Degrees of Computability and Randomness

Anthony William Morphett

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy

The University of Leeds
School of Mathematics
August 2009

Jointly authored publications

Work from the following publication has been included in this thesis:

G. Barmpalias and A. Morphett. Non-cupping, measure and computably enumerable splittings. *Mathematical Structures in Computer Science* **19** (2009), 25–43.

The published version of this paper is included as Appendix A of this thesis.

Theorem 1.1 of the paper was suggested by the first author, and the proof was developed jointly by both authors. Section 4 of the paper was initially written by the second author and later revised by both authors. It appears as chapter 4 of this thesis.

The ideas of section 3 of the paper were suggested by the first author (motivated by discussions with André Nies). The basic technique of the construction of Theorem 3.1 was developed by the second author. An earlier draft of Theorem 3.1 was written by the second author, and the final version was written by the first author. A version of Theorem 3.1 written by the second author appears in chapter 3 of this thesis as Theorem 3.1. Theorem 3.2 of the paper was the work of the first author and does not appear in this thesis.

Acknowledgements

I would like to thank my supervisor Prof. S. Barry Cooper for his support, encouragement and numerous discussions both mathematical and non-mathematical during my time in Leeds. I would also like to thank to George Barmpalias and Andy Lewis for stimulating discussions and collaborations, from which I and the work in this thesis have benefited greatly.

Thanks to the Marie Curie programme for funding my studies in Leeds as well as travel to numerous conferences and research secondments via the MATHLOGAPS scheme. Many thanks particularly to Prof. Dugald Macpherson; I am grateful for his help with funding, visas and travel reimbursements, and for his efforts organising the regular MATHLOGAPS summer schools and more generally as MATHLOGAPS coordinator. Thanks also to Dr. David Salinger for his support and especially his assistance regarding my visa.

I am grateful to Prof. Andrea Sorbi of Siena and Prof. Schwichtenberg of Munich for their generous hospitality during my secondment visits to their departments. Thanks also to Prof. Doug Cenzer of Gainesville and Prof. Alexandra Soskova of Sofia for arranging additional conference travel grants.

My thanks to everyone in the School of Mathematics for providing a welcoming community and a stimulating and exciting research and study environment.

Special thanks to my wife Avalon for her dedicated companionship and continuing support, which I greatly appreciate.

Abstract

In this thesis we will study LR-reducibility, a weakening of Turing reducibility arising naturally in the context of relative randomness. We will focus particularly on the LR-degrees of c.e. and Δ^0_2 sets. In chapter 2 we present a technique for lower cone avoidance in full approximation constructions, and use it to construct a c.e. LR-degree that is LR-incomparable with a given intermediate Δ^0_2 LR-degree. We also give a weak upward density result, showing that every low Δ^0_2 LR-degree is bounded by an incomplete c.e. LR-degree.

In chapter 3, we consider splittings of c.e. sets into two c.e. sets of the same LR-degree; an analogue of the notion of mitoticity previously studied in the context of Turing degrees. We show that there are c.e. sets that cannot be split into two c.e. sets of the same LR-degree, and that such sets may be Turing complete or low, and can be made to avoid nontrivial upper cones of c.e. LR-degrees. We also show that this notion differs from Turing nonmitoticity on both the low-for-random and non-low-for-random c.e. sets.

Chapter 4 presents a construction of an LR-complete c.e. set that is non-cuppable in the c.e. Turing degrees. This is a strengthening of an earlier result of Barmpalias and Montalbán [**7**], and can also be seen as a strengthening of a theorem of Harrington (in [**53**]).

In chapter 5 we introduce a new notion of promptness for c.e. sets, *prompt non-low-for-randomness*, which can be seen as an LR-analogue of prompt simplicity. We investigate the Turing degrees of promptly non-low-for-random sets, and compare the property of prompt non-low-for-randomness to the traditional property of prompt simplicity.

Contents

CHAPTER 1

Introduction

1. Introduction & outline

Algorithmic randomness has been an important and interesting field of study within computability theory.[1] Although the roots of randomness go back to work by von Mises [**72**] on selection rules and by Kolmogorov [**34**] and Solomonoff [**69**] in the context of finite strings, the study of randomness of infinite sequences in its current form traces its origins to Martin-Löf's work [**50**] of 1966. In this paper, Martin-Löf developed the concept of Martin-Löf tests and the notion of Martin-Löf randomness, which has become one of the most popular notions in the recent study of algorithmic randomness. Martin-Löf's eponymous test notion (of which a formal definition is given in section 3.1) was intended to formalise the idea of *effective nullsets* - those measure 0 subsets of Cantor space which can be approximated in a computably enumerable way. Since there are only countably many such nullsets, this yields a well-defined notion of randomness: each such nullset can be seen as the set of strings possessing some 'special property' or 'distinguishing feature' that can be computably approximated, so any string that avoids all such computably approximable nullsets and the corresponding special properties is random, as far as can be effectively detected.

Several other important developments soon followed Martin-Löf's work. Schnorr [**66**] established a connection between randomness of an infinite string in the sense of Martin-Löf and complexity of its finite initial segments in the sense of Kolmogorov. Specifically, Schnorr proved that a sequence X is Martin-Löf random iff there is a constant c such that, for all n,

$$K(X \upharpoonright n) \geq n - c,$$

[1] We defer a discussion of notation, definitions etc until sections 2 and 3, though our notation and terminology is standard. References for background, history, unexplained notation and definitions are Nies [**60**] and Downey and Hirschfeldt [**20**].

where $X \upharpoonright n$ denotes the first n bits of X and $K(\sigma)$ is the prefix-free Kolmogorov complexity of the finite string σ.[2] Schnorr also gave a characterisation of Martin-Löf randomness in terms of *martingales*: X is Martin-Löf random iff no c.e. martingale succeeds on X.[3] Schnorr's work thus unified three different and complementary approaches to randomness, those of effective measure, compressibility and effective gambling strategies. Each approach to randomness has its own advantages and yields its own insights into the class of Martin-Löf random sequences.

The compressibility approach was taken by Chaitin [**11**], whose work on prefix-free complexity yielded a concrete example of a random real number, the halting probability Ω.[4] The halting probability is particularly interesting as it is c.e.[5] and Turing equivalent to the halting problem. Such c.e. random reals, and their special status of Turing-completeness, has been studied for instance in [**10**], [**37**] and [**21**].

Because of its flexibility, the martingale approach has proved useful for generalising or strengthening Martin-Löf randomness. Schnorr [**66**] defined the notions of computable and Schnorr randomness by requiring that the martingales be computable (rather than just c.e.) and that they succeed at a computable rate, respectively. Computable randomness and Schnorr randomness were further studied for example in [**19**],

---

[2] A partial computable function (machine) $M :\subseteq 2^{<\omega} \to 2^{<\omega}$ is *prefix-free* if its domain is prefix-free; that is,

$$M(\sigma)\downarrow \Rightarrow M(\tau)\uparrow \text{ for all } \tau \subset \sigma.$$

All prefix-free machines can be computably listed, and hence there is a universal prefix-free machine U. The *prefix-free Kolmogorov complexity* of a finite string σ, denoted $K(\sigma)$, is the length of the shortest string τ such that $U(\tau) = \sigma$. This is the *length of the shortest program that outputs* σ, or the *shortest description* of σ (in the programming language of the universal machine U). For a comprehensive survey of Kolmogorov complexity in connection with algorithmic randomness, see Downey and Hirschfeldt [**20**]. For a treatment of Kolmogorov complexity more generally, see Li and Vitányi [**45**].

[3] A *martingale* is a function $p : 2^{<\omega} \to \mathbb{Q}$ satisfying the condition

$$q(\sigma) = \frac{1}{2}\big(q(\sigma^\frown 0) + q(\sigma^\frown 1)\big).$$

A martingale is a formalisation of a betting strategy in a fair game; if we think of a gambler as betting on the bits of an infinite sequence X, $q(X \upharpoonright n)$ is the gambler's capital after betting on the first n bits according to the strategy q. The martingale q *succeeds* on an infinite sequence X if

$$\limsup_n q(X \upharpoonright n) = \infty.$$

A c.e. martingale is one in which the set $\{x \in \mathbb{Q} : x \leq q(\sigma)\}$ is uniformly c.e. in σ.

[4] Let U be a universal prefix-free machine. The halting probability

$$\Omega := \sum_{\sigma : U(\sigma)\downarrow} 2^{-|\sigma|} = \mu(\text{dom } U)$$

is the probability that the universal machine U will halt if it is fed random bits as its input.

[5] A c.e. real is a real number $x \in [0,1]$ such that the lower cut $\{q \in \mathbb{Q} : q \leq x\}$ is c.e. This differs from the notion of a c.e. set; if we think of the binary representation of a c.e. real x as giving the characteristic function of a set X, the set X need not be c.e. We will not use the notion of c.e. reals any further; when we write c.e. we mean it in the sense of a c.e. set.

[**23**]. Another way to generalise the martingale approach to randomness is via non-monotonicity. Kolmogorov [**33**] and Loveland [**46**] proposed non-monotonic betting strategies, in which the strategy is allowed to choose the next position in the sequence on which to bet. This was studied more recently in [**57**] and [**52**]. Still another topic arising from martingales is effective dimension. An effective version of Hausdorff dimension was originally suggested by Lutz [**47**], and Mayordomo [**51**] established connections between Kolmogorov complexity and Lutz's original formulation of effective dimension in terms of martingales.

Although the compressibility and martingale approaches to randomness and the various strengthenings or weakenings of Martin-Löf randomness are interesting and fruitful, we will not concern ourselves with them here. We will restrict our attention to Martin-Löf's original notion of randomness in terms of Martin-Löf tests, though we note that many important concepts such as low for randomness and LR-reducibility can be thought of equally well in terms of Kolmogorov complexity or martingales as in terms of Martin-Löf tests. Comprehensive treatments of the various notions of randomness may be found in the references [**20**] and [**60**].

Work on Martin-Löf randomness continued through the 1980s with important contributions by Solovay [**70**], Kučera [**35**], Gacs [**27**] and van Lambalgen [**71**], to name only a few. However it was in the late 1990s that interest in algorithmic randomness blossomed. One topic of particular interest was relative randomness, Martin-Löf randomness relativised to an oracle. The notions of Martin-Löf tests and randomness can naturally be relativised to an arbitrary oracle; one can then study the information content of an oracle X by examining the randomness notion obtained by relativising to X. In particular, relative randomness gave rise to various lowness properties, capturing ways in which an oracle can have low information content as far as randomness is concerned. Zambella [**77**] defined the notion of *low for randomness*: an oracle A is low for random if all (unrelativised) random sequences are also random relative to A. Such an oracle is no assistance for detecting patterns or approximating nullsets, compared to the unrelativised case. Kučera and Terwijn [**39**] constructed a noncomputable c.e. set that is low for random. Muchnik [**56**] defined and studied the class of *low for K*, those oracles which are no assistance as far as Kolmogorov complexity is concerned.[6] Others [**22, 59**] studied the K-trivials, sequences with the smallest possible initial segment complexity.[7] A fourth lowness property, being a *basis for randomness*[8], was suggested by Kučera [**36**]. This work culminated in [**59**] and [**28**] with proofs that these classes co-incide; in fact, they form an ideal in the Turing degrees that has since been studied extensively [**38, 58**].

Relativising notions such as low for randomness naturally leads to reducibilities and associated degree structures. Nies [**59**] defined the *LR-reducibility* $\leq_{LR}$, which is the main topic of this thesis: oracle A is LR-reducible to B, $A \leq_{LR} B$, if the class of Martin-Löf randoms relative to B is contained in the class of randoms relative to A. Intuitively, the oracle B is at least as useful, computationally, for detecting patterns as the oracle A. LR-reducibility extends Turing reducibility, with a least degree consisting of exactly the low for random sets. This gives LR-reducibility a claim to being a very natural notion for studying information content in the context of relative randomness.

In this thesis we will mainly study the reducibility $\leq_{LR}$, its degree structure, its connections with Turing reducibility, and related notions. We will focus particularly on the LR-degrees of c.e. and Δ^0_2 sets. In chapter 2 we prove some results about the structure of the Δ^0_2 LR-degrees (that is, LR-degrees containing Δ^0_2 sets). Specifically, we present a technique for lower cone avoidance in full approximation constructions, and use it to construct a c.e. LR-degree that is LR-incomparable with a given intermediate Δ^0_2 LR-degree. We also give a weak upward density result, showing that every low Δ^0_2 LR-degree is bounded by an incomplete c.e. LR-degree.

In chapter 3, we consider splittings of c.e. sets into two c.e. sets of the same LR-degree. We look at an LR-degree analogue of the notion of mitoticity studied in the context of c.e. Turing degrees by Lachlan [**41**], Ladner [**43**] and others. We show that there are c.e. sets that cannot be split into two c.e. sets of the same LR-degree, and that such sets may be Turing complete or low, and can be made to avoid nontrivial upper

[6] A set A is *low for K* if there is a constant c such that

$$K(\sigma) \leq K^A(\sigma) + c$$

for all strings σ. That is, the oracle A does not help (beyond the fixed constant c) in compressing data. K^A is the prefix-free Kolmogorov complexity relative to oracle A.

[7] A is *K-trivial* if the complexity of initial segments of A is as small as possible: there is a constant c such that

$$K(A \upharpoonright n) \leq K(n) + c$$

for all n.

[8] A is a basis for randomness if there is an X such that X is random relative to A and $A \leq_T X$.

cones of c.e. LR-degrees. We also show that this notion differs from Turing nonmitoticity on both the low-for-random and non-low-for-random c.e. sets.

Chapter 4 presents a construction of an LR-complete c.e. set that is non-cuppable in the c.e. Turing degrees. This is a strengthening of an earlier result of Barmpalias and Montalbán [**7**] that there is a cappable LR-complete c.e. set, and can also be seen as a strengthening of a theorem of Harrington (in [**53**]).

In chapter 5 we introduce a new notion of promptness for c.e. sets, *prompt non-low-for-randomness*, which can be seen as an LR-analogue of prompt simplicity. We investigate the Turing degrees of promptly non-low-for-random sets, and compare the property of prompt non-low-for-randomness to the traditional property of prompt simplicity.

2. Notation and conventions

2.1. Cantor space and strings. Let $2^{<\omega}$ denote the set of all finite binary strings, and 2^{ω} denote Cantor space of infinite binary sequences. We call members of $2^{<\omega}$ *strings*, and members of 2^{ω} *reals*. We identify reals with subsets of $\mathbb{N}$ in the usual way, and sometimes use the terms *real*, *set* and *oracle* synonymously. We typically use the letters σ, τ, ρ for strings and A, X etc. for reals. We sometimes use σ, τ also for finite sets of strings, which represent clopen subsets of 2^{ω}. For $X \in 2^{\omega}$ and $n \in \mathbb{N}$, $X \upharpoonright n$ denotes the initial segment of X of length n, a finite string. For strings σ, τ, we write $\sigma \subseteq \tau$ to denote that σ is an initial segment of τ, and $\subset$ to denote a strict initial segment. We also write $\sigma \subset A$ for $A \in 2^{\omega}$ to indicate that σ is an initial segment of the real A. $\sigma|\tau$ denotes that σ and τ are incomparable, i.e. $\sigma \not\subseteq \tau$ and $\tau \not\subseteq \sigma$. A set of strings is *prefix-free* if for any two distinct strings σ, τ in the set, we have $\sigma|\tau$. The length of a string is denoted $|\sigma|$. We obtain the standard bijection between $2^{<\omega}$ and $\mathbb{N}$ by ordering the finite strings first by length and then lexicographically.

Cantor space 2^{ω} is equipped with the usual topology, generated by basic clopen sets $[\sigma] = \{X \in 2^{\omega} : \sigma \subset X\}$ for $\sigma \in 2^{<\omega}$. We can extend the notation $[\cdot]$ in the obvious way to sets of strings: if $C \subseteq 2^{<\omega}$ then

$$[C] = \{X \in 2^{\omega} : \sigma \subset X \text{ for some } \sigma \in C\}.$$

To simplify presentation we will often omit the brackets and denote by σ both the string and the clopen set (and similarly for sets of strings). It will be clear from context which is intended. The Lebesgue measure on 2^ω is denoted by μ. Note that the measure of (the clopen set corresponding to) a string σ is $\mu(\sigma) = 2^{-|\sigma|}$, and for a prefix-free set C of strings it is $\mu(C) = \sum_{\sigma \in C} 2^{-|\sigma|}$.

Suppose that $V \subseteq 2^\omega$ is a clopen set. When we say *the least* (or *leftmost*) *clopen set* $C \subseteq V$ *of measure* q we mean the clopen set C, if it exists, such that $C \subseteq V$, $\mu(C) = q$, and if $Z \in V - C$ then $Z > X$ for all $X \in C$, where $<$ is the lexicographical ordering on reals. That is, C is the leftmost clopen subset of V of size q. If we require a particular representation of the clopen set C by finite strings, ie. a finite prefix-free $D \subseteq 2^{<\omega}$ such that $C = [D]$, we may take the first such D under a standard listing of finite prefix-free sets of strings.

2.2. Turing functionals and operators as c.e. sets. It is convenient to consider computably enumerable (c.e.) sets to be the fundamental objects of computability theory, since we can obtain other objects such as Turing reductions or Σ^0_1 classes from c.e. sets. In several constructions we will build Turing functionals or oracle Σ^0_1 classes as c.e. sets of axioms. We briefly discuss how such objects can be represented as c.e. sets.

Let $\langle \cdot, \cdot \rangle$ be a standard pairing function, which is a computable bijection from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$. From this we can obtain codings of tuples of all sizes in a standard way. We can consider a Turing reduction Φ to be a consistent c.e. set of axioms of the form

$$\langle x, z, \tau \rangle \text{ for } x, z \in \mathbb{N} \text{ and } \tau \in 2^{<\omega}, \tag{1}$$

which asserts that

$$\Phi^X(x) \downarrow = z \text{ if } \tau \subset X$$

for $X \in 2^\omega$. In this context consistent means that if $\langle x, z, \tau \rangle, \langle x, z', \tau' \rangle$ are both axioms in the set then

$$\tau | \tau' \text{ or } z = z'.$$

Any such consistent c.e. set W of axioms certainly defines a (possibly partial) Turing reduction; moreover, from an enumeration of any c.e. set W we can obtain a Turing reduction by discarding any numbers enumerated into W which are not of the form (1) or which are not consistent with any axioms previously enumerated. Conversely, given a Turing reduction Ψ we can obtain an equivalent c.e. set of axioms by the usual procedure of enumerating all possible computation paths.

In cases when we are interested in Turing reductions as operators from 2^ω to 2^ω, it is convenient to consider Turing reductions as c.e. sets of axioms of the form $\langle \sigma, \tau \rangle$ for $\sigma, \tau \in 2^{<\omega}$ which assert that

$$\sigma \subseteq \Phi^X \text{ if } \tau \subset X.$$

In this case consistent means that if $\langle \sigma, \tau \rangle, \langle \sigma', \tau' \rangle$ are both axioms in the set then

$$\tau \subseteq \tau' \Rightarrow \sigma \subseteq \sigma'.$$

From such a set we obtain a Turing reduction whose domain (for each oracle X) is $\mathbb{N}$ or an initial segment of $\mathbb{N}$. Conversely, from a Turing reduction Ψ we can obtain a c.e. set of axioms for a Turing reduction Φ such that $\Psi^X(x) \simeq \Phi^X(x)$ on that part of the domain of Ψ^X which is an initial segment of $\mathbb{N}$.

A *c.e. operator* is a uniform procedure for obtaining an A-c.e. set W^A from an oracle A. A c.e. operator can be considered as a c.e. set W of axioms of the form

$$\langle n, \tau \rangle \text{ for } n \in \mathbb{N}, \tau \in 2^{<\omega}$$

if we are interested in sets of numbers, or of the form

$$\langle \sigma, \tau \rangle \text{ for } \sigma, \tau \in 2^{<\omega}$$

if we are interested in sets of strings, which assert, respectively, that $n \in W^A$ or $\sigma \in W^A$ if $\tau \subset A$.

Let $A \in 2^\omega$. An *A-Σ^0_1 class* V^A is a subset of 2^ω which is Σ^0_1 relative to the oracle A. That is, there is an A-computable relation R on $2^{<\omega}$ such that

$$X \in V^A \Leftrightarrow \exists n\, R(X \restriction n).$$

Such a class V^A is generated by an A-c.e. set of strings W^A such that $X \in V^A$ iff $X \supset \sigma$ for some $\sigma \in W^A$, although many different A-c.e. sets may generate the same A-Σ^0_1 class of reals. However, when we talk of an A-Σ^0_1 class V^A, we will always understand it to be generated by a fixed A-c.e. set, and for convenience we often identify the class V^A with the c.e. set of strings.

An *oracle Σ^0_1 class* V is a uniform procedure for obtaining an A-Σ^0_1 class V^A from an oracle A. Since A-Σ^0_1 classes are generated by c.e. sets of strings, such a uniform procedure is in fact a c.e. operator. In particular, an oracle Σ^0_1 class V can be thought of as a c.e. set of axioms $\langle\sigma,\tau\rangle$ as above, but asserting in the context of Σ^0_1 classes that if $\tau \subset A$ and $\sigma \subset X$ then $X \in V^A$. It is often convenient to assume without loss of generality that for a fixed A the set

$$\{\sigma \in 2^{<\omega} : \langle\sigma,\tau\rangle \in V \text{ and } \tau \subset A\}$$

is prefix-free, and that if $\langle\sigma,\tau\rangle$ is enumerated into V at stage s then $|\sigma| = |\tau| = s$.

For a set $W \subseteq 2^{<\omega}$, define the weight function

$$\text{weight } W = \sum_{\sigma\in W} 2^{-|\sigma|}.$$

Note that weight $W \geq \mu([W])$, with equality exactly when W is prefix-free. Say that $W \subseteq 2^{<\omega}$ is *bounded* if weight $W < \infty$. If V is an oracle Σ^0_1 class, say that V is *bounded* if there is a rational $q < 1$ such that $\mu(V^X) < q$ for all oracles X.

Bounded oracle Σ^0_1 classes and c.e. operators are a particularly useful tool when working with LR-reducibility, because of Theorem 1.5. In many constructions we will need a computable listing of all bounded oracle Σ^0_1 classes or c.e. operators, along with a bound q. Let $\langle\tilde{V}_e, q_e\rangle$ for $e \in \mathbb{N}$ be a listing of all pairs of a (not necessarily bounded) oracle Σ^0_1 class $\tilde{V}$ and a dyadic[9] rational q in the interval $(0,1)$. We can obtain a bounded oracle Σ^0_1 class V_e from the pair $\langle\tilde{V}_e, q_e\rangle$ by discarding any axioms $\langle\sigma,\tau\rangle$ which are enumerated into $\tilde{V}_e$ and which would make $\mu(V^X_e) \geq q_e$ for any $X \in 2^\omega$. If V is an oracle Σ^0_1 class such that $\mu(V^X) < q$ for some dyadic $q \in (0,1)$, then certainly the pair $\langle V, q\rangle$ will appear somewhere in the listing $(\langle V_e, q_e\rangle)_{e\in\mathbb{N}}$. In a similar way we may obtain a listing of all bounded c.e. operators.

Suppose that V is an oracle Σ^0_1 class (given by a prefix-free c.e. operator) and A is a c.e. set. From enumerations $V[s]$ and $A[s]$ of V and A, we can approximate V^A in a Σ^0_2 way. Let $V^A[s]$ denote $V[s]^{A[s]}$, the approximation to V^A at stage s, which is a clopen set uniformly computable in s. If a string $\sigma \in V^A[s]$, then there is an axiom $\langle\sigma,\tau\rangle \in V[s]$ such that $\tau \subset A[s]$. In this case we say that σ is in $V^A[s]$ *via the axiom* $\langle\sigma,\tau\rangle$. Say that the computation $\sigma \in V^A[s]$ *became valid at* $t \leq s$ if t is the least such that $\langle\sigma,\tau\rangle \in V[t]$ and $\tau \subset A[t]$. When we say *the oldest string in* $V^A[s]$ *satisfying some condition* P we mean the unique string, if it exists, such that $\sigma \in V^A[s]$, and for any other $\sigma' \in V^A[s]$, either the computation $\sigma \in V^A[s]$ became valid before the computation $\sigma' \in V^A[s]$, or

---

[9] A rational whose binary expansion is finite, ie. a finite sum of powers of 2.

if they became valid at the same stage then σ preceeds σ' in the length/lexicographical ordering of strings.

3. Randomness and LR-reducibility

3.1. Martin-Löf randomness. A *Martin-Löf test* is a sequence $(U_i)_{i\in\mathbb{N}}$ of uniformly Σ^0_1 classes such that $\mu(U_i) \leq 2^{-i}$. Each U_i is called a member of the test. The intersection $\cap_i U_i$ is an effective measure 0 set (nullset). We say that a real $X \in 2^\omega$ is *captured* by the test (U_i) if $X \in \cap_i U_i$. X is *Martin-Löf random* if $X \notin \cap_i U_i$ for all Martin-Löf tests $(U_i)_{i\in\mathbb{N}}$; that is, X is not captured by any Martin-Löf test.

The motivation for Martin-Löf randomness is that a real is 'random' if it does not have any 'special properties' or 'patterns' that can be effectively approximated. A pattern or property is 'special' if it is atypical in the measure sense, that is, if the class of reals with that property has measure 0. Each Martin-Löf test approximates (the nullset of reals satisfying) one such special property. Hence if X is not captured by any Martin-Löf test, then it does not have any special property that can be effectively approximated, and is, as far as can be effectively determined, random.

By weakening or strengthening the notion of Martin-Löf test, we can obtain other notions of randomness. For instance, if we require that $\mu(U_i) = 2^{-i}$, then we obtain *Schnorr randomness* (after Schnorr [**66**]); if we require only that $\mu(U_i) \to 0$ as $i \to \infty$ without a uniform bound on $\mu(U_i)$ then we obtain a stronger randomness notion known as *weak 2-randomness.* We will not be concerned with other types of randomness in this thesis however, and we will use the word 'random' to just mean Martin-Löf random.

From an enumeration of any c.e. set W we can obtain a Martin-Löf test $(U_i)_{i\in\mathbb{N}}$ by treating W as a set of axioms of the form $\langle i, \sigma\rangle$ for $i \in \mathbb{N}$ and $\sigma \in 2^{<\omega}$, asserting that $\sigma \in U_i$, and discarding any axioms that would make $\mu(U_i) > 2^{-i}$. Moreover, every Martin-Löf test arises from some c.e. set in this way. Hence there is a uniform listing of all Martin-Löf tests. From this listing we can construct a *universal Martin-Löf test*: let $(U^k_i)_{i\in\mathbb{N}}$ be the k'th Martin-Löf test in the listing, and let

$$U_i = \bigcup_k U^k_{i+k+1}. \tag{2}$$

Clearly the U_i are uniformly Σ^0_1 and $\mu(U_i) \leq 2^{-i}$, hence they form a Martin-Löf test. We have $\cap_i U^k_i \subseteq \cap_i U_i$ for all k, so that $X \in 2^\omega$ is random iff $X \notin \cap_i U_i$.

The definitions of Martin-Löf tests and randomness may be relativised to an arbitrary oracle A. An *oracle Martin-Löf test* is a uniform sequence of oracle Σ^0_1 classes $(U_i)_{i\in\mathbb{N}}$ such that $\mu(U_i^A) \leq 2^{-i}$ for any oracle A. As in (2) above, we can obtain a universal oracle Martin-Löf test $(U_i)_{i\in\mathbb{N}}$. A real $X \in 2^\omega$ is *A-random* if $A \notin \cap_i U_i^A$ for the universal oracle Martin-Löf test $(U_i)_{i\in\mathbb{N}}$.

3.2. Low-for-randomness. Low for randomness, and equivalent notions such as K-triviality, have been a significant area of recent research (for instance, [**59, 22**]). An oracle $A \in 2^\omega$ is *low for random* if all (unrelativised) random reals are also A-random. That is,

$$\forall X \in 2^\omega \qquad X \text{ is random } \Rightarrow X \text{ is } A\text{-random.}$$

Informally, the oracle A is no assistance, compared to the unrelativised case, for detecting patterns in reals. The universal A-Martin-Löf test does not capture any reals that aren't already captured by the unrelativised universal Martin-Löf test.

Certainly all computable sets are low for random. A noncomputable low for random c.e. set was constructed by Kučera and Terwijn [**39**] using the now well-known cost function method. A series of work by Downey, Hirschfeldt, Nies and Stephan [**22**], Nies and Hirschfeldt [**59**], Nies [**59**] and Hirschfeldt, Nies and Stephan [**28**] culminated in the co-incidence of the class of low for randoms with several other classes of reals: the K-trivials, low for K, and the bases for randomness. We list some important facts about the class of low for random reals (some of which are best proved by first proving the corresponding fact for the K-trivials and then invoking the equivalence between K-triviality and low for randomness). All low for randoms are Δ^0_2, and superlow (that is, $A' \equiv_{tt} \emptyset'$). The low for randoms form an ideal in the Δ^0_2 or c.e. Turing degrees; that is, the low for random Turing degrees are closed downwards under $\leq_T$ and under join. Every Δ^0_2 low for random set is computable in some low for random c.e. set; hence in fact the ideal of low for random Turing degrees is generated by the c.e. low for randoms. For proofs of these facts, a thorough treatment of the notions of K-triviality, low for K and bases for randomness, and their equivalence with low for randomness, we refer to Nies [**60**] or Downey and Hirschfeldt [**20**].

