{L}_{\omega_1,\omega}$ which we denote by $\mathcal{CL}_{\omega_1,\omega}$ and define the hyperarithmetic sets to be those sets definable in this language. The approaches are straightforwardly equivalent but the infinitary formula we develop will be a useful way to organize some constructions. However, our treatment is somewhat cursory so we refer the reader to the literature [4, 11, 1, 5] for a formal definition of computable Σ_α and Π_α formula. Here we simply take the existence of an acceptable system of indexes for these formula for granted and offer the following definition sketch.

Definition 1.3.13.

- A $\Sigma^0_0(X)$ or $\Pi^0_0(X)$ formula is a finite formula in the language of arithmetic with the addition of a set constant X containing no quantifiers.
- A $\Sigma^0_{\alpha+1}(X)$ formula is a formula of the form $\exists x\varphi$ where $\varphi \in \Pi^0_\alpha(X)$.
- A $\Pi^0_{\alpha+1}(X)$ formula is a formula of the form $\forall x\varphi$ where $\varphi \in \Sigma^0_\alpha(X)$.
- A $\Sigma^0_\lambda(X)$ formula for λ a limit is of the form $\bigvee\!\!\!\bigvee_{i\in\omega} \varphi_i$ where φ_i is a computable sequence of $\Pi^0_\alpha(X)$ formulas for $\alpha <_{\mathcal{O}} \lambda$ and $\bigvee\!\!\!\bigvee$ is understood as an infinite disjunction.
- A $\Pi^0_\lambda(X)$ formula for λ a limit is of the form $\bigwedge\!\!\!\bigwedge_{i\in\omega} \varphi_i$ where φ_i is a computable sequence of $\Sigma^0_\alpha(X)$ formulas for $\alpha <_{\mathcal{O}} \lambda$ and $\bigwedge\!\!\!\bigwedge$ is understood as an infinite conjunction.
- $\mathcal{CL}_{\omega_1,\omega}$ is the collection of $\Sigma^0_\kappa(X)$ formulas for $\kappa <_{\mathcal{O}} \omega_1^{ck}$.

Rather than defining a new notation for computable Σ_α/Π_α formula we abuse notation and write $\varphi \in \Sigma_\alpha/\in \Pi_\alpha$ whenever φ is equivalent to a computable Σ_α/Π_α. When we need to make it clear that a given formula φ is literally a computable Σ_α or Π^0_α formula rather than merely equivalent to one we will say that φ is **syntactically** $\Sigma^0_\alpha/\Pi^0_\alpha$. We define the Σ^0_α and Π^0_α sets in terms of the formulas thusly.

Definition 1.3.14. Given a set $S \subseteq \omega$ we define

$$\begin{aligned}
S \in \Sigma^0_\alpha(X) &\iff \left(\exists\varphi \in \Sigma^0_\alpha(X)\right)(\forall y \in \omega)\,(y \in S \iff \varphi(y)) \\
S \in \Pi^0_\alpha(X) &\iff \left(\exists\varphi \in \Pi^0_\alpha(X)\right)(\forall y \in \omega)\,(y \in S \iff \varphi(y)) \\
S \in \Delta^0_\alpha(X) &\iff S \in \Sigma^0_\alpha(X) \wedge S \in \Pi^0_\alpha(X)
\end{aligned}$$

Definition 1.3.15. A $\Sigma^0_\alpha(X)/\Pi^0_\alpha(X)$ index for a set S is an index for a $\Sigma^0_\alpha(X)/\Pi^0_\alpha(X)$ formula defining S as in 1.3.14. A $\Delta^0_\alpha(X)$ index for S is a pair $\langle\!\langle e_1, e_2\rangle\!\rangle$ with e_1 a $\Sigma^0_\alpha(X)$

index for S and e_2 a $\Sigma^0_\alpha(X)$ index for $\omega \setminus S$. An index for a total function is simply an index for it's graph.

The following two results are well known extensions of results for arithmetic sentences which we present without proof.

Proposition 1.3.16. *A set S is in $\Sigma^0_{1+\alpha}$ if and only if $S = \mathcal{W}_e^{0^{(\alpha)}}$ for some e. Furthermore there is a computable bijection between $\Sigma^0_{1+\alpha}$ indexes and indexes for computably enumerable sets in $0^{(\alpha)}$.*

Lemma 1.3.17. *Suppose $\varphi, \psi \in \Sigma^0_\kappa(Y)$ or $\Pi^0_\kappa(Y)$ then both $\varphi \wedge \psi$ and $\varphi \vee \psi$ are equivalent to formulas in $\Sigma^0_\kappa(Y)\,/\Pi^0_\kappa(Y)$. Furthermore if $\varphi \in \Sigma^0_\kappa(Y)$ then $\exists x \varphi$ is equivalent to a $\Sigma^0_\kappa(Y)$ formula and likewise if $\varphi \in \Pi^0_\kappa(Y)$ then $\forall x \varphi$ is equivalent to a $\Pi^0_\kappa(T)$ formula. Moreover indexes for the resultant formulas can be found effectively.*

To prove results about Σ^0_α and Π^0_α relations will commonly make use of transfinite induction and in these arguments it will be important that we can always express bounded existential quantification without increasing the complexity of the formula.

Lemma 1.3.18. *If $\psi \in \Pi^0_\kappa(Y)$ then for every k there is a formula $\varphi_k \in \Pi^0_\kappa(Y)$ so that $\varphi_k \leftrightarrow (\exists x < k)\, \psi(x)$ and an index for φ_k is uniformly computable from an index for ψ.*

Proof.

Case 1: $\kappa <_{\mathcal{O}} \omega$

Follows trivially by transformation to predicate normal form.

Case 2: $\kappa = \alpha +_{\mathcal{O}} 1, \alpha \geq_{\mathcal{O}} \omega$

For some $\hat{\psi} \in \Sigma^0_\alpha(Y)$.

$$(\exists x < k)\, \psi(x) \iff (\exists x < k)\, (\forall y)\, \hat{\psi}(x, y)$$

Trivially then

$$(\exists x < k)\, \psi(x) \implies \left(\forall \zeta \in \omega^k\right) (\exists x < k)\, \hat{\psi}(x, \zeta(x))$$

For the converse note that

$$\neg\, (\exists x < k)\, (\forall y)\, \hat{\psi}(x, y) \implies (\forall x < k)\, (\exists y)\, \neg\hat{\psi}(x, y)$$

This in turn implies that for some function $\zeta \in \omega^k$

$$\left(\exists \zeta \in \omega^k\right) (\forall x < k)\, \neg\hat{\psi}(x, \zeta x)$$

Hence

$$(\exists x < k)\,\psi(x) \iff \left(\forall \zeta \in \omega^k\right)(\exists x < k)\,\hat{\psi}(x, \zeta(x))$$

By standard arguments this final formula is equivalent to a formula in $\Pi^0_\kappa(Y)$ and clearly every step we took was effective.

Case 3: $\kappa = \lim_{n \to \infty} \lambda_n$, a proper limit

For some sequence of formulas $\hat{\psi}_i \in \Pi^0_{\lambda_n}(Y)$ we have

$$(\exists x < k)\,\psi(x) \iff (\exists x < k) \bigwedge\!\!\!\bigwedge_{y=0}^{\omega} \hat{\psi}_i(x)$$

Now let

$$\psi^*_{\ulcorner \sigma \urcorner} = \bigvee_{m<k} \bigwedge_{z \in <|\sigma|} \hat{\psi}_{\sigma(z)}(m)$$

$$\psi' = \bigwedge\!\!\!\bigwedge_{n=0}^{\infty} \psi^*_n$$

By lemma 1.3.17 ψ' is in $\Pi^0_\kappa(T)$ and the construction of ψ' is obviously effective. Hence we need only prove that ψ' is equivalent to $(\exists x < k)\,\psi(x)$ to finish the lemma.

Suppose $\psi(x)$ holds for some $x < k$ then so does ψ' since each ψ^*_n has a disjunct consisting only of formulas of the form $\hat{\psi}_i(x)$. Conversely if $\psi(x)$ fails to hold then for each $x < k$ there is some formula $\hat{\psi}_{i_x}$ such that $\hat{\psi}_{i_x}(x)$ doesn't hold. But then $\psi^*_{\langle i_0, \ldots, i_{k-1} \rangle}$ must also fail and thus so must ψ'

□

Since we will make extensive use of Π^0_α singletons we also observe the following lemma.

Lemma 1.3.19. *There is a computable bijection taking the index for a Π^0_α formula $\varphi(X)$ to the index for a Π^0_α set $T_X \subset 2^{<\omega}$ such that for every $X \in \mathcal{P}(\omega)$ $X \in [T] \iff \varphi(X)$. The same result holds if we replace $2^{<\omega}$ with $\omega^{<\omega}$ and allow X to range over ω^ω.*

Proof. We demonstrate that given a Π^0_α definition of a tree T we can effectively go to a formula $\varphi(X)$ and vice versa. The bijection follows from a standard back and forth argument.

Given a Π^0_α set $T \subset 2^{<\omega}$ then for some $\psi \in \Pi^0_\alpha$

$$\sigma \in T \iff \psi(\sigma)$$

Hence

$$X \in [T] \iff (\forall l)\, \psi(X \restriction_l)$$

By lemma 1.3.17 this formula is in Π^0_α. Conversely given a formula $\varphi(X) \in \Pi^0_\alpha$ we assume, without loss of generality, that the only atomic formulas in φ are of the form $X(t) = t'$ where t, t' are terms. We define a Π^0_α formula $\psi(\sigma)$ by replacing every atomic formula of the form $X(t) = t'$ appearing in φ with the formula $\sigma(t) = t' \vee |\sigma| < t + 1$. Clearly ψ defines a Π^0_α tree whose infinite paths are just the solutions to φ. The effectiveness of the argument is evident and the result for ω^ω follows by a similar argument. $\square$

Chapter 2

Degrees and Their Moduli

2.1 Historical Background

One traditional area of interest in computability theory is the relation between the rate of growth of a (total) function (identified with it's graph) and it's Turing degree [13, 14, 21]. Originally motivated by Post's program [22] this topic has long since become a subject of study in it's own right and continues to be an active area of exploration to this day [2]. While a fair amount of work has been done in this area most of it shares the same basic form: classify the degrees of functions which are 'large' for some fixed notion of large (usually a dominating property relative to some class of functions). This approach has been very productive but it limits the sort of questions that can be addressed. While results like Martin's identification of the functions dominating all total computable functions with the high degrees [13] show us that functions with a sufficiently high rate of growth have substantial computational power, i.e., can compute complex sets, this approach is ill-suited to exploring the general relation between a set's computational properties and the rates of growth sufficient to compute that set.

However, not all work on rate of growth has been in this style. Jockusch and Soare introduced a fascinating alternative approach in [8] by introducing the notion of a recursively encodable set.

Definition 2.1.1. A set A is recursively encodable if for every infinite set $X \subset \omega$ there is a set $Y \subset X$ with $Y \geq_{\mathbf{T}} A$.

Intuitively if no matter how sparse one makes X or how trickily it's defined there is always a subset of X computing A this must occur for a reason. In particular A must be computable by all sufficiently sparse sets or equivalently by all sufficiently fast growing functions. We won't give a proof here (though theorem 2.4.1 will yield the same result) but the effective Galvin-Prikry forcing Solovay uses in [20] justifies this interpretation. Thus we can understand a recursively encodable set as a set that is computable by all sufficiently fast growing functions.

This notion flips the standard approach to rate of growth questions on it's head. Rather than classifying the degrees of the functions that are fast growing relative to some fixed conception this asks whether any rate of growth is sufficient to compute a given set. It would therefore seem this would be a promising alternative avenue of investigation into the relation between Turing degree and rate of growth. However, it wasn't long before Solovay [20] identified the recursively encodable sets with the Δ^1_1 sets yielding an elegant result but seemingly exhausting this avenue of exploration.

But, classifying the recursively encodable sets only tells us what sets are computable by sufficiently fast growing functions. It does not tell us how fast must a function grow to compute a particular set or otherwise let us relate particular rates of growth to computational power. As we will see the notion of a moduli of computation for a set X introduced by Groszek and Slaman can be thought of specifying a rate of growth sufficient to compute X. This will allow us to build a much more fine grained picture of the general relation between rate of growth and Turing degree than the notion of a recursively encodable set alone allowed us to see.

2.2 Moduli of Computation

Slaman and Groszek [6] define the a modulus of computation as follows.

Definition 2.2.1 (Slaman and Groszek)**.** $f \in \omega^\omega$ is a **modulus** (of computation) for $X \subset \omega$ if every $g \gg f$ Turing computes X. If there is a single computable functional Φ such that $g \gg f \implies \Phi^2(g) = X$ then we say that Φ witnesses that f is a **uniform modulus** (of computation).

It might not be immediately apparent that a modulus can be thought of as determining a rate of growth sufficient to compute X but by considering finite modifications to the function g we arrive at the following observation.

Observation 2.2.2. *If f is a modulus for X and $g \gg^* f$ then $g \geq_{\mathbf{T}} X$.*

Thus so long as g grows faster than the modulus for X g can compute X. At this point it is natural to ask whether there are any non-computable sets with a modulus.

Observation 2.2.3. *If $\theta^1(x) = \max_{i \leq x} \mu t \left(\Phi_i(\emptyset; i)\downarrow_t\right)$ then θ^1 is a uniform modulus of $0'$.*

We can straightforwardly extend this observation to every degree $\underset{\sim}{d}$ with $\underset{\sim}{d} \leq_{\mathbf{T}} 0'$ by noting that the degrees with moduli are downward closed.

Lemma 2.2.4. *If X has a (uniform) modulus and $Y \leq_{\mathbf{T}} X$ then Y has a (uniform) modulus.*

Proof. Let f be the modulus for X. If $g \gg f$ then g can compute X and hence g can compute Y. If f is a uniform modulus for X then there is a single procedure for any such g to compute X and by composing this with the reduction from X to Y this yields a uniform procedure for any such g to compute Y. □

2.3 Self-Moduli

Given the framework provided by moduli of computation a natural object of interest is the class of sets possessing the simplest possible moduli, i.e., those sets which are guaranteed to be computable by functions with the least possible rate of growth. Obviously if f is a modulus for X f must be able to compute X but observation 2.2.3 demonstrates that in some cases f can be computable from X. This class, i.e, the sets which can compute their own modulus, will be a major object of interest in our investigations.

Definition 2.3.1 (Slaman and Groszek)**.** Say $f \in \omega^\omega$ is a **self-modulus** (of computation) if f is a modulus for some set X with $f \equiv_T X$. f is a **uniform self-modulus** if it is a uniform modulus for X.

The function from observation 2.2.3 already provides us with an example of a non-computable uniform self-modulus so the natural question to ask now is: what kinds of degrees have self-moduli? Slaman and Groszek demonstrated that every Δ^0_2 degree has a uniform self-modulus and we relativize their proof to give a nice closure condition for uniform self-moduli.

Lemma 2.3.2. *Suppose that f is a uniform self-modulus and that $Z \in \Delta^0_2(f)$ computes f then there is a self-modulus $f^* \equiv_T Z$. Moreover, an index for the reduction witnessing that f^* is a uniform self-modulus can be uniformly computed from a $\Delta^0_2(f)$ index for f^* and an index for the reduction witnessing f is a uniform self-modulus.*

Proof. By the limit lemma [19] and out identification of total functions with their graphs there is a function $r \leq_{\mathbf{T}} f$ with

$$\lim_{s\to\infty} r(x,s) = Z(x) = \begin{cases} 1 & \text{if } x \in Z \\ 0 & \text{if } x \notin Z \end{cases}$$

Now define

$$f^*(\langle\!\langle x,s\rangle\!\rangle) = \begin{cases} f(x) & \text{if } s = 0 \\ \min\{t > s | r(x,t) = Z(x)\} & \text{if } s > 0 \end{cases}$$

Evidently $f^* \leq_{\mathbf{T}} Z \oplus f$ and by assumption $Z \geq_{\mathbf{T}} f$ hence $f^* \leq_{\mathbf{T}} Z$. Now suppose $g \gg f^*$. As $g(\langle\!\langle x,0\rangle\!\rangle) \geq f^*(\langle\!\langle x,0\rangle\!\rangle) = f(x)$ using the uniform self-modulus for f we can calculate f from g. Using f we compute $Z(x)$ by searching for an s large enough that for some fixed $i \in \{0,1\}$ and every t with $s \leq t \leq g(\langle\!\langle x,s\rangle\!\rangle)$ we have $r(x,t) = i$. Since $g(\langle\!\langle x,s\rangle\!\rangle) \geq f^*(\langle\!\langle x,s\rangle\!\rangle)$ for at least one t between s and $g(\langle\!\langle x,s\rangle\!\rangle)$ $r(x,t) = Z(x)$. Therefore $Z(x) = i$. Note that the stage at which $\lim_{s\to\infty} r(x,s)$ achieves it's limit is such an s guaranteeing this procedure converges for every x.