3.3. LR-reducibility and LR-degrees. Now we come to the key notion of this thesis.

DEFINITION 1.1. Let $A, B \in 2^\omega$. The real A is LR-reducible to B, $A \leq_{LR} B$, if the class of B-random reals is contained in the class of A-random reals. That is,

$$\forall X \in 2^\omega \qquad X \text{ is } B\text{-random} \Rightarrow X \text{ is } A\text{-random}.$$

$A <_{LR} B$ denotes $A \leq_{LR} B$ but $B \not\leq_{LR} A$. A is LR-equivalent to B, denoted $A \equiv_{LR} B$, if $A \leq_{LR} B$ and $B \leq_{LR} A$. An *LR-degree* is an equivalence class under the equivalence relation $\equiv_{LR}$. The LR-degree of the real A is denoted $\deg_{LR}(A)$. An LR-degree is c.e. or Δ^0_2 if it contains a c.e. or Δ^0_2 set, respectively.

It is more intuitive to phrase the definition of $A \leq_{LR} B$ in contrapositive: if X is not random relative to A then it is not random relative to B. Informally, oracle B is at least as good at detecting patterns as oracle A.

LR-reducibility was first defined by Nies in 2005 [**59**]. It can be seen as a partial relativisation of the notion of low for random.[10] Clearly $A \leq_T B$ implies $A \leq_{LR} B$. The converse fails however; Kučera and Terwijn's [**39**] noncomputable low-for-random set A satisfies $A \leq_{LR} \emptyset$ but $A \not\leq_T \emptyset$. Hence LR-reducibility is a proper weakening of Turing reducibility. However it still has some properties in common with Turing reducibility. For instance, both $\leq_{LR}$ and $\leq_T$ are Σ^0_3 predicates (this is clear for $\leq_{LR}$, for instance, from Theorem 1.5 below). Also, some techniques from the study of the c.e. and Δ^0_2 Turing degrees can be adapted to work with LR-reducibility. Examples of this are Barmpalias, Lewis and Stephan [**6**] and Barmpalias, Lewis and Soskova [**5**], in which the familiar techniques of Sacks restraints and Sacks coding from the c.e. Turing degrees are adapted to the context of c.e. LR-degrees. We will use Sacks restraints in the LR-context in chapter 2.

We mention without detailed proof some key facts and significant results in the study of LR-degrees. The least LR-degree $\mathbf{0}_{LR}$ consists of exactly the low for random sets. If $A \equiv_{LR} B$ then A is low for random relative to B and B is low for random relative to A. Each LR-degree is countable (ie, contains only countably many sets). This follows from the fact that if $A \equiv_{LR} B$ then A and B are low for random relative to each other, and thus $A' \equiv_{tt} B'$ by relativising the fact that all low for randoms are superlow. This was first observed by Nies [**59**]. However, some LR-degrees have uncountably many predecessors. Barmpalias, Lewis and Soskova [**5**] and independently

[10] Note however that a full relativisation of the statement "A is low for random" to an oracle B would be that "all B-randoms are $A \oplus B$-random", ie X is B-random $\Rightarrow X$ is $A \oplus B$-random.

Miller and Yu [**55**] showed that there are uncountably many sets $A \leq_{LR} \emptyset'$. Since each LR-degree is countable, there are uncountably many LR-degrees below that of $\emptyset'$. Barmpalias [**3**] later improved this to show that there are uncountably many reals $\leq_{LR}$ any non-low-for-random Δ^0_2 real. Not every nonzero LR-degree has uncountably many predecessors though; Miller [**54**] showed that reals which are low for the halting probability Ω[11] bound only countably many LR-degrees. In particular all 2-randoms[12] are low for Ω and so bound only countably many LR-degrees.

Kjos-Hanssen, Miller and Solomon [**31**] showed that $\leq_{LR}$ is equivalent to another reducibility arising from relative randomness, LK-reducibility $\leq_{LK}$. Nies [**59**] defined $A \leq_{LK} B$ if there is a constant c such that for all strings σ,

$$K^B(\sigma) \leq K^A(\sigma) + c$$

where $K^X(\sigma)$ is the prefix-free Kolmogorov complexity of the string σ, relative to the oracle X. Intuitively, the oracle B is at least as useful for compressing finite data as A. The reducibility $\leq_{LK}$ has also been studied in [**54**]. Since $\leq_{LR}$ and $\leq_{LK}$ co-incide, one could re-interpret all results about $\leq_{LR}$ in terms of prefix-free Kolmogorov complexity, instead of in terms of Martin-Löf tests and measure.

As Turing reducibility implies LR-reducibility, each LR-degree is a union of Turing degrees. Barmpalias, Lewis and Soskova [**5**] studied the Turing degrees inside LR-degrees, and showed that every LR-degree contains infinite chains and infinite antichains of Turing degrees. They showed that this holds in the c.e. case also: every c.e. LR-degree contains infinite chains and antichains of c.e. Turing degrees. Each LR-degree is closed under $\oplus$, the join operation in the Turing degrees. This follows from the fact that if $A \equiv_{LR} B$ then A and B are low for random relative to each other, and the fact that low for randoms are closed under $\oplus$ relativised to A or B.

However, $\oplus$ does not give a join (least upper bound) operation in the LR-degrees: Kučera and Terwijn [**39**] constructed a promptly simple low for random set A, and by a well-known result from [**1**] the promptly simple c.e. Turing degrees co-incide with the low cuppable c.e. Turing degrees. Therefore there is a low B such that $A \oplus B \equiv_T \emptyset'$. B cannot be low for random since the low for randoms are closed under $\oplus$ and all low

[11] A real A is low for Ω if every (equivalently, some) random of c.e. Turing degree is also A-random. This notion was defined by Nies, Stephan and Terwijn [**61**], and further studied in an excellent paper by Downey, Hirschfeldt, Miller and Nies in [**21**] and by Miller [**54**].
[12] X is 2-random if it is random relative to the halting problem $\emptyset'$.

for randoms are low (and in particular cannot be $\geq_T \emptyset'$). Moreover $B \not\equiv_{LR} \emptyset'$ since $B' \not\equiv_{tt} \emptyset''$. Hence $B \oplus A \not\equiv_{LR} B$ even though $A \equiv_{LR} \emptyset$. It is unknown if every pair of LR-degrees has a least upper bound, ie. if the LR-degrees form an upper semi-lattice.

Barmpalias, Lewis and Soskova [5] also studied the c.e. LR-degrees as a structure. They prove analogues of some theorems from the c.e. Turing degrees, such as the Sacks splitting theorem: that any non-low-for-random c.e. set A can be split into two disjoint c.e. sets B, C such that $A \not\leq_{LR} B$ and $A \not\leq_{LR} C$. Barmpalias, Lewis and Stephan [6] continue the study of c.e. LR-degrees, proving a weak density theorem:

THEOREM 1.2 (Barmpalias, Lewis and Stephan [6]). *Let A, B be c.e. sets such that $A <_{LR} B$ and $A \leq_T B$. Then there is a c.e. set C such that $A <_{LR} C <_{LR} B$ and $A \leq_T C \leq_T B$.*

The proof of Theorem 1.2 is an adaptation of that of the Sacks density theorem for the c.e. Turing degrees. It is not known if this holds without the requirement that $A \leq_T B$.

Hirschfeldt, Nies and Stephan proved the following theorem connecting computable enumerability, randomness and incompleteness.

THEOREM 1.3 (Hirschfeldt, Nies and Stephan [28]). *Suppose A is c.e., Z is random and $\emptyset' \not\leq_T A \oplus Z$. Then Z is A-random.*

It follows from this theorem that the Δ^0_2 and c.e. LR-degrees differ.

COROLLARY 1.4 (Folklore). *There is a Δ^0_2 LR-degree that does not contain any c.e. sets.*

PROOF. Let Z be a low random; such a random exists by the Low Basis Theorem (see Soare [68] §VI.5.13) applied to the complement of a member of a universal Martin-Löf test. We claim that the LR-degree of Z does not contain any c.e. set. Suppose that $Z \equiv_{LR} A$ for some c.e. A. Since each LR-degree is closed under $\oplus$, $A \oplus Z$ is low and in particular $A \oplus Z <_T \emptyset'$. Therefore by Theorem 1.3, Z is random relative to A. But Z is not random relative to itself, contradicting that $A \equiv_{LR} Z$. □

In fact, the Low Basis Theorem guarantees that Z is superlow, $Z' \leq_{tt} \emptyset'$, and hence Z is ω-c.e. We may therefore deduce the stronger result that the ω-c.e. LR-degrees differ from the c.e. LR-degrees. It is not known if the n-c.e. LR-degrees differ from the c.e. LR-degrees or from the ω-c.e. LR-degrees.

Barmpalias [**2**] has established that the structures of c.e. and Δ^0_2 LR-degrees are not elementarily equivalent to the structures of c.e. and Δ^0_2 Turing degrees, respectively. The elementary difference that Barmpalias gave between the c.e. LR-degrees and c.e. Turing degrees is the existence of minimal pairs: Lachlan [**40**] and independently Yates [**74**] showed that there are minimal pairs of c.e. Turing degrees, whereas Barmpalias [**2**] established that there are no minimal pairs of c.e. LR-degrees. The elementary difference between the Δ^0_2 LR-degrees and Δ^0_2 Turing degrees is the existence of minimal degrees: Sacks [**64**] constructed a minimal Δ^0_2 Turing degree, and Barmpalias [**2**] showed that there is no minimal Δ^0_2 LR-degree (in fact, every Δ^0_2 LR-degree bounds a nonzero c.e. LR-degree).

4. LR-reducibility and bounded coverings

4.1. Characterisations of LR-reducibility in terms of bounded coverings. We now present in Theorem 1.5 some important characterisations of $\leq_{LR}$. These characterisations provide a concrete way for us to work with LR-reducibility, giving us a means to construct or diagonalise against possible LR-reductions.

THEOREM 1.5 (Kjos-Hanssen [**30**], Kjos-Hanssen, Miller and Solomon [**31**]). *Let $A, B \in 2^\omega$. The following are equivalent:*

(i) $A \leq_{LR} B$;

(ii) for every A-Σ^0_1 class W^A with $\mu(W^A) < 1$, there is a B-Σ^0_1 class V^B such that

$$W^A \subseteq V^B \text{ and } \mu(V^B) < 1;$$

(iii) for some member U of a universal oracle Martin-Löf test, there is a B-Σ^0_1 class V^B such that

$$U^A \subseteq V^B \text{ and } \mu(V^B) < 1;$$

(iv) for every A-c.e. set of strings W^A with weight $W^A < 1$, there is a B-c.e. set of strings V^B such that

$$W^A \subseteq V^B \text{ and weight } V^B < 1;$$

(v) for every A-c.e. set of strings W^A with weight $W^A < \infty$, there is a B-c.e. set of strings V^B such that

$$W^A \subseteq V^B \text{ and weight } V^B < \infty.$$

Parts (i),(ii),(iii) of Theorem 1.5 are due to Kjos-Hanssen [**30**]. Parts (iv) and (v) are due to Kjos-Hanssen, Miller and Solomon [**31**]. We give a proof of the theorem in section 4.3.

Theorem 1.5 provides a convenient way to work with LR-reductions. If we wish to construct a set A to be $\not\leq_{LR}$ a given set B, by (ii) it suffices to diagonalise against bounded B-Σ^0_1 classes by constructing an A-Σ^0_1 class T^A with $\mu(T^A) < 1$ and $T^A \not\subseteq V^B$ for any bounded oracle Σ^0_1 class V. Alternately, if we wish to ensure that $A \leq_{LR} B$, by (iii) it suffices to construct a B-Σ^0_1 class V^B with $\mu(V^B) < 1$ and $U^A \subseteq V^B$, for some universal oracle Martin-Löf test member U. Conditions (ii) and (iii) will be the primary tool for working with LR-reducibility in chapters 2, 3 and 4. In chapter 5 it will be convenient to work with LR-reducibility and low-for-randomness via (iv) and (v).

The difference between (ii) and (iv) is that W^A is considered as a set of reals in (ii) and as a set of strings in (iv). Clauses (iv) and (v) are in the spirit of Solovay's formulation of Martin-Löf randomness. A *Solovay test* is a c.e. set of strings S such that weight $S < \infty$. A real $X \in 2^\omega$ is captured by the test S if there are infinitely many strings $\sigma \in S$ with $\sigma \subset X$. Solovay [**70**] showed that X is Martin-Löf random iff X is not captured by any Solovay test. This holds when relativised to an arbitrary oracle.

4.2. Preparation for the proof of Theorem 1.5. We first give some lemmas which will be useful for the proof of Theorem 1.5 in section 4.3. To prove the equivalence of (i), (ii) and (iii) of Theorem 1.5 we will follow the presentation in Barmpalias, Lewis and Soskova [**5**]. Our proof of the equivalence of (ii), (iv) and (v) is adapted from Nies [**60**] Lemma 5.6.4, which in turn follows Simpson [**67**].

LEMMA 1.6. If a Π^0_1 class P contains a random, then P has positive measure.

PROOF. Suppose $\mu(P) = 0$. Let P_s be a computable sequence of clopen sets such that $P = \bigcap_s P_s$. In particular $\mu(P_s)$ is computable. For $i \in \mathbb{N}$ let k_i be the least such that $\mu(P_{k_i}) \leq 2^{-i}$. Then $(P_{k_i})_{i\in\mathbb{N}}$ is a Martin-Löf test which captures all reals in P. Therefore any real in P is not random. $\square$

In fact, if P contains a random then $\mu(P)$ is random, but we will not need this stronger result.

LEMMA 1.7. Let U be a member of a universal oracle Martin-Löf test and let $\sigma \in 2^{<\omega}$. If $[\sigma] \not\subseteq U^A$ then

$$\mu_\sigma(U^A) := \frac{\mu(U^A \cap [\sigma])}{\mu([\sigma])} < 1.$$

PROOF. Suppose $[\sigma] \not\subseteq U^A$. Then $[\sigma] - U^A$ is a nonempty A-Π^0_1 class containing a random. Hence it has positive measure by Lemma 1.6 relativised to A, and in particular $\mu_\sigma(U^A) < 1$. □

For sets of strings $X, Y \subseteq 2^{<\omega}$, let

$$XY := \{\sigma\tau : \sigma \in X, \tau \in Y\}. \tag{3}$$

If X, Y are c.e. in some oracle A then XY is also c.e. in A.

LEMMA 1.8. Let $A, B \in 2^\omega$, and suppose that X, Y are A-Σ^0_1 classes with measure < 1, V is a B-Σ^0_1 class with measure < 1 and $XY \subseteq V$. Then there is a B-Σ^0_1 class W with measure < 1 such that either $X \subseteq W$ or $Y \subseteq W$.

PROOF. Suppose first that there is a $\sigma \in X$ such that $\mu_\sigma(V) < 1$. Then for $W = \{\tau : \sigma\tau \in V\}$ we have $Y \subseteq W$ and $\mu(W) < 1$. Suppose next that $\mu_\sigma(V) = 1$ for all $\sigma \in X$. Let q be a rational > 0 such that $\mu(V) < 1 - q$. Let $W = \{\sigma : \mu_\sigma(V) > 1 - q\}$. We have $X \subseteq W$ and $\mu(W) \leq \mu(V) \cdot (1-q)^{-1} < 1$. □

A *tail* of a real X is any final segment of X.

LEMMA 1.9. If P is a Π^0_1 class of positive measure, then P contains a tail of every random.

PROOF. Let X be random and let S be a c.e. set of strings such that $[S] = 2^\omega - P$. Let q be a rational < 1 such that $\mu(S) < q$ and let k_i be the least such that $q^{k_i} \leq 2^{-i}$. Then S^{k_i} is a Martin-Löf test (where S^n is as in (3)), so $X \notin \cap_i S^{k_i}$. Thus $X \notin S^n$ for some n. Let n be the least such. Then $X \in S^{n-1}$ so $X \supset \tau$ for some string $\tau \in S^{n-1}$. Let Z be the tail of X obtained from X after discarding τ. $Z \notin S$ so $Z \in P$ as required. □

We will use this analysis fact in proving (iv) of Theorem 1.5.

LEMMA 1.10. Let $(a_i)_{i\in\mathbb{N}}$ be a sequence of real numbers such that $0 \leq a_i < 1$. Then

$$\sum_i a_i < \infty \Leftrightarrow \prod_i 1 - a_i > 0.$$

For a proof, see for instance Nies [**60**] Lemma 5.6.4.

4.3. Proof of Theorem 1.5.

Now we can give the proof of Theorem 1.5.

PROOF OF THEOREM 1.5. We will prove (i) $\Rightarrow$ (ii) $\Rightarrow$ (iii) $\Rightarrow$ (i) and then (ii) $\Rightarrow$ (v) $\Rightarrow$ (iv) $\Rightarrow$ (ii).

(i) $\Rightarrow$ (ii). Assume $A \leq_{LR} B$, that all B-randoms are A-random. Let W^A be an A-Σ^0_1 class with $\mu(W^A) < 1$. Fix a member $\tilde{U}$ of a universal oracle Martin-Löf test, and let U_i be the universal oracle Martin-Löf test from (2). We claim that

$$\exists m \in \mathbb{N}, \sigma \in 2^{<\omega} \text{ such that } [\sigma] \not\subseteq \tilde{U}^B \text{ and } U^A_m \cap [\sigma] \subseteq \tilde{U}^B. \tag{4}$$

Assume the contrary. We construct by finite extensions a real X which is in U^A_m for all m but $X \notin \tilde{U}^B$. Such an X is B-random but not A-random, contradicting $A \leq_{LR} B$. Let τ_{-1} be the empty string and assume inductively that τ_{i-1} is defined and $[\tau_{i-1}] \not\subseteq \tilde{U}^B$. By the negation of (4), $U^A_i \cap [\tau_{i-1}] \not\subseteq \tilde{U}^B$. Therefore there is a $\tau \supset \tau_{i-1}$ such that $[\tau] \subseteq U^A_i$ and $[\tau] \not\subseteq \tilde{U}^B$. Let τ_i be such a τ. Then the real $X = \cup_i \tau_i$ is B-random but not A-random. This establishes (4).

Let m and σ be as in (4), and let

$$\tilde{V}^B = (2^\omega - [\sigma]) \cup ([\sigma] \cap \tilde{U}^B).$$

Then $U^A_m \subseteq \tilde{V}^B$ and by Lemma 1.7, $\mu_\sigma(\tilde{U}^B) < 1$ and so $\mu(\tilde{V}^B) < 1$. (We have actually showed (iii) for the specific case when U is one component of the universal test (2).) Let q be a rational < 1 such that $\mu(W^A) \leq q$, and let k_i be the least such that $q^{k_i} \leq 2^{-i}$. Then $(W^A)^{k_i}$ gives an A-Martin-Löf test (where $(W^A)^n$ is as in (3)). By the definition (2) of the universal test U_i, we have $(W^A)^{k_i} \subseteq U^A_m \subseteq \tilde{V}^B$ for some i. By Lemma 1.8, there is a B-Σ^0_1 class V^B with $W^A \subseteq V^B$ and $\mu(V^B) < 1$.

(ii) $\Rightarrow$ (iii) is immediate.

(iii) $\Rightarrow$ (i). Assume that $U^A \subseteq V^B$ for some U and V as in (iii). Taking complements, we have $\overline{V^B} := 2^\omega - V^B \subseteq \overline{U^A} := 2^\omega - U^A$. Let X be B-random. Then $\overline{V^B}$ contains a tail Z of X by Lemma 1.9 relativised to B, since $\mu(\overline{V^B}) > 0$. So $\overline{U^A}$ contains Z also, and hence Z is A-random. But X is A-random iff Z is A-random, so X is A-random.

(ii) $\Rightarrow$ (v). We follow Nies **[60]** Lemma 5.6.4 in the special case where f is the weight function. Assume (ii). Let $g(\sigma) = \sum_{\tau<\sigma} |\tau|$ (where the ordering $<$ is the usual length-lexicographical ordering of strings). Let

$$E_\sigma = \{Z \subseteq \mathbb{N} : \exists i \in Z, g(\sigma) \leq i < g(\sigma) + |\sigma|\}.$$

E_σ is a clopen set and $\mu(E_\sigma) = 1 - 2^{-|\sigma|}$. It is easy to check that $\mu(E_\sigma \cap E_\tau) = \mu(E_\sigma)\mu(E_\tau)$ for $\sigma \neq \tau$, and more generally

$$\mu(\bigcap_{\sigma\in I} E_\sigma) = \prod_{\sigma\in I} \mu(E_\sigma) = \prod_{\sigma\in I} 1 - 2^{-|\sigma|}$$

for $I \subseteq 2^{<\omega}$.

Let W^A be a bounded A-c.e. set of strings, and let

$$P = \bigcap_{\sigma\in W^A} E_\sigma.$$

P is an A-Π^0_1 class and $\mu(P) > 0$ by Lemma 1.10 and the fact that W^A is bounded. By (ii) (taking complements), there is a B-Π^0_1 class Q such that $Q \subseteq P$ and $\mu(Q) > 0$. Let

$$V = \{\sigma : Q \subseteq E_\sigma\}.$$

V is a B-c.e. set, and $W^A \subseteq V$. Finally, weight $V = \sum_{\sigma\in V} 2^{-|\sigma|} < \infty$ by Lemma 1.10 because

$$\prod_{\sigma\in V} 1 - 2^{-|\sigma|} = \prod_{\sigma\in V} \mu(E_\sigma) = \mu(\bigcap_{\sigma\in V} E_\sigma) \geq \mu(Q) > 0.$$

(v) $\Rightarrow$ (iv). Assume (v). Let W^A be an A-c.e. set with weight $W^A < 1$. By (v), there is a B-c.e. set $\tilde{V}^B$ with weight $\tilde{V}^B < \infty$ and $W^A \subseteq \tilde{V}^B$. By discarding finitely many strings from $\tilde{V}^B - W^A$, we may obtain a B-c.e. set V^B with weight $V^B < 1$ and $W^A \subseteq V^B$.

(iv) $\Rightarrow$ (ii) is immediate since every A-Σ^0_1 class of measure < 1 is generated by a prefix-free A-c.e. set of weight < 1. ☐

5. Priority arguments and tree constructions

We assume familiarity with priority arguments, in the form of finite injury constructions and tree constructions. Tree constructions will be particularly important in chapters 2 and 4. Our notation will mostly follow Cooper [**15**]; further background on the priority method may be found in a standard reference such as Soare [**68**]. Formally, we can consider a tree T as a suitable subset of words $\Sigma^{<\omega}$ in an ordered alphabet Σ, and a node as a member of that set T. However, we will not specify our trees so formally. We prefer instead to define them inductively by specifying the types of nodes (or labels) of which the tree is comprised, listing the outcomes of each node type in a certain order, and describing how to assign labels to nodes according to their length.

We consider our trees to grow downwards. In particular, we have this in mind when we use terminology like 'above' and 'below'. Suppose that α, β are nodes on a tree. We say that α is above β if $\alpha \subset \beta$, and α is below β if $\alpha \supset \beta$. We use the notation $\alpha <_L \beta$ to denote that α is to the left of β in the lexicographical ordering (induced by the ordering of outcomes, or more formally by the ordering of the alphabet Σ). $\alpha < \beta$ denotes that $\alpha <_L \beta$ or $\alpha \subset \beta$, in which case we also say that α has *higher priority* than β. $\alpha \leq \beta$ has the obvious meaning of $\alpha < \beta$ or $\alpha = \beta$, and similarly for $\leq_L$. For a node α with $|\alpha| > 0$, $\alpha^- = \alpha \restriction |\alpha| - 1$ denotes the parent of α.

We will perform our constructions in stages. Typically, at the start of stage $s+1$ we are given the values possessed by parameters such as A, V^B, etc at the end of stage s, which we denote with the suffix $[s]$. Any enumerations, definitions, etc that we make during stage $s+1$ are in order to define $A[s+1]$, $V^B[s+1]$ etc.

6. The recursion theorem

The Recursion Theorem of Kleene [**32**] is a convenient tool for simplifying constructions. We will use it in Theorem 2.13 and in chapter 5 in the form of the Slowdown Lemma 5.4. Let $(\Phi_e)_{e\in\mathbb{N}}$ be a standard listing of all partial computable functions.

THEOREM 1.11 (Recursion Theorem, Kleene [**32**]). Let $f : \mathbb{N} \to \mathbb{N}$ be a total computable function. There is a fixed point $e \in \mathbb{N}$ such that

$$\Phi_e = \Phi_{f(e)}.$$

PROOF. Let g be a computable function such that

$$\Phi_{g(i)}(x) \simeq \Phi_{f(\Phi_i(i))}(x)$$

for all $x \in \mathbb{N}$. That is, $g(i)$ is the index of a partial computable function that first computes $\Phi_i(i)$, and if $\Phi_i(i)\downarrow$, then computes $\Phi_{f(\Phi_i(i))}(x)$. Let k be an index for the partial computable function g, ie. such that

$$g(x) \simeq \Phi_k(x).$$

Then

$$\Phi_{g(k)}(x) \simeq \Phi_{f(\Phi_k(k))}(x) = \Phi_{f(g(k))}(x).$$

Hence $e = g(k)$ is the required fixed point. □

Since c.e. sets are the domains of partial computable functions and vice versa, we may alternately phrase the Recursion Theorem in terms of the standard listing of c.e. sets $(W_e)_{e \in \mathbb{N}}$: if f is a computable function then there is a fixed point e such that

$$W_e = W_{f(e)}.$$

A small modification of the previous proof yields the following version of the Recursion Theorem.

THEOREM 1.12 (Recursion Theorem with parameters). Let $n \in \mathbb{N}$ and $f : \mathbb{N}^{n+1} \to \mathbb{N}$ be a total computable function. There is a computable function $k : \mathbb{N}^n \to \mathbb{N}$ such that

$$W_{k(y_1,\dots y_n)} = W_{f(k(y_1,\dots y_n),y_1,\dots y_n)}.$$

CHAPTER 2

Structural results in the c.e. and Δ_2^0 LR-degrees

In this chapter we present some results about structural properties of the c.e. and Δ_2^0 LR degrees. First we give a technique for lower cone avoidance in the c.e. and Δ_2^0 LR degrees, and combine this with upper cone avoidance via Sacks restraints to construct a c.e. set which is LR incomparable with a given Δ_2^0 set of intermediate LR degree.

Next we combine measure-guessing with an LR-incompleteness strategy to construct an LR-incomplete c.e. set which is LR-above a given low Δ_2^0 set. This is in contrast to the Turing degrees, in which there is a Δ_2^0 degree which is Turing incomparable with all intermediate c.e. degrees.

Finally we discuss how to combine the techniques of the first two theorems in the c.e. case, and the obstacles to combining them in the more general case of Δ_2^0 sets.

1. Working with an LR-incomplete set

We outline a technique for working with an LR-incomplete c.e. or Δ_2^0 set in full-approximation constructions. The technique is a method for leveraging the LR-incompleteness of a set A to limit the changes in the approximation of A, effectively imposing 'restraints' on A which can be utilised by other requirements in a construction.

Suppose that the c.e. set A is LR-incomplete and F^A is an A-Σ_1^0 class; then by Theorem 1.5 we have

$$U^{\emptyset'} \subseteq F^A \Rightarrow \mu(F^A) = 1$$

for a member U of a universal oracle ML-test. If we attempt to trace $U^{\emptyset'}$ into an A-Σ_1^0 class F^A, then we are guaranteed that $\emptyset'$ will change more frequently than A, frequently enough to ensure $\mu(F^A) = 1$. We can use this to our advantage to provide restraints on A. Suppose that during a construction we wish to restrain $A \restriction u$ at stage s. We can take a string ρ from $U^{\emptyset'}[s]$ which is not yet in F^A, and enumerate ρ into F^A with use u. Then we wait for a $\emptyset'$-change below the use of the computation $\rho \in U^{\emptyset'}[s]$. If the $\emptyset'$-change never occurs, then we never proceed further with this attempt, and the restraint is unsuccessful; we say that the attempt is *stalled*. However in this case we have that $\rho \in U^{\emptyset'}$ permanently; this can happen for at most $\mu(U^{\emptyset'})$ worth of strings ρ, and this can be made as small as necessary by choosing a suitably small U. Otherwise, a $\emptyset'$-change eventually occurs. At this point, we have $\rho \in F^A$ but $\rho \notin U^{\emptyset'}$. If A later changes below u, then the attempt at restraining $A \upharpoonright u$ is unsuccessful. However we can argue that sufficiently many attempts will be successful (ie A will not change below u after the $\emptyset'$-change) to ensure that $\mu(F^A) = 1$.

If our requirement is such that it requires a finite measure worth of restraints for satisfaction, then we can argue that it will be satisfied with the above method. Suppose that it is not satisfied. We will make infinitely many attempts at restraining $A \upharpoonright u$ for some u, each attempt corresponding to a string from $U^{\emptyset'}[s]$. Infinitely many will correspond to the true strings of $U^{\emptyset'}$. Since a $\emptyset'$-change never occurs for these attempts, we trace $U^{\emptyset'}$ into F^A. But then, by the LR-incompleteness of A, we are assured that measure $1-\mu(U^{\emptyset'})$ worth of attempts will succeed, providing enough restraint to satisfy the requirement.

We can think of this technique in the following way. We are given an approximation $A[s]$ of A such that $A = \lim_s A[s]$. Each time we take a string ρ from $U^{\emptyset'}$ and put it into $F^A[s]$ with some use $\tau = A[s] \upharpoonright u$, we are requesting that $A \supset \tau$ in the limit; the measure of ρ is the strength of the request. If in fact $\tau \subset A$, then the request is successful. Since we will threaten to make $U^{\emptyset'} \subseteq F^A$, the LR-incompleteness of A guarantees that enough requests will be successful to ensure that $\mu(F^A) = 1$. Whether a request is successful or not depends only on whether $\tau \subset A$ in the limit. It does not matter (as far as this basic strategy is concerned) whether A is approximated in a c.e. or Δ^0_2 way. Thus the technique can be used with both c.e. and Δ^0_2 sets A.

We will first use this technique in a lower cone avoidance strategy, as part of the construction of a c.e. set B that is LR-incomparable to a given LR-intermediate Δ^0_2 set A. In section 3.5 we sketch another use of the technique in combination with the LR-incompleteness strategy of section 3.

2. A c.e. LR-degree incomparable with a given intermediate Δ^0_2 LR-degree

Barmpalias, Lewis and Stephan [6] use an oracle construction to construct a Δ^0_2 set B that is LR-incomparable with a given Δ^0_2 set A of intermediate LR degree. We strengthen this result to make B c.e., using a full approximation construction.

The analogous theorem for the Turing degrees, namely that for every Δ_2^0 set A of intermediate Turing degree there is (uniformly in A) a c.e. set B Turing incomparable with A, was proved by Sacks [**65**] using a coding strategy combined with Sacks restraints. A presentation may be found in Odifreddi [**62**] XI.3.7.

THEOREM 2.1. Let A be a Δ_2^0 set such that $\emptyset <_{LR} A <_{LR} \emptyset'$. There is (uniformly in A) a c.e. set B such that $A|_{LR}B$.

Let A be a Δ_2^0 set such that $\emptyset <_{LR} A <_{LR} \emptyset'$, given as a computable approximation $A[s]$ such that

$$\lim_s A(x)[s] = A(x) \quad \text{for all } x.$$

Let $\langle V_e, p_e \rangle$ be a listing of all LR-operators, that is, pairs $\langle V, p \rangle$ of an oracle Σ_1^0 class V and a dyadic rational $p \in (0,1)$ such that

$$\mu(V^X) \leq p \text{ for all } X \in 2^\omega.$$

Let U be a fixed member of a universal oracle Martin-Löf test. We will construct the required c.e. set B, as well as an oracle Σ_1^0 class T, to satisfy the requirements

$$P_e: \qquad T^B \not\subseteq V_e^A$$

$$S_e: \qquad U^A \not\subseteq V_e^B$$

for all $e \in \mathbb{N}$. In fact, we will uniformly build a sequence $T_{\alpha,i}$ of oracle Σ_1^0 classes, where $i \in \mathbb{N}$ and α ranges over nodes of the tree of strategies defined later. We can set $T^B = \cup_{\alpha,i} T_{\alpha,i}^B$ which is a B-Σ_1^0 class. We will ensure that

$$\sum_{\alpha,i} \mu(T_{\alpha,i}^B) < 2^{-1},$$

so by Theorem 1.5, the requirements P_e ensure that $B \not\leq_{LR} A$. We will use a strategy based on the technique outlined at the start of the chapter to meet the P requirements; for the S requirements we will use a variation of Sacks restraints adapted for LR degrees, first used by Barmpalias, Lewis and Soskova [**5**].

2.1. Outline of the S-strategy. We use Sacks restraints, adapted to LR-reductions. Sacks restraints were first used by Sacks [**65**] in the context of c.e. Turing degrees (see Soare [**68**] for the standard presentation). The technique was adapted to LR-degrees by Barmpalias, Lewis and Soskova [**5**].

In the Turing case, suppose we are building B and want to ensure that $\Phi^B \neq A$ for a Turing functional Φ and a noncomputable c.e. or Δ_2^0 set A. We can monitor the length of agreement of $\Phi^B = A$, and whenever we see a new computation $\Phi^B(n) = A(n)$ converge, we restrain B on the use u of $\Phi^B(n)$ to preserve that part of the computation. If the restraint is respected and B does not change below u after the restraint is imposed, then we can argue that $\Phi^B \neq A$. If $\Phi^B = A$, then we would be able to compute $A(n)$ by finding a stage in the construction when the length of agreement is above n; at this stage the Φ^B-side of the computation will never change so the approximation to $A(n)$ must be correct. Thus A would be computable, which is a contradiction.

In the case of LR-reductions, we have a fixed member U^A of a universal Martin-Löf test relative to A, and a bounded B-Σ_1^0 class V^B. We want to ensure that $U^A \not\subseteq V^B$. Suppose at some stage we see a string $\sigma \in U^A$ and $\sigma \subseteq V^B$ with use u. We can restrain B up to u in order to preserve the computation $\sigma \subseteq V^B$. Assuming that the restraint is respected, we can also enumerate the string σ into a Σ_1^0 class G. If $U^A \subseteq V^B$, then we will eventually do this for every string $\sigma \in U^A$. Thus $U^A \subseteq G$. But since the B-restraint is respected, each string in G is also in V^B, so $G \subseteq V^B$. Since V^B has measure < 1, so does G. But then U^A is contained in a Σ_1^0 class of bounded measure, which would mean that A is low-for-random, a contradiction. So eventually there must be some string $\sigma \in U^A$ but $\sigma \not\subseteq V^B$, and we succeed in diagonalising against V.

2.2. Outline of the P-strategy. Formal details of the P-strategy are given later. We omit the subscript e in the following discussion. We will diagonalise against the LR-operator $\langle V, p\rangle$ by putting a clopen set σ into T^B, waiting for $\sigma \subseteq V^A[s]$, and then removing σ from T^B by enumerating into B if we see $\sigma \subseteq V^A[s]$. If σ is never $\subseteq V^A[s]$ then P is satisfied and the requirement contributes at most $\mu(\sigma)$ to $\mu(T^B)$. If eventually $\sigma \subseteq V^A[s]$ and A does not later change below the use of the computation $\sigma \subseteq V^A[s]$, then $\mu(V^A)$ increases permanently by $\mu(\sigma)$ but $\mu(T^B)$ does not increase. With suitable choice of $\mu(\sigma)$, requirement P must be satisfied after finitely many repetitions of this strategy as $\mu(V^A)$ cannot increase above p.

If however A does later change below the use of $\sigma \subseteq V^A[s]$, then σ may no longer be $\subseteq V^A$ and the attack σ is unsuccessful. We can use the method described above using the LR-incompleteness of A to impose restraints on A and guarantee that sufficiently many attacks will be successful to satisfy P. Since $A \not\geq_{LR} \emptyset'$, if $U^{\emptyset'} \subseteq F^A$ for some A-c.e. class F^A and member U of a universal oracle Martin-Löf test, then $\mu(F^A) = 1$. When we want to schedule an attack at stage s, we take a string $\rho \in U^{\emptyset'}[s]$ and choose σ with $\mu(\sigma) = \mu(\rho)$. When the attack σ appears successful because $\sigma \subseteq V^A[t]$, we would like to restrain A on the use u of σ. We put ρ into an A-c.e. class F^A with the same use u, then we wait for a $\emptyset'$-change to remove ρ from $U^{\emptyset'}$. If this $\emptyset'$-change never occurs, then $\rho \in U^{\emptyset'}$ permanently and the attack σ is considered unsuccessful; however this can happen for at most $\mu(U^{\emptyset'})$ worth of attacks. Otherwise, $\emptyset'$ eventually changes and ρ is removed from $U^{\emptyset'}$. Then we can remove σ from T^B via a B-change. If later an A-change removes σ from V^A, then the attack σ is unsuccessful. However, since we threaten to trace $U^{\emptyset'}$ into F^A, we are guaranteed that enough ρ's will be permanently in F^A, and hence σ's permanently in V^A, to ensure that $\mu(F^A) = 1$. Since each $\rho \in F^A$ corresponds to an attack σ of the same measure in V^A, we can argue that the P requirement must eventually be satisfied as $\mu(V^A) = 1$ is impossible.

Each attack σ is tied to a computation $\rho \in U^{\emptyset'}[s]$, and the outcome of the attack σ depends on the outcome of the computation $\rho \in U^{\emptyset'}[s]$. If the computation $\rho \in U^{\emptyset'}[s]$ is not permanent, then the attack σ will be removed permanently from T^B and will either succeed or fail, depending on whether $\rho \in F^A$ (and $\sigma \subseteq V^A$). If the computation $\rho \in U^{\emptyset'}[s]$ is permanent, then σ will be permanently in T^B. The attack σ may end up permanently pending (if $\sigma \subseteq V^A$), or permanently waiting (if σ is never $\subseteq V^A$). However we also might have $\sigma \subseteq V^A[s]$ at infinitely many s but $\sigma \not\subseteq V^A$ in the limit; in this case, σ may rotate infinitely often between waiting and pending. In this case σ is permanently in T^B but $\sigma \not\subseteq V^A$ so the P-requirement is satisfied, and σ itself does not cause any enumerations into B. However, during the stages when σ is pending, we will schedule other attacks for P. Each time σ changes from pending to waiting, any attacks scheduled after σ must be removed from T^B. So although σ itself will not cause any B-enumerations, attacks scheduled during σ's pending periods might cause infinitely many B-enumerations, which could conflict with the B-restraints of weaker-priority S-requirements. This conflict is resolved by having the P-strategy play an infinitary outcome each time σ moves between pending and waiting; the S-strategies below the infinitary outcome will only believe a computation if its use is below that of any attacks scheduled after σ.

We will argue that if P is not satisfied then $U^{\emptyset'} \subseteq F^A$, F^A has measure 1, and so V^A must have measure 1 also since every string in F^A corresponds to a successful attack in V^A. For this argument to work, we require that successful attacks are disjoint. This is slightly complicated by the Δ_2^0 approximation of A. At a stage s_0 we may have an attack σ which is succeeding, ie $\sigma \subseteq V^A[s_0]$ and its corresponding string $\rho \in F^A[s_0]$. At $s_1 > s_0$ the attack σ might be failing ($\sigma \not\subseteq V^A[s_1]$ and $\rho \notin F^A[s_1]$) because the approximation to A has changed. We might now schedule a new attack σ', which might overlap (as a clopen set) with σ. However, later on at $s_2 > s_1$ the approximation to A might change back to its state at s_0, and σ would become succeeding again. If we keep working with σ', we risk having two non-disjoint attacks. To account for this, when we create a new attack σ' we record the state of all earlier attacks, in the form of a suitable initial segment γ of $A[s]$. We choose $|\gamma|$ longer than the use of any computations relevant to earlier attacks σ which might later become succeeding if A reverts back to an earlier approximation. We will only work with σ' at stages when $\gamma \subset A[s]$. We call γ the *state* of the attack σ'. Note that this is not necessary if A is in fact c.e., since a c.e. approximation cannot revert to a previous state.

Since we want to illustrate the technique outlined in section 1 in generality, the form of the P-strategy we are using here is slightly more general than is necessary for this specific construction. To meet the P-requirements of this theorem we could trace strings from $U^{\emptyset'}$ directly into T^B, rather than via an intermediate set F^A. We could thus slightly simplify the notation by eliminating the class F^A. However the notion of state and the infinitary outcomes would still be necessary. In order to establish notation that is suitable for more general applications of this technique, such as in the discussion of section 3.5, we will not make this simplification in this construction.

2.3. The priority tree. The construction takes place on an $\omega+1$ branching tree, with nodes labelled either $P_{e,i}$ or S_e for some $e, i \in \mathbb{N}$. Nodes are labelled according to their length: if $|\alpha| = 2e$ then α is labelled S_e and has a single outcome 0 (that is, there is a single child node $\alpha^\frown 0$ extending α on the tree). If $|\alpha| = 2\langle e, i\rangle + 1$ then α is labelled $P_{e,i}$ and has $\omega+1$ outcomes (children)

$$\infty(0) <_L \infty(1) <_L \infty(2) <_L \dots <_L f.$$

The ordering of outcomes $<_L$ induces an ordering on the tree; for nodes α, β, $\alpha <_L \beta$ indicates that α is to the left of β, and $\alpha < \beta$ indicates that $\alpha <_L \beta$ or $\alpha \subset \beta$. We refer to nodes labelled $P_{e,i}$ for some e, i as *P-nodes*, or as P_e-nodes if the index e is significant, and similarly for S-nodes.

During the construction we will define *approximations to the true path* TP_s, which are the nodes of the tree which are active at stage s. Say that s is an *α-stage*, or alternatively that α is *accessible* at s, if $\alpha \subseteq \mathrm{TP}_s$.

2.4. P-requirements. Each P-node pursues an independent copy of the P-strategy. Fix a computable listing $\alpha_0, \alpha_1, \dots$ of all P-nodes on the tree. Recall that p_e is an upper bound on the measure $\mu(V_e^X)$ of the e'th LR-operator. If α is a P_e-node (for some $e \in \mathbb{N}$), let n be its position in the above ordering. Let m_α be the least number m such that $2^{-m} < 2^{-n-2} \cdot p_e$. Each P-node α has a counter $c(\alpha, s)$ which is the number of times that node α has been reset by the end of stage s. At any stage, α works with oracle Σ^0_1-classes T_α and F_α, and a member $U^{\emptyset'}_\alpha$ of a universal Martin-Löf test relative to the halting problem. Actually, α works with a sequence of oracle Σ^0_1-classes $T_{\alpha,i}$ and Martin-Löf test members $U^{\emptyset'}_{\alpha,i}$, $i \in \mathbb{N}$. Each time α is reset it empties F_α, abandons the previous $T_{\alpha,i}, U^{\emptyset'}_{\alpha,i}$ and starts working with $T_{\alpha,i+1}, U^{\emptyset'}_{\alpha,i+1}$ instead. To be precise, at stage s, α will work with $T_{\alpha,c(\alpha,s)}$ and $U^{\emptyset'}_{\alpha,c(\alpha,s)}$, where $U_{\alpha,i}$ is the $m_\alpha + i + 1$'th member of the universal oracle Martin-Löf test. For brevity we write T_α and U_α to refer to the appropriate class $T_{\alpha,i}, U_{\alpha,i}$ which is in use at the time; T_α and U_α may be considered pointers to $T_{\alpha,c(\alpha,s)}$ and $U_{\alpha,c(\alpha,s)}$ at each stage s. If α is reset only finitely often, then T_α, U_α are eventually fixed. We will ensure that $\mu(T^X_{\alpha,i}[s]) \leq 2^{-m_\alpha - i - 1}$ for all $X \in 2^\omega$ and all s, so that, setting $T = \bigcup_\alpha \bigcup_i T_{\alpha,i}$, we have

$$\mu(T^B) \leq \sum_\alpha \sum_i 2^{-m_\alpha - i - 1} \leq \sum_\alpha 2^{-m_\alpha} \leq \frac{1}{2}, \tag{5}$$

where α ranges over all P-nodes.

Let α be a P_e-node. The node α will attempt to meet its requirement by putting certain clopen sets of reals (*attacks*) σ into T^B_α, waiting until $T^B_\alpha \subseteq V^A_\alpha$, then removing the clopen set σ from T^B_α and attempting to restrain A to keep $\sigma \subseteq V^A_\alpha$. Each time, α causes the measure of V^A_α to increase by $\mu(\sigma)$, while $\mu(T^B_\alpha)$ does not increase. Since $\mu(V^A_\alpha)$ is bounded by $p_\alpha < 1$, this can only happen finitely often (with a suitable choice of $\mu(\sigma)$) before some attack satisfies the requirement because $\sigma \not\subseteq V^A_\alpha$.

An *attack* σ is a finite prefix-free set of strings (representing a clopen set of reals) which we treat as a single unit. It is possible that two attacks may be created at different times in the construction which have the same set of strings, but in this case we consider them to be distinct attacks (formally, we may consider an attack as the finite set of strings along with the stage at which it is scheduled, though we will not do this explicitly). When we put an attack into $T^B_\alpha[s]$ we put each string from σ into $T^B_\alpha[s]$ with the same use; we write $\sigma \in T^B_\alpha[s]$ to mean that each string in σ is in $T^B_\alpha[s]$ with the same use. At most one attack is scheduled (created) at each stage. If an attack σ is scheduled by node α then we say that σ is an α-attack. The lifecycle of an attack is as follows. It is *scheduled* at some stage t. At future stages it is either *current* or not current; if it is current then it is either *waiting* for V^A_α, *pending* $\emptyset'$-permission, *succeeding* or *failing*. These terms are defined later.

We consider each $U_{\alpha,i}$ as a c.e. set of axioms. An axiom is a pair $\langle \rho, \tau \rangle$ asserting that $\rho \in U^X$ if $\tau \subset X$. Since we are working with $U^{\emptyset'}$ as we approximate $\emptyset'$, we are only interested in those axioms such that $\tau \subseteq \emptyset'[s]$ at some stage. An axiom $\langle \rho, \tau \rangle$ is *valid* at stage s if the tuple $\langle \rho, \tau \rangle$ has been enumerated into U by stage s and $\tau \subset \emptyset'[s]$. Fix α, i and let

$$\langle \rho_0, \tau_0 \rangle, \langle \rho_1, \tau_1 \rangle \ldots \tag{6}$$

be a list of the computations (axioms) from $U_{\alpha,i}$ which are valid at some stage, ordered first by the least stage at which they are valid and then by the usual length/lexicographical ordering on the ρ_j. When an α-attack is created, it is associated with one of these computations. Suppose that attack σ is associated with axiom $\langle \rho_k, \tau_k \rangle$. We write $\mathrm{rank}(\sigma)$ to denote the position k of the axiom in the above list, $\rho(\sigma)$ to denote the string ρ_k, $\hat{\rho}(\sigma)$ to denote the computation $\langle \rho_k, \tau_k \rangle$, and $u(\sigma)$ to denote $|\tau_k|$.

Attack σ also has a *state* $\gamma(\sigma)$, defined when σ is scheduled, which is an initial segment of A at the stage when σ is scheduled. The attack σ is *current* at a later stage $s+1$ if $\gamma(\sigma) \subset A[s]$; at any stage, we only work with attacks which are current.

When we wish to restrain a computation $\sigma \subseteq V^A_\alpha[s]$, we will put the string $\rho(\sigma)$ into $F^A_\alpha[s+1]$ by defining a new computation with use v larger than that of the computation $\sigma \subseteq V^A_\alpha[s]$. We say that the axiom $\langle \rho, A[s] \upharpoonright v \rangle$ is in F_α *on account of* σ. Every axiom in F_α is on account of some attack. Two distinct axioms $\langle \rho, \tau \rangle, \langle \rho', \tau' \rangle \in F_\alpha$ may be on account of different attacks even if $\rho = \rho'$ (although then $\tau \neq \tau'$). We say that $\rho \in F_\alpha^A[s]$ on account of σ if $\rho \in F_\alpha^A[s]$ due to an axiom $\langle \rho, \tau \rangle \in F_\alpha[s]$ on account of σ.

Schedule an α-attack at stage $t+1$ by taking the least k (if it exists) such that $\langle \rho_k, \tau_k \rangle$ is valid at $t+1$ and $\rho_k \notin F_\alpha^A[t]$, and choosing the least clopen set $\sigma \subseteq 2^\omega - V_\alpha^A[t]$ with $\mu(\sigma) = \mu(\rho_k)$. Set $\rho(\sigma) = \rho_k$, $\hat{\rho}(\sigma) = \langle \rho_k, \tau_k \rangle$, $u(\sigma) = |\tau_k|$ and $\mathrm{rank}(\sigma) = k$. Let w be the maximum use of any computation $\langle \rho', \tau' \rangle \in F_\alpha[t]$ on account of any attack σ' with $\mathrm{rank}(\sigma') < k$, and define the state $\gamma(\sigma)$ to be $A[t] \upharpoonright w$. Put σ into $T_\alpha^B[t+1]$ with fresh use. Note that, when scheduling an attack, a suitable choice for σ will always exist since

$$\mu(\rho_k) \leq \mu(U_\alpha^{\emptyset'}[t]) \leq 1 - p_\alpha \leq \mu(2^\omega - V_\alpha^A[t])$$

by choice of m_α. It is possible that sometimes k will not exist (if $U_\alpha^{\emptyset'}[t] \subseteq F_\alpha^A[t]$). In this case, do nothing; no attack is scheduled at $t+1$.

At certain stages $s+1$ we will *implement* σ by putting $\rho(\sigma)$ into $F_\alpha^A[s+1]$ with some use v; that is, enumerating a new axiom $\langle \rho(\sigma), A[s] \upharpoonright v \rangle$ into $F_\alpha[s+1]$. We declare that the new computation is on account of σ.

The attack σ which was scheduled at $t+1$ is *failing* at $s+1 > t+1$ if it is current at $s+1$, $\emptyset'[s] \upharpoonright u(\sigma) \neq \emptyset'[t] \upharpoonright u(\sigma)$ and $\rho(\sigma)$ is not in $F_\alpha^A[s]$ on account of σ. σ is *succeeding* at $s+1$ if it is current at $s+1$, $\emptyset'[s] \upharpoonright u(\sigma) \neq \emptyset'[t] \upharpoonright u(\sigma)$ and $\rho(\sigma)$ is in $F_\alpha^A[s]$ on account of σ.

Attack σ is *waiting* at stage $s+1$ if it is current at $s+1$, $\emptyset'[s] \upharpoonright u(\sigma) = \emptyset'[t] \upharpoonright u(\sigma)$, but $\rho(\sigma)$ is not in $F_\alpha^A[s]$ on account of σ. This is the case when $\sigma \not\subseteq V_\alpha^A[s]$. σ is *pending* at stage $s+1$ if it is current at $s+1$, $\emptyset'[s] \upharpoonright u(\sigma) = \emptyset'[t] \upharpoonright u(\sigma)$ and $\rho(\sigma)$ is in $F_\alpha^A[s]$ on account of σ. In this case we are waiting for a $\emptyset'$-change before we remove σ from T_α^B.

It is possible that for some σ we may have $\sigma \subseteq V_\alpha^A[s]$ for infinitely many s but $\sigma \not\subseteq V_\alpha$ in the limit. Such a σ may be implemented infinitely often, with $A[s]$ always changing below the use of the new F_α computation. In this case, $\sigma \in T_\alpha^B$ permanently but $\sigma \not\subseteq V_\alpha^A$ so the requirement P_α is satisfied. Although σ itself will not cause infinitely many enumerations into B, attacks of rank $> \mathrm{rank}(\sigma)$ may cause infinitely many B-enumerations since any such attack will not be permanently current and will eventually need to be removed from T_α^B. To allow the lower-priority negative requirements to work with these potentially infinitary enumerations, each time attack σ is implemented we access an infinitary outcome $\alpha^\frown\infty(k)$ for $k = \mathrm{rank}(\sigma)$. The negative requirements below this outcome will only believe a computation if the use of the computation is less than that of any α-attack of rank $> k$ which is in $T^B_\alpha[s]$.

Let β be an S-node below some infinitary outcome $\alpha^\frown\infty(n)$. The node β will only believe a computation if the use of the computation is less than the use of any α-attack σ which is currently in $T^B_\alpha[s]$ and has $\mathrm{rank}(\sigma) > n$. Precisely, suppose that β is an S-node and $\tau \subseteq V^B[s]$ with use u for some clopen set τ and oracle Σ^0_1-class V. The computation $\tau \subseteq V^B[s]$ is *β-believable* if there does not exist a P-node α such that

- $\alpha^\frown\infty(n) \subseteq \beta$ for some n, and
- there is an α-attack σ with $\mathrm{rank}(\sigma) > n$ and $\sigma \in T^B_\alpha[s]$ with use $\leq u$.

If in fact $\tau \subseteq V^B$ and β is on the true path, then eventually the computation will be β-believable.

We observe that the construction can be considerably simplified in the case that the set A is low ($A' \equiv_T \emptyset'$). In this case we can avoid the infinitary outcomes by a technique similar to Robinson guessing (see Soare [**68**] §XI.3). We can take a computable function $h : \mathbb{N} \times \mathbb{N} \to \{0, 1\}$ such that

$$\lim_s h(\sigma, e, s) = \begin{cases} 1 & \text{if } \sigma \subseteq V^A_e \\ 0 & \text{otherwise} \end{cases}$$

and believe a computation $\sigma \subseteq V^A_e[s]$ only if $h(\sigma, e, s) = 1$. In this case, if $\sigma \subseteq V^A_e[s]$ at infinitely many s but not in the limit, then σ will only be implemented finitely often before $h(\sigma, e, s)$ reaches its limiting value of 0. Hence the infinitary outcomes are not necessary, and we could use the strategy in a finite injury construction.

2.5. S-requirements. Let α be an S-node. Define the *length of agreement*

$$l(\alpha, s) = \max\Big\{i : \langle\rho_i, \tau_i\rangle \in U[s] \text{ and } \forall j \leq i(\rho_i \subseteq V^B_\alpha[s] \text{ by an } \alpha\text{-believable computation, or } \rho_i \notin U^A[s])\Big\}.$$

Since A is Δ^0_2, we also need the *modified length of agreement*

$$m(\alpha, s) = \max\Big\{i : \exists\alpha\text{-stage } t \leq s\Big(i \leq l(\alpha, t) \ \wedge\ B[s] \upharpoonright u = B[t] \upharpoonright u\Big)\Big\}$$

where u is the maximum use of all computations $\rho_j \subseteq V^B_\alpha[t]$ for $j \leq i$ with $\rho_j \in U^A[t]$.

Each S-node has a restraint r_α which is initially 0 and is set explicitly during the construction. Let $R(\alpha, s) = \max\{r_\beta[s] : \beta < \alpha\}$ be the total restraint imposed by nodes of higher priority than α. To *reset* an S-node α at stage s means to set $r_\alpha[s] = 0$.

2.6. Some conventions. We assume that when a string is put into any $T^B_{\alpha,i}$ with some use u, it remains there until the number u is explicitly enumerated into B. In particular, the string remains in $T^B_{\alpha,i}$ even if numbers $< u$ enter B.

We use the 'hat-trick' for the enumeration of U^A. Let $a_0 = 0$ and for $s > 0$ let a_s be the least number such that $A(a_s)[s] \neq A(a_s)[s-1]$, or $a_s = s$ if such number does not exist. Let

$$\widehat{U^A}[s] := U^{A \restriction a_s}[s] = \{\sigma : \sigma \text{ is in } U^A[s] \text{ with use} \leq a_s\}.$$

Henceforth we omit the hat and write $U^A[s]$ to mean $\widehat{U^A}[s]$. The hat-trick ensures that $\sigma \in U^A[s]$ for all but finitely many s iff $\sigma \in U^A$.

2.7. The construction. Initially, B and all classes $T_{\alpha,i}, F_\alpha$ are empty, and r_α is zero for all S-nodes α.

At stage 0, do nothing.

At stage $s+1$, define TP_{s+1} inductively as below. After TP_{s+1} is defined, reset all nodes $\beta >_L \mathrm{TP}_{s+1}$.

Suppose that $\mathrm{TP}_{s+1} \restriction n$ is defined, for $n \geq 0$. If $n = s+1$ then stop defining TP_{s+1}. Otherwise let $\alpha = \mathrm{TP}_{s+1} \restriction n$ and go to the appropriate case below.

• **α is a P-node.** Let s' be the previous stage when α was accessible, or 0 if never. Check if any of the following hold:

(I) there is an α-attack σ such that σ is in $T^B_\alpha[s]$ but σ is not current at $s+1$;
(II) there is a current α-attack σ that is failing or succeeding at $s+1$ and $\sigma \in T^B_\alpha[s]$;
(III) there is a current α-attack σ that is pending or waiting at $s+1$ but $\sigma \notin T^B_\alpha[s]$;
(IV) there is a current α-attack σ which is waiting at $s+1$ and $\sigma \subseteq V^A_\alpha[s]$ (as sets of reals);
(V) no current α-attacks are waiting at $s+1$.

Go to the least case below which holds.

(I) or (II) hold. For each such σ, remove σ from $T_\alpha^B[s+1]$ by enumerating the use of the computation $\sigma \in T_\alpha^B[s]$ into $B[s+1]$. Let k be the minimum rank of all such σ, and reset $\alpha^\frown\infty(k)$ and all nodes of lower priority. Stop defining TP_{s+1}.

(III) holds. For each such σ, add σ to $T_\alpha^B[s+1]$ with fresh use. Stop defining TP_{s+1}.