The reduction described above is evidently uniform in terms of a f index for r and an index witnessing f is a uniform self-modulus. By lemma 1.3.16 a $\Delta^0_2(f)$ index for Z is interchangeable with an index for r as an f computable function this is sufficient to establish the second half of the lemma. □

Corollary 2.3.3 (Slaman and Groszek)**.** *If $X \in \Delta^0_2$ then X has a uniform self-modulus.*

Proof. Immediate from lemma 2.3.2 and the fact that $\underset{\sim}{0}$ has a uniform self-modulus. □

By also showing the class of sets with a uniform self-modulus is closed under effective joins we can demonstrate that there are self-moduli all the way up the hyperarithmetic hierarchy.

Lemma 2.3.4. *Given a sequence of functions $\langle f_i\rangle_{i\in\omega}$ and a computable sequence of indexes $\langle e_i\rangle_{i\in\omega}$ such that $\Phi^2_{e_i}$ witnesses f_i is a uniform self-modulus of X_i then $f = \bigoplus_{i\in\omega} f_i$ is a* *uniform self-modulus of* $X = \bigoplus_{i\in\omega} X_i$*and an index for the reduction witnessing this can be effectively found from an index for the function* $i \mapsto e_i$*.*

Proof. Suppose $g \gg f$ we exhibit a procedure to compute $X(z)$ from g for an arbitrary $z = \langle\!\langle i, y\rangle\!\rangle$. By the interpretation of $\bigoplus_{i\in\omega} f_i$ we adopted for functions in convention 1.1.2 part 12 we can conclude that $g^{[i]} \gg f_i$ and thus $\Phi^2_{e_i}(g^{[i]}; y) = X_i(y) = X(\langle\!\langle i, y\rangle\!\rangle)$. The uniformity follows directly from the uniformity of each self-modulus f_i and the uniformity of the pairing function. □

We could easily weaken the assumptions in the preceding lemma. For instance instead of requiring an outright computable function enumerating the reductions we might only demand that there is a uniform means of recovering e_{i+1} from f_i. However, this result is already sufficient to guarantee the uniform self-moduli go all the way up the hyperarithmetic hierarchy and, as we are about to see, even include all the REA sets. Thus attempts to enlarge the class of iteratively generated sets with uniform self-moduli are unlikely capture any larger natural class of degrees. Later, however, we will turn to non-iterative methods to completely classify the degrees of uniform self-moduli but now we demonstrate that the REA sets indeed have uniform self-moduli.

Proposition 2.3.5. *Every REA set* X *has a uniform self-modulus* f*. Furthermore an index witnessing this fact can be uniformly computed from an REA index for* X*.*

Proof. We approach this as if we were offering a standard inductive argument. First we note that if X is a 0-REA set then $X = \emptyset$ thus the lemma trivially holds for $\kappa = 0$. Therefore we assume the proposition holds for every γ with $\gamma <_{\mathcal{O}} \kappa$ and show it also holds for κ. If $\kappa = \alpha + 1$ then there must be some REA set $\hat{X}$ and a pseudo-jump operator $\mathcal{J}_u$ such that $\mathcal{J}_u\left(\hat{X}\right) = \hat{X} \oplus \mathcal{W}^{\hat{X}}_{e'} = X$. Thus by the inductive assumption we can apply lemma 2.3.2 to show there is a uniform self-modulus of degree $\underset{\sim}{d}$.

In the case κ is a proper limit then by definitions 1.3.10 and 1.3.11 there is a uniformly in κ computable sequence $\langle\langle\!\langle \beta_k, c\rangle\!\rangle\rangle_{k\in\omega}$ of REA indexes sets with $X = \bigoplus_{i\in\omega} X_i$. By our inductive hypothesis there is a function f_k and a reduction Φ_{e_k} computing X_k from and $g \gg f_k$. If we could assume there was a a single computable function h with $h(\langle\!\langle \beta_k, c\rangle\!\rangle) = e_k$ for every k then by 2.3.4 we would be done. However, the inductive hypothesis only guarantees that for every $\beta <_{\mathcal{O}} \kappa$ we have a function h_β which operates as desired on pairs $\langle\!\langle \beta', c'\rangle\!\rangle$ with $\beta' <_{\mathcal{O}} \kappa$ and we need a single function to extend all the way up through κ. Intuitively this shouldn't be a problem because we always extend our translation from REA indexes to indexes for reductions in a uniform fashion so we should be able to offer one definition of this function that works for all $\beta <_{\mathcal{O}} \kappa$. However, rigorously demonstrating this requires us to deploy the fixed point lemma and approach the problem in a slightly different fashion. We demonstrate this approach in this lemma to illustrate the method but will omit it in later proofs.

Instead of trying to induct directly we now attempt to define the computable operations that let us extend our computable translation to higher ordinals. Thus given some notation for a limit ordinal λ we simply suppose we have a computable function h_λ that translates indexes for a REA set of level less that λ as we desired above. Now if $\Phi_t = h_\lambda$ then by the argument we couldn't use in the prior paragraph and the uniformity of lemma 2.3.4 there is a computable function p so that $p(t)$ is an index for $h_{\lambda +_{\mathcal{O}} 1}$. Alternatively suppose that for some notation $\alpha +_{\mathcal{O}} 1$ the index of our translation $h_{\alpha +_{\mathcal{O}} 1}$ is t. As our argument for successor stages in the first paragraph was evidently effective there is some computable function s so that $s(t)$ is an index for a function $h_{\alpha +_{\mathcal{O}} 2}$. It only remains to show that we can put these methods of extending our translation together to form a single computable function.

Now given an arbitrary REA index $a = (\kappa, e)$ we define a computable function c, making use of our conventions 1.3.1 and 1.3.11, satisfying:

$$\Phi_{c(e)}(\emptyset; a) = \begin{cases} e_0 & \text{if } \kappa = 0 \text{ where } (\forall k)\,(\Phi_{e_0}(g; k) = 0) \\ \Phi_{s(e)}(\emptyset; a) & \text{if } \kappa \text{ (is a successor notation)} \\ \Phi_{p(t)}(\emptyset; a) & \text{if } \kappa \text{ (is a limit notation)} \end{cases}$$

By the fixed point theorem [19] there is some e so that $\Phi_{c(e)} \simeq \Phi_e$. We claim that if a is an index for a REA set X then $\Phi_e(\emptyset; a)$ is an index for a uniform self-modulus $f \equiv_T X$. This is trivial if a codes for a 0-REA set, i.e., $\emptyset$, and if we assume that Φ_e works for all indexes of γ-REA sets for $\gamma <_{\mathcal{O}} \kappa$ the definition of $\Phi_{c(e)}$ guarantees $\Phi_{c(e)}$ and hence Φ_e must also work for κ. This completes the proof. □

The proof technique used in proposition 2.3.5 are fairly standard in constructions dealing with ordinal notations. The general pattern is that some computable function must be extended to all ordinal notations while maintaining some property but to show this property can be maintained at limit stages seems to require that one have already shown that there is a single computable function that works for every stage below the limit. As the argument here illustrated we can overcome this difficulty by showing that there is a uniform means to transform a computable function defined on the ordinals less than κ to one that is also defined on κ and letting the fixed point theorem do the work. In the future we will simply observe that the relevant steps are uniform and let the interested reader refer back to this proof.

Corollary 2.3.6. *There is a uniform self-modulus of* $0^{(\alpha)}$ *for every notation* α. *Also every Kleene H-set is of the same degree as a uniform self-modulus.*

Proof. $0^{(\alpha)}$ is a α-REA set as are all Kleene H-sets. □

Corollary 2.3.7. *Every* Δ^1_1 *set has a uniform modulus.*

Proof. By theorem 1.3.8 if $X \in \Delta^1_1$ then $X \leq_{\mathbf{T}} 0^{(\alpha)}$ for some $\alpha <_{\mathcal{O}} \omega_1^{ck}$. By corollary 2.3.6 this entails X is below a set with a uniform modulus and thus by lemma 2.2.4 X has a uniform modulus. □

2.4 Definability From Fast Growing Functions

An obvious question to ask now is what degrees have a modulus. This question takes on particular import since the property of having a modulus and that of being recursively encodable seem to both capture some notion of being computable from every fast enough growing function. If these didn't turn out to capture the same class of sets one might worry that our understanding of the concepts was flawed. But indeed a theorem of Groszek and Slaman [6] shows that the sets with a self-modulus are just the Δ^1_1 sets, which by Solovay's result [20] are exactly the recursively encodable sets.

Theorem 2.4.1 (Slaman and Groszek)**.** *X has a modulus iff* $X \in \Delta^1_1$.

We approach this problem by breaking it up into pieces. First we demonstrate the claim holds for sets with a uniform modulus and then show that sets with a modulus have a uniform modulus as well.

Lemma 2.4.2. *X has a uniform modulus iff X is* Δ^1_1.

Proof. $\Delta^1_1 \Rightarrow$ uniform modulus by corollary 2.3.7. To show the other direction we offer a Π^1_1 and Σ^1_1 definition of X in terms of the witness Φ_r for the uniform modulus $f_X \in \omega^\omega$ of X.

$$X(y) = 1 \iff (\exists f \in \omega^\omega)\,(\forall \sigma \gg f)\left[\Phi_r(\sigma; y)\downarrow \implies \Phi_r(\emptyset; y) = 1\right] \tag{2.4.2.1}$$

$$\iff (\forall g \in \omega^\omega)\,(\exists \sigma \gg g)\left[\Phi_r(\sigma; y)\downarrow = 1\right] \tag{2.4.2.2}$$

To check the $\Rightarrow$ direction of equation (2.4.2.1) we note that setting $f = f_X$ satisfies the equation. By the definition of a uniform modulus $g \gg f_X \implies \Phi_r(g; y)\downarrow = X(y)$ and since any $\sigma \gg f_X$ extends to such a g the right hand side must hold. Conversely assume the right hand side holds for $f = \bar{f}$ and choose some function $h \gg f_X, \bar{f}$. Again by the definition of a uniform self-modulus some initial segment $h\restriction_n$ of h causes $\Phi(\sigma; y)\downarrow = X(y)$ so by assumption $X(y) = 1$.

We verify equation (2.4.2.2) in a similar manner. If the right side of equation (2.4.2.2) holds there must be some σ, g with $\sigma \gg g \gg f_X$ with $\Phi_r(\sigma; y)\downarrow = 1$. Conversely if $X(y) = 1$ and $g \in \omega^\omega$ we can always choose $\sigma \gg g, f_X$ to make $\Phi_r(\sigma; x)\downarrow$. Hence $X \in \Sigma^1_1$ and $X \in \Pi^1_1$ so $X \in \Delta^1_1$.

□

Lemma 2.4.3. *X has a modulus if and only if X has a uniform modulus.*

Proof. Assume that X has a modulus f. Following the approach for Groszek and Slaman we try to build a $g \gg f$ so that $g \not\geq_{\mathbf{T}} X$ in violation of the assumption that f is a modulus. Our failure to do so will guarantee the existence of a uniform modulus f^* of X. We build g using conditions of the form (σ, h) where $\sigma \in \omega^{<\omega}$ and $h \in \omega^\omega, h \gg f$ and $\sigma \gg h$. We view these conditions as a partially ordered set with the ordering given by

$$(\sigma_1, h_1) \succ (\sigma_2, h_2) \iff \sigma_1 \supset \sigma_2 \wedge h_1 \gg h_2$$

While we could frame this argument in a slightly more compact form in terms of Hechler forcing writing out the argument as a construction is more enlightening. We start with our initial condition $(\sigma_0, h_0) = (\emptyset, f)$ and suppose we have defined (σ_n, h_n) we first try to choose (σ_{n+1}, h_{n+1}) so that $\sigma_{n+1} \supsetneq \sigma_n$ and:

$$(\sigma_{n+1}, h_{n+1}) \succ (\sigma_n, h_n) \wedge (\exists x)\,(\Phi_n(\sigma_{n+1}; x)\downarrow \neq X(x)) \tag{2.4.3.1}$$

Otherwise we try to satisfy

$$(\sigma_{n+1}, h_{n+1}) \succ (\sigma_n, h_n) \wedge (\exists x) \left(\forall \left(\tau, \hat{h}\right) \succ (\sigma_{n+1}, h_{n+1})\right) \left(\Phi_n(\sigma'; x)\uparrow\right) \tag{2.4.3.2}$$

Suppose for every n we are able to satisfy either equation (2.4.3.1) or equation (2.4.3.2). As σ_{n+1} is always longer than σ_n, $\bigcup_{n\in\omega} \sigma_n = g$ gives us a well-defined total function with $g \gg f$. Furthermore if we satisfied equation (2.4.3.1) at stage e then $\Phi_e(g) \mid X$ and if we satisfied equation (2.4.3.2) then $\Phi_e(g)$ isn't total. But this would mean that $g \not\geq_{\mathbf{T}} X$ contradicting our assumption. Therefore there is some stage, say e, in our construction at which we are incapable of satisfying either equation. Let $f^* = h_e$ and define $\Phi(g)$ as follows.

$$\Phi(g; x) = \Phi_e(\tau; x) \text{ where } \tau \supset \sigma_e \wedge \tau \gg g \wedge \Phi_e(\tau; x)\downarrow \tag{2.4.3.3}$$

Obviously $\Phi(g; x)$ is a computable functional and if $g \gg h_e$ then $\Phi(g; x)\downarrow$. Otherwise g would be a witness to the satisfiability of equation (2.4.3.2). Furthermore $\Phi(g) \nmid X$ as otherwise it would be a counterexample to the failure of equation (2.4.3.1). Hence Φ witnesses that h_e is a uniform modulus for X. □

Proof of theorem 2.4.1. $X \in \Delta^1_1$ if and only if (by lemma 2.4.2) X has a uniform modulus if and only if (by lemma 2.4.3) X has a modulus. □

Corollary 2.4.4. *X has a modulus if and only if X is recursively encodable.*

Proof. By a result of Solovay [20] the recursively encodable sets are just the Δ^1_1 sets and by theorem 2.4.1 so are the sets with a modulus. □

Chapter 3

Categorizing (Uniform) Self-Moduli

3.1 Categorizing Uniform Moduli

We've seen that a uniform modulus for X serves as a witnesses that $X \in \Delta^1_1$ which is interesting in itself but so far we haven't said much about the relation between sets and their uniform modulus. Since it is precisely the prospect of relating a set to the rates of growth sufficient to compute the set that makes moduli a more promising tool than recursively encodable sets for analyzing computational properties of fast growing functions this is an obvious direction to explore. However, rather than jumping in blindly and trying to build a systematic picture of moduli in total generality in one go we start by examining the uniform self-moduli and expand our results about these objects to moduli more generally.

Theorem 3.1.1. *$f \in \omega^\omega$ is a uniform self-modulus iff if and only if f is a Π^0_1 singleton.*

Proof. $\Rightarrow$Suppose Φ_u witnesses that f is a uniform self-modulus. We claim that f is the unique solution to the Π^0_1 formula

$$\Psi(h) = (\forall \sigma \gg f)\,(\forall t)\,(\forall x)\,(\neg \Phi_u(\sigma; x)\downarrow_t \neq h(x)) \tag{3.1.1.1}$$

Evidently f is a solution to Ψ and if $f' \neq f$ then for some x and $g \gg f', f$ then $\Phi_u(g; x)\downarrow = f(x) \neq f'(x)$. Thus there is some $\sigma \subset g$ witnessing that $\neg\Psi(f')$. Hence f is the unique solution to Ψ.

$\Leftarrow$Let $\Psi(h) = \forall z R(h\restriction_z, z)$ be the Π^0_1 formula witnessing that f is a Π^0_1 singleton. We describe a uniform procedure for computing f from any $g \gg f$. Define $T_g = \{\sigma \in \omega^{<\omega} | \sigma \ll g\}$. Since $g \gg f$, $f \in [T]$ and if $f \in [T]$ then Ψf so f is the unique path through T. Since T_g is finitely branching by König's lemma [15] if $\sigma \in T_g$ doesn't extend to an infinite path then there are only finitely many elements in T_g extending σ. Therefore to compute $f\restriction_x$ from g we simply search for an n so that all paths of length n in T_g extend a common initial segment of length x. Obviously this computation can be done uniformly in g. □

This theorem characterizes the degrees containing uniform self-moduli in terms of the definability of elements of ω^ω. A simple lemma lets us translate between this and the definability of members of 2^ω.

Lemma 3.1.2. *A degree $\underset{\sim}{d}$ contains a Π^0_1 singleton $f \in \omega^\omega$ if and only if it contains a Π^0_2 singleton $F \in 2^\omega$.*

Proof. ⇒Define $F = \{\langle\langle x, y\rangle\rangle | f(x) = y\}$. Suppose $\psi(h) = \forall x \varphi(h\restriction_x, x)$ witnesses that f is a Π^0_1 singleton then, identifying $h\restriction_x$ with the corresponding element of $\omega^{<\omega}$ in turn identified with it's code, define

$$\Psi(X) = (\forall x)(\exists \vec{y})\left((\forall x' < x)\left[\langle\langle x, \vec{y}_x\rangle\rangle \in X\right] \wedge |\vec{y}| = x \wedge \psi(\vec{y}, x)\right) \wedge \\ (\forall x)(\forall y)\left(\forall y'\right)\left(y \neq y' \wedge \langle\langle x, y\rangle\rangle \in X \implies \langle\langle x, y'\rangle\rangle \notin X\right)$$

Obviously Ψ witnesses that F is a Π^0_2 singleton and evidently $f \equiv_T F$.