(IV) holds. For the least such σ, let v be the maximum of $|\gamma(\sigma)|$ and the use of the computation $\sigma \subseteq V_\alpha^A[s]$, and implement σ by defining $\rho(\sigma) \in F_\alpha^A[s+1]$ with use v. Let $k = \mathrm{rank}(\sigma)$ and let $\mathrm{TP}_{s+1} \restriction n+1 = \alpha^\frown\infty(k)$.

(V) holds. Schedule a new α-attack at $s+1$. Stop defining TP_{s+1}.

None of (I)-(V) hold. Let $\mathrm{TP}_{s+1} \restriction n+1 = \alpha^\frown f$.

- **α is an S-node.** Let r be the maximum use of all computations $\rho_j \subseteq V_\alpha^B[s]$ for $j \leq m(\alpha, s)$ with $\rho_j \in U^A[s]$. If $r > r_\alpha[s]$ then set $r_\alpha[s+1] = r$, reset all nodes of lower priority than α and stop defining TP_{s+1}. Otherwise let $\mathrm{TP}_{s+1} = \alpha^\frown 0$.

End of construction.

2.8. Verification. First we deal with the P-requirements. We give some lemmas to clarify the relations between attacks.

LEMMA 2.2. Let α be a P-node. If σ is an α-attack such that $\mathrm{rank}(\sigma) = k$ and σ is current at stage $s+1$, then for every $j < k$ such that the j'th computation from the list (6) is valid at $s+1$ there is an α-attack σ' with $\rho(\sigma') = \rho_j$ and $\mathrm{rank}(\sigma') \leq j$ which is current at $s+1$.

PROOF. Suppose that there is a $j < k$ such that the j'th computation $\langle \rho_j, \tau_j \rangle$ is valid at $s+1$ but there is no current attack with rank j. Let $t+1 < s+1$ be the stage when σ was scheduled. Since $\mathrm{rank}(\sigma) > j$, the computation $\langle \rho_j, \tau_j \rangle$ is valid at $t+1$, and we must have $\rho_j \in F_\alpha^A[t]$. But then there must be a current α-attack σ' pending at $t+1$ with $\rho(\sigma') = \rho_j$ and $\mathrm{rank}(\sigma') \leq j$. But $\gamma(\sigma') \subseteq \gamma(\sigma)$, and σ is current at $s+1$, so σ' must also be current at $s+1$. □

The next lemma states that no two attacks associated with the same computation from $U_\alpha^{\emptyset'}$ are simultaneously current, and that any two attacks which both have strings in $F_\alpha^A[s]$ are disjoint.

LEMMA 2.3. For any P-node α and any stage s, if distinct α-attacks σ, σ' are both current at s then $\text{rank}(\sigma) \neq \text{rank}(\sigma')$. If $\rho(\sigma), \rho(\sigma')$ are both in $F_\alpha^A[s]$ on account of σ, σ' respectively, then $\rho(\sigma) \neq \rho(\sigma')$ and $\sigma \cap \sigma' = \emptyset$ (as sets of reals).

PROOF. Let σ, σ' be as in the claim. Assume for contradiction that $\text{rank}(\sigma) = \text{rank}(\sigma')$, and let $t+1, t'+1$ be the stages when σ, σ' respectively are scheduled. Suppose w.l.o.g. that $t < t'$. Then $\gamma(\sigma) \subseteq \gamma(\sigma') \subset A[t']$, so σ must be current at $t'+1$, and is either waiting, pending, failing or succeeding. σ cannot be waiting at $t'+1$ or the attack σ' would not be scheduled. Nor can σ be failing or suceeding, as the computation $\hat{\rho}(\sigma)$ $(= \hat{\rho}(\sigma'))$ is valid at $t'+1$. Finally if σ is pending at $t'+1$ then $\rho(\sigma) \in F_\alpha^A[t']$, and σ' would be scheduled with $\rho(\sigma') \neq \rho(\sigma)$. So $\text{rank}(\sigma) \neq \text{rank}(\sigma')$.

Suppose that $\rho(\sigma), \rho(\sigma')$ are in $F_\alpha^A[s]$ on account of σ, σ'. Then both σ, σ' are current at $s+1$ by choice of the use of the F_α computations, so $\text{rank}(\sigma) \neq \text{rank}(\sigma')$ by the above. Suppose w.l.o.g. that $\text{rank}(\sigma) < \text{rank}(\sigma')$ and let $t+1, t'+1$ be the stages when σ, σ' are scheduled. By Lemma 2.2, $t < t'$. Then σ must be pending, failing or succeeding at $t'+1$. If σ is pending or succeeding at $t'+1$, then $\sigma \subseteq V_\alpha^A[t']$ and σ' will be chosen disjoint from σ. If σ is failing at $t'+1$, then we cannot have $\rho(\sigma) \in F_\alpha^A[s]$ on account of σ since $A[t']$ and $A[s]$ agree on the use of any such computation. So σ and σ' are disjoint. $\square$

The next lemma verifies that $\mu(T^B) < 1$.

LEMMA 2.4. For any P-node α and $s \in \mathbb{N}$,

$$\mu(T_\alpha^B[s]) \leq 2^{-m_\alpha - c(\alpha,s)-1}.$$

PROOF. Suppose that $T_\alpha^B[s]$ is not empty, and let $t+1$ be the greatest stage $\leq s$ when an attack is added to $T_\alpha^B[t+1]$. At stage $t+1$, (I) and (II) do not hold for any α-attack, as otherwise nothing would be added to $T_\alpha^B[t+1]$. So for every attack $\sigma \in T_\alpha^B[t+1]$, σ is pending or waiting at $t+1$ and $\rho(\sigma) \in U_\alpha^{\emptyset'}[t]$. Since $\mu(\sigma) = \mu(\rho(\sigma))$ and $T_\alpha^B[s] \subseteq T_\alpha^B[t+1]$, we have $\mu(T_\alpha^B[s]) \leq \mu(U_\alpha^{\emptyset'}[t]) \leq 2^{-m_\alpha - c(\alpha,s)-1}$. $\square$

Note that since A is Δ_2^0 and the string $\gamma(\sigma)$ is finite, each attack σ is eventually either permanently current (that is, σ is current at all $s >$ some s_0) or permanently not current, depending on whether $\gamma(\sigma) \subset A$.

The following lemma describes the fate of α-attacks for α on the true path.

LEMMA 2.5. Suppose that α is a P-node which is accessible infinitely often and is reset only finitely often. Let s_0 be the last stage at which α is reset (or 0 if never). Suppose that σ is an α-attack scheduled at $t+1 > s_0$ such that σ is eventually permanently current and no α-attack σ' with $\operatorname{rank}(\sigma') < \operatorname{rank}(\sigma)$ is implemented infinitely often. Then exactly one of the following holds:

(A) σ is implemented infinitely often;

(B) $\sigma \not\subseteq V^A_\alpha[s']$ at every α-stage s' after some stage s;

(C) σ is permanently pending after some s (that is, σ is pending at every $s' > s$);

(D) σ is permanently succeeding after some s;

(E) σ is permanently failing after some s.

PROOF. We use induction on $\operatorname{rank}(\sigma)$. Let σ be as in the claim and assume inductively that there is a stage s_1 such that σ is permanently current after s_1 and all permanently current α-attacks of rank $< \operatorname{rank}(\sigma)$ scheduled after s_0 satisfy one of (B)-(E) for $s = s_1$.

First consider the case that the computation $\hat{\rho}(\sigma)$ is not permanent. Then σ is implemented only finitely often (possibly never), with finitely many computations $\langle \rho(\sigma), \tau_0 \rangle, \ldots \langle \rho(\sigma), \tau_n \rangle$ in F_α on account of σ. If $A \supset \tau_i$ for some i then eventually σ is permanently succeeding and (D) holds; otherwise σ is permanently failing and (E) holds.

Next consider the case that the computation $\hat{\rho}(\sigma)$ is permanent. If σ is implemented infinitely often then (A)·holds and (B), (C) cannot hold. Suppose that σ is implemented only finitely often and (B) does not hold; that is, $\sigma \subseteq V^A_\alpha[s']$ for infinitely many α-stages s'. Each time this occurs after s_1, σ will be implemented unless some existing computation $\langle \rho(\sigma), \tau \rangle \in F_\alpha[s']$ on account of σ is valid. Since σ is implemented only finitely often, and as A is Δ^0_2, eventually the approximation $A[s]$ will settle on the use of these computations. After this point, one of the computations must be permanently valid, and $\rho(\sigma) \in F^A_\alpha$ permanently on account of σ. Thus (C) holds. □

As usual, the true path consists of the leftmost infinitely-often visited nodes. Since the tree is infinitely branching, and since we sometimes stop defining TP_s early, we must verify that the true path exists. This lemma verifies that P-nodes do not cause the true path to be finite; Lemma 2.9 does the same for S-nodes. We simultaneously verify that a P-node on the true path satisfies its P-requirement.

LEMMA 2.6. Suppose that α is a P-node that is reset only finitely often and is accessible infinitely often. Then $T_\alpha^B \not\subseteq V_\alpha^A$. Furthermore, either some outcome $\alpha^\frown\infty(i)$ is accessible infinitely often, or after some stage s_0 outcome $\alpha^\frown f$ is accessible at every α-stage.

PROOF. Suppose first that some α-attack is implemented infinitely often; let σ be the least α-attack by rank which is implemented infinitely often. Since σ is permanently current, and the computation $\hat{\rho}(\sigma)$ is permanently valid, σ is permanently in T_α^B. However, $\sigma \not\subseteq V_\alpha^A$, since then σ would be implemented only finitely many times before some computation $\rho(\sigma) \in F_\alpha^A$ on account of σ was permanent. So $T_\alpha^B \not\subseteq V_\alpha^A$ and P_e is satisfied. Furthermore, every time σ is implemented, outcome $\infty(k)$ is accessed, where $k = \text{rank}(\sigma)$.

Suppose then that no α-attack is implemented infinitely often. Inductively, by Lemma 2.5, every permanently current α-attack satisfies one of (B)-(E). We claim that some attack satisfied (B), and hence $T_\alpha^B \not\subseteq V_\alpha^A$, only finitely many α-attacks are scheduled, and $\alpha^\frown f$ is accessible at all but finitely many α-stages. Suppose for contradiction that there is no attack which satisfies (B). Then infinitely many α-attacks are scheduled. We first argue that $U_\alpha^{\emptyset'} \subseteq F_\alpha^A$.

We argue by induction on the U_α computations $\langle \rho_i, \tau_i \rangle$ that for every string $\rho \in U_\alpha^{\emptyset'}$ there is an α-attack σ which is permanently pending or permanently succeeding with $\rho(\sigma) = \rho$. For $\rho_i \in U_\alpha^{\emptyset'}$, assume inductively that there is a stage s_1 such that for every computation $\rho_j \in U_\alpha^{\emptyset'}, j < i$ there is an attack σ_j with $\rho(\sigma_j) = \rho_j$ which is permanently pending or succeeding by s_1. Let s_2 be a stage such that the computation $\rho_i \in U_\alpha^{\emptyset'}[s_2]$ is correct, and any computations $\rho_j \in U_\alpha^{\emptyset'}[s_2]$ for $j < i$ are correct.

By choice of s_2, if an α-attack is scheduled at $s > s_2$ and there is no attack σ pending at s with $\rho(\sigma) = \rho$, then the newly scheduled attack will have rank i. Furthermore, $\gamma(\sigma')$ is fixed for all such attacks scheduled after s_2. Therefore eventually there must be an attack σ which is permanenly current and has $\rho(\sigma) = \rho$; by Lemma 2.5 and the assumptions it must eventually be permanently pending. This establishes that $U_\alpha^{\emptyset'} \subseteq F_\alpha^A$.

Since $\emptyset' \not\leq_{LR} A$, we must have $\mu(F_\alpha^A) = 1$. By Lemma 2.3, every string ρ in F_α^A corresponds to an attack of the same measure in V_α^A, and distinct strings correspond to disjoint attacks. Therefore $\mu(V_\alpha^A) = 1 > p_\alpha$, a contradiction.

Some α-attack must therefore satisfy (B) of Lemma 2.5, and hence $T_\alpha^B \not\subseteq V_\alpha^A$ and P_e is satisfied. Since condition (V) in the construction holds for only finitely many α-stages, only finitely many α-attacks are scheduled. Since each attack satisfies one of (I)-(IV) only finitely often, outcome f is accessed at all but finitely many α-stages. $\square$

Now we deal with the S-requirements. First we verify that an S-node's restraint is respected.

LEMMA 2.7. For an S-node α and stage s, if $r_\alpha[s] \neq 0$ and α is not reset at stage $s+1$ then $B[s+1] \upharpoonright r_\alpha[s] = B[s] \upharpoonright r_\alpha[s]$.

PROOF. Suppose that α is not reset at $s+1$, $r = r_\alpha[s] \neq 0$ but $B[s+1] \upharpoonright r \neq B[s] \upharpoonright r$. Some P-node β must enumerate a number $x < r$ into $B[s+1]$ in order to remove an attack σ from $T_\beta^B[s+1]$. We must have $\beta < \alpha$ since all nodes of lower priority than α are reset when r_α is set nonzero, and thereafter any attacks would be put into T with use $> r$. Let σ be the least β-attack by rank which is removed from $T_\beta^B[s+1]$, and let $k = \mathrm{rank}(\sigma)$. Since all nodes of lower priority than $\beta^\frown\infty(k)$ are reset at $s+1$, we must have $\beta^\frown\infty(j) \subseteq \alpha$ for some $j < k$. Let $t+1$ be the greatest stage $< s$ when σ was put into $T_\beta^B[t+1]$, and let $s'+1$ be the greatest stage $< s+1$ when r_α was increased above x. If $t+1 < s'$ then $\sigma \in T_\beta^B[s']$, and since $\mathrm{rank}(\sigma) > j$ and $x < r$ the computation $\rho \subseteq V_\alpha^B[s']$ with use r would not be α-believable. So r_α would not be set nonzero at $s'+1$, contradicting the choice of s'. If $t+1 > s'$ then σ would be put into $T_\beta^B[t+1]$ with fresh use $x > r$, contradicting $x < r$. Also $s' = t$ is impossible since we stop defining TP_{t+1} once β takes action. So $B[s] \upharpoonright r = B[s+1] \upharpoonright r$. $\square$

The next lemma verifies that true computations will eventually be believable, with respect to an S-node on the true path.

LEMMA 2.8. Let α be an S-node such that α is accessible infinitely often and α is reset only finitely often. Suppose that $\tau \subseteq V_\alpha^B$. Then there is a stage s_0 such that the computation $\tau \subseteq V_\alpha^B[s]$ is α-believable at all $s \geq s_0$.

PROOF. Suppose that α is as in the claim, and let s_0 be the first α-stage such that $\tau \subseteq V_\alpha^B[s_0]$ with use u, and $B[s_0] \upharpoonright u = B \upharpoonright u$. Then the computation $\tau \subseteq V_\alpha^B$ is believable at s_0. If it were not, then there must be a P-node β with $\beta^\frown\infty(k) \subseteq \alpha$ for some k, and a β-attack σ with $\mathrm{rank}(\sigma) > k$ and $\sigma \in T_\beta^B[s_0]$ with use $< u$. Note that

β's computation $\rho_k \in U^{\emptyset'}$ must be permanent, since otherwise $\beta^\frown\infty(k)$, and hence α, would not be accessible infinitely often. Let $t+1$ be the stage when σ was scheduled. By Lemma 2.2, $\rho_k \in F_\beta^A[t]$ on account of some attack with rank $\leq k$, so $\gamma(\sigma)$ is greater than the use of $\rho_k \in F_\beta^A[t]$. By Lemma 2.6 there is a stage $s_1 \geq s_0$ when some β-attack σ' with $\rho(\sigma') = \rho_k$ is implemented. At s_1, the computation $\rho_k \in F_\beta^A[t]$ is no longer valid, so σ cannot be current at s_1. Therefore σ must have been removed from T_β^B before s_1 but after s_0 via a B-enumeration, which contradicts the choice of s_0. So the computation $\tau \in V_\alpha^B$ is believable by s_0. $\square$

Now we verify that the restraint $m(\alpha, s)$ reaches a limit for nodes on the true path.

LEMMA 2.9. Let α be an S-node such that α is accessible infinitely often and α is reset only finitely often. Then

$$\lim_s m(\alpha, s) < \infty.$$

PROOF. By Lemma 2.7, α's restraint is respected at all $s \geq$ some s_0. After s_0, $m(\alpha, s)$ does not decrease. Suppose that $\lim_s m(\alpha, s) = \infty$. Enumerate a Σ_1^0-class G as follows: put the string ρ_i into G at α-stage $s \geq s_0$ if $\rho_i \in U^A[s]$ and $m(\alpha, s) \geq i$. Then $U^A \subseteq G$, since eventually $m(\alpha, s) \geq i$ for every $\rho_i \in U^A$. Also, since the restraint $r(\alpha, s)$ is greater than the use of $\rho_i \subseteq V_\alpha^B[s]$ and is respected after s, we have $G \subseteq V_\alpha^B$. So $\mu(G) \leq \mu(V_\alpha^B) < 1$. But this gives $A \leq_{LR} \emptyset$ since G is Σ_1^0, which contradicts $\emptyset <_{LR} A$. Therefore $m(\alpha, s) < i$ for some i and all s. $\square$

LEMMA 2.10. The true path $\mathrm{TP} = \liminf_s \mathrm{TP}_s$ exists and is infinite, and each node on it is reset only finitely often.

PROOF. The root node is on TP_s for all s and is never reset. Inductively assume that $\alpha = \liminf_s \mathrm{TP}_s \upharpoonright n$ for $n > 0$, and that α is reset only finitely often. If α is a P-node, then by Lemma 2.6 there is some outcome (either f or $\infty(k)$ for some k) that is accessible infinitely often. If α is an S-node, then Lemma 2.9 guarantees that the child $\alpha^\frown f$ is accessible at all but finitely many α-stages, since the definition of TP_s is only ended at α if r_α and $m(\alpha, s)$ increase. Therefore $\beta = \liminf_s \mathrm{TP}_s \upharpoonright n+1$ exists.

Now we verify that β is reset only finitely often. The situations where β might be reset are when (I) or (II) holds for some P-node $\gamma \subset \beta$, when some S-node $\delta \subset \beta$ increases its restraint, or when $\mathrm{TP}_s <_L \beta$. The last can happen only finitely often by induction assumption. By Lemma 2.9, each S-node $\delta \subset \beta$ increases its restraint only finitely often.

Suppose then that γ is a P-node and $\gamma^\frown f \subseteq \beta$. By Lemma 2.6 there is an attack σ satisfying (B) at all γ-stages after some s_0. After s_0, no new attacks will be scheduled by γ, and existing attacks can cause β to be reset only finitely often after s_0.

If $\gamma^\frown\infty(k) \subseteq \beta$ for some k, then k is the least such that some γ-attack of rank k is implemented infinitely often. By Lemma 2.2, there are only finitely many γ-attacks of rank $< k$, and they can cause β to be reset due to (I) or (II) only finitely often. □

LEMMA 2.11. Each requirement P_e is satisfied.

PROOF. Let α be the P_e-node on TP. By Lemma 2.6, there is an α-attack σ that is permanently in T_α^B but $\sigma \not\subseteq V_\alpha^A$. (In the case that $\alpha^\frown\infty(k) \subset TP$, then σ is implemented infinitely often so no computation $\sigma \subseteq V_\alpha^A[s]$ is permanent.) Since $T_\alpha^B \subseteq T^B$, we have $T^B \not\subseteq V_\alpha^A = V_e^A$. □

LEMMA 2.12. Each requirement S_e is satisfied.

PROOF. Let α be the S_e-node on the true path. By Lemma 2.8 and the use of the hat-trick for U^A, we have

$$U^A \subseteq V_e^B \Leftrightarrow \lim_s m(\alpha, s) = \infty.$$

By Lemma 2.9, $\lim_s m(\alpha, s) < \infty$. So $U^A \not\subseteq V_e^B$. □

This completes the proof of Theorem 2.1. □

3. C.e. LR-degrees above low LR-degrees

In this section we show that above any low Δ_2^0 LR degree there is an incomplete c.e. LR degree. This is in contrast with the Δ_2^0 Turing degrees, in which there is a low Δ_2^0 degree which is incomparable with all intermediate c.e. degrees (the proof of this is sketched in section 3.4). Thus it highlights a difference of the position of the c.e. degrees within the Δ_2^0 LR-degrees, as compared to the Δ_2^0 Turing degrees.

We first tried to prove the considerably stronger result that above *any* LR-incomplete Δ_2^0 LR-degree A there is an LR-incomplete c.e. degree. Unfortunately there are obstacles to performing the construction in this general case, which we discuss in section 3.6. It is not known if this more general result holds. However the construction does work in the specific case when A is c.e. and LR-incomplete. We outline the modifications necessary for this case in section 3.5.

THEOREM 2.13. Let A be a low Δ^0_2 set. Then there is a c.e. set B such that

$$A \leq_{LR} B <_{LR} \emptyset'.$$

Let A be a low Δ^0_2 set, given by a computable approximation $A[s]$ such that $\lim_s A(x)[s] = A(x)$ for all x. Let $\langle V_e, p_e \rangle$ be a listing of all LR-operators, and U be the second member of a universal oracle ML-test (so $\mu(U^X) \leq \frac{1}{2}$ for all $X \in 2^\omega$). We construct c.e. sets B, D and oracle Σ^0_1-classsses E, H_e for all $e \in \mathbb{N}$ to satisfy the requirements:

$$N_e: \quad H_e^D \not\subseteq V_e^B$$

$$R: \quad U^A \subseteq E^B \quad \text{and} \quad \mu(E^B) < 1$$

By Theorem 1.5, requirement R ensures that $A \leq_{LR} B$. Since the H_e are uniformly Σ^0_1, their union $H = \cup_e H_e$ is also an oracle Σ^0_1 class. We will ensure that $\mu(H_e^D) < 2^{-e-1}$, and thus

$$\mu(H^D) \leq \sum \mu(H_e^D) \leq 2^{-e-1} < 1.$$

If N_e is satisfied for each e, then $H^D \not\subseteq V_e^B$, and $D \not\leq_{LR} B$ by Theorem 1.5, since $\mu(H^D) < 1$. Therefore in particular $\emptyset' \not\leq_{LR} B$.

Notice that we do not include any requirements to explicitly make $B \not\leq_{LR} A$. If desired, we could include P-requirements as in Theorem 2.1 to explicitly ensure $A <_{LR} B$. How to do this is discussed briefly at the end of the section. However, we may instead invoke the upward density of the c.e. LR-degrees, namely Theorem 1.2, to obtain a c.e. set C with $B <_{LR} C <_{LR} \emptyset'$. Of course, if the LR-degree of A does not contain any c.e. sets then we automatically have $A <_{LR} B$ from the requirements above. (This is the interesting case since if A is $\equiv_{LR}$ to some c.e. set then we can just invoke the upward density of the c.e. LR-degrees in the first place.)

R **strategy.** Fix U as the second member of a universal oracle ML-test, so $\mu(U^A) < \frac{1}{2}$. We simply trace U^A into E^B: whenever a new computation $\rho \in U^A[s]$ appears, we put ρ into $E^B[s]$ with large use. If a B-change inadvertently removes a string ρ from $E^B[t]$ while $\rho \in U^A[t]$ is still valid, we put ρ back into E^B with the same use as previously. If an A-change invalidates the computation $\rho \in U^A[s]$, we remove ρ from E^B via a B-enumeration, if such an enumeration is not prevented by an active restraint. It is up to the N-requirements to ensure that their restraints do not prevent too many strings from being removed from E^B, so we can ensure that $\mu(E^B) < 1$.

N_e **strategy.** Recall that $\langle V_e, p_e \rangle$ is a list of all LR-operators; that is, an oracle Σ^0_1 class V_e and a dyadic rational p_e such that $\mu(V_e^X) \leq q_e$ for all $X \in 2^\omega$. We need to diagonalise against V_e, forcing measure into V_e^B without causing $\mu(H_e^D)$ to permanently increase. The basic strategy is to put a clopen set δ into H_e^D and wait until $\delta \subseteq V_e^B$. When this occurs we restrain B on the use of the computation and remove δ from H_e^D by enumerating into D. Thus $\mu(V_e^B)$ permanently increases by $\mu(\delta)$ but $\mu(H_e^D)$ does not. With a suitable choice of $\mu(\delta)$, after finitely many repetitions we will have some δ which is not covered by V_e^B, since $\mu(V_e^B)$ cannot increase above p_e. This δ will be permanently in H_e^D, and nothing else will be added to H_e^D. So $\mu(H_e^D)$ permanently increases by $\mu(\delta)$ only once.

This is complicated by the fact that B-restraints conflict with the R-strategy. Each B-restraint captures certain junk intervals in E^B, in the sense that the B-restraint will prevent the R-strategy from removing some intervals from E^B if an A-change removes them from U^A. Such strings that are in E^B but not in U^A are *junk*; we must make sure that the total junk measure captured by B-restraints is small so that $\mu(E^B) < 1$.

We can separate the N_e-requirement into finitely many subrequirements $N_{e,i}$, and assign each N-subrequirement a quota ϵ. We ask that $N_{e,i}$'s restraints contribute at most ϵ measure of junk to E^B. We will allow an N-node α to impose a restraint r at stage s only if the total junk measure that would be captured, those strings in $E^B - U^A$ with use $< r$, is within the quota ϵ. However, the junk captured by the restraint r may later increase as the construction proceeds, as strings may be removed from U^A after the restraint is imposed. Although we can easily ensure that the restraint initially captures at most ϵ of junk, we must also ensure that the junk does not later grow too large.

This is dealt with by *measure-guessing*: we place the construction on a tree of strategies, and equip each N-node α with a backing measure-guessing node. The backing node supplies the N-node α with an approximation to $\mu(U^A)$, in the form of a rational interval $[q_\alpha, q_\alpha+\epsilon_\alpha)$. The node α works only at stages when $\mu(U^A[s]) \in [q_\alpha, q_\alpha+\epsilon_\alpha)$; that is when the 'measure guess' that $\mu(U^A) \in [q_\alpha, q_\alpha + \epsilon_\alpha)$ appears correct. By ordering the nodes with lower intervals to the left, and by applying the hat-trick to the approximation of U^A, we can ensure that the true path (the path of leftmost infinitely-often visited nodes) consists of those nodes whose measure guess is correct, ie $\mu(U^A) \in [q_\alpha, q_\alpha + \epsilon_\alpha)$ in the limit. A node may only impose a B-restraint r if the measure of junk captured by r is less than ϵ_α. If the junk captured by the B-restraint later increases by more than ϵ_α, the hat-trick applied to the approximation of U^A guarantees that the node will be reset, as in that case the approximation $\mu(U^A[s])$ drops below q_α. The result is that a node never captures more than 2ϵ much junk: ϵ much which was present when it first imposed restraint, and ϵ much which may have been added after the restraint was imposed. This technique was first used by Cholak, Greenberg and Miller [**12**] and later by Barmpalias and Montalbán [**7**]. We will use the same technique in Theorem 4.1 of chapter 4.

We arrange the priority tree so that each level of the tree is occupied by a single requirement, and all the nodes of that level have the same quota. We need one additional condition to keep the junk captured by restraints under control. In a traditional tree construction, there is no bound to the number of nodes on any level of the tree that may be imposing restraint simultaneously. In our case, each such node on a particular level would potentially be contributing the same amount ϵ of junk to E^B, threatening our desire to keep $\mu(E^B) < 1$. The solution is to ensure that *at most one node on each level of the tree is imposing restraint at any time.* To satisfy this, all nodes on one level of the tree will work with the same clopen set δ, on the task of ensuring that $\delta \subseteq V_e^B$. If a node β on the same level but to the left of α has already imposed a B-restraint to preserve $\delta \subseteq V_e^B$, then α does not need to do anything since (from α's point of view) β has already satisfied $\delta \subseteq V_e^B$. If no node left of α has imposed a restraint, and α sees $\delta \subseteq V_e^B[s]$ (via a computation which does not capture too much junk), then α may impose a restraint and remove δ from H_e^D.

There is a risk here however that the junk captured by α's restraint later grows above the quota ϵ_α, and α is reset when $\mu(U^A[s])$ drops below q_α. Then we will have to start again with δ, putting δ back into H_e^D and waiting for $\delta \subseteq V_e^B[s]$ again. In fact, if α is to the right of the true path, this may happen infinitely often; in this case, $\delta \not\subseteq V_e^B$ as no computation $\delta \subseteq V_e^B[s]$ is permanent, but nor is $\delta \subseteq H_e^D$. So δ does not contribute towards requirement N_e. This problem only arises if every time a node α on level i of the tree imposes a restraint, its measure guess proves wrong and it is reset. We can solve this problem by attempting to restrain A, to prevent an A-change from causing $\mu(U^A[s])$ to drop below q_α.

Each time an $N_{e,i}$-node α wishes to impose B-restraint and remove δ from H^D, α will attempt to restrain $A \upharpoonright u$ for u such that $\mu(U^{A\upharpoonright u}) \geq q_\alpha$. If the restraint is successful, then $\mu(U^A)$ will never drop below q_α and α will never be reset. As long as the restraint is eventually successful after finitely many attempts, we will avoid the problem described above. To restrain A, we utilise the fact that A is low. (In section 3.5 we describe how to modify this strategy to work in the case where A is c.e. and LR-incomplete rather than low.) When α is ready to remove δ from H^A, it defines a computation $\Phi^A(z)[s]\downarrow$ with use u as above. Since A is low, we can argue that computations $\Phi^A(z)[s]$ can only be spoiled by A-changes finitely often before the computation is permanent and the A-restraint succeeds.

3.1. The priority tree and notation. For each e, let k_e be the least number such that $2^{-k_e} < 2^{-e-2} \cdot (1 - p_e)$. Requirement N_e will use 2^{k_e} subrequirements $N_{e,i}$, $0 \leq i \leq 2^{k_e} - 1$. Fix a listing of all N subrequirements

$$N_{0,0}, \dots N_{0,2^{k_0}-1}, N_{1,0}, \dots N_{1,2^{k_1}-1}, \dots \tag{7}$$

namely, all the N_0 subrequirements in order, followed by the N_1 subrequirements, etc.

The construction takes place on a finitely branching tree, defined below, consisting of nodes labelled G or $N_{e,i}$ for some $e, i \in \mathbb{N}$ according to their length. Nodes of even length (including the root node) are labelled G; nodes of odd length $2n+1$ are labelled $N_{e,i}$ where $N_{e,i}$ is the n'th entry in the list (7). Nodes labelled $N_{e,i}$ for some e, i are referred to as N-nodes; nodes labelled $N_{e,i}$ for a fixed e are referred to as N_e-nodes. N-nodes have a single outcome 0 (a single child node $\alpha^\frown 0$ on the tree). G-nodes have four outcomes $x_0 <_L x_1 <_L x_2 <_L x_3$, corresponding to subintervals of the half-unit interval $[0, \frac{1}{2})$. Each node α is associated with an interval $[q_\alpha, q_\alpha + \epsilon_\alpha)$, where $q_\alpha, \epsilon_\alpha$ are dyadic rationals. For the root node $\emptyset$ we have $q_\emptyset = 0, \epsilon_\emptyset = \frac{1}{2}$; for other nodes the interval is defined inductively. Suppose that $[q_\alpha, q_\alpha + \epsilon_\alpha)$ is defined. If α is an N-node then it has only one child $\alpha^\frown 0$; let $q_{\alpha^\frown 0} = q_\alpha$ and $\epsilon_{\alpha^\frown 0} = \epsilon_\alpha$. If α is a G-node, then $\epsilon_{\alpha^\frown x_i} = \frac{1}{4}\epsilon_\alpha$ and $q_{\alpha^\frown x_i} = q_\alpha + i\frac{1}{4}\epsilon_\alpha$ for $0 \leq i \leq 3$; that is, we evenly subdivide $[q_\alpha, q_\alpha + \epsilon_\alpha)$ and assign the subintervals in order to $x_0 \ldots x_3$. We subdivide into four to ensure that

$$\sum_{\gamma \in Z} 2\epsilon_\gamma < \epsilon_\alpha \tag{8}$$

for any set Z of nodes longer than α containing at most one node of each length. This means that, even if every subrequirement below α has a node γ imposing restraint, and capturing up to ϵ_γ of junk, the total junk is still within α's quota ϵ_α, so α will be able to act.

The ordering $<_L$ on $\{x_0, \ldots x_3\}$ induces an ordering on the tree: for nodes α, β, $\alpha <_L \beta$ indicates that α is to the left of β, and $\alpha < \beta$ indicates that $\alpha <_L \beta$ or $\alpha \subset \beta$.

As described previously, we will use the lowness of A to ensure that the restraints imposed by N-nodes are eventually permanent. Since A is low, the jump

$$A' = \{\langle e, z\rangle : \Phi_e^A(z)\downarrow\}$$

is limit computable; that is, there is a total computable function $g : \mathbb{N} \times \mathbb{N} \to \{0,1\}$ such that

$$\forall x \ \lim_s g(x,s) \text{ exists and equals } A'(x).$$

In the construction, we will construct a Turing functional Γ (as a consistent c.e. set of axioms). By the Recursion Theorem 1.11, we may assume that we know in advance an index of the functional Γ; that is, a number e such that $\Gamma = \Phi_e$.[1] Define $h : \mathbb{N} \times \mathbb{N} \to \{0,1\}$ by $h(x,s) = g(\langle e, x\rangle, s)$; then $\Gamma^A(x)\downarrow$ iff $\lim_s h(x,s) = 1$.

For each e, divide 2^ω evenly into 2^{k_e} many subintervals $I_{e,0}, I_{e,1} \ldots I_{e,2^{k_e}-1}$. Subrequirement $N_{e,i}$ works with interval $I_{e,i}$. Assign each N-node α a unique number z_α from $\mathbb{N}$. An $N_{e,i}$-node α pursues the following strategy. When all higher-priority N_e subrequirements are finished, α puts $I_{e,i}$ into H_e^D and waits until $I_{e,i} \subseteq V_e^B[s]$. When this occurs it restrains B on the use of this computation to preserve $I_{e,i} \subseteq V_e^B$ (if the restraint does not capture too much measure in E^B). Then α defines a computation $\Gamma^A(z_\alpha)[s]\downarrow$ (if it is not already defined), to attempt to prevent $\mu(U^A[s])$ from dropping below q_α which would cause α to be reset. When the Δ_2^0 approximation h of A' indicates

[1] Formally, we can consider e to be a free parameter, on which the function h and indeed the whole construction depend. The construction is well-defined for all values of e (although it will only do what we want it to do for certain values of e). We thus have a uniform procedure for obtaining a Turing functional Γ_e from a number e; that is, a computable function f such that $\Phi_{f(e)} = \Gamma_e$ (where $(\Phi_e)_{e\in\mathbb{N}}$ is a canonical listing of Turing functionals). The Recursion Theorem 1.11 guarantees an e such that $\Phi_e = \Phi_{f(e)} = \Gamma_e$. For that particular value of e, we do in fact have $\Gamma^A(x)\downarrow$ iff $\lim_s h(x,s) = 1$ as desired.

that the computation $\Gamma^A(z_\alpha)$ is permanent, we remove $I_{e,i}$ from H_e^D by enumerating into D. If our attempted A-restraint fails and α is later reset, then we have to re-add $I_{e,i}$ to H_e^D. However, each time this happens an old computation $\Gamma^A(z_\alpha)$ is invalidated. If it happens infinitely often then $h(z_\alpha, s) = 0$ for all but finitely many s, so $I_{e,i}$ can only be removed from H_e^D finitely often before it is permanently in H_e^D. Then we can argue that $I_{e,i} \not\subseteq V_e^B$ and N_e is satisfied.

Each N-node α has a parameter r_α which is the restraint that α wishes to impose on B. Let $R_\alpha[s] = \max_{\beta<\alpha} r_\beta[s]$ be the total restraint imposed by N-nodes of higher priority than α.

As in Theorem 2.1, we assume that when a string is put into any H_e^D with some use u, it remains there until the number u is explicitly enumerated into D. This assumption does not apply to E^B however. Each time we put a string into $H_e^D[s]$ it will be while carrying out the instructions for some N-node. If an interval I is in $H_e^D[s]$, then we say that *I is in $H_e^D[s]$ on account of α* if I was most recently put into H_e^D while carrying out the instructions for α.

We again use the hat-trick for the enumeration of U^A. Let $a_0 = 0$, and for $s > 0$ let a_s be the least number such that $A(a_s)[s] \neq A(a_s)[s-1]$, or $a_s = s$ if such number does not exist. Let

$$\widehat{U^A}[s] = U^{A\upharpoonright a_s}[s] = \{\sigma : \sigma \text{ is in } U^A[s] \text{ with use } \leq a_s\}.$$

Henceforth we omit the hat and write $U^A[s]$ to mean $\widehat{U^A}[s]$. In this case, the hat-trick ensures that there are infinitely many true stages, at which $U^A[s] \subseteq U^A$ and $\mu(U^A[s]) \leq \mu(U^A)$.

In the construction, we will explicitly define the approximation to the true path TP_s. When we take action for a node on TP_s, we will stop defining TP_s, so TP_s will not always have length s. This means that we cannot rely on a node α being reset due to $\mathrm{TP}_s <_L \alpha$ when its measure guess becomes wrong. Hence at each stage we must explicitly reset all nodes whose measure guess has become wrong. For convenience, we do this resetting only at even stages of the construction, and perform the other tasks of the construction only at odd stages. We can assume that we are given approximations of A, U, V_i etc that change only on even stages. That is, $A[2s] = A[2s+1]$ for all s, and similarly for U, V_i (as sets of axioms).

3.2. The construction. To *reset* an $N_{e,i}$-node α at stage $s+1$ means to set $r_\alpha[s+1] = 0$, and if $I_{e,i}$ is in $H_e^D[s]$ on account of α then remove $I_{e,i}$ from $H_e^D[s+1]$ by enumerating its use into $D[s+1]$.

Initially $B[0] = D[0] = H[0] = \emptyset$ and $r_\alpha[0] = 0$ for all N-nodes α. At stage $s+1$ we are given $A[s], B[s]$ etc and any changes we make are in order to define $B[s+1]$ etc.

At stage $s+1$ where $s+1$ is even, reset any N-nodes α such that $\mu(U^A[s]) < q_\alpha$ and $r_\alpha[s] \neq 0$.

At stage $s+1$ where $s+1$ is odd, perform steps 1 and 2 in order.

Step 1. We define the approximation to the true path TP_{s+1} and take action for some node on TP_{s+1}. Suppose inductively that $\mathrm{TP}_{s+1} \upharpoonright n$ is defined, for $n \geq 0$. If $n = s+1$ then stop defining TP_{s+1} and go to step 2. Otherwise let $\alpha = \mathrm{TP}_{s+1} \upharpoonright n$ and go to the appropriate case below.

- **α is a G-node.** Inductively, $\mu(U^A[s]) \in [q_\alpha, q_\alpha + \epsilon_\alpha)$. Let i be such that $\mu(U^A[s]) \in [q_{\alpha^\frown x_i}, q_{\alpha^\frown x_i} + \epsilon_{\alpha^\frown x_i})$ and let $\mathrm{TP}_{s+1} \upharpoonright n+1 = \alpha^\frown x_i$.
- **α is an $N_{e,i}$-node for some $e, i \in \mathbb{N}$.** Say that α is *active* at stage $s+1$ if $\forall j < i$ there is an $N_{e,j}$-node $\beta < \alpha$ with

$$r_\beta[s] \neq 0 \text{ and } h(z_\beta, s) = 1.$$

Go to the least case below which holds.

(I) *Higher priority subrequirements have finished and α is ready to start.* α is active; there is no $N_{e,i}$-node $\alpha' \leq_L \alpha$ with $r_{\alpha'}[s] \neq 0$; and $I_{e,i} \notin H_e^D[s]$. Then put $I_{e,i}$ into $H_e^D[s+1]$ with large use. Stop defining TP_{s+1} and go to step 2.

(II) *α's interval $I_{e,i}$ has appeared in $V_e^B[s]$ with a believable computation and we are ready to restrain B.* α is active; there is no $N_{e,i}$-node $\alpha' \leq_L \alpha$ with $r_{\alpha'}[s] \neq 0$; $I_{e,i} \in H_e^D[s]$ and $I_{e,i} \subseteq V_e^B[s]$ with use u such that

$$\mu(E^{B \upharpoonright u}[s] - E^{B \upharpoonright R_\alpha}[s] - U^A[s]) < \epsilon_\alpha. \tag{9}$$

Then set $r_\alpha[s+1] = u$. If $\Gamma^A(z_\alpha)[s] \uparrow$ then let v be the least such that

$$\mu(U^{A \upharpoonright v}[s]) \geq q_\alpha, \tag{10}$$

and define a new computation $\Gamma^{A[s] \upharpoonright v}(z_\alpha) \downarrow$ with use v. Reset all N-nodes of lower priority than α, stop defining TP_{s+1} and go to step 2.

(III) *The computation* $\Gamma^A(z)$ *appears permanent so we can remove* $I_{e,i}$ *from* H_e^D. There is an $N_{e,i}$-node $\alpha' \leq_L \alpha$ such that $r_{\alpha'}[s] \neq 0$, $I_{e,i} \in H_e^D[s]$ and $h(z_{\alpha'}, s) = 1$. Then remove $I_{e,i}$ from $H_e^D[s+1]$ by enumerating the use into $D[s+1]$. Stop defining TP_{s+1} and go to step 2.

(IV) Otherwise, set $\mathrm{TP}_{s+1} \upharpoonright n+1 = \alpha^\frown 0$, the unique child of α, and continue defining TP_{s+1}.

Step 2. Let $R = \max_\alpha r_\alpha[s+1]$ be the total restraint imposed by all nodes after step 1. Enumerate $R+1$ into $B[s+1]$ to remove some junk intervals from $E^B[s] - U^A[s]$.

End of construction.

3.3. Verification. First we verify that $H_e^D[s]$ contains at most one of the intervals $I_{e,i}$ at any time, and thus $\mu(H^D) < 1$.

LEMMA 2.14. For all e and at any stage s, either $H_e^D[s] = \emptyset$ or $H_e^D[s] = I_{e,i}$ for some i.

PROOF. Suppose on the contrary, that $H_e^D[s]$ contains both $I_{e,i}$ and $I_{e,j}$ for some $i \neq j$. Suppose that $I_{e,i}$, $I_{e,j}$ were added to $H_e^D[s]$ at stage s_0, s_1 by N_e-node α, β, respectively. Note that at most one interval is added to H_e^D at each stage (since the stage is ended if (I) holds); thus we may assume that $s_0 < s_1$. We consider the possibilities for the position of α relative to β.

If $\beta <_L \alpha$, then $\mu(U^A[s_1]) < q_\alpha$, and α would have been reset at the (even) stage $s_1 - 1$. Thus $I_{e,i}$ could not be in $H_e^D[s_1]$.

If $\beta \geq \alpha$ and $|\beta| > |\alpha|$, then β must satisfy (I) at s_1; in particular, there must be an $N_{e,i}$-node $\alpha' \leq \beta$ with $r_{\alpha'}[s_1] \neq 0$ and $h(z_{\alpha'}, s_1) = 1$. But then, $\alpha'' = \beta \upharpoonright |\alpha|$ would satisfy (III) at s_1, $I_{e,i}$ would be removed from $H_e^D[s_1+1]$ and β would not be accessible at s_1.

Finally, $|\beta| < |\alpha|$ and $\beta \not<_L \alpha$. In this case, at s_0 there is some $N_{e,j}$-node $\beta' \leq \alpha$ with $r_{\beta'}[s_0] \neq 0$. Since β satisfies (I) at $s_1 > s_0$, the node β' must have been reset at some s', $s_0 < s' < s_1$. But then α would be reset at s' also, and $I_{e,i}$ would have been removed from $H_e^D[s'+1]$. □

Since $H^D = \cup_e H_e^D$, it follows that $\mu(H^D) \leq \sum_e \mu(H_e^D) \leq 2^{-k_e} < 1$.

Next we verify that each restraint does indeed capture no more than its quota of junk. For an N-node α and $s > 0$, let

$$J_\alpha[s] = E^{B\restriction r_\alpha}[s] - E^{B\restriction R_\alpha}[s] - U^A[s-1]$$

be the junk intervals restrained by α at the end of stage s.

LEMMA 2.15. For any N-node α and for any even s,

$$\mu(J_\alpha[s]) \leq 2\epsilon_\alpha.$$

PROOF. Suppose that $r_\alpha[s] \neq 0$, and let $t+1$ be the greatest stage $\leq s$ when r_α was set nonzero. Write $r = r_\alpha[s]$. If new strings are added to E^B after t then they are added with fresh use, and if $B[s] \restriction r \neq B[t] \restriction r$ then α would be reset between $t+1$ and s. Thus $E^{B\restriction r}[s] = E^{B\restriction r}[t]$. Also, $R_\alpha[t] = R_\alpha[s]$ as otherwise α would have been reset. So,

$$\begin{aligned} \mu(J_\alpha[s]) &= \mu(E^{B\restriction r}[s] - E^{B\restriction R_\alpha}[s] - U^A[s-1]) \\ &\leq \mu(E^{B\restriction r}[t] - E^{B\restriction R_\alpha}[t] - U^A[t]) + \mu(U^A[t] - U^A[s-1]). \end{aligned} \tag{11}$$

The first term of (11) is the junk that was captured by α when it imposed its restraint; the second is that which becomes junk after the restraint was imposed. By (9), the first term is less than ϵ_α. Suppose that $\mu(U^A[t] - U^A[s-1]) \geq \epsilon_\alpha$. But then by the hat-trick there would be an even stage t' with $t < t' \leq s$ such that $\mu(U^A[t']) \leq \mu(U^A[t]) - \epsilon_\alpha \leq q_\alpha$. At t', α would be reset, contradicting the definition of t. □

Since we sometimes end the definition of TP_s before it reaches length s, we must verify that the true path $\liminf_s \mathrm{TP}_s$ exists. The following lemma simultaneously verifies that the true path exists, is infinite, and that the N-nodes on the true path reach a limit state.

LEMMA 2.16. Suppose that α is an $N_{e,i}$-node and the leftmost node of length $|\alpha|$ that is accessible infinitely often. Then $\mu(U^A) \in [q_\alpha, q_\alpha + \epsilon_\alpha)$, α is reset only finitely often, the definition of TP_s is ended at α only finitely often, and exactly one of the following hold:

(i) there is a $j \leq i$ and an $N_{e,j}$-node $\beta \leq \alpha$ and a stage s_0 such that $I_{e,j} \in H_e^D$ permanently after s_0 and $I_{e,j} \not\subseteq V_e^B$;

(ii) there is an $N_{e,i}$-node $\beta \leq_L \alpha$ and a stage s_0 such that $r_\beta[s_0] \neq 0$, β is not reset after s_0, $I_{e,i} \subseteq V_e^B$ and $I_{e,i} \notin H_e^D[s]$ at any $s \geq s_0$.

PROOF. Suppose that α is the leftmost node of length $|\alpha|$ which is accessible infinitely often, and assume inductively that there is a stage s_0 such that $\mathrm{TP}_s \not<_L \alpha$ and the definition of TP_s is not ended at any $\beta \subset \alpha$ for all $s > s_0$, no N-node $\beta \subset \alpha$ is reset after s_0, and all N-nodes $\beta \subset \alpha$ satisfy (i) or (ii) by s_0.

First we establish that $\mu(U^A) \in [q_\alpha, q_\alpha + \epsilon_\alpha)$. Let $\gamma = \alpha^-$ be the parent of α. Inductively, $\mu(U^A) \in [q_\gamma, q_\gamma + \epsilon_\gamma)$, and by the hat-trick, there are infinitely many true stages when $\mu(U^A[s]) \in [q_\gamma, q_\gamma + \epsilon_\gamma)$. At every true stage after s_0, we also have $\mathrm{TP}_s \supset \gamma$. If $\mu(U^A) < q_\alpha$, then at all true stages after some $s_1 \geq s_0$ some outcome of γ to the left of α would be accessible, contradicting that α is the leftmost node of length $|\alpha|$ which is accessible infinitely often. If $\mu(U^A) \geq q_\alpha + \epsilon_\alpha$ then there is a u such that $\mu(U^{A\restriction u}) \geq q_\alpha + \epsilon_\alpha$ and a stage $s_2 \geq s_0$ when $A \restriction u$ has settled. After s_2, α will never be accessible, again a contradiction. So $\mu(U^A) \in [q_\alpha, q_\alpha + \epsilon_\alpha)$.

Next we verify that α is reset only finitely often. By assumption on α, it is reset only finitely often by step 2 when $\mathrm{TP}_s <_L \alpha$. As $\mu(U^A) \in [q_\alpha, q_\alpha + \epsilon_\alpha)$, eventually $\mu(U^A[s])$ does not drop below q_α, so α is is reset only finitely often at even stages. By the induction assumption, eventually all N-nodes $\beta \subset \alpha$ reach a limit state and (II) will not hold for any such β after s_0. These are the only places in the construction where α may be reset.

Next we show that α reaches a limit state, satisfying (i) or (ii). If some N_e-node $\beta \subset \alpha$ satisfies (i) then α also satisfies (i) after s_0. Thus (I) or (II) of the construction will hold only finitely often for α, since α is only active finitely often. Otherwise, suppose that every N_e-node $\beta \subset \alpha$ satisfies (ii) by s_0. If some $N_{e,i}$-node $\alpha' <_L \alpha$ has $r_{\alpha'}[s_0] \neq 0$, then that restraint is permanent and α satisfies (i) due to α'. Then $\Gamma^A(z_{\alpha'}) \downarrow$ and eventually $h(z_{\alpha'}, s) = 1$ for all sufficently large s; so α will eventually satisfy (III) and $I_{e,i}$ will be removed permanently from H_e^D.

Suppose that no $N_{e,i}$-node $\alpha' <_L \alpha$ has $r_{\alpha'}[s_0] \neq 0$. If (II) holds for α at any stage after s_0, then α will impose a permanent restraint, $\Gamma^A(z_\alpha) \downarrow$ and $I_{e,i}$ will eventually be removed permanenly from H_e^D via (III). Then α will satisfy (ii). It suffices now to show that if (II) never holds for α then $I_{e,i} \not\subseteq V_e^B$ and $I_{e,i} \in H_e^D$ permanently.

Suppose that α never imposes a permanent restraint. We argue that $I_{e,i}$ is permanently in H_e^D after some $s_1 \geq s_0$. At any α-stage s after s_0, if $I_{e,i} \notin H_e^D[s]$ then it will be put into $H_e^D[s+1]$ as α will satisfy (I). We argue that, because A is low, $N_{e,i}$-nodes to the right of α will satisfy (III) only finitely often. Suppose that some $N_{e,i}$-node right of α satisfies (III) infinitely often; let β be the leftmost such node. If β satisfies (III) at s then some $N_{e,i}$-node $\beta' \leq_L \beta$ must have previously satisifed (II), imposed a restraint, and defined a computation $\Gamma^A(z_{\beta'})[s]$ (if it wasn't already defined). Thus no $N_{e,i}$-node imposes a permanent restraint, and no computation axiom $\Gamma^A(z_\gamma)[s]$ for any $N_{e,i}$-node γ is permanently valid. But the Δ_2^0 approximation of A ensures that each computation axiom is eventually permanently valid or permanently invalid. Thus infinitely many different computation axioms $\Gamma^A(z_\gamma)[s]$ must be defined during the construction, for some $N_{e,i}$-node γ. But then, $\Gamma^A(z_\gamma) \uparrow$ in the limit, and since A is low, $h(z_\gamma, s) = 0$ for all but finitely many s. Thus (III) can hold at only finitely many stages, and $I_{e,i}$ is permanently in H_e^D. Hence α satisfies (i) (with $\alpha' = \alpha$).

Finally we verify that if $I_{e,i} \subseteq V_e^B$ then eventually the measure condition (9) is satisfied and α will satisfy (ii). Let v be the use of the computation $I_{e,i} \subseteq V_e^B$, and let s_2 be the second α-stage after s_0 such that

$$B[s_2] \upharpoonright v = B \upharpoonright v \text{ and } E^{B \upharpoonright v}[s_2] \cap U^A[s_2] = E^{B \upharpoonright v} \cap U^A \text{ (as sets of strings)}$$

(such s_2 exists because of the hat-trick). Every string in $E^{B \upharpoonright v}[s_2] - E^{B \upharpoonright R_\alpha}[s_2] - U^A[s_2]$ is in $J_\gamma[s_2]$ for some $\gamma > \alpha$; as otherwise it would be removed in step 2 of the construction contradicting the choice of s_2. Let

$$Z = \{\gamma : \gamma > \alpha \text{ and } r_\gamma[s_2] \neq 0\}$$

be the lower-priority nodes with nonzero restraint at s_2. Then

$$\begin{aligned}
\mu(E^{B \upharpoonright v}[s_2] - E^{B \upharpoonright R_\alpha}[s_2] - U^A[s_2]) &\leq \sum_{\gamma \in Z} \mu(J_\gamma[s_2]) \\
&\leq \sum_{\gamma \in Z} 2\epsilon_\gamma \\
&\leq \epsilon_\alpha
\end{aligned}$$

by Lemma 2.15 and (8). Thus eventually (9) is satisfied, so if $I_{e,i} \subseteq V_e^B$ then α satisfies (i). □

Having established that each N-node reaches a limit state, we can easily verify that each requirement is satisfied.

LEMMA 2.17. For all e there is an $I_{e,i}$ such that $I_{e,i} \in H_e^D$ and $I_{e,i} \not\subseteq V_e^B$. Thus each requirement N_e is satisfied.

PROOF. If some $N_{e,i}$-node on the true path satisfies (i) of Lemma 2.16, then N_e is satisfied by $I_{e,i}$ for the least such i. To see that there is always such an i for each e, note that if not, every N_e-node on TP must satisfy (ii), and thus $I_{e,i} \subseteq V_e^B$ for each i. But there are 2^{k_e} many such $I_{e,i}$, each has measure 2^{-k_e}, and they are pairwise disjoint. Then $\mu(V_e^B) \geq 2^{k_e} \cdot 2^{-k_e} = 1$, which contradicts $\mu(V_e^B) \leq q_e < 1$. □

Finally we verify that requirement R is satisfied.

LEMMA 2.18. $A \leq_{LR} B$.

PROOF. By the definition of E, once an interval appears in U^A via a permanent computation it will henceforth always be in E^B with the same use. Thus $U^A \subseteq E^B$. We must verify that $\mu(E^B) < 1$. Since $\mu(U^A) \leq \frac{1}{4}$, it suffices to show that

$$\mu(E^{B\upharpoonright n}[s] - U^A[s]) \leq \frac{1}{2}$$

for all $n \in \mathbb{N}$ and sufficiently large s. Fix n and let s_0 be a stage such that

$$B \upharpoonright n[s_0] = B \upharpoonright n \text{ and } E^{B\upharpoonright n}[s_0] - U^A[s_0] = E^{B\upharpoonright n} - U^A$$

(as sets of strings). Then for all $s \geq s_0$ we have

$$E^{B\upharpoonright n}[s] - U^A[s] \subseteq \bigcup_\alpha J_\alpha[s],$$

and by Lemma 2.15 and the fact that at any time there is at most one node of each length with nonzero restraint, $\mu(E^{B\upharpoonright n}[s] - U^A[s]) \leq \sum_e 2^{-e-2} = \frac{1}{2}$. □

This completes the proof of Theorem 2.13. □

In section 3.5 we sketch how this construction can be adapted to the case where A is c.e. and LR-incomplete (rather than low), and discuss the obstacles to the most general case where A is Δ^0_2 and LR-incomplete.

The key aspect of this construction is the fact that *movement of TP_s depends only on the approximation of A*. That is, if α is accessible at t, and later $\text{TP}_s <_L \alpha$ for

$s > t$, then there must have been an A-change between s and t which removed some strings from U^A and made the approximation to $\mu(U^A)$ decrease. This is the fact that allows us to restrain A in order to prevent α from being reset. This seems to be a significant limitation on the technique; it is unclear how, if at all, this construction could be combined with other requirements which involve branching on the tree of strategies that does not depend solely on A, for instance minimal pair or non-cupping-style requirements in which the outcomes depend on convergence of computations rather than on the approximation to A. However, it can be combined with the P-strategy of Theorem 2.1. Although sometimes the P-strategy causes the construction to move to the left independently of A-changes, for instance when an attack is implemented for the first time, Lemma 2.6 ensures that if infinitary outcomes are accessed infinitely often, then infinitely often this will be due to an A-change, after an attack changes from pending to waiting. This is sufficient for the N-strategy to succeed; we just need to modify condition (10) so that v is also larger than the use of any computations $\sigma \subsetneq V^A[s]$ for pending P-attacks σ which would cause the N-node to be reset if re-implemented.

3.4. Differences with the Turing degrees. Theorem 2.13 displays a difference between the c.e. and Δ_2^0 Turing degrees and the c.e. and Δ_2^0 LR-degrees. Yates **[75]** constructed a Δ_2^0 Turing degree which is incomparable with all c.e. Turing degrees except $\mathbf{0}$ and $\mathbf{0}'$. Yates used an oracle construction with a $\emptyset'$ oracle to construct the required Δ_2^0 set A. It is possible to adapt his construction to also ensure that A is low (the proof is sketched below). Hence there is a low Δ_2^0 Turing degree that is incomparable with all intermediate c.e. Turing degrees. In particular, it has no incomplete c.e. Turing degree above it. This is in contrast to Theorem 2.13 which shows that every low Δ_2^0 LR-degree is bounded by an LR-incomplete c.e. LR-degree.

It is possible to further adapt Yates's construction to make the set A be low and non-low-for-random in addition to being incomparable with all intermediate c.e. Turing degrees. We note that if we apply Theorem 2.13 to such an A, we obtain a c.e. set B such that $A <_{LR} B$ but $A \not\leq_T B$. Hence the construction of Theorem 2.13 does not automatically produce $B \geq_T A$.