⇐Suppose that $\Psi(X) = \forall x \exists y \varphi(X\restriction_y, x, y)$ witnesses that $F \in 2^\omega$ is a Π^0_2 singleton. Define

$$f(x) = \langle\langle \ulcorner X\restriction_z \urcorner, z\rangle\rangle \text{ where } z = \min\{y | \varphi(X\restriction_y, x, y)\} \tag{3.1.2.1}$$

$$\psi(h) = (\forall x)\left(\exists \vec{X} < h(x)\right)(\exists z < f(x))\left(h(x) = \langle\langle \vec{X}, z\rangle\rangle \wedge |\vec{X}| = z \wedge \varphi(\vec{X}, x, z)\right) \tag{3.1.2.2}$$

Clearly ψ witnesses that f is a Π^0_1 singleton and evidently $f \equiv_T F$. □

Corollary 3.1.3. *$X \subset \omega$ has a uniform self-modulus if and only if X is a Π^0_2 singleton*

Proof. Immediate from lemma 3.1.2, theorem 3.1.1 and the closure of the Π^0_2 singletons under Turing degree. □

Actually we can generalize theorem 3.1.1 to classify those sets X within κ jumps of a uniform modulus for X.

Theorem 3.1.4. *There is some uniform modulus $f \leq_{\mathbf{T}} X^{(\kappa)}$ for $\kappa \in \mathcal{O}$ if and only if X is a $\Pi^0_{2+\kappa}$ singleton.*

To prove this result we first relativize the fact that $0^{(\kappa)}$ contains a uniform self-modulus to show that $X^{(\alpha)}$ contains a uniform self-modulus relative to X.

Lemma 3.1.5. *For every $X \subset \omega$ $\kappa <_{\mathcal{O}} \omega_1^{ck}$ there is a function $\theta^\kappa_X \equiv_T X^{(\kappa)}$ and a single reduction computing $X^{(\kappa)}$ from $g \oplus X$ for any $g \gg \theta^\kappa_X$.*

Proof. Straightforward relativization of corollary 2.3.6. □

proof of theorem 3.1.4. ⇒This proof is the same as that of theorem 3.1.1 except we replace $\sigma \gg f$ with $\sigma \gg \Phi_e(X^{(\kappa)})$ for some x which gives us a $\Pi^0_{2+\kappa}$ sentence.

⇐Suppose X is the unique solution of the $\Pi^0_{2+\kappa}$ formula Ψ. By lemma 1.3.19 this is equivalent to X being the unique path through a $\Pi^0_{2+\kappa}$ downward closed tree T. Suppose $\forall x \exists y \varphi(\sigma, x, y) \iff \sigma \in T$. Set $f = \sup \theta^\kappa, h$ where θ^κ is a uniform modulus of $0^{(\kappa)}$ and $h(\langle\langle x, l\rangle\rangle) = \min\{y | \varphi(X\restriction_l, x, y)\}$. Now if $g \gg f$ then g can compute the tree $T_g = \{\sigma | \forall l < |\sigma| \forall x < |\sigma| \exists y < g(\langle\langle x, l\rangle\rangle)\varphi(\sigma\restriction_l, x, y)\}$. Since clearly X is the unique infinite path through T_g by König's lemma g uniformly computes X. □

3.2 Exploring Self-Moduli

We've now developed a substantial collection of self-moduli (all the Π^0_1 singletons) but we don't have any example yet of a degree that lacks a self-modulus.

Theorem 3.2.1. *If $G \subset \omega$ is a 2-generic then G has no self-modulus.*

Proof. Given a 2-generic G and a total function $f = \Phi_e(G)$ we try to build $g \gg f$ with $g \not\geq_{\mathbf{T}} G$ by Hechler conditions much like we did in lemma 2.4.3 in our presentation of Slaman and Groszek's argument. As before we deal with pairs (σ, h) with $\sigma \gg h$ but now we require h be computable. We start our construction with $\sigma_0 = \emptyset$ and $h_0 = 0$. At stage $n + 1$ we try to find a string σ_{n+1} and a computable function h_{n+1} with $\sigma_n \subsetneq \sigma_{n+1}$ and $h_{n+1} \gg h_n$ and $\sigma_{n+1} \gg f$ such that

1. $\Phi^2_n(\sigma_{n+1}) \mid G$
2. For some x $(\forall \tau \supseteq \sigma_{n+1})\left(\tau \gg h_{n+1} \implies \Phi^2_{n+1}(\tau; \bar{x})\uparrow\right)$

If we can't satisfy either of these conditions we simply set $h_{n+1} = h_n$ and extend σ_n with an arbitrary $\sigma_{n+1} \supsetneq \sigma_n$ with $\sigma_{n+1} \gg h_{n+1}$ and $\sigma_{n+1} \gg f$. Now if we satisfy either prong it's clear that our function $g = \cup_{n\in\omega}\sigma_n$ doesn't compute G by way of Φ^2_{n+1} and as $g \gg f$ this would complete the lemma if it held for every n. Therefore assume that we can't satisfy either condition at stage $n+1$. From our assumption we can infer that G must force the following formulas.

$$G \Vdash (\forall x)\,(\exists s)\,\Phi_e(G;x)\downarrow_s \tag{3.2.1.1}$$

$$G \Vdash \neg\,(\exists x, j)\,(\forall \tau \supseteq \sigma_n)\,(\forall t) \\ \left[(\forall z < |\tau|)\,(h_n(y)\downarrow_t < \tau(y) \wedge \Phi_e(G;z)\downarrow_t \leq \tau(z)) \implies \Phi^2_{n+1}(\tau;x)\not\downarrow_t\right] \tag{3.2.1.2}$$

Note that as G is 2 generic it is sufficent to show these formula actually hold for G. Equation (3.2.1.1) simply asserts that f is total which is required for f to be a self-modulus of G. Equation (3.2.1.2) simply asserts that our commitment $\sigma_n \subset g$ doesn't guarantee the non-totality of $\Phi_{n+1}(g)$ which must hold by the failure to satisfy the second prong.

Now select some l long enough that $G\restriction_l$ forces all both these formula. Now given any $x > l$ pick some υ_1 and υ_2 both extending $G\restriction_l$ with $\upsilon_1(x) \neq \upsilon_2(x)$. Thus if there were a single $\tau \supseteq \sigma_n$ and some υ'_1 and υ'_2 such that $\tau \gg h_n$ and $\tau \gg \Phi_e(\upsilon'_1;,)\Phi_e(\upsilon'_2)$ with $\operatorname{dom}\Phi_e(\upsilon'_1;,)\operatorname{dom}\Phi_e(\upsilon'_2) \supset \operatorname{dom}\tau$ with $\Phi^2_{n+1}(\tau;x)\downarrow$ this would contradict equation **??** since both υ'_1 and υ'_2 could be extended to 2-generics. On the other hand if there was no such τ then we could define the computable function

$$h_{n+1}(y) = \max \Phi_e(H^1;y), \Phi_e(H^2;y), h_n(y)$$

where

$$H^i = \cup_{n\in\omega} H^i_n$$

$$H^i_0 = \upsilon'_i$$

$$H^i_{n+1} = \epsilon \supset H^i_n \text{ which is first such observed to satisfy } (\forall z < n+1)\,\Phi_e(\epsilon;()z)\downarrow$$

This function will certainly be total since each H^i_n extends $G\restriction_l$ so by equation (3.2.1.1) there is some 2-generic G' extending them make $\Phi_e(G')$ total. However, by the argument above if $\tau \supsetneq \sigma_n$ and $\tau \gg h_{n+1}(y)$ then $\Phi^2_{n+1}(\tau;x)\uparrow$ But this directly contradicts the failure to satisfy the second condition. Thus for every n there is a σ_n satisfying one of the two conditions thus guaranteeing that G has no self-modulus.

□

Chapter 4

A Non-Iterative Self-Modulus

4.1 Theorem

Theorem 4.1.1. *For each $\alpha \leq \omega_1^{ck}$ there is a non-computable uniform self-modulus f such that if $X \in \Delta_\alpha \wedge X \leq_{\mathbf{T}} f$ then X is recursive.*

To prove this theorem for $\alpha > 2$ explicitly defining the action of the reduction witnessing that f is a uniform self-modulus quickly becomes too complicated to manage. Instead we take inspiration from our identification of uniform self-moduli with Π^0_1 singletons in ω^ω.

Lemma 4.1.2. *Suppose T is a computable (downward closed) tree and f is the unique infinite path through T then f is a uniform self-modulus.*

Proof. By lemma 1.3.19 it is sufficient to observe that every computable tree is also a Π^0_1 tree. Hence if f is the unique path through a computable tree it is a Π^0_1 singleton. $\square$

Note that every tree we encounter in this chapter will be closed under initial segments.

4.2 Requirements

In order to prove the theorem we will try to build a computable tree T with a unique path f satisfying the following requirements for every $e, i \in \omega$.

$$R_{e,i}\text{: Either } (\exists x)\neg\left(\Phi^2_e(0^{(\alpha)};x)\downarrow = \Phi^2_i(f;x)\downarrow\right) \text{ or } (\exists i')\left(\Phi^2_i(f) \simeq \Phi^2_{i'}\right)$$

N_e: $(\exists x) \neg (\Phi_e(\emptyset; x)\downarrow = f(x))$

To keep things simple we will build the tree T in stages and decide at most one membership question about T per stage.

Rule 4.2.1. At stage s of our construction we will decide whether $\sigma \in T$ where $\ulcorner \sigma \urcorner = s$.

Since we will often want to refer to the state of the tree at stage s we introduce convenient notation for this object.

Notation 4.2.2. For a tree T set $T[s] = \{\sigma \in T | \ulcorner \sigma \urcorner \leq s\}$

Note that by our stipulation in subsection 1.1.1 our coding function satisfies lemma 1.1.4 so if $\sigma \subset \sigma'$ we always decide $\sigma \in T$ before $\sigma' \in T$. The property has the convenient consequence that our stage s approximation to T is $T[s]$ and if $\ulcorner \sigma \urcorner \leq s$ then $\sigma \in T \iff \sigma \in T[s]$.

We will satisfy $R_{e,i}$ in the standard fashion: we will attempt to satisfy the first prong of the requirement so throughly that if we fail then we it must be because we've second prong. The basic strategy for satisfying $R_{e,i}$ is to wait until a stage s at which we observe $\sigma_0, \sigma_1 \in T[s]$ forming a Φ^2_i split and ensure that $f \supset \sigma_1$ if $\Phi^2_e(0^{(\alpha)}) \supseteq \sigma_0$ and otherwise $f \supset \sigma_0$. To ensure that the requirements don't interfere with each other so much that our construction fails a requirement $R_{e,i}$ won't be allowed to pick just any Φ^2_i split but will instead have to wait for a split where σ_0 and σ_1 agree on some initial segment ρ being used by higher priority machinery. Thus at best what we will be able to guarantee is that if for every initial segment of the true path $f\restriction_n$ is extended by $\sigma_0, \sigma_1 \in T$ forming a Φ^2_i split then $(\exists x)\left(\neg \Phi^2_e(0^{(\alpha)}; x)\downarrow = \Phi^2_i(f; x)\downarrow\right)$. This, however, is sufficient to show that we satisfy $R_{e,i}$.

Lemma 4.2.3. *Suppose there is no Φ^2_i split $\sigma_0, \sigma_1 \in T$ with $\sigma_0, \sigma_1 \supseteq f\restriction_n$ then $R_{e,i}$ is satisfied.*

Proof. We compute $\Phi^2_i(f; x)$ by running the algorithm that searches for a $\sigma \in T$ with $\sigma \supseteq f\restriction_n$ and $\Phi^2_i(\sigma; x)\downarrow$ the result of which it outputs. If $\Phi^2_i(f; x)\downarrow$ then for some $m \geq n$ $\Phi^2_i(f\restriction_m; x)\downarrow$ so if out computation doesn't converge then neither does $\Phi^2_f(i; x)$ thus satisfying the first half of the requirement. Alternatively if our algorithm halts and disagrees with $\Phi^2_i(f; x)\downarrow$ than $f\restriction_m$ and σ would form a Φ^2_i split extending $f\restriction_n$ in violation of the assumption. Hence if $\Phi^2_f(i)$ is total we satisfy the second prong of the requirement. □

With this lemma in hand we can modify our requirements to better reflect our construction.

$$R_{e,i}\text{: } (\exists n)\Big((\exists \sigma_0, \sigma_1 \in T)\left[\sigma_0, \sigma_1 \supset f\restriction_n \wedge \Phi_i^2(\sigma_0) \mid \Phi_i^2(\sigma_1)\right] \implies (\exists x)\left[\neg\Phi_e^2(0^{(\alpha)}; x)\downarrow = \Phi_i^2(f; x)\downarrow\right]\Big)$$

$$N_e\text{: } (\exists x)\neg\left(\Phi_e(\emptyset; x)\downarrow = f(x)\right)$$

Lemma 4.2.4. *Suppose T is a computable tree with a unique infinite path $f \in [T]$ satisfying requirement 4.2 then f satisfies theorem 4.1.1.*

Proof. By lemma 4.2.3 we can instead suppose f satisfies requirement 4.2. Since f satisfies N_e for every e it is not equal to any computable function and by satisfying $R_{e,i}$ for every pair (e, i) f avoids computing any set $X \leq_{\mathbf{T}} 0^{(\alpha)}$ that doesn't satisfy $X \equiv_T \underset{\sim}{0}$. Finally by lemma 4.1.2 f is a uniform self-modulus. □

4.3 Markers & Assignments

To organize our construction we place markers on nodes in our tree to indicate that a particular node is reserved for a certain purpose in our construction. Since at a given stage s we won't know which branches in $T[s]$ extend to an infinite path we must try to satisfy the requirements on all branches. Thus we will assign a single marker to different locations on different paths and it may occupy different locations at different stages. Therefore we formally define a marker

Definition 4.3.1. A **marker** $\mathfrak{m}$ is a computable function taking a stage s to a finite, pairwise incompatible set of nodes, $\mathfrak{m}_s$. Also we set $\mathfrak{m}_\infty = \bigcup_{t\in\omega}\bigcap_{t'>t}\mathfrak{m}_{t'}$

We will frequently abuse notation and speak of markers as if they occupied only a single location. Thus we will often use a statement like "$\mathfrak{m}_s$ is located at σ" to mean that $\sigma \in \mathfrak{m}_s$ or, when context restricts our attention to initial segments of some ρ, to mean that $\sigma \in \mathfrak{m}_s(\rho)$. We use the symbol $\mathfrak{m}$ when we wish to refer to an arbitrary marker but our construction will make use of four different types of markers and it will be convenient to distinguish between them. The types of markers we use and their intended application are listed below. Note that typographic clarity demands we avoid placing an excessive number of indexes on each symbol so we don't assume that the value of the subscripts and superscripts of the symbols below uniquely determine a marker.

$\boldsymbol{\alpha}_{\langle e\rangle,}$**:** Used to diagonalize against Φ_e to meet requirement N_e.

$\boldsymbol{r}^*_{\langle e,i\rangle}$**:** A collection of markers used to occupy the nodes injured when $R_{e,i}$ acts to reserve a Φ_i^2 split.

$\boldsymbol{r}^j_{\langle e,i\rangle}$**:** This marker is placed at the node τ_j where $\tau_k, k \in \{0,1\}$ forms one of the Φ_i^2 splits reserved for $R_{e,i}$. $\boldsymbol{r}^j_{\langle e,i\rangle}$ serves to tag the nodes in the split and initiate the Re, i module.

ξ^{φ}**:** This marker does the actual work to ensure the unwanted side of the Φ_i^2 split for $R_{e,i}$ is terminated. As choosing the correct path is a $0^{(\alpha+1)}$ question the markers in our construction will often hire employee markers to help them reach the right conclusion and the formula φ encodes the orders given to this employee.

We now explain how these markers are assigned to nodes in our tree.

Definition 4.3.2. The **request queue**, $\mathcal{Q}_s$ is the set of ordered pairs $(\boldsymbol{m}, \sigma)$ such that a **request to cover** σ with $\boldsymbol{m}$ (or alternatively **enumerate** $\boldsymbol{m}$ **above** σ) was enumerated at $t \leq s$ and was not removed from $\mathcal{Q}$ at any t' with $t \leq t' \leq s$. We also adopt the notation:

$$\mathcal{Q}_\infty = \bigcup_{s\in\omega} \bigcap_{t>s} \mathcal{Q}_s \tag{4.3.2.1}$$

$$\mathcal{Q}_{\hat{s}}(\rho) = \{\boldsymbol{m} | (\boldsymbol{m}, \sigma) \in \mathcal{Q}_{\hat{s}} \wedge \sigma \subseteq \rho \wedge (\forall \tau \in \boldsymbol{m}_{\hat{s}}) (\tau \mid \rho)\} \text{ for } \hat{s} \in \omega \cup \{\infty\} \tag{4.3.2.2}$$

As the name suggests a request to cover σ with $\boldsymbol{m}$ will result in $\boldsymbol{m}$ containing a T-cover of σ. Particularly we use the priority of the enumerated markers to control when they are assigned to the tree as follows. We will implement our finite injury argument by assigning priorities to requirements and markers in our construction and only allowing those requirements that are important than the markers they wish to remove from T to act. This will be described in the next section but we need to introduce the notion of priority to explain our rule for assigning markers to nodes.

Notation 4.3.3. We call the integer we associate to every marker and requirement the **priority** of the marker/requirement and denote this number by the application of the 'function'[1] $\mathcal{P}$. To avoid the ambiguity inherent in saying "higher priority" we will say one marker/requirement Q is **more important** than Q' to indicate that $\mathcal{P}(Q) < \mathcal{P}(Q')$.

[1] $\mathcal{P}$ isn't truly a function as requirements aren't truly elements in the object language. Technically $\mathcal{P}$ is merely a notational convenience analogous to a macro in a programming language.

Rule 4.3.4. If, at stage s of the construction, the node σ is allowed to enter $T[s]$ we assign to σ the marker in $\mathcal{Q}(\sigma)$ with the numerically least priority that isn't already assigned to some $\sigma' \subset \sigma$.

If any marker anywhere in our construction could enumerate requests for arbitrary markers to cover random nodes our construction would quickly spiral into overwhelming complexity. We first limit this ability by giving an explicit and exhaustive enumeration of requests for the markers $\boldsymbol{\alpha}_{\langle e \rangle,}$.

Rule 4.3.5. At the beginning of the construction for every $e \in \omega$ we enumerate a request for $\boldsymbol{\alpha}_{\langle e \rangle,}$ to be enumerated above $\emptyset$. These are the only requests for the markers $\boldsymbol{\alpha}_{\langle e \rangle,}$ made in the construction.

We further limit requests by restricting the relation between $\boldsymbol{m}$ and the nodes σ' it can request be covered. This limit reflects the idea that a request enumerated by $\boldsymbol{m}$ should be seen as a delegation of part of the task $\boldsymbol{m}$ was assigned to a subordinate since it must have been work that $\boldsymbol{m}$ could have done itself if sufficiently capable (had access to a powerful oracle).

Rule 4.3.6. If $\boldsymbol{m}$ enumerates a request at stage s for $\boldsymbol{m}'$ to cover σ' then there exist σ, y with $\sigma^\frown\langle y \rangle = \sigma'$ and $\sigma \in \boldsymbol{m}_s$. Also $\boldsymbol{m}'$ is of the form ξ^{φ} while $\boldsymbol{m}$ is of any form except $\boldsymbol{r}^{*}_{\langle e,i \rangle}$.

Later in our construction it will be important to easily indicate how the markers relate in terms of their requests.

Definition 4.3.7. Say that the marker $\boldsymbol{m}$ is a **child** of $\boldsymbol{m}'$ denoted $\boldsymbol{m} \lhd_1 \boldsymbol{m}'$ if $\boldsymbol{m}'$ was the marker that requested $\boldsymbol{m}$. We say $\boldsymbol{m}$ is a **descendant** of $\boldsymbol{m}'$ if $\boldsymbol{m} \lhd_\omega \boldsymbol{m}'$ where $\lhd_\omega$ is the transitive closure of $\lhd_1$. We define **parent** and **ancestor** analogously.

At this point we now have the tools to clarify the meaning of our notation for markers. Without inducing confusion and inordinate proliferation of indexes on our symbols we can't guarantee there is a bijective correspondence between the indexes of the symbols we use and the actual markers but for some markers it's important that we be able to uniquely refer to them in a simple fashion so we make the following stipulations.

Notation 4.3.8.

- $\boldsymbol{r}^{j'}_{\langle e',i' \rangle} = \boldsymbol{r}^{j}_{\langle e,i \rangle} \iff e = e' \wedge i = i' \wedge j = j'$
- $\boldsymbol{\alpha}_{\langle e \rangle,} = \boldsymbol{\alpha}_{\langle e' \rangle,} \iff e = e'$

- $\xi^{\varphi} = \xi^{\varphi'}$ if and only they are both symbols represent the marker resulting from the same request in $\mathcal{Q}$.

4.4 Priorities & Injuries

The priorities we will use during the construction are the following.

$$\mathcal{P}\left(R_{e,i}\right) = \mathcal{P}\left(\boldsymbol{r}^{*}_{\langle e,i\rangle}\right) = \mathcal{P}\left(\boldsymbol{r}^{j}_{\langle e,i\rangle}\right) = 3 \cdot \langle e\rangle ij + 1 \tag{4.4.0.1a}$$

$$\mathcal{P}\left(\boldsymbol{n}_{\langle e\rangle,}\right) = 3 \cdot e \tag{4.4.0.1b}$$

$$\mathcal{P}\left(\xi^{\varphi}\right) = 3 \cdot \langle\langle \upsilon, t, + \rangle\rangle 2 \qquad \text{where } t \text{ is the stage } \left(\xi^{\varphi}, \upsilon\right) \text{ entered } \mathcal{Q} \tag{4.4.0.1c}$$

While our construction must be infinitary to diagonalize against computations in $0^{(\alpha)}$ this is reflected in the difficulty of recovering f from T so we won't require any infinitary arguments in the placement of our markers. However, along any particular path we still may need to move a given marker finitely many times to make room for the requirements $R_{e,i}$ and this is what we will refer to as an injury.

Definition 4.4.1. A marker $\boldsymbol{m}$ is **injured** at σ during stage s if $\sigma \in \boldsymbol{m}_s$ but $\sigma \notin \boldsymbol{m}_{s+1}$. $\boldsymbol{m}$ is injured along $\epsilon \in \omega^{<\omega} \cup \omega^{\omega}$ at stage s if $\boldsymbol{m}$ is injured at σ during stage s for some $\sigma \nmid \epsilon$.

The only reason we will need to injure a marker is to reserve two nodes τ_0, τ_1 forming a Φ^2_i split for some requirement $R_{e,i}$. During the normal (injury free) parts of our construction we will guarantee the various modules place nicely together by ensuring that if $\sigma \in \boldsymbol{m}$ and no ancestor of $\boldsymbol{m}$ demands that $\sigma \notin T^{\infty}$ then $\boldsymbol{m}$, aided by it's descendants, ensures that for all but one y $\sigma\widehat{\ }\langle y\rangle \notin T^{\infty}$. However, if ρ is longest common initial segment of τ_0, τ_1 then if $\upsilon \in \boldsymbol{m}'$ with $\upsilon \supseteq \rho$ and $\upsilon \not\supseteq \tau_i$ for some i then $\boldsymbol{m}'$ very likely will be prevented from doing it's duty since we can't let any such marker cut off either τ_i nor can we allow any extension of ρ incompatible with both τ_i to grow into an infinite branch. To keep things simple we don't bother to preserve markers assigned to extensions of either τ_i.

Furthermore a marker $\boldsymbol{m}$ operating at $\sigma \in \boldsymbol{m}$ may request that some child marker be enumerated over $\sigma\widehat{\ }\langle y\rangle$ for the purposes of delegating work that is too complex for $\boldsymbol{m}$ to accomplish on it's own. Indeed we will actually let $\boldsymbol{m}$ do this countably many times over $\sigma\widehat{\ }\langle y\rangle$. However, this delegation would be meaningless if $\boldsymbol{m}$ wasn't present to manage the process and ensure everything was completed. However, these unsupervised markers might cause trouble on their own so we also ensure that no employees are ever left unsupervised on our tree.

Rule 4.4.2. Suppose at the end of stage s we observe maximal elements $\tau_0, \tau_1 \in T[s]$ with $\Phi^2_{i,s}(\tau_0) \mid \Phi^2_{i,s}(\tau_1)$ and there exists a minimal string $\rho \subset \tau_0, \tau_1$ satisfying:

$$(\forall \boldsymbol{m})\,(\forall \sigma \in \boldsymbol{m})\,([\mathcal{P}(\boldsymbol{m}) \leq \mathcal{P}(R_{e,i}) \wedge \sigma \nmid \rho] \implies \sigma \subsetneq \rho) \tag{4.4.2.1}$$

$$(\forall e')\left(\mathcal{P}(R_{e,i}) < \mathcal{P}\left(R_{e',i}\right) \implies (\exists j)\left(\exists \tau \in \boldsymbol{r}^j_{\langle e',i\rangle}\right)[\tau \subsetneq \rho]\right) \tag{4.4.2.2}$$

$$\neg\left(\exists \rho' \subseteq \rho\right)\left(\rho' \in \boldsymbol{r}^*_{\langle e,i\rangle,s}\right) \tag{4.4.2.3}$$

Then $R_{e,i}$ removes every marker located at a string $\sigma \subseteq \rho$ as well as any requests in $\mathcal{Q}_s$ enumerated by these markers. Then at every node $\sigma \subseteq \rho$ except τ_0, τ_1 $R_{e,i}$ places a marker $\boldsymbol{r}^*_{\langle e,i\rangle}$. Finally $R_{e,i}$ places $\tau_j \in \boldsymbol{r}^j_{\langle e,i\rangle}$ for $j \in \{0,1\}$.

Definition 4.4.3. If $R_{e,i}, \rho, \tau_0, \tau_1, s$ are as in rule 4.4.2 we say $R_{e,i}$ **acts at** ρ during stage s to preserve τ_0, τ_1. We say $R_{e,i}$ is injured at ρ (along $\epsilon \supseteq \rho$) during stage $t > s$ if $\rho \notin \boldsymbol{r}^*_{\langle e,i\rangle,t+1}$.

As rule 4.4.2 is the exclusive means for injury in our construction we are almost in a position to show that the finite injury component of our construction works properly. First we need to guarantee that if $R_{e,i}$ acts at σ and is never injured at σ then no paths extend σ but not one side of the Φ^2_i-split. We do this by demanding that whenever υ is occupied by a marker $\boldsymbol{r}^*_{\langle e,i\rangle}$ no immediate extensions of υ are allowed to enter T.

Rule 4.4.4. If $\upsilon \in \boldsymbol{r}^*_{\langle e,i\rangle,s}$ and $\ulcorner \upsilon^\frown\langle y\rangle \urcorner = s$ then $\upsilon^\frown\langle y\rangle \notin T$.

Lemma 4.4.5. *Given any $f \in \omega^\omega$ there is some $n \in \omega$ so that either there is no Φ^2_i-split in T extending $f\restriction_n$ or there is some stage s and string $\sigma \subset f$ so that $R_{e,i}$ acts at σ during stage s and no marker of the form $\boldsymbol{r}^*_{\langle e,i\rangle}$ or $\boldsymbol{r}^j_{\langle e,i\rangle}$ gets injured at $\sigma' \supseteq \sigma$ at any stage after s. Furthermore $R_{e,i}$ only acts at finitely many times along f.*

Proof. We first note that unless that once $R_{e,i}$ has acted at σ by equation (4.4.2.3) it won't act at any extension of σ unless $R_{e,i}$ is injured at σ. This follows because by rule 4.4.4 the only extensions of σ not already occupied by markers of the same priority as $R_{e,i}$ will extend the Φ^2_i split. Additionally we note that the priority of the marker assigned to a given string υ can only increase during the construction so $R_{e,i}$ can only act finitely many times along any finite string.

We now prove this lemma by induction on the priority of $R_{e,i}$. First assume that no requirement $R_{e',i'}$ is more important than $R_{e,i}$. As our exclusive rule allowing for injuries never allows the injury of a higher priority marker no marker of the form $\boldsymbol{\mathscr{r}}^*_{\langle e,i\rangle}$ or $\boldsymbol{\mathscr{r}}^j_{\langle e,i\rangle}$ can be injured thus the lemma is easily satisfied. Now assume this lemma holds for all requirements $R_{e',i'}$ with $\mathcal{P}\left(R_{e',i'}\right) < \mathcal{P}\left(R_{e,i}\right)$. Pick a stage large t enough that no more important requirement ever acts after t along f. Now wait until a stage $t' > t$ where all of the finitely many markers with higher priority than $R_{e,i}$ that will ever be assigned locations along f after t have been assigned to some location along f. Let n be the length of the longest initial segment of f in $T[t']$. We know $R_{e,i}$ only acts finitely many times along $f{\upharpoonright}_n$ so the lemma is satisfied unless there is a Φ^2_i split extending $f{\upharpoonright}_n$. If there is such a split then $R_{e,i}$ can act on it since $f{\upharpoonright}_n$ witnesses that some ρ as in rule 4.4.2 exists and no higher priority requirement is available to injure any marker of the form $\boldsymbol{\mathscr{r}}^*_{\langle e,i\rangle}$ or $\boldsymbol{\mathscr{r}}^j_{\langle e,i\rangle}$ at any location extending $f{\upharpoonright}_n$ □

Lemma 4.4.6. *For every $f \in \omega^\omega$ and marker $\boldsymbol{\mathscr{m}}$ there are at most finitely many injuries of $\boldsymbol{\mathscr{m}}$ along f. Also there are only finitely many injuries that occur at any $\sigma \in T$.*

Proof. In order for a requirement $R_{e,i}$ to injure $\boldsymbol{\mathscr{m}}$ along f $R_{e,i}$ must act along f. Thus by lemma 4.4.5 no finite collection of requirements could be responsible for the failure of the lemma. However, there are only finitely many requirements with a higher priority than any particular marker so the first half of the lemma must hold. To verify the second half observe that this would require an infinite chain of markers with descending priorities acting to injure the marker at σ. □

Lemma 4.4.7. *If a request for $\boldsymbol{\mathscr{m}}$ to cover σ is enumerated at stage s and no marker responsible for that request is injured after stage s then $\boldsymbol{\mathscr{m}}_\infty$ contains a T-cover of σ.*

Proof. Suppose not then for some $f \in \omega^\omega, f \supset \sigma$ every assignment of $\boldsymbol{\mathscr{m}}$ along f is eventually injured. This can't happen infinitely many times by lemma 4.4.6. Thus the only way the rule can fail is if after some stage$t$ $\boldsymbol{\mathscr{m}}$ is never assigned to any location along f. Pick $t' > t$ so that every marker of numerically lower priority than $\boldsymbol{\mathscr{m}}$ has been injured for their last time along f and a $t'' > t$ so every marker of numerically lower priority than $\boldsymbol{\mathscr{m}}$ that will ever be assigned along f after t' has been. Now by rule 4.3.4 at stage $t'' + 1$, $\boldsymbol{\mathscr{m}}$ is assigned to a location along f. Contradiction. □

Lemma 4.4.8. *For every $\sigma \in T$ there is some marker $\boldsymbol{m}$ such that $\sigma \in \boldsymbol{m}_\infty$. Furthermore if any marker was ever injured at σ then $\boldsymbol{m}$ is of the form $\boldsymbol{r}^j_{\langle e,i \rangle}$ or $\boldsymbol{r}^*_{\langle e,i \rangle}$.*

Proof. By rule 4.3.5 the set $\mathcal{Q}(\tau)$ will never be empty for any τ so by rule 4.3.4 every node entering T is assigned a marker. Also every injured marker is replaced with a new marker so the first half of the lemma follows from lemma 4.4.6. To see the second half of the lemma holds observe that when σ is injured a marker of the form $\boldsymbol{r}^j_{\langle e,i \rangle}$ or $\boldsymbol{r}^*_{\langle e,i \rangle}$ will be assigned to σ. □

4.5 Marker Action

Our construction can be thought of as occurring on 2 layers. At the management layer we will explicitly enumerate requests for various markers and describe the changes made in case of injury. At the machinery layer various markers will act, both to enumerate requests and to decide what further nodes are in T. More precisely we restrict the influence of a particular marker to enumerating child markers and to deciding what happens to the immediate successors of it's node.

Rule 4.5.1. Suppose $\ulcorner \tau\frown\langle y \rangle \urcorner = s$ and $\tau \in \boldsymbol{m}_s$ then at stage s the module associate with $\boldsymbol{m}$ decides the membership of $\tau\frown\langle y \rangle$ in T.

Notation 4.5.2. We say φ is requested over $\sigma\frown\langle y \rangle$ to abbreviate the claim that the marker assigned to σ enumerates a request to cover $\sigma\frown\langle y \rangle$ by ξ^φ

Therefore we first articulate a way to give orders to the markers ξ^φ which, as the notation suggests, will be by way of sentences $\varphi \in \mathcal{CL}_{\omega_1,\omega}$. The intended interpretation of this formula is that if φ is requested over σ and $\neg\varphi$ then the ξ^φ module will ensure that no infinite path extends σ. On the other hand if φ holds then ξ^φ shouldn't block an infinite branch from extending σ. However, as ξ^φ must avoid proliferating branches that might grow to multiple infinite branches if $\tau \in \xi^\varphi$ then ξ^φ will ensure that all but one immediate extension of τ is killed off. Surprisingly it is actually easier to just go ahead and detail the behavior of ξ^φ rather than offering more theoretical definitions to capture how it behaves.