We sketch the construction of a Δ_2^0 set which is low, non-low-for-random and Turing incomparable with all c.e. sets of intermediate Turing degree. We build A via finite extensions $\alpha_0 \subseteq \alpha_1 \ldots$, so $A = \cup_s \alpha_s$. We have a listing W_e of all c.e. sets, and the

W_e are uniformly computable from $\emptyset'$. Φ_e is a standard listing of Turing functionals. To make A low we use the usual technique of 'forcing the jump': at stage s if there is a string $\tau \supset \alpha_s$ such that $\Phi_e^\tau(e)\downarrow$, then make $A \supset \tau$ by suitable choice of α_{s+1}. The strategy for making $W_e \not\leq_T A$ if W_e is noncomputable is the usual strategy of looking for splittings of Φ_i for each i. To make $A \not\leq_T W_e$ if $\emptyset' \not\leq_T W_e$, we utilise a function $f \leq_T \emptyset'$ such that f is not dominated by any function of degree $< \mathbf{0}'$. We use a bounded search, bounded at stage s by $f(s)$, to search for an x such that $A(x)$ is not yet defined and $\Phi_i^{W_e}(x)\downarrow$. We can then define $A(x) \neq \Phi_i^{W_e}(x)$. Because we use only a bounded search, we must place the requirements in a finite injury setting, to allow a higher priority requirement to take action as soon as its bounded search finds a suitable candidate. This is straightforward and may be found in Odifreddi [**62**] Proposition XI.3.6.

The only remaining requirement is to make A non-low-for-random. Fix a member U of a universal oracle Martin-Löf test, and a listing $\langle V_i, p_i\rangle$ of Σ_1^0 classes along with a rational bound $p_i < 1$ such that $\mu(V_i) \leq p_i$. We can assume that the set U^τ is finite and uniformly computable from $\tau \in 2^{<\omega}$. Given any $\alpha \in 2^{<\omega}$, we want to find an extension $\alpha' \supset \alpha$ such that $U^{\alpha'} \not\subseteq V_i$. Using the $\emptyset'$ oracle we can search for an α' and a string σ such that

$$\sigma \in U^{\alpha'} \text{ and } \sigma \not\subseteq V_i$$

and make $A \supset \alpha'$. Such an α' and σ are guaranteed to exist: let $X \in 2^\omega$ be such that $X \notin V_i$; such X exists since $\mu(V_i) < 1$. There is a $Z \supset \alpha$ such that X is not random relative to Z (for instance, a Z such that $X \leq_T Z$). Since U is a universal test, there is a σ and an n such that $\sigma \subset X$ and $\sigma \in U^{Z\restriction n}$. The string $\alpha' = Z \restriction n$ and σ are as required. We can combine this with the previous strategies in a finite injury setting using a $\emptyset'$ oracle to construct the set A.

3.5. The N-strategy with A LR-incomplete and c.e. It was originally hoped that the construction of Theorem 2.13 could be combined with the LR-incompleteness strategy of section 1 to show that every incomplete Δ_2^0 LR-degree is bounded by an incomplete c.e. LR-degree; that is, that above any LR-incomplete Δ_2^0 set there is an LR-incomplete c.e. set. Unfortunately there are obstacles to performing the construction in this most general case. Indeed, it is not known if every incomplete Δ_2^0 LR-degree is bounded by an incomplete c.e. LR-degree. However, the construction of Theorem 2.13 can be combined with the strategy of section 1 in the specific case when the set A is c.e. (rather than only Δ_2^0) and LR-incomplete. Although this result is not of interest in itself, the technique might be of use for other constructions in the LR-degrees. We now sketch how the construction of Theorem 2.13 may be modified to work with an LR-incomplete c.e. set, and discuss in section 3.6 the difficulties with the more general case when A is an arbitrary LR-incomplete Δ_2^0 set.

As in Theorem 2.13 we will put strings δ into H^D, wait for $\delta \subseteq V^B$, then remove δ from H^D and restrain B on the use. Again, we will use subrequirements $N_{e,i}$, and the N_e-nodes must co-ordinate their actions. All $N_{e,i}$-nodes will work with the same clopen set δ. In this case, when an N-node α wishes to remove δ from H^D and impose a B-restraint, it must use the incompleteness of A to (attempt to) prevent A from changing and the construction moving to the left of α. The method for attempting to preserve A is the same as in the P-strategy of Theorem 2.1. Each clopen set δ will be associated with a string $\rho \in U^{\emptyset'}[s]$. When we have $\delta \subseteq V^B$, we will restrain B and try to prevent A from changing by enumerating ρ into an A-Σ_1^0 class F^A. Once $\emptyset'$ changes and ρ is removed from $U^{\emptyset'}$, we can remove δ from H^D on the assumption that A will not change below the use of $\rho \in F^A$. Since we threaten to make $U^{\emptyset'} \subseteq F^A$, we are guaranteed that our attempted A-restraint will succeed sufficiently often to make $\mu(F^A) = 1$, which assures us of enough successful attacks δ to ensure $H^D \not\subseteq V^B$.

Using this strategy, it is possible that an attempted A-restraint will neither succeed nor fail, when it corresponds to a true string $\rho \in U^{\emptyset'}$. In such a case, we will be unable to remove δ from H^D. Say that such an attempt is *stalled.* Since there are infinitely many such true strings, we cannot fix the size of all δ's in advance. Instead, each subrequirement will correspond to a string ρ from the approximation to $U^{\emptyset'}$, and will choose δ the same size as ρ. Thus the measure of those δ's which we cannot remove from H_e^D is bounded by $\mu(U^{\emptyset'})$. We will need infinitely many subrequirements for each N_e, as we can use each string ρ from the approximation of $U^{\emptyset'}$ at most once, and we do not know in advance which will succeed, which will fail and which will stall.

Note that ρ is associated with δ, and not with any individual N-node. In particular, all $N_{e,i}$-nodes share the same ρ. This is in contrast to the earlier version of the N-strategy where every N-node had its own computation $\Gamma^A(z_\alpha)$.

Since $\mu(H_e^D)$ is bounded by the measure of the Martin-Löf test member $U^{\emptyset'}$ that N_e is working with, N_e must work with a U such that $\mu(U^{\emptyset'}) < 2^{-e-1}$ to ensure $\mu(H^D) < 1$.

We also want $\mu(U^{\emptyset'}) < 1 - \mu(V_e^B)$ as in earlier constructions. Let

$$\langle \tau_0, \rho_0 \rangle, \langle \tau_1, \rho_1 \rangle, \ldots$$

be a listing of the axioms from $U^{\emptyset'}$ that are valid at some stage, in the order in which they become valid (as in (6)). Subrequirement $N_{e,i}$ will work with some computation $\langle \tau, \rho \rangle$ from this list, and a clopen set δ_i such that $\mu(\delta_i) = \mu(\rho)$. All $N_{e,i}$-nodes share the same computation $\langle \tau, \rho \rangle$ and δ_i. The $N_{e,i}$-nodes work as follows. If at stage s there is a higher-priority subrequirement $N_{e,j}$ with $\delta_j \in H_e^D[s]$ but $\delta_j \not\subseteq V_e^B[s]$, then (from $N_{e,i}$'s point of view) N_e appears satisfied and $N_{e,i}$ need not do anything. So $N_{e,i}$ will only work when all higher-priority N_e subrequirements have either succeeded in restraining δ_j into V_e^B, or have abandoned their attempt for reasons described below.

When $N_{e,i}$ first needs to work, it chooses its clopen set δ_i of the same measure as $\mu(\rho)$ and disjoint from any δ_j, $j < i$ for which δ_j is already restrained in $V_e^B[s]$. It puts δ_i into H_e^D, where it remains until a $\emptyset'$-change invalidates the computation $\rho \in U^{\emptyset'}[s]$. Once δ_i is in H_e^D, all $N_{e,i}$-nodes α wait for $\delta_i \subseteq V_e^B[s]$ via an *acceptable* computation whose use v does not capture too much junk in E^B (ie, condition (9)). We say that $N_{e,i}$ is *waiting* at stage s if $\delta_i \in H_e^D[s]$ but $\delta_i \not\subseteq V_e^B[s]$ by a computation satisfying (9). for any $N_{e,i}$-node α. When some α sees $\delta_i \subseteq V_e^B[s]$ via an acceptable computation, α imposes a B-restraint to preserve the computation $\delta_i \subseteq V_e^B[s]$. At this point α would like to remove δ_i from H_e^D, but this would jeopardise the $N_{e,i}$-strategy from the point of view of $N_{e,i}$-nodes to the left of α, if the construction later moves to the left. α must use the LR-incompleteness of A to attempt to restrain a suitable initial segment $A \upharpoonright v$ as in (10) to prevent the construction moving to the left of α. It does this by defining a computation $\rho \in F^A[s+1]$ with use v.

We then wait for an $\emptyset'$-change to invalidate the computation $\rho \in U^{\emptyset'}[s]$. During this period (while the computation $\rho \in U^{\emptyset'}[s]$ is valid and $\rho \in F^A[s]$) we say that $N_{e,i}$ is *pending*. If such a $\emptyset'$-change never occurs, then $\delta_i \in H_e^D$ permanently, even though δ_i may be contained in V_e^B. However, then we must have $\rho \in U^{\emptyset'}$, and such ρ's are bounded in measure by $\mu(U^{\emptyset'}) < 2^{-e-2}$. Otherwise, eventually an $\emptyset'$-change occurs. When this happens we can remove δ_i from H_e^D permanently, on the assumption that the restraint $A \upharpoonright u$ is successful. If the restraint is successful then $N_{e,i}$ has forced $\mu(V_e^B)$ to increase by $\mu(\delta_i)$ while $\mu(H_e^D)$ has not increased; contributing towards N_e. If the restraint is not successful, then TP_s will move to the left and α will be reset. In this case $N_{e,i}$ is unsuccessful. However we can argue that the LR-incompleteness of A guarantees that enough subrequirements will be successful (ie. the restraint will succeed) that N_e will be satisfied.

Specifically, we argue that if N_e is not satisfied, then infinitely many subrequirements will start working. For infinitely many of these, namely, those whose computations $\rho \in U^{\emptyset'}$ are permanent, we will have $\delta_i \subseteq V_e^B$ acceptable for some α on or left of the true path, and thus we have $\rho \in F^A$. But then we have $U^{\emptyset'} \subseteq F^A$, and since A is LR-incomplete, $\mu(F^A) = 1$. Each string $\rho \in F^A$ corresponds to a clopen set $\delta \subseteq V_e^B$, since the B-restraint is maintained as long as the computation $\rho \in F^A[s]$ is valid. Since $\mu(U^{\emptyset'}) < 1 - p_e$ by choice of U, F^A contains more than p_e measure worth of successful ρ's (recall that p_e is a bound on $\mu(V_e^B)$). Further, since $N_{e,i}$ chooses its δ disjoint from those of higher-priority subrequirements, distinct strings in F^A correspond to disjoint clopen sets in V_e^B. But then $\mu(V_e^B) > p_e$, a contradiction. Thus N_e is eventually satisfied: there is some δ permanently in H_e^D but $\not\subseteq V_e^B$.

3.6. Obstacles when A is Δ_2^0 and LR-incomplete. In both Theorem 2.13 and the construction sketched above, we want to restrain segments of A from changing, to avoid N-nodes being reset and some B-computation $\delta \subseteq V^B$ from being destroyed. When we want to restrain A to protect $\delta \subseteq V^B$, we define a computation: $\Gamma^A(z)$ in the first case, $\rho \in F^A$ in the second. If A later changes, then $\delta \subseteq V^B$ may be destroyed by B-enumerations. In the case when A is c.e., if the attempted A-restraint fails then A changes below the use of the computation and the computation is destroyed permanently. The N strategy can then try again with δ. In the case where A is Δ_2^0, the computation might only be destroyed temporarily, and later A might be restored to its previous state. Then we would have $\Gamma^A(z)\downarrow$ or $\rho \in F^A$ but $\delta \not\subseteq V^B$.

This is not a problem in Theorem 2.13 because each N-node uses its own computation $\Gamma^A(z_\alpha)$. Eventually some $N_{e,i}$-node $\beta \geq_L \alpha$ will have a computation $\Gamma^A(z_\beta)\downarrow$ protecting $\delta \subseteq V^B$. In the end there might be some spurious computations $\Gamma^A(\alpha)\downarrow$ for $\alpha <_L \beta$, but these are harmless.

In the LR-incomplete case however, all $N_{e,i}$-nodes must share the same string ρ from $U^{\emptyset'}$ since they all share the same δ and $\mu(\delta)$ is tied to $\mu(\rho)$. Hence they must also share the same computation $\rho \in F^A$. This can be done in the c.e. case, since then the computation $\rho \in F^A$ is valid iff the corresponding B-restraint is respected. In the Δ^0_2 case however, if the computation $\rho \in F^A$ becomes spurious then there would be no other computations available to protect $\delta \subseteq V^B$. Also, the spurious $\rho \in F^A$ would contribute useless measure to F^A which we could not attribute to measure in V^B. In that case we could not argue that $\mu(V^B) \geq \mu(F^A) = 1$.

4. Downward density and other results

We briefly mention some related results. Since this work was done, Barmpalias [2] has proved that LR-below any non-low-for-random Δ^0_2 set A there is a non-low-for-random c.e. set. That is, if $A \not\leq_{LR} \emptyset$ is Δ^0_2 then

$$\exists \text{ c.e. } B \text{ such that } \emptyset <_{LR} B <_{LR} A.$$

This is a dual of Theorem 2.13 in the general case of any non-low-for-random Δ^0_2 set A.

By relativising an earlier splitting theorem from [5], Barmpalias [2] also established the upward density of the Δ^0_2 LR-degrees. That is, if A is Δ^0_2 and $<_{LR} \emptyset'$ then there is a Δ^0_2 set B such that $A <_{LR} B <_{LR} \emptyset'$. It is not known (as discussed earlier) whether the set B can be made c.e.. Theorem 2.13 can be considered a stronger version of Barmpalias's upwards density result in the specific case when A is low.

CHAPTER 3

Nonmitoticity and LR-degrees

1. Splittings and mitoticity

Let A be a c.e. set. A *splitting* of A is a pair of disjoint c.e. sets C, D such that $C \cup D = A$. Note that $C, D \leq_T A$ and $A \leq_T C \oplus D$ for such a splitting. In the context of c.e. Turing degrees, a splitting is called (Turing) *mitotic* if $C \equiv_T D \equiv_T A$, and the set A is called mitotic if it has a mitotic splitting. If it does not have a mitotic splitting then it is (Turing) *nonmitotic.*

Mitoticity and nonmitoticity were first studied in the context of the c.e. Turing degrees. Lachlan [**41**] constructed a nonmitotic c.e. set, and Ladner [**43, 42**] proved further results about nonmitotic sets and their degrees, including the existence of a non-zero completely mitotic c.e. degree, that is, a non-zero c.e. Turing degree in which all the c.e. sets are mitotic. Downey and Slaman [**24**] improved this result and gave an alternate method for constructing completely mitotic Turing degrees.

Some work has also been done on mitoticity in the wtt-degrees. Ladner's construction from [**42**] in fact gives a completely mitotic wtt-degree. Downey [**18**] showed that every array nonrecursive c.e. Turing degree contains a completely mitotic wtt-degree, and Downey and Stob [**25**] point out that not all c.e. Turing degrees contain completely mitotic wtt-degrees. Study of mitoticity in the Turing and wtt-degrees is motivated in part by the connections with notions such as autoreducibility and contiguity; see Downey and Stob [**25**] for a survey on mitoticity and other splitting results.

In this chapter we will consider nonmitoticity in the LR-degrees. Call a c.e. set A *LR-mitotic* if it has a splitting C, D such that $A \equiv_{LR} C \equiv_{LR} D$; otherwise A is LR-nonmitotic. We prove first that there is an LR-nonmitotic c.e. set of Turing degree $\mathbf{0}'$, then describe how to modify the construction to make the set low or to avoid a non-trivial LR-upper cone. We show that the notions of LR-mitoticity and Turing mitoticity differ on the non-low-for-random c.e. sets by constructing a c.e. set that is LR-mitotic but Turing nonmitotic. It is not known if there is a non-zero completely

LR-mitotic LR-degree, though we observe as a corollary of previous results that there is no completely Turing-mitotic LR-degree.

Theorem 3.1 is joint work with George Barmpalias and has been published in [8] (see also Appendix A). The notion of LR-nonmitoticity was motivated by André Nies (in private communications), and was first studied in [8].

2. A non-LR-mitotic Turing complete c.e. set

THEOREM 3.1. There is a c.e. set A which cannot be split into two c.e. sets X, Y such that $X \equiv_{LR} Y \equiv_{LR} A$. Moreover, $A \geq_T \emptyset'$.

We will construct the required set A. Let $\langle X_i, Y_i, V_i, q_i \rangle$ be a listing of all quadruples of disjoint c.e. sets X, Y, an oracle Σ^0_1 class V and a rational $q < 1$ such that $\mu(V^Z) \leq q$ for all $Z \in 2^\omega$. We will construct A as well as a sequence of oracle Σ^0_1 classes T_i to satisfy the requirements

$$R_i: \qquad X_i \cup Y_i = A \Rightarrow T_i^A \not\subseteq V_i^{X_i} \vee T_i^A \not\subseteq V_i^{Y_i}.$$

The T_i are uniformly Σ^0_1, and we will ensure that $\mu(T_i^A) \leq 2^{-i-2}$, so that we can set

$$T^A = \cup_i T_i^A$$

which is a Σ^0_1 class and has measure < 1. By Theorem 1.5, the requirements R_i suffice to ensure that if X, Y is a c.e. splitting of A then $A \not\leq_{LR} X$ or $A \not\leq_{LR} Y$. We will argue at the end that the set A automatically satisfies $A \geq_T \emptyset'$.

The strategy for satisfying R_i is as follows. We will put a clopen set σ of size ϵ into T_i^A and wait until $\sigma \subseteq V_i^{X_i}$ and $\sigma \subseteq V_i^{Y_i}$ with some use v such that $X_i[s] \cup Y_i[s] \upharpoonright v = A[s] \upharpoonright v$ (if this never occurs then R_i is already satisfied). When we see σ in both $V_i^{X_i}$ and $V_i^{Y_i}$, we can remove σ from T^A by enumerating a single number into A, and then restrain $A \upharpoonright v$ to prevent any other numbers from entering A. Since we enumerate only a single number x into A below v, at most one of $X_i \upharpoonright v$, $Y_i \upharpoonright v$ can change later, if they are to be a splitting of A. Hence one of $X_i \upharpoonright v$, $Y_i \upharpoonright v$ must be fixed (assuming that X_i, Y_i are a splitting of A) and thus $\sigma \subseteq V_i^{X_i}$ or $\sigma \subseteq V_i^{Y_i}$ permanently. We have forced one of $V_i^{X_i}$, $V_i^{Y_i}$ to increase in size by at least ϵ. We can now repeat with a new σ of size ϵ. Since $V_i^{X_i}$ and $V_i^{Y_i}$ are bounded in measure by q_i, after at most $2q_i/\epsilon$ many repetitions we will have some σ which is never covered by $V_i^{X_i}$ or $V_i^{Y_i}$, satisfying R_i. At no time do we ever have more than ϵ in T_i^A, so $\mu(T_i^A) \leq \epsilon$. We can combine these strategies in a finite injury setting.

For each i, let $p_i = 2^{-i-2} \cdot (1 - q_i)$; requirement R_i will work with clopen sets of size p_i. It will have parameters σ_i^X, σ_i^Y which are clopen sets, $\sigma_i := \sigma_i^X \cup \sigma_i^Y$, and u_i which is the use of the computation $\sigma_i \in T_i^A$ (when it is defined). Initially they are all undefined, and we denote their value at the end of stage s by the suffix $[s]$. Requirement R_i *requires attention* at stage $s+1$ if $u_i[s], \sigma_i^X[s], \sigma_i^Y[s]$ are undefined or if they are defined and

$$\sigma_i^X \subseteq V_i^{X_i}[s], \sigma_i^Y \subseteq V_i^{Y_i}[s] \text{ with use } v \tag{12}$$

and

$$X_i[s] \cup Y_i[s] \upharpoonright v = A[s] \upharpoonright v. \tag{13}$$

The construction. At stage 0 do nothing; at stage $s+1$ let i be the least such that R_i requires attention at $s+1$.

If $u_i[s], \sigma_i^X[s], \sigma_i^Y[s]$ are undefined then let $\sigma_i^X[s+1]$ be the leftmost clopen set of measure $p_i/2$ which is $\subseteq 2^\omega - V_i^{X_i}[s]$ and $\sigma_i^Y[s+1]$ be the leftmost clopen set of measure $p_i/2$ which is $\subseteq 2^\omega - V_i^{Y_i}[s]$, choose $u_i[s+1]$ fresh (i.e., larger than any numbers used before) and declare $\sigma_i = \sigma_i^X \cup \sigma_i^Y \in T_i^A[s+1]$ with use $u_i[s+1]$.

If R_i satisfies (12) and (13) then enumerate $u_i[s]$ into $A[s+1]$ to remove σ_i from $T_i^A[s+1]$ and set $\sigma_j^X[s+1], \sigma_j^Y[s+1], u_j[s+1]$ undefined for all $j \geq i$.

Verification. Since we defined $p_i < 1 - q_i \leq \mu(2^\omega - V_i^Z)$ for any Z, when we need to choose clopen sets $\sigma_i^X \subseteq 2^\omega - V_i^{X_i}[s]$ and $\sigma_i^{Y_i} \subseteq 2^\omega - V_i^{Y_i}[s]$ they will exist. Note that if $u_i[s]$ and $u_j[s]$ are both defined and $i < j$ then $u_i[s] < u_j[s]$. In particular, if R_j receives attention at stage $s+1$ and enumerates $u_j[s]$ into A, this enumeration will not affect R_i's computation $\sigma_i \in T_i^A[s]$. Also, if R_i enumerates $u_i[s]$ into A then this will be below the use $u_j[s]$ of $\sigma_j \in T_j^A[s]$, so $\sigma_j[s]$ will be removed from T_j^A when σ_j is set undefined. Note that at any stage either $T_i^A[s] = \emptyset$ or $T_i^A[s] = \sigma_i[s]$, which has measure $\leq p_i$. So T^A is an A-Σ_1^0 class and $\mu(T^A) < 1$.

Now we show by induction that each R_i requires attention only finitely often and is eventually satisfied. Assume inductively that s_0 is the last stage at which some $R_{i'}$ for $i' < i$ requires attention. After s_0, no number $< u_i[s]$ will ever be enumerated into A. Except for stage $s_0 + 1$, every time R_i receives attention due to $u_i[s] \uparrow$, it must have received attention at the previous stage due to (12) and (13). But when (12) and (13) hold, we will enumerate only the single number $u_i[s]$ into A. If (12) and (13) hold again at some $t > s$, then one of $X_i[t] \upharpoonright u_i[s] = X_i[s] \upharpoonright u_i[s]$ or $Y_i[t] \upharpoonright u_i[s] = Y_i[s] \upharpoonright u_i[s]$ must hold since only one number $u_i[s]$ has entered A between s and t. But then either $\sigma_i^X[s] \subseteq V_i^{X_i}[t]$ or $\sigma_i^Y[s] \subseteq V_i^{Y_i}[t]$, and in fact this computation is permanent since no number less than the use v will ever enter A after s_0. So one of $\mu(V_i^{X_i}), \mu(V_i^{Y_i})$ has increased by $p_i/2 = \mu(\sigma_i^A) = \mu(\sigma_i^Y)$. This can happen at most $4q_i/p_i$ times since $\mu(V_i^{X_i}), \mu(V_i^{Y_i})$ are bounded by q_i. So R_i can require attention at most $4q_i/p_i$ times after s_0. Requirement R_i is therefore eventually satisfied since if it were not, it would require attention infinitely often which is impossible.

To argue that $\emptyset' \leq_T A$, let f be a computable function such that

$$X_{f(i)} = Y_{f(i)} = \emptyset, q_{f(i)} = 2^{-1}$$

and

$$V_{f(i)}^Z = \begin{cases} \{0\} & \text{if } i \in \emptyset' \\ \emptyset & \text{if } i \notin \emptyset' \end{cases}$$

for any $Z \in 2^\omega$. That is, $V_{f(i)}^Z$ contains the string '0' if $i \in \emptyset'$, or is empty otherwise. By the argument above, u_j reaches a limit for each j, and since u_j can only change if some number $\leq u_j[s]$ enters A at $s+1$, the oracle A can compute a stage t_j such that $u_j[s]$ has reached its limit by t_j. Since we always choose the leftmost clopen set for $\sigma_{f(i)}^X$ and $\sigma_{f(i)}^Y$, we have $i \in \emptyset'$ iff $i \in \emptyset'[t_{f(i)}]$. Hence $\emptyset' \leq_T A$. □

3. Low LR-nonmitotics and cone avoidance

The nonmitoticity strategy of Theorem 3.1 can easily be combined with other requirements in a finite injury setting. We briefly discuss the modifications necessary to combine the nonmitoticity strategy with negative requirements to construct, for instance, a low LR-nonmitotic c.e. set, or an LR-nonmitotic c.e. set that is not LR-above a given non-low-for-random Δ_2^0 set.

THEOREM 3.2. There is a low c.e. set that cannot be split into two c.e. sets of the same LR-degree. Given any non-low-for-random Δ_2^0 set B, there is a c.e. set $A \not\geq_{LR} B$ such that A cannot be split into two c.e. sets of the same LR-degree.

To make the set A low, we can use the usual strategy of restraining A when we see a computation $\Phi_e^A(e)[s]$ converge. Since the higher priority nonmitoticity requirements will act only finitely often, we will have

$$\exists^\infty s\ \Phi_e^A(e)[s]\downarrow \Rightarrow\ \Phi_e^A(e)\downarrow$$

which ensures that $A' \leq_T \emptyset'$ if met for all e.

To make the set A be $\not\geq_{LR}$ a given non-low-for-random Δ_2^0 set B, we can use Sacks restraints as in Theorem 2.1. Either of these negative strategies can be combined with the nonmitotic strategy in a finite injury setting, since each nonmitotic requirement acts only finitely often.

One modification is necessary to the nonmitoticity requirements in the presence of restraints however. If a higher priority negative requirement imposes a restraint on A, this restraint may prevent lower priority nonmitoticity requirements from removing some strings σ from T^A. Such strings would become permanent residents of T^A, contributing measure to $\mu(T^A)$. When this happens the nonmitoticity requirement must abandon σ and start with a new attack σ'. However we will no longer have that at all times T_i^A contains at most one attack. To keep the measure of T_i^A under control, requirement R_i must halve its quota each time a higher priority requirement acts (ie, imposes or changes its restraint). Each time a higher priority requirement acts, R_i must abandon any previous attack σ and must restart with a new attack σ' half the size of σ. In a finite injury setting, this will only happen finitely often before R_i's quota is fixed and then the verification can proceed as in Theorem 3.1. Since T_i^A consists of at most one attack of each size, we still ensure that $\mu(T^A) < 1$ by a suitable choice of the initial quotas. The details are a standard finite injury argument.

Obvious questions remain about which LR-degrees contain LR-nonmitotic c.e. sets. In particular, is there a *completely LR-mitotic LR-degree* - a c.e. LR-degree which does not contain an LR-nonmitotic c.e. set?

4. A Turing-nonmitotic but LR-mitotic non-low-for-random c.e. set

Clearly all LR-nonmitotic c.e. sets are Turing nonmitotic. Note that all low-for-random c.e. sets are trivially LR-mitotic since low-for-randomness is closed downwards under Turing reducibility. Ladner [43] showed that every noncomputable c.e. set computes a nonmitotic c.e. set. We can apply this result to a noncomputable low-for-random c.e. set to obtain a c.e. set which is Turing nonmitotic but (trivially) LR-mitotic. We now show that LR-nonmitoticity and Turing nonmitoticity differ also on non-low-for-random c.e. sets.

THEOREM 3.3. There is a non-low-for-random Turing nonmitotic c.e. set A and disjoint c.e. sets C, D such that

$$A = C \cup D \text{ and } A \equiv_{LR} C \equiv_{LR} D.$$

Let $\langle X_i, Y_i, \Phi_i, \Psi_i \rangle$ be a listing of all quadruples of disjoint c.e. sets X_i, Y_i and Turing functionals Φ_i, Ψ_i. Let V_i be a listing of all Σ_1^0 classes such that $\mu(V_i) < 1$. Fix a member U of a universal oracle Martin-Löf test. We will construct the required set A as well as an oracle Σ_1^0 class T which will satisfy the requirements

$$P_i: \qquad T^A \not\subseteq V_i$$

$$R_i: \qquad X_i \cup Y_i = A \Rightarrow \Phi_i^{X_i} \neq A \vee \Psi_i^{Y_i} \neq A.$$

We also construct the c.e. sets C, D such that $C \cap D = \emptyset$ and $C \cup D = A$, and oracle Σ_1^0 classes V^C, V^D such that

$$U^A \subseteq V^C, U^A \subseteq V^D \text{ and } \mu(V^C), \mu(V^D) < 1. \tag{14}$$

By Theorem 1.5 this ensures that $A \leq_{LR} C, D$; $A \equiv_{LR} C \equiv_{LR} D$ follows since $C, D \leq_T A$.

The requirements P_i are the standard non-low-for-randomness requirements: we put a clopen set σ into T^A, wait until $\sigma \subseteq V_i$ and then remove σ from T^A by enumerating into A. With $\mu(\sigma)$ fixed, after finitely many repetitions $\mu(V_i)$ will not be able to increase any further and we will satisfy P_e.

The requirements R_i are the Turing version of the nonmitoticity requirements. The basic strategy is similar to (in fact, simpler than) that of Theorem 3.1: choose a witness x not yet in A and wait until

$$\Phi^X(x)[s]\downarrow = 0 \text{ and } \Psi^Y(x)[s]\downarrow = 0 \text{ with use } u \text{ such that } X[s] \cup Y[s] \upharpoonright u = A[s] \upharpoonright u.$$

When this occurs, enumerate x into A and restrain A to prevent any other numbers $< u$ from entering A. Since at most one of $X \upharpoonright u, Y \upharpoonright u$ can change, at least one of

$\Phi^X(x), \Psi^Y(x) \neq A(x) = 1$. This is the strategy originally used by Lachlan [**41**] to construct a nonmitotic c.e. set.

Whenever a number is enumerated into A, we will also enumerate it into one of C or D, ensuring that $C \cup D = A$ and $C \cap D = \emptyset$. To meet (14), we will trace strings from U^A into V^C and V^D. That is, if we see $\rho \in U^A$ with use v then we will put ρ into V^C and V^D also with use v. The danger is that if some number $x < v$ is enumerated into A, then ρ might be removed from U^A. Since we can only enumerate x into one of C or D, say C, we will only be able to remove ρ from V^C. If no other numbers $< v$ ever enter A, then we will never be able to remove ρ from V^D and it will become junk. We need to keep this junk small so that $\mu(V^C)$ and $\mu(V^D)$ are < 1.

If the enumeration is on account of a P-requirement, the solution is simple. We can just enumerate two numbers x and $x-1$ into A (even though one is enough to fulfil P's desire of removing σ from T^A). We can put one into C and the other into D. We cannot do this for the nonmitotic requirements though, as they depend on there being only a single enumeration into A below the use u. However we can use the ideas from the 'cost function' construction of a noncomputable non-low-for-random ([**39**], see [**60**]). Let

$$\mathrm{cost}(x, s) = \mu\Big(\big\{\sigma : \sigma \in U^A[s] \text{ with use } > x\big\}\Big).$$

This is the amount of junk that we risk contributing to one of V^C, V^D if we enumerate x into A at stage s. We will give each requirement R_i a quota ϵ, and will only allow R_i to enumerate a number x if $\mathrm{cost}(x, s) \leq \epsilon$. By a familiar argument from cost function constructions, we can argue that R_i will eventually have a suitable x that it can use for diagonalising against Φ^X, Ψ^Y.

Each requirement P_i has a parameter σ_i which is a clopen set and z_i which is a number. They are initially undefined and may be redefined or declared undefined during the construction. It also has a quota 2^{-k_i}, which is the amount P_i is allowed to contribute to T^A. Initially we set $k_i = i + 2$, and k_i may be incremented during the construction.

Each requirement R_i has a parameter x_i which is initially undefined and may be defined or declared undefined during the construction. It also has a quota 2^{-j_i}, initially set to $j_i = i + 2$, which may be incremented during the construction. The suffix $[s]$ indicates the value of a parameter at the end of stage s of the construction.

Requirement P_i *requires attention* at $s+1$ if $\sigma_i[s]\uparrow$ and $z_i[s]\uparrow$, or if $\sigma_i[s]\downarrow$, $\sigma_i \in T^A[s]$ and $\sigma_i \subseteq V_i[s]$. We *take action* for P_i by doing the following. If $\sigma_i[s], z_i[s]\uparrow$ then choose $\sigma_i[s+1]$ to be a clopen set of measure $2^{-k_i}[s]$ that is disjoint from any previous $\sigma_i[t]$ used since the last stage when P_i was injured. Choose $z_i[s+1]$ to be a fresh odd number (ie, larger than any number used so far in the construction) and declare $\sigma_i[s+1] \in T^A[s+1]$ with use $z_i[s+1]+1$. If $\sigma_i[s], z_i[s]\downarrow$ and $\sigma_i[s] \subseteq V_i[s]$, then enumerate $z_i[s]+1$ into A and into C, and enumerate $z_i[s]$ into A and D. Choose a new $\sigma_i[s+1]$ and $z_i[s+1]$ as above.

Requirement R_i *requires attention* at $s+1$ if $x_i[s]\uparrow$, or if $x_i[s]\downarrow$, $\Phi_i^{X_i}(x_i)[s]\downarrow = 0$ and $\Psi_i^{Y_i}(x_i)[s]\downarrow = 0$ with use u, $x_i[s] \notin A$, and $X_i[s] \cup Y_i[s] \upharpoonright u = A[s] \upharpoonright u$. We *take action* for R_i by doing the following. If $x_i[s]\uparrow$ then choose x_i fresh (larger than any numbers used so far in the construction). Otherwise (when R_i requires attention for the second reason), check if

$$\mathrm{cost}(x_i[s], s) \leq 2^{-j_i[s]}. \tag{15}$$

If so, then enumerate $x_i[s]$ into A and into C. If not, then abandon the old $x_i[s]$ and choose $x_i[s+1]$ fresh.

To *injure* a requirement P_i at stage $s+1$ means to set $\sigma_i[s+1]\uparrow$ and $z_i[s+1]\uparrow$ and to set $k_i[s+1] = k_i[s]+1$. To injure a requirement R_i means to set $x_i[s+1]\uparrow$ and $j_i[s+1] = j_i[s]+1$.

We order the requirements in the order $P_0, R_0, P_1, R_1, \dots P_n, R_n \dots$.

The construction. Initially A, C, D are empty and T, V are empty (as sets of axioms). We give the construction of V^C and V^D in advance. At stage s, for each $\rho \in U^A[s]$ with some use u, put ρ into $V^C[s+1]$ and $V^D[s+1]$ (if it is not in there already) with use v, where v is the least even number $\geq u$.

Now for the construction of A, C, D. At stage 0 do nothing. At stage $s+1$, take action for the highest priority requirement that requires attention at stage $s+1$. Injure all lower priority requirements.

End of construction.

Verification. First, note that if a requirement P_i enumerates into A, then it enumerates two numbers $z, z+1$, and z goes into C and $z+1$ into D. Since all strings in V^C and V^D have even use, if a string is removed from U^A because of the enumeration of z or $z+1$, then it is removed from both V^D and V^C. So the P-requirements do not contribute junk to $\mu(V^C), \mu(V^D)$. If requirement R_i enumerates a number x into A at s, then it must satisfy (15). Hence the enumeration of x can contribute at most $2^{-j_i[s]}$ towards $\mu(V^D)$. By the construction, if R_i later enumerates another number, then it must have been injured since s and so its quota 2^{-j_i} would have halved. So each R_i can contribute at most twice its initial quota to $\mu(V^D)$, and by the initial choice of j_i, we have $\mu(V^D - U^A) \leq 2^{-1}$. Since $\mu(U^A) \leq 2^{-2}$, we have $\mu(V^D) < 1$. By the definition of V^C, V^D we have $U^A \subseteq V^C, U^A \subseteq V^D$. Hence $A \leq_{LR} C$ and $A \leq_{LR} D$. Note that actually $V^C = U^A$; in fact, $A \leq_T C$.

We must verify that each requirement requires attention only finitely often, and hence is eventually satisfied. Assume inductively that no requirement of higher priority than P_i requires attention at any stage $s \geq s_0$, and that s_0 is the least such. At s_0, we will define σ_i, z_i and they will remain defined thereafter. Each time P_i receives attention after s_0, it is because $\sigma_i \subseteq V_i$. But each time we choose a new clopen set for σ_i, we choose it disjoint from those previously used. So after receiving attention at most $2^{k_i[s_0]}$ many times after s_0, P_i will be satisfied since $\mu(V_i)$ cannot increase above $q_i < 1$.

Assume now that no requirement of higher priority than R_i requires attention at any stage $s \geq s_0$, and that s_0 is the least such. Note that j_i is fixed after s_0. If R_i requires attention at some stage s after s_0 such that (15) holds, then it will enumerate $x_i[s]$ into A and R_i will be permanently satisfied. We just need to argue that eventually this will occur. Assume to the contrary that R_i requires attention infinitely often after s_0, at stages $s_0 < s_1+1 < s_2+1 < \ldots$. Let $J_k = \{\rho : \rho \in U^A[s_k] \text{ with use } > x_i[s_k]\}$; at each s_k, we have $\mu(J_k) = \text{cost}(x_i[s_k], s_k) > 2^{-j_i}$. Since all lower priority requirements are injured at s_k, no numbers $< s_k$ will enter A after s_k. By the usual assumptions about the use of computations, all $\rho \in J_k$ have use $< s_k$. So $J_k \subseteq U^A$. Since $x_i[s_k+1]$ is chosen fresh, we have $\text{cost}(x_i[s_k+1], s_k) = 0$. At s_{k+1} we again have $\mu(J_{k+1}) = \text{cost}(x_i[s_{k+1}], s_{k+1}) > 2^{-j_i}$. As above, $J_{k+1} \subseteq U^A$. But since U^A is prefix-free, J_k and J_{k+1} must be disjoint. So during each interval $s_k < s \leq s_{k+1}$, $\mu(U^A)$ increases by at least 2^{-j_i}. But this is impossible since $\mu(U^A) < 1$. So eventually there must be an s_k at which (15) holds. Then R_i will be satisfied. $\square$

What we have done in this construction is to make $A \leq_T C$ by coding all enumerations into C, and to make enough extra enumerations into A but not into D to make A be low-for-random but non-computable relative to D.

A corollary of previous results is the fact that every c.e. LR-degree contains a Turing nonmitotic c.e. set. Ingrassia [**29**] (see Downey and Slaman [**24**] for a more accessible proof) showed that the degrees of nonmitotic c.e. sets are dense in the c.e. Turing degrees. That is, given c.e. sets $B <_T C$, there is a Turing nonmitotic c.e. set A with $B <_T A <_T C$. Barmpalias, Lewis and Soskova [**5**] showed that every c.e. LR-degree contains infinite chains of c.e. Turing degrees; in particular it contains c.e. sets $B <_T C$. Combining these two results we obtain the desired fact.

COROLLARY 3.4. Every c.e. LR-degree contains a Turing nonmitotic c.e. set.

As mentioned previously, it is not known if there is a completely LR-mitotic LR-degree; that is, a c.e. LR-degree which does not contain any LR-nonmitotic c.e. set. However Corollary 3.4 establishes the weaker result that there is no completely Turing mitotic LR-degree.

CHAPTER 4

A non-cuppable LR-complete c.e. set

1. Non-cupping and LR-completeness

In this chapter, we turn our attention to the LR-degree of $\emptyset'$. We will prove that the LR-degree of $\emptyset'$ contains a non-cuppable c.e. Turing degree, that is, a c.e. set that cannot be joined to $\emptyset'$ by any incomplete c.e. set. The theorem presented in this chapter is joint work with George Barmpalias, and has been published in [**8**] (see also Appendix A).

THEOREM 4.1. There is a c.e. set A such that $A \geq_{LR} \emptyset'$ and $A \oplus W \not\equiv_T \emptyset'$ for all c.e. sets $W <_T \emptyset'$.

The proof is given in section 2.

LR-completeness, the property of being $\geq_{LR} \emptyset'$, is a notion that has been of interest apart from studies of relative randomness. Dobrinen and Simpson [**17**] defined the notion of *almost everywhere domination*, in connection with the reverse mathematics of certain results from measure theory. A is almost everywhere dominating if the class of $X \in 2^\omega$ such that every total function $g \leq_T X$ is dominated by some total $f \leq_T A$ has measure 1.[1] Kjos-Hanssen, Miller and Solomon [**31**] showed that this notion of almost everywhere domination is equivalent to LR-completeness (see Nies [**60**] 5.6.30). Dobrinen and Simpson pointed out that almost everywhere domination implies highness, Binns, Kjos-Hanssen, Lerman and Solomon [**9**] showed that not all high degrees are almost everywhere dominating, and Cholak, Greenberg and Miller [**12**] showed that there are Turing-incomplete c.e. sets that are almost everywhere dominating (though this also follows from the result from [**5**] that every c.e. LR-degree contains infinite chains of c.e. Turing degrees and the equivalence between LR-completeness and almost everywhere domination).

[1] An equivalent notion is *uniformly almost everywhere dominating* (u.a.e.d.): A is u.a.e.d. if there is a single $f \leq_T A$ such that the measure of those X such that f dominates all total $g \leq_T X$ has measure 1. This was shown to be equivalent to (non-uniform) almost everywhere domination by Kjos-Hanssen [**30**].

LR-completeness has also been studied in comparison to other highness properties such as (regular) highness, superhighness[2] and $\emptyset'$-trivialising [3]. See for example [**67**] or [**4**].

One question that is of interest is whether there is a degree-theoretic property on which the LR-complete and the high Turing degrees can be separated. That is, a formula ϕ in the language of the c.e. Turing degrees such that some high c.e. degrees satisfy ϕ but no LR-complete c.e. degrees satisfy ϕ. Barmpalias and Montalbán [**7**] constructed a cappable c.e. Turing degree that is LR-complete; that is, a c.e. set A which forms half of a minimal pair with another c.e. set and is $\geq_{LR} \emptyset'$. Hence being cappable is not such a property. We strengthen this result by showing that there is a non-cuppable LR-complete c.e. Turing degree. Hence non-cuppability is also not such a property.

Non-cuppable c.e. degrees were first constructed by Yates, Cooper [**13**] (unpublished). Harrington constructed a high non-cuppable c.e. degree, in fact proving the stronger theorem that for every high c.e. degree $\mathbf{a}$ there is a high c.e. degree $\mathbf{b}$ that cannot be cupped to $\mathbf{a}$ by any c.e. degree $\mathbf{c} \not\geq \mathbf{a}$ (see Miller [**53**]). A tree strategy for constructing non-cuppable degrees was sketched in Cooper [**15**], the original constructions of Yates, Cooper and Harrington being pinball constructions. Tree constructions were also given by Li, Slaman and Yang [**44**] and Yang and Yu [**76**], whose basic strategy we will use in Theorem 4.1.

It is well known from work by Ambos-Spies, Jockusch, Shore and Soare [**1**] that the cappable degrees form an ideal within the c.e. Turing degrees. Harrington established that the non-cuppable degrees form a proper subideal of the cappable degrees, by showing that all c.e. degrees either cup or cap and some do both (see Soare [**68**] XII.4.3).

Since the LR-complete sets are a proper subclass of the high sets, Theorem 4.1 can be seen as a strengthening both of Barmpalias and Montalbán's cappable LR-complete c.e. set, and of Harrington's construction of a high non-cuppable c.e. set.

An open question regarding the LR-complete c.e. sets is whether there is a single noncomputable c.e. set that is computable by all LR-complete c.e. sets. A corollary of Theorem 4.1 is that if such a set exists then it must be non-cuppable.

---

[2] A is superhigh if $A' \equiv_{tt} \emptyset''$.

[3] A is $\emptyset'$-trivialising, also known as *almost complete*, if $\emptyset'$ is K-trivial relative to A. See [**20**] for discussion of this notion.

2. Proof of Theorem 4.1

In the following we fix U to be the second member of a universal oracle Martin-Löf test, so that $\mu(U^X) \leq 2^{-1}$ for all $X \in 2^\omega$. We will construct a non-cuppable set A and an A-Σ^0_1 class V^A such that $U^{\emptyset'} \subseteq V^A$ and $\mu(V^A) < 1$. By Theorem 1.5 this ensures that $A \geq_{LR} \emptyset'$.

We adopt the assumptions that, for a Turing functional Γ, $\Gamma^X(z)[s]\downarrow$ only if $\Gamma^X(y)[s]\downarrow$ for all $y < z$, and that $\operatorname{use}\Gamma^X(y)[s] \leq \operatorname{use}\Gamma^X(z)[s] \leq s$ if $\Gamma^X(z)\downarrow$ and $y \leq z$. We consider Turing functionals Γ as c.e. sets of axioms $\langle z, y, \sigma\rangle$ (asserting that $\Gamma^X(z) = y$ for all $X \in 2^\omega$ with $\sigma \subset X$), which are consistent in the sense that if $\langle z, y, \sigma\rangle$ and $\langle z, y', \sigma'\rangle$ are both in the set, for $y' \neq y$, then σ and σ' are incomparable. We will abbreviate $\Gamma^{X\oplus Y}$ as Γ^{XY}.

2.1. Making A non-cuppable.

We describe the basic strategies for constructing a non-cuppable degree, based on [44, 76]. For convenience, we can assume that $\emptyset' \subseteq 2\mathbb{N}$, the even numbers. We will construct Turing functionals Δ_e to ensure that the following holds for all $e \in \mathbb{N}$:

$$N_e: \qquad \Gamma_e^{AW_e} = K \Rightarrow \Delta_e^{W_e} = \emptyset' \tag{16}$$

where $\langle \Gamma_e, W_e\rangle$ ranges over all pairs of a Turing functional and a c.e. set; assuming that $\emptyset' \subseteq 2\mathbb{N}$ we let $K = D \cup \emptyset'$ where $D \subseteq 2\mathbb{N}+1$ is an auxiliary c.e. set that we enumerate. Although $K = D \cup \emptyset'$ is not the 'standard' halting set, we use the letter K nonetheless as we can think of it as a version of the halting problem which we have some control over (via the set D). In the following discussion we omit the index e. The idea is to let Δ^W copy Γ^{AW} by monitoring the Turing reduction Γ^{AW} and restraining A to preserve the agreement of the two computations. The problem with this approach is that the restraint on A may well have limit ∞, in which case very little can be done to make A nontrivial, let alone LR-above $\emptyset'$. The solution is to split N into infinitely many subrequirements M_p which are responsible just for the definition of $\Delta^W(p)$, thus splitting a potentially infinite restraint into infinitely many finite restraints. The strategies for the subrequirements M_p will be coordinated by a master N strategy which will make sure that Δ is consistent and this coordination will be implemented on a tree of strategies.

We can think of N as having two outcomes $\infty <_L f$ (i.e. ∞ is to the left of f) corresponding to whether there are infinitely many expansionary stages in $\Gamma^{AW} = K$ or not, and M_p having outcomes $\infty <_L f$ according to whether $\Gamma^{AW}(p)\downarrow$ or equivalently, $\Delta^W(p)\downarrow$. We will have a uniformly labelled tree of strategies with each strategy N or M_p occupying a single level of the tree. For the consistency of Δ we make sure that at any M_p-level (i.e. occupied by an M requirement) and at any stage at most one node α will be responsible for $\Delta^W(p)\downarrow$ (by preserving A in $\Gamma^{AW}(p)\downarrow$). Any nodes to the right of α may adopt α's Δ-definition but if a node to the left of α wishes to define $\Delta(p)$ it must first cancel the Δ computation that α holds. This happens by enumerating into the auxiliary set D which in turn causes a W-change (provided that the $\Gamma^{AW} = K$ reduction is valid). Eventually, if $\Gamma^{AW} = K$, at each M_p level there will be exactly one node *on or to the left of the true path* which permanently preserves $\Delta^W(p)\downarrow = \emptyset'(p)$. Otherwise some node will witness the partiality of Γ^{AW}. As in any $\emptyset''$ priority argument the restraints imposed on a node on the true path will be finite.

Each M_p-node α has a *flip-point* d, which is the number enumerated into D when we wish to cancel the computation $\Delta^W(p)$. When α is visited, it checks if the computation $\Gamma^{AW}(d)$ has changed since the last time it was visited, and if so it plays outcome ∞. Otherwise we may define $\Delta^W(p) = \Gamma^{AW}(p)$, with W-use $u = \operatorname{use}\Gamma^{AW}(d)$ and restrain $A \upharpoonright u$. If we later want to visit a node β to the left of α, we enumerate the flip-point d into D whilst maintaining α's A-restraint. This enumeration should force a W-change below u, and so α will not hold a Δ-computation anymore (if this does not happen then N will be satisfied by a finite outcome). Then we can drop α's restraint and β can take action. This must happen immediately upon seeing the next N-expansionary stage, otherwise some other node α' to the right of β may act first and define another Δ-computation which prevents β from acting. For this reason when we enumerate d into D we create a link (τ, β) from the N-node τ to β, and when τ is next visited at an expansionary stage we will follow the link straight to β.

2.2. Measure-guessing nodes and LR-completeness. This aspect of the construction is largely the same as in Theorem 2.13, however the presence of links necessitate some small modifications compared to the method of Theorem 2.13. For completeness and so that this chapter might be self-contained, we discuss the strategy for making $\emptyset' \leq_{LR} A$.

To make A LR-complete, it suffices to construct an A-Σ^0_1 class V^A with $U^{\emptyset'} \subseteq V^A$ and $\mu(V^A) < 1$. Without loss of generality we may assume that if $\langle \sigma, \tau \rangle$ is enumerated into U at stage s then $|\sigma| = |\tau| = s$. We will also use the *hat-trick* for $U^{\emptyset'}$: let $k_s = \min\{x : x \in \emptyset'[s] - \emptyset'[s-1]\}$, or $k = s$ if there is no such x and define $\widehat{\emptyset'}[s] = \emptyset'[s] \upharpoonright k$. Then $\widehat{U^{\emptyset'}}[s] = \{\sigma : \langle \sigma, \tau \rangle \in U_s \text{ for some } \tau \subseteq \widehat{\emptyset'}[s]\}$. In the following we assume that $U^{\emptyset'}[s]$ and $\emptyset'[s]$ refer to $\widehat{U^{\emptyset'}}[s]$ and $\widehat{\emptyset'}[s]$ respectively. Infinitely often we have *true stages* s at which $U^{\emptyset'}[s] = U^{\emptyset' \upharpoonright n} \subset U^{\emptyset'}$ for some n, and thus $\mu(U^{\emptyset'}[s]) < \mu(U^{\emptyset'})$.

Whenever an interval σ appears in $U^{\emptyset'}$, we add it to V^A with large A-use u. If a $\emptyset'$-change later removes σ from $U^{\emptyset'}$, we could remove it from V^A by enumerating u into A, provided that u is not restrained by some requirement. The A-change may also remove some legitimate intervals from V^A, but we add these again with the same use as before. This clearly gives $U^{\emptyset'} \subseteq V^A$. The conflict with the non-cupping strategy is that the A-restraints will prevent us from removing some superfluous 'junk' intervals σ from V^A. For the argument to succeed, we must ensure that the total measure of junk intervals $\mu(V^A - U^{\emptyset'}) < \frac{1}{2}$. We assign each requirement (each level of the tree) a quota ϵ, which is the amount of junk measure that requirement is allowed to capture. We implement the negative strategies in such a way that we have at most one node imposing restraint at each level of the tree. A restraint may only be imposed on A if the (current) junk measure that it captures is less than the quota. To ensure that strategies will eventually be able to impose restraints under this restriction, we choose the quota $\epsilon(k)$ of level k of the tree so that $\sum_{j>k} \epsilon(j) < \epsilon(k)$ (in this way the lower priority requirements will not capture more than $\epsilon(k)$ of junk).

To ensure that the strategies do not exceed their junk quota, the predecessor of each N and M node will be a node with a strategy G which measures $\mu(U^{\emptyset'})$ in a Π^0_2 way. The backup nodes G successively subdivide the interval $[0,1)$, assigning each of its outcomes an interval $[q,r)$ which corresponds to a guess that $\mu(U^{\emptyset'}) \in [q,r)$. The construction will make sure that if the backing node of a strategy predicts the right interval $[q,r)$ of $\mu(U^{\emptyset'}[s])$ then the junk measure that it captures will increase by no more than $r-q$ after it acts. If we choose $r-q=\epsilon$, then α will capture at most 2ϵ of junk, which is acceptable if we choose the quotas $\epsilon(k)$ such that $\sum_{k\in\omega} 2\epsilon(k) < \frac{1}{2}$. An analysis of the permanent restraints and the timing of the enumerations into A in the construction will verify that $\mu(V^A - U^{\emptyset'}) < \frac{1}{2}$.

2.3. Combining the strategies. The difficulty in combining the non-cupping and LR-completeness strategies stems from the fact that the non-cupping subrequirements are not independent of each other or of the parent N-node. In previous constructions using the measure-guessing strategy such as [**12**], [**7**] and Theorem 2.13, when a node holds a restraint under a measure guess which proves wrong, we initialise that node and all lower-priority nodes. However here we can only initialise non-cupping parent N-nodes since by initialising an M-node we may make Δ inconsistent. Once a $\Delta^W(p)$ axiom has been enumerated, we must retain the A-restraint until the axiom is invalidated by a W-change or the parent N-node is initialised.

Thus whenever some M-node holds a restraint under a wrong assumption about $\mu(U^{\emptyset'})$ we just try to invalidate the corresponding Δ axiom by enumerating the flip point and waiting for a suitable W-change. The construction will make sure that if this does not happen and N is not reset, the junk measure from the subrequirements of N will be less than the quota of N, even though the junk measure of some M may turn out to be larger than its quota. Overall this satisfies N trivially and with small enough cost. The trick which allows the above quota-junk relation is in the enumeration of $U^{\emptyset'}$: it is prefix-free and if some interval σ leaves $U^{\emptyset'}$ then all intervals which were enumerated after σ leave as well, at the same time.

2.4. Priority Tree and Definitions. The priority tree is a finite branching tree which consists of the parent nodes labelled N_e, the subrequirement nodes labelled $M_{e,p}$, and the measure-guessing backup nodes labelled G. Let $\langle \cdot, \cdot \rangle$ be a monotone 1–1 computable function from $\mathbb{N} \times \mathbb{N}$ onto $\mathbb{N}$. If $|\alpha| = 2\langle e, 0\rangle + 1$ then α is labelled N_e and if $|\alpha| = 2\langle e, p+1\rangle + 1$ then it is labelled $M_{e,2p}$ (by assumption $\emptyset' \subset 2\omega$ and so only even $\Delta^W(p)$ arguments need to be considered). If $|\alpha| = 2e$ then α is labelled G. We write $R < R'$ to indicate that the requirement R occupies an earlier level of the tree than requirement R' (where R, R' are one of $G, N_e, M_{e,p}$).

The N_e-nodes τ have outcomes $\infty <_L f$ and are associated with a functional Δ_τ that is built by the $M_{e,p}$-nodes below τ and is occasionally cleared and started afresh when τ is reset. The $M_{e,p}$ nodes have outcomes $\infty <_L f$ and are associated with a *flip-point* d_α which may change in the course of the construction.

A measure-guessing G-node γ has outcomes $q_0 <_L q_1 <_L q_2 <_L q_3$ which correspond to guesses about an interval in which $\mu(U^{\emptyset'})$ may lie. We inductively assign to each node

α an interval I_α as in Theorem 2.13. Start with $I_\lambda = [0, 2^{-1})$ for the root node λ. For a node α with I_α defined, if α is a G-node then subdivide I_α equally into four subintervals and assign them to $I_{\alpha^\frown q_i}$ in order. If α is an N- or M-node then let $I_{\alpha^\frown x} = I_\alpha$ for $x \in \{\infty, f\}$. We write $q(\alpha)$ for the lower endpoint of I_α and $\epsilon(\alpha)$ for the width of I_α. We refer to $\epsilon(\alpha)$ as α's *resolution* and $q(\alpha)$ as its *measure guess*. Since all nodes of the same label have the same length, we may write $\epsilon(N_e)$ or $\epsilon(M_{e,p})$ to denote $\epsilon(\alpha)$ for any node α labelled N_e or $M_{e,p}$, respectively. For each N or M requirement R we have

$$\sum_{R'>R} 2\epsilon(R') < \epsilon(R) \quad \text{and} \quad \sum_{e\in\omega} 2\epsilon(N_e) < \frac{1}{2} \tag{17}$$

where R' is an N or M requirement.

The ordering $<_L$ on the outcomes can be extended lexicographically to the nodes of the tree. We say that α has *higher priority than* β or write $\alpha < \beta$ if either $\alpha \subset \beta$ or $\alpha <_L \beta$.

We write r_α for the restraint imposed on A by node α, and α^- for the predecessor of α. Also let $R_\alpha = \max\{r_\beta : \beta <_L \alpha \text{ or } \beta \subset \alpha\}$. All parameters have a current value each time they are mentioned in the construction. It is convenient in this chapter to use the suffix $[s]$ to denote the value of a parameter *at the start of* stage s. For an $M_{e,p}$-node α, we write $\tau(\alpha)$ for the unique N_e-node $\tau \subset \alpha$. We refer to τ as α's *parent*, or say that α is *working for* τ. An $M_{e,p}$-node α with parent τ is *enabled* if $\tau^\frown\infty \subset \alpha$ and for every $M_{e,p'}$-node α' with $\tau \subset \alpha' \subset \alpha$, we have $\alpha'^\frown f \subset \alpha$. Otherwise, α is *disabled* (which means that it regards Γ^{AW_e} as partial and no further action is needed for N_e).