4.5.1 ξ^φ Module

Rule 4.5.3. Given a marker ξ^ψ, where ψ is syntactically written as a Σ^0_κ or Π^0_κ sentence (not merely equivalent to one), we define the behavior of ξ^ψ at $\tau \in \xi^\psi_\infty$ by specifying the immediate extensions of τ allowed to enter T and the requests enumerated by ξ^{ψ} based on the properties of ψ.

Case 1: $\kappa = \alpha +_{\mathcal{O}} 1 \wedge \psi = \exists x \hat{\psi}(x)$

Let φ_z be the effectively indexed sequence of formula given by lemma 1.3.18 such that $\varphi_z \in \Pi^0_{\alpha}$ and $\varphi_z \leftrightarrow (\exists x < z)\, \hat{\psi}(x)$. Let $\varphi'_0 = \varphi_0$ and $\varphi'_{k+1} = \varphi_{k+1} \wedge \neg\varphi_k$ Then the marker ξ^{ψ} ensures:

- $(\forall y)\,(\tau\widehat{\ }\langle y\rangle \in T)$
- For every y, φ'_y is requested over $\tau\widehat{\ }\langle y\rangle$.

Case 2: $\kappa = \alpha +_{\mathcal{O}} 1 \wedge \psi = \forall x \hat{\psi}(x)$

The marker ξ^{ψ} ensures:

- $\tau\widehat{\ }\langle y\rangle \in T \iff y = 0$
- For every n, $\hat{\psi}(n)$ is requested over $\tau\widehat{\ }\langle 0\rangle$.

Case 3: $\kappa = \lambda$ a limit and $\psi = \bigvee\!\!\!\bigvee_{i=0}^{\omega} \varphi_i, \varphi_i \in \Sigma^0_{\vec{\lambda}^{\lambda}_n} \cup \Pi^0_{\vec{\lambda}^{\lambda}_n}$

Let $\hat{\varphi}_i$ be the effectively generated formula guaranteed to exist by proposition 1.3.17 so that $\hat{\varphi}_n \iff \bigvee\!\!\!\bigvee_{i=0}^{n} \varphi_i$ where $\hat{\varphi}_n \in \Sigma^0_{\lambda_n} \cup \Pi^0_{\lambda_n}$. Furthermore let $\hat{\varphi}'_0 = \hat{\varphi}_0$ and $\hat{\varphi}'_{n+1} = \hat{\varphi}_{n+1} \wedge \neg\hat{\varphi}_n$. Then the marker ξ^{ψ} ensures:

- $(\forall y)\,(\tau\widehat{\ }\langle y\rangle \in T)$
- For every y, φ'_y is requested over $\tau\widehat{\ }\langle y\rangle$.

Case 4: $\kappa = \lambda$ a limit and $\psi = \bigwedge\!\!\!\bigwedge_{i=0}^{\omega} \varphi_i, \varphi_i \in \Sigma^0_{\vec{\lambda}^{\lambda}_n} \cup \Pi^0_{\vec{\lambda}^{\lambda}_n}$

The marker ξ^{ψ} ensures:

- $\tau\widehat{\ }\langle y\rangle \in T \iff y = 0$
- For every n $\hat{\psi}(n)$ is requested over $\tau\widehat{\ }\langle 0\rangle$.

Case 5: $\kappa = 0$

Let R be a computable predicate effectively recovered from the definition of ψ such that $R = 1 \iff T \models \psi$ and otherwise $R = 0$. The marker ξ^{ψ} ensures:

- $\tau\widehat{\ }\langle y\rangle \in T$ if and only if $R{\downarrow}_y = 1$ and for no $y' < y$, $R{\downarrow}_{y'} = 1$.
- ξ^{ψ} makes no requests.

4.5.2 $\mathfrak{a}_{\langle e\rangle,}$ Module

Our aim with the $\mathfrak{a}_{\langle e\rangle,}$ module is particularly simple. If $\sigma \in \mathfrak{a}_{\langle e\rangle,\infty}$ and $\Phi_e(\emptyset; |\sigma| + 1)\uparrow$ or $\Phi_e(\emptyset; |\sigma| + 1)\downarrow \neq 0$ then $\sigma\widehat{\ }\langle 0\rangle \in T$ and $y \neq 0 \implies \sigma\widehat{\ }\langle y\rangle \notin T$. Otherwise we want to make sure that for some $y \neq 0$, $\sigma\widehat{\ }\langle y\rangle \in T$ and no other extension of σ is allowed to extend to an infinite path.

Rule 4.5.4. If $\mathfrak{a}_{\langle e\rangle,}$ is assigned to σ at stage s then at any stage $t > s$ with $\sigma \in \mathfrak{a}_{\langle e\rangle,t}$ the marker $\mathfrak{a}_{\langle e\rangle,}$ takes one of several actions depending on the value of t. In describing these actions we set $t_n = \ulcorner \sigma\widehat{\ }\langle n\rangle \urcorner$ and $x = |\sigma| + 1$.

Case 1: $t \neq t_n, n \in \omega$

Do nothing.

Case 2: $t = t_0$

Allow $\sigma\widehat{\ }\langle 0\rangle$ to enter T.

Case 3: $t = t_{n+1} \wedge \neg\Phi_e(\emptyset; x)\downarrow_{t_n} = 0$

Do nothing.

Case 4: $t = t_{n+1} \wedge \Phi_e(\emptyset; x)\downarrow_{t_n} = 0$

Allow $\sigma\widehat{\ }\langle n + 1\rangle$ to enter T and request $[\![0 = 1]\!]$ over $\sigma\widehat{\ }\langle 0\rangle$

4.5.3 $\mathfrak{r}^j_{\langle e,i\rangle}$ Module

Given that rule 4.4.4 completely characterized the behavior of the markers of the form $\mathfrak{r}^*_{\langle e,i\rangle}$ it only remains to explain how the module for $\mathfrak{r}^j_{\langle e,i\rangle}$ behaves during the construction. Surprisingly, even while $\mathfrak{r}^j_{\langle e,i\rangle}$ must answer the most complicated question in the construction this module is no more complex than the one for $\mathfrak{a}_{\langle e\rangle,}$.

Rule 4.5.5. Suppose $R_{e,i}$ acts at ρ during stage s assigning $\mathfrak{r}^j_{\langle e,i\rangle}$ to τ_j respectively with τ_j and $\bar{x}$ chosen so that $\Phi^2_i(\tau_j; \bar{x}) = j$. We let ψ_1 be the $\Sigma^0_{\alpha+1}$ sentence asserting that $\Phi^2_e(0^{(\alpha)}; \bar{x})\downarrow = 0$ and $\psi_0 = \neg\psi_1$ and simultaneously describe the behavior of $\mathfrak{r}^j_{\langle e,i\rangle}$ acting at τ_j for $j \in \{0, 1\}$.

- $\mathfrak{r}^j_{\langle e,i\rangle}$ allows $\tau_j\widehat{\ }\langle y\rangle$ into T where y is the least number such that $\ulcorner \tau_j\widehat{\ }\langle y\rangle \urcorner > s$ but denies $\tau_j\widehat{\ }\langle y'\rangle$ for $y' \neq y$.
- $\mathfrak{r}^j_{\langle e,i\rangle}$ requests that ξ^{ψ_j} cover $\sigma\widehat{\ }\langle y\rangle$

Note that the reason it was necessary to use $\tau_j{}^\frown\langle y\rangle$ rather than $\tau_j{}^\frown\langle 0\rangle$ in the above rule is that rule 4.4.2 only guarantees that $\boldsymbol{r}^{j}_{\langle e,i\rangle}$ is placed at a maximal element in $T[s]$. Thus it's possible that a finite number of immediate extensions of τ_j have already been denied entry in T by the time $\boldsymbol{r}^{P}_{\langle e,i\rangle}j$ takes over at τ_j. This isn't a worry for ξ^φ or $\boldsymbol{a}_{\langle e\rangle}$, as rule 4.3.4 ensures they are always placed at a virgin node on the tree.

4.6 Verifying The Construction

An important assumption at work in the description of the basic modules above is the idea that we can 'cancel' an immediate extension $\sigma^\frown\langle y\rangle$ of a node σ either by directly blocking them from entering the tree or by enumerating a request that some ξ^φ cover $\sigma^\frown\langle y\rangle$ where φ is false. We then capture the intuitively uncanceled extensions of σ with the following notation.

Definition 4.6.1. Given σ fix $\boldsymbol{m}$ so $\sigma \in \boldsymbol{m}_\infty\varphi$ and if $\boldsymbol{m} \neq \boldsymbol{r}^*_{\langle e,i\rangle}$ define:

$$\Upsilon_\sigma = \left\{\sigma^\frown\langle y\rangle \in T \middle| \left(\nexists \xi^\psi\right)\left(\left(\xi^\psi, \sigma^\frown\langle y\rangle\right) \in \mathcal{Q}_\infty \wedge \neg\psi\right)\right\} \tag{4.6.1.1}$$

Otherwise if $\sigma \in \boldsymbol{r}^*_{\langle e,i\rangle,\infty}$ set

$$\Upsilon_\sigma = \{\tau_j | \sigma \subset \tau_j \wedge \psi_j\} \text{ where } \psi_j \text{ is as in rule 4.5.5} \tag{4.6.1.2}$$

Ultimately we want to use this definition to inductively prove the existence of a unique true path through T satisfying all the requirements but before that can happen we need to verify that false sentences φ genuinely do prevent infinite paths.

Lemma 4.6.2. *Given* $\sigma \in \xi^\psi_\infty$ *then*

$$\neg\psi \implies \Upsilon_\sigma = \emptyset \implies (\nexists f \in [T])\,(\sigma \subset f) \tag{4.6.2.1}$$

$$\psi \implies |\Upsilon_\sigma| = 1 \tag{4.6.2.2}$$

Proof. Obviously if $\varphi \in \Sigma^0_0 \cup \Pi^0_0$ then by rule 5 of rule 4.5.3 $|\Upsilon_\sigma| = 1$ if $T \models \varphi$ and $\Upsilon_\sigma = \emptyset$ otherwise. Therefore we suppose the lemma holds for all $\varphi \in \Sigma^0_\beta(T) \cup \Pi^0_\beta(T)$ for $\beta <_\mathcal{O} \kappa$ and show it must also hold for $\varphi \in \Sigma^0_\kappa(T) \cup \Pi^0_\kappa(T)$.

Case 1: $\kappa = \alpha +_\mathcal{O} 1 \wedge \psi = \exists x \hat{\psi}(x)$

Let $\hat{x} = \min\left\{x \middle| T \models \hat{\psi}(x)\right\}$. Note that if $\neg\psi$ then $\hat{x}$ is undefined and hence for every n $\neg\varphi'_n$ where φ'_n is as in rule 1. Hence if $\neg\psi$ then $\Upsilon_\sigma = \emptyset$. By lemma 4.4.7 since ξ^ψ is never injured at σ there will be some node σ' along every infinite path through T extending $\sigma^\frown\langle n\rangle$ with the marker $\xi^{\varphi'_n}$ assigned to it which, by our inductive assumption, shows equation (4.6.2.1) holds. On the other hand if $\neg\psi$ then $\hat{x}$ is the unique value such that $\neg\varphi'_{\hat{x}}$. Hence $|\Upsilon_\sigma| = 1$

Case 2: $\kappa = \alpha +_{\mathcal{O}} 1 \wedge \psi = \forall x\hat{\psi}(x)$

Since rule 2 lets only $\sigma^\frown\langle 0\rangle$ into T by a similar argument $\Upsilon_\sigma \neq \emptyset$ iff $|\Upsilon_\sigma| = 1$ iff $\neg\varphi$ for every formula φ requested over $\sigma^\frown\langle 0\rangle$. However, the formulas requested over $\sigma^\frown\langle 0\rangle$ were just the formulas $\hat{\psi}(n)$ for $n \in \omega$ which all hold if and only if $\neg\psi$.

Case 3: $\kappa = \lambda$ a limit

Same argument as the existential case if ψ is an infinite disjunction and same as the universal case if ψ is an infinite conjunction.

□

Lemma 4.6.3. *For all $\sigma \in \omega^{<\omega}$, $|\Upsilon_\sigma| \leq 1$.*

Proof. We already established this if $\sigma \in \xi^\varphi_{infty}$ for some ξ^φ. By lemma 4.4.8 the following three cases exhaust the alternatives.

Case 1: $\sigma \in \boldsymbol{\alpha}_{\langle e\rangle,\infty}$

If $\boldsymbol{\alpha}_{\langle e\rangle,}$ only allows $\sigma^\frown\langle 0\rangle$ to enter T the conclusion is immediate. But by rule 4.5.4 $\boldsymbol{\alpha}_{\langle e\rangle,}$ will allow $\sigma^\frown\langle y\rangle \in T$ for at most one y, requesting $[\![0 = 1]\!]$ over $\sigma^\frown\langle 0\rangle$ if it does.

Case 2: $\sigma \in \boldsymbol{\tau}^j_{\langle e,i\rangle,\infty}$

This is immediate as $\boldsymbol{\tau}^j_{\langle e,i\rangle}$ is always assigned to a maximal node in the current tree and only allows a single immediate successor of that node to enter.

Case 3: $\sigma \in \boldsymbol{\tau}^*_{\langle e,i\rangle,\infty}$

The only situation of possible concern is when σ is extended by both τ_0 and τ_1 but since ψ_0 is the negation of ψ_1 they can't both be true.

□

Lemma 4.6.4. *Suppose that $f \in [T]$ and that $f \supset \rho \implies f \supset \Upsilon_\rho$ then the requirement $R_{e,i}$ is satisfied.*

Proof. If there is no $\rho \subset f$ that was acted on by $R_{e,i}$ and not later injured then $R_{e,i}$ is satisfied by lemma 4.4.5. Therefore suppose there is such a ρ. Since ρ is extended by both sides τ_0, τ_1 of the corresponding Φ^2_i split definition 4.6.1 tells us $\Upsilon_\rho = \{\tau_j | \psi_j\}$ where ψ_j is as defined in rule 4.5.5. But ψ_1 holds if and only if $\Phi^2_e(0^{(\alpha)})$ agrees with τ_0 at a location τ_0, τ_1 disagree. So surely if ψ_1 is satisfied then, as the two sentences are negations of each other, $f \supset \tau_1$ and the requirement $R_{e,i}$ is satisfied. Similarly if ψ_0 holds then $f \supset \tau_0$ and the requirement is satisfied as well. □

Lemma 4.6.5. *Suppose that $f \in [T]$ and that $f \supset \rho \implies f \supset \Upsilon_\rho$ then the requirement N_e is satisfied.*

Proof. By rule 4.3.5 the construction starts with a request for $\boldsymbol{\alpha}_{\langle e\rangle,}$ to cover $\emptyset$ and since this request wasn't made by a marker lemma 4.4.7 tells us that $\boldsymbol{\alpha}_{\langle e\rangle,\infty}$ contains a T-cover of $\emptyset$. Hence there is some $\rho \subset f$ with $\rho \in \boldsymbol{\alpha}_{\langle e\rangle,\infty}$. Now rule 4.5.4 tells us that if we never observe $\Phi_e(\emptyset; |\rho|+1)\downarrow = 0$ then $\rho^\frown\langle 0\rangle$ is the only immediate extension of ρ on T and $\boldsymbol{\alpha}_{\langle e\rangle,}$ enumerates no requests over $\rho^\frown\langle 0\rangle$ so $\Upsilon_\rho = \rho^\frown\langle 0\rangle$. Thus in this case $\neg\Phi_e(\emptyset; |\rho|+1)\downarrow = f(|\rho|+1)$. Alternatively if $\Phi_e(\emptyset; |\rho|+1)\downarrow = 0$ then at some stage $\boldsymbol{\alpha}_{\langle e\rangle,}$ notices this fact and allows $\rho^\frown\langle y\rangle \in T$ and enumerates a request over $\sigma^\frown\langle 0\rangle$ for a false sentence. In this case$\Upsilon_\sigma = y$ so $\Phi_e(\emptyset; |\rho|+1)\downarrow = 0 \neq f(|\rho|+1)$. Either way N_e is satisfied. □

Proof of theorem 4.1.1. Set $\sigma_0 = \emptyset$ and choose $\sigma_{n+1} \in \Upsilon_{\sigma_n}$. We claim that $f = \bigcup_{n\in\omega} \sigma_n$ satisfies the theorem. By lemma 4.6.5 and lemma 4.6.4 we see that f satisfies the requirements so it's left only to show that f is the unique path through a computable tree T. By construction T is computable and lemma 4.6.3 tells us that if $f' \in \omega^\omega, f' \neq f$ then there is some $\tau \subset f'$ so that either $\Upsilon_\tau = \emptyset$ or the unique member of Υ_τ is incompatible with f'. However, by definition of Υ_τ this can happen only in three ways. Either f extends some node not in T, f extends some ρ that was acted on by $R_{e,i}$ and never injured but fails to extend the τ_j whose associated sentence ψ_j is true, or extends some τ' with some $(\xi^\varphi, \tau') \in \mathcal{Q}_\infty$ where φ is false. By the module for $\boldsymbol{\tau}^j_{\langle e,i\rangle}$ the second case entails either the first or third case and by lemma 4.6.2 the third case entails the first. Hence if $f \in [T]$ is total then f is the unique such path.

We suppose, by way of induction, that for any $\sigma' \subseteq \sigma_n$ if $(\xi^{\varphi}, \sigma') \in \mathcal{Q}_\infty$ then φ is true. By our stipulation in rule 4.3.6 if σ_n satisfies the inductive hypothesis and $\sigma_n \in \xi^{\varphi}_{\infty}$ or $\sigma_n \in \boldsymbol{\mathscr{u}}_{\langle e\rangle,\infty}$ then so does σ_{n+1}. By lemma 4.4.8 the only other case to consider is where $\sigma_n \in \boldsymbol{\mathscr{r}}^*_{\langle e,i\rangle,\infty}$ but in this situation if $\sigma_{n+1} \in \Upsilon_{\sigma_n}$ then there are no requests $(\xi^{\varphi}, \sigma') \in \mathcal{Q}_\infty$ for $\sigma_n \subsetneq \sigma' \subseteq \sigma_{n+1}$ as σ_{n+1} only skips across nodes occupied by $\boldsymbol{\mathscr{r}}^*_{\langle e,i\rangle}$.

Now suppose, by way of contradiction, that n is the least such that $\Upsilon_{\sigma_n} = \emptyset$. By lemma 4.6.2 we can't have $\sigma_n \in \xi^{\varphi}_{\infty}$. If $\sigma_n \in \boldsymbol{\mathscr{u}}_{\langle e\rangle,\infty}$ then by inspection of rule 4.5.4 in either of the two possible cases $\Upsilon_{\sigma_n} \neq \emptyset$. If $\sigma_n \in \boldsymbol{\mathscr{r}}^*_{\langle e,i\rangle,\infty}$ then, as the corresponding sentences are defined so $\psi_0 = \neg\psi_1$ at least one must be true. Moreover, by the definition of σ_n it must be extended by both τ_0 and τ_1 so $\Upsilon_{\sigma_n} \neq \emptyset$. But if $\sigma_n \in \boldsymbol{\mathscr{r}}^j_{\langle e,i\rangle,\infty}$ then $n \neq 0$ and $\sigma_{n-1} \in \boldsymbol{\mathscr{r}}^*_{\langle e,i\rangle,\infty}$ so j must be such that ψ_j is true meaning that the single extension of σ_n allowed by rule 4.5.5 must be in Υ_{σ_n}. By lemma 4.4.8 this was an exhaustive list of the possibilities thus contradicting our supposition.

Given that $\sigma_n \notin \Upsilon_{\sigma_n}$ it follows that f is the unique total function through T satisfying all the requirements. The theorem follows from this by lemma 4.2.4.

□

Chapter 5

Nonuniform Moduli

5.1 Distance From Uniformity

Thanks to Slaman and Groszek's proof of theorem 2.4.1 we know that every degree with a modulus of computation has a uniform modulus of computation. As any uniform modulus of computation is Δ^1_1 we trivially know that any degree $\deg d$ with a modulus has a uniform modulus that is hyperarithmetic. Obviously we can construction hyperarithmetic degrees who lack any simple uniform modulus, e.g., no uniform modulus for $0^{(\alpha)}$ will be computable by any $\beta <_{\mathcal{O}} \alpha$. But while any uniform modulus of $0^{(\alpha)}$ is very complex in absolute terms corollary 2.3.6 tells us that $0^{(\alpha)}$ has a uniform self-modulus, i.e., $0^{(\alpha)}$ has a uniform modulus that's extremely simple relative to $0^{(\alpha)}$. In fact every example we've seen so far of a modulus has been a uniform modulus.

This naturally leads to the following question.

Question 5.1.1. *Is there an $\alpha < \omega_1^{ck}$ so that if f is a modulus for X then there is a $f' \leq_{\mathbf{T}} f^{(\alpha)}$ with f' a uniform modulus of X*

That is must every modulus for a set be 'close' to some uniform modulus for that set. Given our interest in self-moduli we may as well make the question even more specific and search for a self-modulus that is very non-uniform.

Question 5.1.2. *Is there an $\alpha < \omega_1^{ck}$ so that if f is a self-modulus then there is a $f' \leq_{\mathbf{T}} f^{(\alpha)}$ with f' a uniform modulus of f*

We show the existence of a non-uniform self-modulus and show that we can push the closest uniform modulus any integer number of jumps away giving a partial answer to these questions.

Theorem 5.1.3. *For each $n < \omega$ there is a a self-modulus f^{n+2} such that no $\hat{f} \leq_{\mathbf{T}} f^{(n)}$ is a uniform modulus for f.*

5.2 Preliminaries

5.2.1 Tools

Before we can even start thinking about the rest of the proof we need some means to avoid having a nearby uniform modulus. We've already seen how to avoid having any self-modulus at all by being Cohen generic but that result was too strong. We need to allow the possibility of a self-modulus while pushing away uniform moduli. In light of theorem 3.1.4 we can understand this as saying we don't want to be special, i.e., we aren't an isolated path on any Π^0_{n+2} tree. We accomplish this goal by trying to be sufficiently generic on some perfect tree.

Lemma 5.2.1. *If $g \in \omega^\omega$ is $\alpha+2$ generic on some perfect tree $T \subset \omega^{<\omega}$ then any $f \leq_{\mathbf{T}} g^{(\alpha)}$ is not a uniform modulus of g.*

Proof. Now suppose, by way of contradiction, that g is locally $\alpha + 2$ generic on T and that Φ_j witnesses $f = \Phi_i(g^{(\alpha)})$ is a uniform modulus of g. Now let $\varphi_1(h)$ be the $\Sigma^0_{\alpha+1}$ formula asserting that there is some string τ above $\Phi_i(h^{(\alpha)})$ witnessing that $\Phi_j(\tau)$ computes the wrong value. Formally,

$$\varphi_1(h) = \left(\exists \tau \gg \Phi_i(h^{(\alpha)})\right) (\exists x) \left(\Phi_j(\tau; x)\downarrow_{|\tau|} \neq h(x)\right)$$

As Φ_j witnesses that f is a uniform modulus of g we must have $g \not\models \varphi_1$. As g is $\alpha + 2$ generic on T there is an n_1 such that $g_{n_1} \Vdash_T \neg\varphi_1$.

Now let $\varphi_2(h) \in \Sigma^0_{\alpha+2}$ assert that $\Phi_j(h)$ is total. Since $g \models \varphi_2$ there is a $n_2 > n_1$ so that $g{\restriction}_{n_2} \Vdash_T \varphi_2$. By the perfectness of T, pick some $g' \supset g{\restriction}_{n_2}$ with $g' \in [T]$ and g' $\alpha + 2$ locally generic on T but $g \neq g'$. Thus for some $m > n_2$ $g'(m) \neq g(m)$. As $g' \supset g{\restriction}_{n_2}$ $g' \Vdash_T \varphi_2$ and $g' \Vdash_T \neg\varphi_1$. By the genericity of g' we know $f' = \Phi_i(g'^{(\alpha)})$ is total so pick some $h \in \omega^\omega$ with $h \gg f$ and $h \gg f'$. Since Ψ witnesses that f is a uniform modulus of g for some k $\Phi_j(h{\restriction}_k)\downarrow_k = g(m) \neq g'(m)$. Thus $g' \models \varphi_1$. Contradiction. □

We will also need to define a canonical uniform self-modulus for $0^{(n)}$ We do this by extending the definition of $\theta^1(x)$ from observation 5.2.3 to cover arbitrary n.

Definition 5.2.2.

- $\theta^0 x = 0$
- $\theta^1(x) = \max_{i \le x} \min \{t | \Phi_x(\emptyset; x)\downarrow_t\}$
- $\theta_s^{n+1}(x) = \theta^n(x) + \max_{i \le x} \min_s \{s \le t | \Phi_{0^{(\bar{n})}}(i; i)\downarrow_s\}$
- $\theta^{n+1}(x) = \lim_{s \to \infty} \theta_s^{n+1}(x)$

Using the same reasoning deployed in section 2.3 we observe several properties of these functions that will be useful later.

Observation 5.2.3.

1. *$\theta^n(x)$ is increasing in both n and x.*
2. *θ^m is computable in $0^{(m)}$*
3. *θ^n is as uniform self-modulus of $0^{(n)}$*

We also need to define some basic terminology to deal with constructing generic functions.

Definition 5.2.4.

- A string $\epsilon \in \omega^{<\omega} \cup \omega^\omega$ meets a set $S \subset \omega^{<\omega}$ (at x) if there is some $\tau \in S$ ($|\tau| = x$) with $\epsilon \supseteq \tau$.
- A string $\epsilon \in \omega^{<\omega} \cup \omega^\omega$ meets a function $\Gamma : \omega^{<\omega} \mapsto \omega^{<\omega}$ (at x) if $\epsilon \supset \Gamma\epsilon\restriction_x$ for some x.
- A string **densely meets** a function f (set S) if it meets f (S) at infinitely many x.

5.2.2 Strategy

With this result in hand the difficulty in proving the theorem now lies in making sure that f^{n+2} is actually a self-modulus while making it $n+2$ generic on some tree. We know that every set with a modulus has some uniform modulus so we will need to provide some non-uniform means for every 'small' g with $g \gg f^{n+2}$ to compute f^{n+2} and a uniform method for 'large' g to do so. By building f^{n+2} recursively in $0^{(n+2)}$ we guarantee that every g dominating θ^{n+2}, a uniform self-modulus for $0^{(n+2)}$, computes f^{n+2}. The trick is dealing with the small g.

The idea here is to ensure that the locations where g is 'small' let us recover information about f^{n+2}. More specifically it is to encode f^{n+2} into the locations where $f^{n+2}(x) < \theta^m(x)$, i.e., f^{n+2} is m-small, for every $m \leq n+2$. Thus if $g \gg f^{n+2}$ it must (non-uniformly) fall into one of 2 (well really $n+2$) categories. Either $g \gg^* \theta^{n+2}$ and $g \geq_{\mathbf{T}} 0^{(n+2)} \geq_{\mathbf{T}} f^{n+2}$ or there is a maximal $k < n+2$ with $g \gg^* \theta^{n+2}$. In the later case $g \gg^* \theta^k$ so $g \geq_{\mathbf{T}} 0^{(k)}$ and because we arrange $\theta^{k+1}(x)$ to admit a $0^{(k)}$ approximation from below g will be able to enumerate an infinite list of locations at which it is $k+1$-small, i.e., $g(x) < \theta^{k+1}(x)$. Because we code the information about f into these locations this will enable g to compute f^{n+2}.

The difficult part part of the proof is encoding information about f^{n+2} while still retaining sufficient genericity. Our approach is to build f^{n+2} so that it lies on a sequence of trees $T_{n+2} \subseteq T_{n+1} \subseteq \ldots \subseteq T_1$ each with the property that for any x there is at most one string $\sigma \in T_m$ of length $x+1$ with $\sigma(x) < \theta^m(x)$. We can think of this as coding information about the initial segments of f^{n+2} into the locations f^{n+2} is m-small using T_m as a codebook. If we could build $T_{m+1} \leq_{\mathbf{T}} 0^{(m)}$ then it would be relatively straightforward to recover f from $g \gg f$. As described above for some $k < n+2$ g would compute $0^{(k)}$ and list off infinitely many locations with $g(x) < \theta^{k+1}(x)$ so if our codebook was computable in $0^{(k)}$ then g could simply use that information to decode initial segments of f^{n+2} from the locations at which g (and hence f) is $k+1$-small. However, we can't hope to make $T_{m+1} \leq_{\mathbf{T}} 0^{(m)}$ since $T_{m+1} \geq_{\mathbf{T}} 0^{(m+1)}$ by virtue of the uniqueness property.

The above approach to computing f^{n+2} fell short only by a single jump so by the limit lemma if $g \gg f^{n+2}$ and $g(x) < \theta^{k+1}(x)$ then g can make a guess $f^{n+2}\restriction_x$ for every stage s in a manner that will eventually settle on the correct answer. Our basic strategy then is to ensure that if s is large enough for g to notice that $g(x) < \theta^{k+1}(x)$ then s is also large enough to give the correct answer for $f^{n+2}\restriction_x$. Since the function f^{n+2} needs to satisfy the same properties as f^{n+1} does plus some additional ones we will want to approach this inductively, i.e, build the function f^1 on T_1 and f^2 on $T_2 \subset T_1$. This isn't too difficult if we merely want to guarantee the uniqueness property but we must also make sure that f^{n+2} is $n+2$ generic on some perfect tree which we choose to make T_{n+2} as well.

Just as if we were building f^{n+2} to be Cohen generic, i.e., locally generic on $\omega^{<\omega}$, we will achieve genericity for f^{n+2} by building it using finite extensions which we try to extend to meet Σ^0_{n+2} sets. To do this in the controlled fashion necessary to satisfy our constraints while retaining the ability to use the work from f^{n+1} in f^{n+2} we define a function Γ^{n+2} which will dictate how to extend the the initial segments of f^{n+2} to meet the appropriate Σ^0_{n+2} sets. In doing so this function will incorporate the advice of Γ^k for $k < n+2$ so as to also meet Σ^0_{n+1} sets. Our basic strategy will be to define $\Gamma^{n+2}(\sigma)$ as the limit of $\Gamma^{n+2}_s(\sigma)$ and demand that $\Gamma^{n+2}_s(\sigma)$ always look $n+2$-large at any stage it changes it's value. Thus at the stage g observes that $g(x)$ is small the function $\Gamma^{n+2}_s(\sigma)$ has already achieved it's limit for the appropriate σ. The details, however, are considerable more complex.

5.3 Construction

We start with the definitions for f^m and T_m in terms of the function Γ^m to illuminate the choices we make in the construction of Γ^m.

Definition 5.3.1. Define $f^m = \cup_i f^m_i$ where $f^m_0 = \emptyset$ and $f^m_{i+1} = \Gamma^m(f^m_i)$.

Definition 5.3.2. $T_n \subset \omega^{<\omega}$ is the smallest set containing $\emptyset$ such that

- If $\sigma \in T_n$ then any σ' extending σ with $\sigma' \setminus \sigma \gg \theta^n$ is in T_n as well.
- If $\tau \in T_n$ then $\Gamma^n(\tau) \in T_n$.

Note that T_m is not defined to be closed under initial segments. The tree formed by closing T_m under initial segments would work just as well but by defining T_m we keep the relation between T_m and Γ^m simpler and avoid book keeping problems later.

Now we are prepared to construct the function $\Gamma^n = \lim_{s\to\infty} \Gamma^{n+1}_s(\sigma)$. The basic approach is to let $\Gamma^{n+1}_s(\sigma)$ search for s steps using $0^{(n)}$ as an oracle trying to find an extension of σ that looks $(n+1)$-large at stage s and that meets the next $\Sigma^0_1\big(0^{(n+1)}\big)$ set. However, there are several complications which we will explain along the way.

5.3.1 Building Γ^n

Define $\Gamma^0(\sigma) = \sigma$ for every σ. To define $\Gamma^{n+1}(\sigma)$ assume that for all $\bar{k} \leq n$ $\Gamma^{\bar{k}}$ has been defined to be a total function and construct Γ^{n+1}. We do so by offering a guess, $\Gamma^{n+1}_s(\sigma)$, at $\Gamma^{n+1}\sigma$ for every stage s and then setting $\Gamma^{n+1}(\sigma) = \lim_{s\to\infty} \Gamma^{n+1}_s(\sigma)$.

Construction 5.3.3. If $s \leq |\sigma|$ set $\Gamma^{\bar{m}}_s(\sigma) = \sigma$. Otherwise assume that $\Gamma^{\bar{m}}_t$ is defined on all arguments for $t \leq s$ and $\Gamma^{\bar{m}}_{s+1}(\tau)$ is defined for all τ with $\ulcorner\tau\urcorner < \ulcorner\sigma\urcorner$ and compute $\Gamma^{\bar{m}}_{s+1}(\sigma)$ by executing the following steps in order unless told to stop.

STEP 1: For each τ with $\ulcorner\tau\urcorner < \ulcorner\sigma\urcorner$ examine $\Gamma^{n+1}_s(\tau)$ and $\Gamma^{n+1}_{s+1}(\tau)$. If for any such τ we have $\Gamma^{n+1}_s(\tau) \neq \Gamma^{n+1}_{s+1}(\tau)$ then set $\Gamma^{n+1}_{s+1}(\sigma) = \sigma$ otherwise go to the next step.

STEP 2: If $\Gamma^{n+1}_s(\sigma) \subset \Gamma^{n+1}_{s+1}(\tau)$ for some τ with $\ulcorner\tau\urcorner < \ulcorner\sigma\urcorner$ leave $\Gamma^{n+1}_{s+1}(\sigma)$ equal to $\Gamma^{n+1}_s(\sigma)$ and stop. Otherwise continue to the next step.

> In order to be appropriately generic we must guarantee that if σ can be extended on T_{n+1} to meet the next Σ^0_{n+1} set then $\Gamma^{n+1}(\sigma)$ meets this set. However, since we require $\Gamma^{n+1}(\sigma)$ to look big when enumerated there is the risk that by the time $\Gamma^{n+1}(\sigma)$ realizes it can meet a certain set it is no longer able to do so. We therefore adopt the solution of allowing $\Gamma^{n+1}_{s_1+1}(\sigma)$ to swallow $\Gamma^{n+1}_{s_1}(\sigma')$ where $\sigma' \supseteq \sigma$ even if this would otherwise not be allowed. Although this weakens our assumption that we only select extensions that look big at the time no problem will result since we recover the same function if g believes that the true path is given by $\Gamma^{n+1}_{s_0}(\sigma) = \Gamma^{n+1}_{s_1}(\sigma)$.

STEP 3: If any τ with $\ulcorner\tau\urcorner < \ulcorner\sigma\urcorner$ $\Gamma^{n+1}_s(\tau) \neq \Gamma^{n+1}_{s+1}(\tau)$ then reset $\Gamma^{n+1}_s(\sigma)$ to σ.

> We always want to make sure the earlier strings reach their limits first so we can assume that we have the correct value for Γ^τ when computing Γ^σ with $\ulcorner\sigma\urcorner > \ulcorner\tau\urcorner$.

STEP 4: If $\Gamma^{n+1}_s(\sigma)(|\sigma|) < \theta^{n+1}(|\sigma|)[s+1]$ then set $\Gamma^{n+1}_{s+1}(\sigma) = \sigma$ and stop. Otherwise continue to the next step.

STEP 5: If for some τ with $\ulcorner\tau\urcorner < \sigma$ and $x \geq |\sigma|$ with $\Gamma^{n+1}_{s+1}(\tau)(x), \Gamma^{n+1}_s(\sigma)(x)$ are both defined and less than $\theta^{n+1}(x)[s+1]$ and $\Gamma^{n+1}_{s+1}(\tau)\restriction_x \mid \Gamma^{n+1}_s(\sigma)\restriction_x$ then set $\Gamma^{n+1}_{s+1}(\sigma) = \sigma$. Otherwise go on to the next step.

> Guarantees that at most one path on the tree is small at any location.

STEP 6: Look for the first $\tau \supset \sigma$ with $\ulcorner\tau\urcorner \leq s+1$ such that

1. There is a sequence of strings σ_k, ν_k

$$\tau = \sigma_n \supset \nu_{n-1} \supset \sigma_{n-1} \supset \nu_{n-2} \ldots \supset \nu_1 \supset \sigma_1 \supset \sigma$$

   where ν_k is either $\Gamma^n(\sigma_k)$ or $\Gamma^{n+1}_s(\sigma_k)$.

   > The segments ν_k allow the inclusion of segments from Γ^n and swallowed segments from earlier stages.

2. If $\nu_{k+1} = \Gamma^n(\sigma_k)$ and $\nu_{k+2} = \Gamma^n(\sigma_{k+1})$ then $\sigma_{k+1} \neq \emptyset$.

   > We never want to include two applications of Γ^n adjacent to each other lest we create a long stretch below θ^{n+1} that doesn't include any locations x where $\Gamma^{n+1}(\sigma)(x) \not> \theta^{(}_s x)$ when enumerated.

3. If $\tau(x) < \theta^n(x)$ and $x \geq |\sigma|$ then $x \in \operatorname{dom} \nu_k \setminus \sigma_k$ for some k with $\nu_k = \Gamma^n(\sigma_k)$
4. If $\tau(x) < \theta^{n+1}(x)[s+1]$ then $x \in \operatorname{dom} \sigma$ or $x \in \operatorname{dom} \nu_k \setminus \sigma_k$ for some k

   > So long as x doesn't fall in the exceptions this will let us recover $\Gamma^{n+1}(\sigma)$ from any stage t at which $\Gamma^{n+1}(\sigma)(x) < \theta^{(}_t x)$.

5. For some $i \leq s+1$, τ meets $\mathcal{W}^{0^{(n)}}_i[s+1]$ but σ does not and there is no τ' with this property for some $j < i$.

   > We need to meet the Σ^0_{n+1} sets to make ourselves locally generic.

If such a τ is found set $\Gamma^{n+1}_{s+1}(\sigma) = \tau$ otherwise leave $\Gamma^{n+1}_{s+1}(\sigma) = \Gamma^{n+1}_s(\sigma)$.

5.3.2 Construction Properties

We begin by verifying some basic properties of our construction above.

Lemma 5.3.4. *Suppose $f \in \omega^\omega$ densely meets Γ^m. Then for every $k < m$ f densely meets Γ^k*

We prove this simultaneously we the next lemma.

Lemma 5.3.5. *$\Gamma^m(\sigma)$ is a total function computable in $0^{(m)}$ with $\Gamma^m(\sigma) \neq \sigma$.*

Proof. Inductively assume that lemma 5.3.4 and lemma 5.3.5 hold for all $k \leq n$. To show they hold for $n+1$ suppose, by way of contradiction, that Γ^{n+1} is not total and let σ be the string with the least code so that $\lim_{s\to\infty} \Gamma^{n+1}_s(\sigma)$ diverges. Hence we can pick t be a stage large enough that for any τ with $\ulcorner\tau\urcorner < \ulcorner\sigma\urcorner$ $\Gamma^{n+1}_s(\tau)$ has permanently achieved it's limit and set $r = \max_{\{\tau|\ulcorner\tau\urcorner<\ulcorner\sigma\urcorner\}} |\Gamma^{n+1}_t(\tau)|$. Pick $t' > t$ such that $\theta^{n+1}(x)[t']$ has also permanently achieved it's limit. Now if $s > t'$ the only step in the construction able to change the value of $\Gamma^{n+1}_s(\sigma)$ is step 6 but that doesn't happen unless it can extend to meet a Σ^0_{n+1} set with a smaller index. Since the integers are well-ordered this can only happen finitely many times. Contradiction.

Hence Γ^{n+1} is total and inspection of the construction shows that $\Gamma^{n+1}_s \leq_{\mathbf{T}} \Gamma^n \oplus 0^{(n)}$ thus by the limit lemma [19] and the inductive hypothesis we know $\Gamma^{n+1} \leq_{\mathbf{T}} 0^{(n+1)}$. Furthermore we can conclude $\Gamma^n(\sigma) \neq \sigma$ since there is always some Σ^0_n set that σ could be extended to meet. This leaves only lemma 5.3.4 to verify. To see this observe that since $\Gamma^n \leq_{\mathbf{T}} 0^{(n)}$ there is an Σ^0_n set consisting only of those strings meeting Γ^n at least k times. As step 6 always allows Γ^{n+1} to include applications of Γ^n if f densely meets Γ^{n+1} it also densely meets Γ^n which by induction establishes lemma 5.3.4 . □

Corollary 5.3.6. *$T_m \leq_{\mathbf{T}} 0^{(m)}$ and $f^m \leq_{\mathbf{T}} 0^{(m)}$*

Proof. Follows from the definition and lemma 5.3.5. □

Lemma 5.3.7. *If $m < n$ then $T_m \supset T_n$*

Proof. Suppose this holds for n we show it also holds for $n+1$. Note that it is sufficient to show that $T_n \supset T_{n+1}$. If this doesn't hold by the well-foundness of $\subset$ on $\omega^{<\omega}$ pick some $\tau \in T_{n+1}, \tau \notin T_m$ but if $\sigma \subsetneq \tau$ then $\sigma \in T_{n+1} \implies \sigma \in T_n$. If there is some such σ so that $\tau \setminus \sigma \gg \theta^{n+1}$ then as observation 5.2.3 tells us that $\theta^{n+1} \gg \theta^n$ so by definition $\tau \in T_n$. This would be a contradiction so by our supposition we can conclude that there is some $\sigma \in T_{n+1}$ with $\tau = \Gamma^{n+1}(\sigma)$. Yet by parts 1 and 3 of step 6 $\tau = \sigma_n \supset \nu_{n-1} \supset \ldots \supset \sigma$ where each $\sigma_k \setminus \nu_{k-1} \gg \theta^n$ and $\nu_k = \Gamma^n(\sigma_n)$ hence $\tau \in T_n$. Hence $T_n \supset T_{n+1}$ □

5.4 Verifying The Theorem

5.4.1 Recovering f

Lemma 5.4.1. *If $x > |\sigma|, |\tau|$ and $\Gamma^n_s(\tau)(x), \Gamma^n_s(\sigma)(x) < \theta^n_s(x)$ then $\Gamma^n_s(\tau)\restriction_x = \Gamma^n_s(\sigma)\restriction_x$*

Proof. Suppose this fails and let $\Gamma^n_s(\sigma)(x), \Gamma^n_s(\tau)(x)$ witness with $\ulcorner\tau\urcorner < \ulcorner\sigma\urcorner$. By step 5 of the construction this can only occur if $\Gamma^n_{s-1}(\sigma) \neq \Gamma^n_s(\sigma)$. However, by □

Lemma 5.4.2. *If $x > |\sigma|, |\tau|$ and $\Gamma^n(\tau)(x), \Gamma^n(\sigma)(x) < \theta^n(x)$ then $\Gamma^n(\tau)\restriction_x = \Gamma^n(\sigma)\restriction_x$*

Proof. Suppose this fails and let $\Gamma^n(\sigma)(x), \Gamma^n(\tau)(x)$ witness. Let t be large enough that $\Gamma^n(\sigma)[t], \Gamma^n(\tau)[t]$ have achieved their limits and $\Gamma^n(\sigma)(x)[t], \Gamma^n(\tau)(x)[t] < \theta^n(x)[t]$. Since $\Gamma^n_t(\tau)\restriction_x \neq \Gamma^n_t(\sigma)\restriction_x$ the construction would have continued past step 2 and step 5 of the construction would have reset $\Gamma^n(\sigma)$ or $\Gamma^n(\tau)(x)$ at stage t. ⇒⇐ □

We now observe that f is a self-modulus.

Notation 5.4.3. Define$L^n(x) = 1 + \sup_{|\sigma|<x} |\Gamma^n(\sigma)|$ for $n > 0$ and $\Gamma^0(x) = 1$.

Lemma 5.4.4. *Suppose for every $m \leq n$ there are only finitely many x satisfying*

$$\left(\forall x' \in [x, L^m(x)]\right)\left(g(x') < \theta^{m+1}(x')\right) \tag{5.4.4.1}$$

then $g \geq_{\mathbf{T}} 0^{(n+1)}$.

Proof. Suppose, by way of induction, this holds for every $n' < n$ and let g satisfy the hypothesis of the lemma for n. By our inductive assumption $g \geq_{\mathbf{T}} 0^{(n)}$. Thus by lemma 5.3.5 L^n $(\leq_{\mathbf{T}})$ g. Let x_0 be large enough that no $x > x_0$ satisfies (5.4.4.1) for $m = n$. Then for $x > x_0$, $\min_{x' \in [x, L^m(x)]} g(x') \geq \min_{x' \in [x, L^m(x)]} \theta^{n+1}(x') \geq \theta^{n+1}(x)$. Hence g computes some function $g' \gg \theta^{n+1}$ which by observation 5.2.3 is sufficient to show that $g \geq_{\mathbf{T}} 0^{(n+1)}$. □

Lemma 5.4.5. *If $g \gg f^{n+1}$ then $g \geq_{\mathbf{T}} f^{n+1}$.*

Proof. If there are only finitely many x satisfying (5.4.4.1) for $m \leq n$ then $g \geq_{\mathbf{T}} 0^{(n+1)} \geq_{\mathbf{T}} f$ so we suppose that $m \leq n$ is the smallest integer for which the hypothesis fails. To compute $f^{n+1}\restriction_{x_0}$ we computably search in $0^{(m)} \leq_{\mathbf{T}} g$ for some $x > x_0$ and some s so that

$$\left(\forall x'\right)\left(x \leq x' \leq L^m(x) \implies g(x') < \theta^{m+1}_s(x')\right)$$

and then pick the string σ with minimal code so that $x \geq |\sigma|$ such that $\Gamma^{m+1}_s(\sigma)(x) < \theta^{m+1}_s(x)$. We claim that $f\restriction_x = \Gamma^{m+1}_s(\sigma)$.

Now by part 2 of step 6 and the definition of f^{n+1} there is some σ' so that $|\sigma'| < x$ and $|\Gamma^{m+1}(\sigma')| \geq L^m(x)$ where $\Gamma^{m+1}(\sigma') \subset f^{n+1}$ and $f(x) > \theta^{m+1}_t(x')$ at a stage where $\Gamma^{m+1}_t(\sigma') = \Gamma^{m+1}(\sigma')$. This holds since, by construction, the only way for f to dip below

θ^{m+1} is by inclusion of a segment given by Γ^{m+1} and L^m is too long to be contained in an inclusion of a single segment Γ^m and no two such segments occur consecutively. Given that $\theta^n_s(x') > \theta^n_t(x')$ for all x' between x and $L^m(x)$ we know that $s > t$. Therefore for all τ with $\ulcorner\tau\urcorner < \ulcorner\sigma'\urcorner$ we know by step 3 that $\Gamma^{m+1}_s(\tau) = \Gamma^{m+1}(\tau)$ since otherwise they would have been reset. Hence by lemma 5.4.2 we know that $\Gamma^{m+1}(\sigma) \subseteq \Gamma^{m+1}(\sigma')$. Therefore $\Gamma^{m+1}_s(\sigma) \subset f$. Thus no matter which case we consider $g \geq_{\mathbf{T}} f$. □

5.4.2 Genericity

The construction of Γ^m does everything possible to meet the least Σ^0_m set still unmeet so for any Σ^0_m set if $\nu \in T_m$ meets Γ^m enough times then ν should meet this set or witness that it is impossible to meet. We can make this idea precise as follows.

Lemma 5.4.6. *Suppose τ is in T_{n+1} and τ meets Γ^{n+1} at least j times then either $\Gamma^{n+1}(\tau)$ meets $\mathcal{W}^{0^{(n+1)}}_j[|\tau|]$ or no $\tau' \supset \tau$ with $\tau' \in T_{n+1}$ meeting $\mathcal{W}^{0^{(n)}}_j$.*

Proof. Inductively assume that the lemma holds for every $i < j$ and suppose that τ doesn't meet $\mathcal{W}^{0^{(n)}}_j$. By step 6 in the construction either for some $i < j$ $\Gamma^{n+1}(\tau)$ meets $\mathcal{W}^{0^{(n)}}_{|\tau'|,i}$ but τ does not or for there is a stage t so that every later stage $s > t$ the construction no $\sigma \supset \tau$ meeting $\mathcal{W}^{0^{(n)}}_i[s]$ satisfies the conditions in step 6 at stage s. By our inductive hypothesis the first possibility is ruled out.

But if there were a $\sigma \in T_{n+1}$ with $\sigma \supset \Gamma^{n+1}(\tau)$ meeting $\mathcal{W}^{0^{(n+1)}}_i$ then σ meets the conditions in step 6 for τ so $\Gamma^{n+1}(\tau)$ must meet $\mathcal{W}^{0^{(n)}}_j$. Contradiction. □

If we were dealing with Cohen genericity (i.e. T_m was computable) this lemma would be enough to show that any real in $[T_m]$ densely meeting Γ^m was m generic. However, in general an $f \in [T]$ which either meets or strongly avoids on T (no extension on T of some initial of f meets) every Σ^0_m set doesn't have to be very T generic at all. This remains true even though our notion of being k locally generic on T only requires we force every Σ^0_k formula or it's negation using the local forcing relation not the $\Sigma^0_k(T)$ formulas. Thus we need some stronger principle to achieve genericity. The intuition that we deploy is that since we form $T_{n+1} \subset T_k$ as the result of trying to meet Σ^0_{n+1} sets on T_k the local forcing relation on T_{n+1} will be substantially similar to the local forcing relation on T_k. To transform this intuition into a solid proof we introduce a relation $\Vdash^s_k$ which we will demonstrate is equivalent to the standard forcing relation for sufficiently generic reals on

T_k. Note that for the rest of this chapter we adopt the common notational convenience of identifying (syntactically) Π^0_k sentences with the negation of sentences in Σ^0_k for the purpose of working with the forcing relation.

Definition 5.4.7. We define the relation $\sigma \Vdash^s_m \varphi$ for $\sigma \in T_m$ and $\varphi \in \Sigma^0_m \cup \Pi^0_m$ by induction on m.

1. If $\varphi \in \Sigma^0_0 = \Pi^0_0$ then $\sigma \Vdash^s_0 \varphi \iff \sigma \models \varphi$.
2. If $\varphi \in \Sigma^0_n \cup \Pi^0_n$ for $n < m$ then $\sigma \Vdash^s_m \varphi \iff \sigma \Vdash^s_n \varphi$.
3. If φ properly in Σ^0_{m+1} and $\varphi = \exists n \psi(n)$ then
$$\sigma \Vdash^s_{m+1} \varphi \iff (\exists n)\, \sigma \Vdash^s_m \psi(n) \tag{5.4.7.1}$$
4. If φ properly in Π^0_{m+1}, $\varphi = \neg\psi$ and j is a canonically given index for $\mathcal{W}^{0^{(m)}}_j = \{\tau | \tau \Vdash^s_{m+1} \psi\}$ then
$$\sigma \Vdash^s_{m+1} \neg\varphi \iff (\nexists \sigma' \supset \sigma)\left(\sigma' \in T_{m+1} \wedge \sigma' \Vdash^s_{m+1} \varphi\right) \wedge$$
$$\sigma \text{ meets } \Gamma^{m+1} \text{ at least } j \text{ times.}$$

We adopt the same notations for $\Vdash^s_m$ as we do for genuine forcing. In particular if $f \in \omega^\omega$ then $f \Vdash^s_m \varphi$ if and only if for some n $f\restriction_n \Vdash^s_m \varphi$.

Lemma 5.4.8. *Not both $\sigma \Vdash^s_m \varphi$ and $\sigma \Vdash^s_m \neg\varphi$.*

Proof. By part 2 of the definition of $\Vdash^s_m$ it's safe to assume that φ is properly in $\Sigma^0_m \cup \Pi^0_m$. But if either $\sigma \Vdash^s_m \varphi$ or $\sigma \Vdash^s_m \neg\varphi$ hold then $\sigma \in T_m$. Hence $\sigma \Vdash^s_m \varphi$ is a direct contradiction to the requirement for $\sigma \Vdash^s_m \neg\varphi$. □

Lemma 5.4.9. *The relation $\sigma \Vdash^s_m \varphi$ for $\varphi \in \Sigma^0_m \cup \Pi^0_m$ is uniformly computable in $0^{(m)}$ and hence well defined.*

Proof. The only step that does not follow by a trivial induction is dealing with Π^0_{m+1} sentences but this follow from lemma 5.4.6 since if $\varphi \in \Sigma^0_{m+1}$ and σ meets Γ^{m+1} at least j times then $\sigma \Vdash^s_{m+1} \neg\varphi \iff \sigma \neg \Vdash^s_{m+1} \varphi$. □

Lemma 5.4.10. *If $f \in [T_m]$ densely meets Γ^m then for all $\varphi \in \Sigma^0_m$ either $f \Vdash^s_m \varphi$ or $f \Vdash^s_m \neg\varphi$.*

Proof. This is trivial for $m = 0$ so we suppose the lemma holds for all $m \leq n$ and show it holds for $m = n + 1$ as well.

If f is as in the statement of the lemma for $m = n+1$ then by lemma 5.3.4 f must densely meet Γ^n. Therefore, as lemma 5.3.7 tells us $T_{n+1} \subset T_n$ if $\psi \in \Sigma^0_n$ the inductive hypothesis implies either $f \Vdash^s_{n+1} \psi$ or $f \Vdash^s_{n+1} \neg\psi$. Now let $\varphi = (\exists k)\, \psi(k)$ for some $\psi \in \Pi^0_n$ and $S = \{\sigma \in T_n | (\exists k)\, (\sigma \Vdash^s_n \psi(k))\}$. By lemma 5.4.9 the relation $\Vdash^s_n$ is computable in $0^{(n)}$ as is T_n hence S is a Σ^0_{n+1} set. If f meets S then for some k $f \Vdash^s_n \psi(k)$ and by definition $f \Vdash^s_{n+1} \varphi$ so suppose not. Let j be the canonical index such that $\mathcal{W}^{0^{(m)}}_j = S$ and an l large enough that $f\restriction_l$ meets Γ^{n+1} at least $j+1$ times. By lemma 5.4.6 it follows that no $\sigma \in T_{n+1}$ with $\sigma \supset f\restriction_l$ meets S. Thus $f \Vdash^s_{n+1} \neg\varphi$. □

Lemma 5.4.11. *If $f \in [T_m]$ densely meets Γ^m then*

1. *If f is m locally generic on T_m then for any $\varphi \in \Sigma^0_m \cup \Pi^0_m$, $f \Vdash^*_{T_m} \varphi \iff f \Vdash^s_m \varphi$.*

2. *If $f \in [T_{m+1}]$ and f is m locally generic on T_{m+1} then f is m locally generic on T_m*

3. *If $f \in [T_{m+1}]$ and f is m locally generic on T_m then f is m locally generic on T_{m+1}.*

Proof. We prove these claims via simultaneous induction on m.

1. Trivially this claim holds for $m = 0$ so we inductively assume the lemma holds for n and show it holds for $n + 1$. Now suppose that f is $n + 1$ locally generic on T_{n+1} then by part 2 applied to n we can infer that f is n locally generic on T_n so $f \Vdash^*_{T_n} \varphi \iff f \Vdash^s_n \varphi$ for $\varphi \in \Sigma^0_n \cup \Pi^0_n$. But for $\varphi \in \Sigma^0_n \cup \Pi^0_n$, $f \Vdash^s_n \varphi \iff f \Vdash^s_{n+1} \varphi$ hence

$$f \Vdash^*_{T_{n+1}} \varphi \iff f \models \varphi \iff f \Vdash^*_{T_n} \varphi \iff f \Vdash^s_n \varphi \iff f \Vdash^s_{n+1} \varphi \qquad (5.4.11.1)$$

Now assume that φ is properly in $\Sigma^0_{n+1} \cup \Pi^0_{n+1}$ and $f \Vdash^*_{T_{n+1}} \varphi$. To see that $f \Vdash^s_{n+1} \varphi$ consider theses cases:

Case 1: $\varphi = \exists x \psi(x)$

There is some l and x so that $f\restriction_l \Vdash^*_{T_{n+1}} \psi(x)$ and hence $f \Vdash^s_n \psi(x)$ and thus by definition $f \Vdash^s_{n+1} \varphi$.

Case 2: $\varphi = \neg\psi$

There is some l so that

$$(\forall \sigma \in T_{n+1})\left(\sigma \supset f\upharpoonright_l \implies \neg\sigma \Vdash^*_{T_{n+1}} \psi\right) \tag{5.4.11.2}$$

Hence by the previous case

$$(\forall \sigma \in T_{n+1})\left(\sigma \supset f\upharpoonright_l \implies \neg\sigma \Vdash^s_{n+1} \psi\right) \tag{5.4.11.3}$$

which by definition yields that $f \Vdash^s_{n+1} \varphi$.

But as f is $n+1$ locally generic on T_{n+1} for every formula $\varphi \in \Sigma^0_n \cup \Pi^0_n$ we have either $f \Vdash^*_{T_{n+1}} \varphi \wedge f \Vdash^s_{n+1} \varphi$ or $f \Vdash^*_{T_{n+1}} \neg\varphi \wedge f \Vdash^s_{n+1} \neg\varphi$. Since by 5.4.8 we can't have both $f \Vdash^s_{n+1} \varphi$ and $f \Vdash^s_{n+1} \neg\varphi$ the relations $\Vdash^*_{T_{n+1}}$ and $\Vdash^s_{n+1}$ must agree on f.

2. We assume that part 1 holds and prove this claim by induction on the complexity of the formulas being forced. In particular we note that $f \Vdash^*_{T_{m+1}} \phi \implies f \Vdash^*_{T_m} \phi$ trivially holds if $\phi \in \Sigma^0_0 \cup \Pi^0_0$ so we assume that it holds for $\phi \in \Sigma^0_n \cup \Pi^0_n$ and observe it also holds for ϕ properly in $\Sigma^0_{n+1} \cup \Pi^0_{n+1}$. To see this fix such a ϕ and suppose $f \Vdash^*_{T_{m+1}} \phi$ and then demonstrate $f \Vdash^*_{T_m} \phi$ by considering the following cases.

Case 1: $\phi = \exists x \psi(x)$

For some x, $f \Vdash^*_{T_{m+1}} \psi(x)$ and by our inductive hypothesis $f \Vdash^*_{T_m} \psi(x)$.

Case 2: $\phi = \neg\varphi$

For some l

$$(\nexists \sigma \supset f\upharpoonright_l)\left(\sigma \in T_{m+1} \wedge \sigma \Vdash^*_{T_{m+1}} \varphi\right)$$

Let j be a canonical $0^{(n+1)}$ index for $S = \left\{\sigma \in T_{n+1} \middle| \sigma \Vdash^s_{n+1} \varphi\right\}$ and pick l' large enough that $f\upharpoonright_{l'}$ has met Γ^{n+1} at least $j+1$ times. By lemma 5.4.6 it follows that no extension of $f\upharpoonright_{l'}$ meets S and thus $f\upharpoonright_{l'} \Vdash^s_{n+1} \neg\varphi$. By part 1 it follows that $f\upharpoonright_{l'} \Vdash^*_{T_m} \neg\varphi$.

3. Inductively suppose that $f \Vdash^*_{T_m} \phi \implies f \Vdash^*_{T_{m+1}} \phi$ for $\phi \in \Sigma^0_k \cup \Pi^0_k$ with $k \leq n < m$. So suppose ϕ properly in $\Sigma^0_{n+1} \cup \Pi^0_{n+1}$ and $f \Vdash^*_{T_m} \phi$ we show that $f \Vdash^*_{T_{m+1}} \phi$.

Case 1: $\phi = \exists x \psi(x)$

For some x, $f \Vdash^*_{T_m} \psi(x)$ so by the inductive hypothesis $f \Vdash^*_{T_{m+1}} \psi(x)$ thus $f \Vdash^*_{T_{m+1}} \phi$.

Case 2: $\phi = \neg\varphi$

For some l

$$(\forall\sigma \supseteq f\upharpoonright_l)\left(\sigma \in T_m \implies \neg\sigma \Vdash^*_{T_m} \varphi\right) \tag{5.4.11.4}$$

Now suppose, by way of contradiction

$$\left(\exists\sigma' \in T_{m+1}\right)\left(\sigma' \supseteq f\upharpoonright_l \wedge \sigma' \Vdash^*_{T_{m+1}} \varphi\right) \tag{5.4.11.5}$$

Let $f' \supset \sigma'$ be m locally generic on T_{m+1} where σ' witnesses (5.4.11.5). Since $f' \Vdash^*_{T_{m+1}} \varphi$ by genericity $f' \models \varphi$ but by part 2 f' is also m locally generic on T_m hence $f' \Vdash^*_{T_m} \varphi$. Hence

$$(\exists\sigma \in T_m)\left(f\upharpoonright_l \subset \sigma \subset f' \wedge \sigma \Vdash^*_{T_m} \varphi\right) \tag{5.4.11.6}$$

This contradicts (5.4.11.4) hence

$$\left(\forall\sigma' \in T_{m+1}\right)\left(\sigma' \supseteq f\upharpoonright_l \implies \neg\sigma' \Vdash^*_{T_{m+1}} \varphi\right) \tag{5.4.11.7}$$

By definition of strong forcing we can thus conclude $f \Vdash^*_{T_{m+1}} \varphi$.

□

Lemma 5.4.12. *If $f \in [T_m]$ and f densely meets Γ^m then f is m locally generic on T_m.*

Proof. Assume, by way of induction, this claim holds for n with $n + 1 = m$. Given f as in the lemma 5.3.4 we can apply the inductive hypothesis to conclude that f is n locally generic on T_n. Making use of number 3 of lemma 5.4.11 we can conclude that f is n locally generic on T_{n+1}. Now fix φ properly in Σ^0_{n+1} with $\varphi = \exists x\psi(x)$. By lemma 5.4.10 we need only consider the following two cases.

Case 1: $f \Vdash^s_{n+1} \varphi$

For some x, $f \Vdash^s_n \psi(x)$ and by number 1 of lemma 5.4.11 we can conclude that $f \Vdash^*_{T_n} \psi(x)$ and by number 3 of the same theorem we can conclude $f \Vdash^*_{T_{n+1}} \psi(x)$ and thus $f \Vdash^*_{T_{n+1}} \varphi$.

Case 2: $f \Vdash^s_{n+1} \neg\varphi$

For some l

$$(\nexists\sigma \in T_{n+1})\left(\sigma \supset f\upharpoonright_l \implies \neg\sigma \Vdash^s_{n+1} \varphi\right) \tag{5.4.12.1}$$

Now suppose, for a contradiction, that

$$\left(\exists\sigma' \in T_{n+1}\right)\left(\sigma' \supset f\restriction_l \wedge \sigma \Vdash^*_{T_{n+1}} \varphi\right) \tag{5.4.12.2}$$

If σ' witnesses this then pick some f' $n+1$ locally generic on T_{n+1} with $f' \supset \sigma'$. By number 1 of lemma 5.4.11 we can conclude that $f' \Vdash^s_{n+1} \varphi$ but then for some σ

$$f\restriction_l \subset \sigma \subset f' \wedge \sigma \in T_{n+1} \wedge \sigma \Vdash^s_{n+1} \varphi \tag{5.4.12.3}$$

This is a contradiction hence we can conclude that $f \Vdash^*_{T_{n+1}} \neg\varphi$

□

Lemma 5.4.13. *f^m is m locally generic on T_m.*

Proof. Immediate from the definition of f^m and lemma 5.4.12. □

Proof. By lemma 5.4.5 we know that f^{n+2} is a self-modulus and by lemma 5.4.13 that f^{n+2} is n locally generic on some perfect tree. Hence by 5.2.1 no f^* computable by n jumps of f^{n+2} is a uniform modulus of f^{n+2}. As n was arbitrary this demonstrates the theorem. □

Bibliography

[1] C. J. Ash and J. Knight. *Computable structures and the hyperarithmetical hierarchy*, volume 144 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 2000. 12

[2] Peter A Cholak, Noam Greenberg, and Joseph S. Miller. Uniform almost everywhere domination. *Journal of Symbolic Logic*, 71:1057–1072, 2006. 16

[3] A. Church. The constructive second number class. *Bulletin of the American Mathematical Society*, 44:224–232, 1938. 8

[4] S. S. Goncharov and J. F. Knight. Computable structure and non-structure theorems. *Algebra and Logic*, 41(6):351–373, 2002. 12

[5] Sergey S. Goncharov, Valentina S. Harizanov, Julia F. Knight, and Richard A. Shore. Π^1_1 relations and paths through $\mathcal{O}^*$. *J. Symbolic Logic*, 69(2):585–611, 2004. 12

[6] Marcia J. Groszek and Theodore A. Slaman. Moduli of computation. Lecture delivered at "Logic, Computability and Randomness", January 2007. 17, 22

[7] Carl G. Jockusch and Richard A Shore. Pseudo-jump operators II: Transfinite iterations, hierarchies and minimal covers. *The Journal of Symbolic Logic*, 49:1205–1236, December 1984. 11

[8] Carl G. Jockusch and Robert I Soare. Encodability of Kleene's $\mathcal{O}$. *The Journal of Symbolic Logic*, 38:437–440, December 1973. 16

[9] Jr. Jockusch, Carl G. and Richard A. Shore. Pseudo jump operators. i: The r. e. case. *Transactions of the American Mathematical Society*, 275(2):599–609, 1983. 11

[10] S. C. Kleene. On notation for ordinal numbers. *The Journal of Symbolic Logic*, 3(4):150–155, 1938. 8

[11] E. G. K. Lopez-Escobar. Remarks on an infinitary language with constructive formulas. *The Journal of Symbolic Logic*, 32(3):305–318, sep 1967. 12

[12] David Luckham and Hilary Putnam. On minimal and almost-minimal systems of notations. *Transactions of the American Mathematical Society*, 119(1):86–100, 1965. 10

[13] Donald A Martin. Classes of recursively enumerable sets and degrees of unsolvability. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 12:295–310, 1966. 16

[14] T. G McLaughlin. On the relations between some rate-of-growth conditions. *Proceedings of the American Mathematical Society*, 69:151–155, April 1978. 16

[15] P. G. Odifreddi. *Classical recursion theory. Vol. II.* North-Holland Publishing Co., Amsterdam, 1999. 7, 26

[16] Piergiorgio Odifreddi. Forcing and reducibilities. *The Journal of Symbolic Logic*, 48:288–310, June 1983. 2, 7

[17] H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, New York, 1967. 2, 8, 10

[18] Gerald Sacks. *Higher Recursion Theory.* Perspectives in Mathematical Logic. Springer–Verlag, Heidelberg, 1990. 2, 8, 10

[19] Robert I. Soare. *Recursively Enumberable Sets and Degrees: A Study of Computable Functions and Computably Generated Sets (Perspectives in Mathematical Logic).* Springer-Verlag New York, Inc., New York, NY, USA, 1987. 2, 19, 21, 51

[20] Robert M Solovay. Hyperarithmetically encodable sets. *Transactions of the American Mathematical Society*, 239:99–122, May 1978. 17, 22, 24

[21] S. Tennenbaum. Degree of unsolvability and the rate of growth of functions. In *Proc. Sympos. Math. Theory of Automata*, pages 71–73. Polytechnic Press of Polytechnic Inst. of Brooklyn, Brooklyn, N.Y., 1963. 16

[22] C. E. M. Yates. Three theorems on the degrees of recursively enumerable sets. *Duke Math. J*, 32(3):461–468, 1965. 16