2.5. Construction. Set $A[0] = \emptyset, \Delta_\tau = \emptyset$ for all N-nodes τ, and $d_\alpha \uparrow, r_\alpha = 0$ initially for all M-nodes α. When a parameter is assigned a value, it retains that value until explicitly given a new value. To *reset* an N-node τ means to empty Δ_τ, set $r_\beta = 0$ and $d_\beta \uparrow$ for any M-nodes β working for τ, and remove any links to or from τ or any M-node β working for τ. To *reset* an M-node α means to remove any links to it and if $r_\alpha \neq 0$ and $d_\alpha \downarrow$, enumerate d_α into D, setting $d_\alpha \uparrow$. To *reset* a G-node means to remove any links to it. The construction will explicitly declare certain nodes α to be *accessible* at each stage, which does not merely mean that $\alpha \subset \mathrm{TP}_s$. If α is an N-node, it will also declare certain stages to be α-*expansionary*. We give the enumeration of V^A

during the stages s of the construction in advance:

(18) **Enumeration of V^A.** For each $\langle\sigma,\rho\rangle \in U[s]$ with $\rho \subset \emptyset'[s]$ but $\sigma \notin V^A[s]$, if $\sigma \in V^A[t]$ with use u for some $t < s$ take the largest such t and if $\langle\sigma,\rho'\rangle \in U[t]$, $\rho' \subset \emptyset'[s]$, then enumerate σ into $V^A[s+1]$ with use u. Otherwise, put σ into $V^A[s+1]$ with fresh use.

The construction will occasionally call the following routine, which is needed in order to access certain outcomes x of nodes α.

(19) **Routine $L(\alpha, x, s)$.** Reset all N-nodes which are to the left of $\alpha^\frown x$. Then consider the longest node $\tau \subset \alpha$ which has label N_e for some $e \in \mathbb{N}$ and there is some $M_{e,p}$-node $\beta \supset \tau$ with $\beta >_L \alpha^\frown x$ and $r_\beta[s] \neq 0$. If τ exists, let β be the shortest node as above, enumerate d_β into D (if $d_\beta \downarrow$), set $d_\beta \uparrow$, create a link (τ, α) associated with outcome x and go to step 4. Otherwise let $\mathrm{TP}_{s,t+1} = \alpha^\frown x$ and go to step 3.

At stage s, perform the following steps in order.

Step 1. *(Reset some nodes)* Look for the highest priority node α such that some $\beta \supseteq \alpha$ has been accessed since α was last reset and $\mu\big(U^{\emptyset'}[s]\big) < q(\alpha)$. If there is such, reset α and all nodes of lower priority than α.

Step 2. *(Drop some restraints)* For each M-node α with $r_\alpha \neq 0$ and $W \upharpoonright r_\alpha[s] \neq W \upharpoonright r_\alpha[t]$, where t is the stage at which the restraint r_α was last set, set $r_\alpha = 0$ and reset $\alpha^\frown f$ and all nodes of lower priority than $\alpha^\frown f$.

Step 3. *(Define TP_s in substages)* Let $\mathrm{TP}_{s,0} = \lambda$. Let t be the largest number such that $\mathrm{TP}_{s,t} \downarrow$. If $|\mathrm{TP}_{s,t}| \geq s$ then go to step 4. Otherwise let $\alpha = \mathrm{TP}_{s,t}$ and check if

(20) there is an M-node $\beta \leq_L \alpha$ with $\tau(\beta)^\frown\infty \subset \alpha, r_\beta \neq 0$ and $d_\beta \uparrow$.

The check for (20) is to ensure that an M-node does not try to act while a higher-priority M-node is awaiting a W-change. This could potentially happen if a parent node τ was bypassed by a link, rather than accessing outcome $\tau^\frown f$. Preventing this helps simplify the verification later.

If (20) holds, go to step 4; otherwise declare α accessible and go to the relevant clause below.

• **α is a G-node.** Let $[a_0, a_1), \ldots [a_3, a_4)$ be the intervals corresponding to the outcomes of α and $\epsilon = a_1 - a_0$ be the resolution of α. Let $g_\alpha(s)$ be the largest $t < s$ such that $\alpha \subset \mathrm{TP}_t$, or 0 if such t does not exist. Let

$$\nu = \nu(\alpha, s) = \min\{\mu(U^{\emptyset'}[k]) : g_\alpha(s) < k \le s \;\&\; \mu(U^{\emptyset'}[k]) \in [a_0, a_4)\} \tag{21}$$

(Lemma 4.3 verifies that ν always exists) and let i be such that $\nu \in [a_i, a_{i+1})$, and run routine $L(\alpha, q_i, s)$.

• **α is an $M_{e,p}$-node.** If α is a disabled $M_{e,p}$-node, let $\mathrm{TP}_{s,t+1} = \alpha^\frown\infty$ and go to step 3. Otherwise do as follows. For brevity let $d = d_\alpha, \tau = \tau(\alpha), W = W_e, \Gamma = \Gamma_e$, $u = \operatorname{use}\Gamma^{AW}(d)[s]$ (if defined) and

$$h_\alpha(s) = \max\{t \le s : \nu(\alpha^-, s) = \mu(U^{\emptyset'}[t])\} \tag{22}$$

where α^- is the predecessor of α. *$h_\alpha(s)$ is the stage for which the measure-guessing G-node of α gave its outcome.* If $d\uparrow$ choose a fresh value for d.

M1. If $\Delta_\tau^W(p)[s]\downarrow$ let $\mathrm{TP}_{s,t+1} = \alpha^\frown f$ and go to step 3; if $\Delta_{\tau'}^W(p)[s]\downarrow$ for some N_e-node $\tau' <_L \alpha$ then define $\Delta_\tau^W(p) = \Delta_{\tau'}^W(p)$ with the same use, let $\mathrm{TP}_{s,t+1} = \alpha^\frown f$ and go to step 3.

M2. Otherwise if $\Gamma^{AW}(d)[s]\uparrow$ or if $A \upharpoonright u[s] \ne A \upharpoonright u[t]$ or $W \upharpoonright u[s] \ne W \upharpoonright u[t]$ for the last stage t when α was accessible, or if α has never been accessible before, then run routine $L(\alpha, \infty, s)$.

M3. Otherwise, if

$$\mu\Big(V^{A\upharpoonright u}[s] - V^{A\upharpoonright R_\alpha}[s] - U^{\emptyset'}[h_\alpha(s)]\Big) < \epsilon(\alpha) \tag{23}$$

we define $\Delta_\tau^W(p) = \Gamma^{AW}(p)[s]$ with use u, impose restraint $r_\alpha[s+1] = u$, and go to step 4.

M4. In any other case go to step 4.

• **α is an N_e-node.** Let

$$l(\alpha, s) = \min\Big(\{n : \Gamma_e^{AW_e}(n)[s] \ne K(n)[s]\} \\ \cup \{d : d \text{ was enumerated into } D \text{ in step 1 or 2}\}\Big),$$

and say that stage s is *α-expansionary* if $l(\alpha, s) > l(\beta, t)$ for all N_e-nodes $\beta \leq_L \alpha$ and all $t < s$ such that β was accessible at t. If s is not α-expansionary, then let $\mathrm{TP}_{s,t+1} = \alpha^\frown f$ and go to step 3. Otherwise, if there is a link (α, β) associated with outcome x of β which was created at stage $t < s$, remove it and run routine $L(\beta, x, s)$. Otherwise run routine $L(\alpha, \infty, s)$.

Step 4. Set $\mathrm{TP}_s = \alpha$ for the longest α which was declared accessible in step 3. Reset all nodes $>_L \mathrm{TP}_s$ and enumerate into A the least number which is not in A and is greater than all $r_\beta[s+1]$ for all M-nodes β.

2.6. Verification. In the following, whenever we say 'M-node' we mean an enabled M-node, as disabled M-nodes have no effect on the construction. A basic fact which stems from the hat-trick in the enumeration of $U^{\emptyset'}$ and will be used repeatedly in the verification is the following: if $s_0 < t \leq s_1$ are stages and $\mu(U^{\emptyset'})$ takes its minimum value in $(s_0, s_1]$ at t, then $U^{\emptyset'}[t] \subseteq U^{\emptyset'}[s]$ for all $s \in (s_0, s_1]$.

LEMMA 4.2. Links can never be nested or crossing. That is, if (τ, α) and (τ', α') are two distinct links both present at stage s, with $\tau \subset \alpha \subset \beta$ and $\tau' \subset \alpha' \subset \beta$ for some node β, then $\alpha \subset \tau'$ or $\alpha' \subset \tau$. Furthermore, at the end of any stage s, there is at most one link (τ, α) with $\tau \subset \alpha \subseteq \mathrm{TP}_s$, and such a link was created at stage s.

PROOF. By induction on the stages. Note that initially there are no links and at any stage at most one link is created. Suppose that the claim holds at stage s and a link (τ, α) is created at stage $s+1$. Then α is accessible at stage $s+1$ or a link was travelled to α, and any links (τ', α') with $\tau' \subset \alpha' \subseteq \alpha$ present at the start of stage $s+1$ have been travelled and removed. If there was a link (τ'', α'') at the start of stage $s+1$ for some $\tau'' \subset \alpha \subset \alpha''$, then that link would have been travelled and α would not be accessible. Thus the new link cannot be crossing or nested within an existing link. Finally any links (τ, α) with $\tau \subset \alpha \subset \mathrm{TP}_{s+1}$ which are present at the start of stage $s+1$, would be travelled and removed during the definition of TP_{s+1} in step 3. Since at most one link is created under routine (19), the last claim of the lemma holds. ☐

The following lemma verifies that a G-node will always have a valid outcome to play when it is accessible. Note that we need to verify this because of the presence of links in the construction; in Theorem 2.13 such a lemma was not necessary.

LEMMA 4.3. *Suppose a G-node γ is accessible at stage s_0 and let $s_1 = g_\gamma(s_0)$ be the greatest stage $< s_0$ such that $\gamma \subset \mathrm{TP}_{s_1}$ (or 0 if such stage does not exist). Then there is some t with $s_1 < t \leq s_0$ and $\mu(U^{\emptyset'}[t]) \in I_\gamma$. Thus, when γ is accessible in step 3, ν (as in (21)) will exist.*

PROOF. Let γ, s_0 and s_1 be as in the lemma. The proof is by simultaneous induction on the length of γ and the stage s_0. For the root node the claim is trivial, so let $|\gamma| > 1$ and suppose that the claim is true for all G-nodes shorter than γ and at all stages $\leq s_0$. Let $\gamma' = \gamma \restriction |\gamma| - 2$ be the last G-node above γ and note that if γ has never been accessed before, a suitable t must exist or else γ' would not have chosen the outcome leading to γ. Suppose then that γ has been accessed before. If γ' is also accessible at s_0, since $\gamma' \subset \gamma$ we have $g_{\gamma'}(s_0) \geq s_1$ and by hypothesis there is a suitable t with $g_{\gamma'}(s_0) < t \leq s_0$ and $\mu(U^{\emptyset'}[t]) \in I_\gamma$.

If γ' is not accessible at s_0, then there must be a link (τ, β) at s_0, with $\tau \subset \gamma' \subseteq \beta \subset \gamma$. Also by induction hypothesis there must be a stage $t_0 < s_0$ such that γ' is accessible at t_0 and $\mu(U^{\emptyset'}[t]) \in I_\gamma$ for some t with $g_{\gamma'}(t_0) < t \leq t_0$. We can assume that t_0 is the greatest stage $< s_0$ with the above property. If t_2 is the stage at which the link (τ, β) was created we have $t_2 \geq t_0$. Now $\mathrm{TP}_s \not\supseteq \gamma$ for $t_0 \leq s \leq t_2$, as otherwise t_0 would not be the greatest with the above property. Also $\mathrm{TP}_s \not\supseteq \gamma$ for $t_2 < s < s_0$ as otherwise the link would be travelled and removed before s_0, because by Lemma 4.2 links cannot be nested. So $s_1 < t_0$ and $s_1 \leq g_{\gamma'}(t_0)$ since $\gamma' \subset \gamma$, which means that $s_1 < t \leq s_0$. □

By the construction, if an $M_{e,p}$-node α has $r_\alpha[s] \neq 0$ and $d_\alpha \downarrow$, then d_α has not been enumerated into D via resetting or routine (19). Conversely, $r_\alpha[s] \neq 0$ and $d_\alpha \uparrow$ indicates that the construction has attempted to invalidate α's $\Delta^W(p)$ computation. The definition of τ-expansionary stage and the check for (20) in step 3 ensures that no M_e-node of lower priority than α will be accessible again until the $\Delta^W(p)$ computation is invalidated.

A restraint r_α is called *permanent at stage s* if $r_\alpha[s] = r_\alpha[t] \neq 0$ for all $t \geq s$; it is called *permanent* if it is permanent at some stage. Let P be the set of nodes with permanent restraints.

For an M-node α, let $J_\alpha[s] = \{\sigma \in V^A[s+1] - U^{\emptyset'}[s] : R_\alpha[s+1] \leq \operatorname{use}\sigma < r_\alpha[s+1]\}$, which is the junk intervals that are restrained at stage s by α but not by any higher-priority node at the *end* of stage s. For an N_e-node τ, let $Q_\tau[s] = \bigcup J_\alpha[s]$, where the union is taken over all M_e-nodes α which are either $\supset \tau$ or $<_L \tau$. The following lemma shows that if the junk captured by an M-node becomes greater than the node's quota 2ϵ then the node is reset; and although an M-node may sometimes capture more than its quota of junk (if the junk is never released via step 2), the total junk captured by nodes belonging to an N-node remains within the N-node's quota.

LEMMA 4.4. Let β be an M-node and s a stage such that $r_\beta[s+1] \neq 0$ and $d_\beta[s+1]\downarrow$ (so β has not been reset since r_β was set $\neq 0$). Then $\mu(J_\beta[s]) < 2\epsilon(\beta)$. Let τ be an N-node. Then $\mu(Q_\tau[s]) < 2\epsilon(\tau)$ for all s.

PROOF. Suppose β and s are as in the first claim. Let t be the stage when $r_\beta[s+1]$ was set. At t, $V^{A\restriction r}[t] = V^{A\restriction r}[s+1]$ for $r = r_\beta[s+1]$ as new intervals in V^A have use chosen fresh. So,

$$\begin{aligned} \mu(J_\beta[s]) &= \mu\big(V^{A\restriction r_\beta}[s+1] - V^{A\restriction R_\beta}[s+1] - U^{\emptyset'}[s]\big) \\ &\leq \mu\big(V^{A\restriction r_\beta}[t] - V^{A\restriction R_\beta}[t] - U^{\emptyset'}[h_\beta(t)]\big) \\ &\quad + \mu\big(U^{\emptyset'}[h_\beta(t)] - U^{\emptyset'}[s]\big) \end{aligned} \tag{24}$$

where the first term of (24) is the junk that β captured when it imposed its restraint $r_\beta[s+1]$, and the second is the measure which appears to be in $U^{\emptyset'}$ at $h_\beta(t)$ but later is removed from $U^{\emptyset'}$. By (23) the first term is less than $\epsilon(\beta)$. Suppose that $\mu(U^{\emptyset'}[h_\beta(t)] - U^{\emptyset'}[s]) \geq \epsilon(\beta)$. We have $U^{\emptyset'}[h_\beta(t)] - U^{\emptyset'}[t] = \emptyset$, as otherwise (by the canonical enumeration of $U^{\emptyset'}$) there would be a stage $t', h_\beta(t) < t' \leq t$ with $\mu(U^{\emptyset'}[t']) < \mu(U^{\emptyset'}[h_\beta(t)])$, which contradicts (22). So we must have $\mu(U^{\emptyset'}[t] - U^{\emptyset'}[s]) \geq \epsilon(\beta)$. But then, again by the canonical enumeration of $U^{\emptyset'}$ there would be a stage $t', t < t' \leq s$ such that $\mu(U^{\emptyset'}[t']) \leq \mu(U^{\emptyset'}[h_\beta(t)]) - \epsilon(\beta)$, and β would be reset at t' by step 1 of the construction. So $\mu(U^{\emptyset'}[h_\beta(t)] - U^{\emptyset'}[s]) < \epsilon(\beta)$, and $\mu(J_\beta[s]) < 2\epsilon(\beta)$.

Next, let τ be an N_e-node; we need only consider the case where there is some M_e-node $\beta \supset \tau$ with $J_\beta[s] \neq \emptyset$. Let Z denote the set of M_e-nodes $\beta' \supset \tau$ or $<_L \tau$ with $r_{\beta'}[s+1] \neq 0$, and let β be the longest; by assumption $\beta \supset \tau$. Let t be the stage when $r_\beta[s+1]$ was set $\neq 0$. At t, $d_{\beta'}[t+1]\downarrow$ for all $\beta' \in Z$, as otherwise β would not be accessible at t. Also $\mu(J_\beta[t]) < \epsilon(\beta)$ by (23). So by the first part of the lemma and (17), $\mu(Q_\tau[t]) < 2\epsilon(\tau)$. Also, $d_{\beta'}[t'+1]\downarrow$ for all $t < t' \leq s$ and $\beta' \in Z, \beta' <_L \tau$, as otherwise τ would be reset, contradicting the definition of t. So if $\mu(Q_\tau[t']) \geq 2\epsilon(\tau)$

at some $t < t' \leq s$ it must be because $\sum_{\tau \subset \beta' \in Z} \mu(J_{\beta'}[t']) > \epsilon(\tau)$. But then by the canonical enumeration of $U^{\emptyset'}$ there would be a stage t'' such that $t < t'' \leq t'$ and $\mu(U^{\emptyset'}[t'']) < \mu(U^{\emptyset'}[h_\beta(t)]) - \epsilon(\tau)$. In such a case τ would be reset at step 1, again contradicting the definition of t. So $\mu(Q_\tau[s]) < 2\epsilon(\tau)$. □

In the following lemma we prove simultaneously that the true path $\mathrm{TP} = \liminf_s \mathrm{TP}_s$ is infinite, that every node on it has infinitely many chances to act, and that eventually the measure condition (23) will be satisfied for each M-node on TP.

LEMMA 4.5. If α is the leftmost node of length $|\alpha|$ such that $\alpha \subseteq \mathrm{TP}_s$ for infinitely many s, then

(i) α is reset only finitely often; if it is an M-node then eventually the flip-point d_α is fixed;
(ii) α is accessible infinitely often;
(iii) there is some extension $\beta \supset \alpha$ with $\beta \subseteq \mathrm{TP}_s$ for infinitely many s.

Thus $\mathrm{TP} = \liminf_s \mathrm{TP}_s$ is infinite.

PROOF. First of all, if $|\alpha| = 0$ then $\alpha \subseteq \mathrm{TP}_s$ for all s so (i)-(iii) of the lemma implies by induction that TP is infinite. Then it remains to assume that α is the leftmost node of length $|\alpha|$ such that $\alpha \subseteq \mathrm{TP}_s$ infinitely often and (inductively) that the lemma holds for all $\beta \subset \alpha$, and show claims (i)-(iii).

For the first claim note that there are four places in the construction where α may be reset: in step 1, step 2, step 3 (through the routine L) and step 4. Let s_0 be the second stage such that $\alpha \subseteq \mathrm{TP}_{s_0}$, $\mathrm{TP}_s \not<_L \alpha$ $\forall s > s_0$, any computations $\Delta^W_{\tau(\beta)}(p) \downarrow$ of nodes $\beta <_L \alpha$ that exist at s_0 are permanent and no nodes above or to the left of α are reset after s_0. After s_0, α will not be reset in step 4. If α was reset after s_0 at step 3 then it would be because routine $L(\beta, x, s)$ was run for some $\beta \subset \alpha$ such that $\beta^\frown x <_L \alpha$. But this would mean that either $\mathrm{TP}_s <_L \alpha$ for some $s > s_0$ or α is not $\subseteq \mathrm{TP}_s$ infinitely often, a contradiction.

If α was reset by step 2, by the choice of s_0 there must be some M-node β such that $\beta^\frown f \subset \alpha$ which had a computation $\Delta^W_{\tau(\beta)}(p) \downarrow$ and this was spoilt after s_0. But then the corresponding Γ computation (which has larger use) would be spoilt and the construction would define TP_s to the left of α at M2, a contradiction. Suppose that α was reset in step 1 after stage s_0. By the choice of s_0 there must be a node $\beta \subset \alpha$

and a stage $s_1 > s_0$ such that $\mu(U^{\emptyset'}[s_1]) < q(\beta)$. But in that case after stage s_1 the construction would define TP_s to the left of α, before it defines it below α, a contradiction. Finally suppose that α is an M-node and d_α was changed after stage s_0. Since α is not reset after s_0 there must be some $\beta \subset \alpha$ which ran routine $L(\beta, x, s_1)$ for $s_1 > s_0$ and $\beta^\frown x <_L \alpha$. But in that case the construction would define TP_s to the left of α, before it defines it below α, a contradiction.

For claim (ii), notice that since by hypothesis $\alpha \subseteq \mathrm{TP}_s$ for infinitely many s, the only way that α may stop being accessible after some stage is that for all sufficiently large stages there is a link (τ, β) with $\tau \subset \alpha \subset \beta$. Suppose, for a contradiction, that this is the case and after stage s_0 α is never accessible again. Let $Y[s]$ be the finite set of Δ-computations that are held by M-nodes below α at $s \geq s_0$. Note that if $\mathrm{TP}_t \supseteq \alpha$ for $t \geq s_0$ then by Lemma 4.2 a link must be created at t as otherwise the next time $\alpha \subseteq \mathrm{TP}_s$, α would not be covered by a link and would be accessible. Thus no new computations can be added to Y after s_0 as if a Δ-definition is made then no link is created at that stage. Also, by the construction there are no Δ-computations held by nodes $>_L \alpha$ at the end of a stage s when $\alpha \subseteq \mathrm{TP}_s$. Finally a link is only travelled if the Δ-computation for which it was created has been invalidated. So any link covering α at $s \geq s_0$ is created because of a computation in Y, which is removed from Y when the link is travelled. Since Y is finite and non-increasing, after finitely many stages Y will be empty and α will be accessible when next $\mathrm{TP}_s \supseteq \alpha$.

For claim (iii), since α is accessible infinitely often the only way the claim could fail is if, whenever α is accessible after some finite stage $s_0 > |\alpha|$, step 3 is ended without any $\alpha^\frown x$ being declared accessible. Suppose this is the case. Then whenever α is accessible after s_0, step 3 is ended by routine L, or by M3 or M4 if α is an M-node, or because of (20).

At s_0 there are only finitely many $\Delta(p)$ definitions held by nodes β below α. If (20) holds at $s > s_0$ for some $\alpha^\frown x$, it is because one such β was reset while $\tau(\beta)$ was covered by a link. But the link is removed after being travelled, and the next time $\tau(\beta)^\frown\infty \subset \alpha$ is accessible, β's $\Delta(p)$ definition will have been invalidated and r_β set to 0 at step 2. Since no β below α is accessible after s_0, this can happen only finitely often for the finitely many $\Delta(p)$ computations below α. So it will not happen after some stage s_1.

If step 3 is ended after s_1 due to a routine $L(\alpha, x, s)$ for some outcome x of α, according to the induction hypothesis for α the routine will eventually define $\mathrm{TP}_{s,t} = \alpha^\frown x$ and so $\mathrm{TP}_s \supseteq \alpha^\frown x$ at some stage s. If step 3 is ended because of M3 applied to α, then either the Δ-definition made there is permanent (in which case $\alpha^\frown f \subseteq \mathrm{TP}_s$ at some later stage s) or it is not, in which case routine $L(\alpha, \infty, s)$ will be called and the previous argument applies.

Finally, suppose that whenever an $M_{e,p}$-node α is accessible after some s_1, case M4 applies and step 3 is ended at α. We show that eventually the measure condition (23) is satisfied and M3 will apply, a contradiction. At s_1, there are only finitely many nodes $\supset \alpha$ with restraints, and no nodes below α are accessible after s_1. Let s_2 be the second stage after s_1 such that

- any non-permanent restraints below α have been dropped;
- all nodes β above or left of α have settled; ie β is not reset after s_2 and if $r_\beta[s_2] \neq 0$ then $r_\beta[s_2]$ is permanent;
- $\Gamma^{AW}(d_\alpha)\downarrow$ and the use is correct;
- $V^{A\restriction u}[s_2] - V^{A\restriction R_\alpha}[s_2] - U^{\emptyset'}[s_2] = V^{A\restriction u} - V^{A\restriction R_\alpha} - U^{\emptyset'}$, where u is as in M3;
- α is accessible at s_2.

Such stage exists by the induction hypothesis and the fact that new intervals in V^A have use chosen fresh. Every interval in $V^{A\restriction u}[s_2] - V^{A\restriction R_\alpha}[s_2] - U^{\emptyset'}[s_2]$ is in $J_\beta[s_2]$ for some $\beta \supset \alpha$, as otherwise it would be removed in step 4 contradicting the choice of s_2. Letting $E = \{\beta : \beta \supset \alpha \text{ and } r_\beta[s_2] \neq 0\}$, we have

$$\mu(V^{A\restriction u}[s_2] - V^{A\restriction R_\alpha}[s_2] - U^{\emptyset'}[s_2]) = \sum_{\beta \in E} \mu(J_\beta[s_2]).$$

Write $E = F \cup G$ where

$$F = \{\beta \in E : \tau(\beta) \subset \alpha\}; \quad G = \{\beta \in E : \alpha \subset \tau(\beta)\}.$$

Note that at s_2, every node β in F has $d_\beta[s_2 + 1]\downarrow$; as otherwise β has been reset at some $t, s_0 \leq t \leq s_2$, and by choice of s_2 r_β is never set to 0 and β's Δ-definition is never invalidated. But then $\tau(\beta)$ has only finitely many expansionary stages, contradicting that $\tau(\beta)^\frown\infty \subset \alpha$ is accessible infinitely often by induction hypothesis.

Observe that the first clause of Lemma 4.4 holds for any $\beta \in F$ and $s = s_2$, and the second for $\tau = \tau(\beta)$ for any $\beta \in G$ and $s = s_2$. So by (17),

$$\begin{aligned}\mu(V^{A\restriction u}[s_2] - V^{A\restriction R_\alpha}[s_2] - U^{\emptyset'}[s_2]) &= \sum_{\beta\in F}\mu(J_\beta[s_2]) + \sum_{\tau\in\{\tau(\beta):\beta\in G\}}\mu(Q_\tau[s_2]) \\ &< \sum_{\beta\in F} 2\epsilon(\beta) + \sum_{\tau\in\{\tau(\beta):\beta\in G\}} 2\epsilon(\tau) \\ &< \epsilon(\alpha).\end{aligned}$$

Thus (23) will hold at s_2, α will make a $\Delta(p)$ definition which will be permanent, and $\alpha^\frown f$ will be accessible at some stage after s_2. □

LEMMA 4.6. All non-cupping requirements N_e are satisfied.

PROOF. Let τ be the N_e-node on TP. It is clear from the construction that $\tau^\frown\infty \subset$ TP iff there are infinitely many τ-expansionary stages. By Lemma 4.5 and the construction, if α is an M_e-node with $\tau^\frown\infty \subset \alpha \subset$ TP then

- $\alpha^\frown\infty \subset \mathrm{TP} \Rightarrow \Gamma^{AW}(d_\alpha)\uparrow$, and
- $\alpha^\frown f \subset \mathrm{TP} \Rightarrow \Delta_\tau^{W_e}(p)\downarrow$.

To show that for each e the requirement N_e is satisfied assume that $\Gamma_e^{AW_e} = K$ and let τ be the N_e-node on TP. Since $\Gamma_e^{AW_e} = K$ there are infinitely many τ-expansionary stages. First note that by the construction, Δ_τ is consistent, i.e. at each stage s if $\langle\sigma, n, x\rangle, \langle\rho, n, y\rangle \in \Delta_\tau[s]$ and $\sigma \subseteq \rho$ then $x = y$. Also by Lemma 4.5 and the fact that all strategies appear along the true path, the function Δ_τ^W is total and the restraints imposed by each M_e-node below τ when it makes a definition ensure that $\Delta_\tau^W(p) = \Gamma_e^{AW_e}(p) = \emptyset'(p)$ for each $p \in \mathbb{N}$. Thus $W \geq_T \emptyset'$ and N_e is satisfied. □

LEMMA 4.7. $\emptyset' \leq_{LR} A$.

PROOF. We must verify that $U^{\emptyset'} \subseteq V^A$ and $\mu(V^A) < 1$. Once an interval σ appears in $U^{\emptyset'}$ with correct $\emptyset'$-use, according to (18) in any later stage it will be in V^A with the same A-use. Thus eventually it will permanently belong to V^A and $U^{\emptyset'} \subseteq V^A$.

To verify $\mu(V^A) < 1$, since $\mu(U^{\emptyset'}) < \frac{1}{2}$ it suffices to show that $\mu(V^{A\restriction n}[s] - U^{\emptyset'}[s]) < \frac{1}{2}$ for all $n \in \mathbb{N}$ and all $s \geq$ some s_0. Fix n and let s_0 be a stage such that $A \restriction n[s_0] = A \restriction n$

and $V^{A\restriction n}[s_0] - U^{\emptyset'}[s_0] = V^{A\restriction n} - U^{\emptyset'}$. Then for all $s \geq s_0$ we have

$$V^{A\restriction n}[s] - U^{\emptyset'}[s] \subseteq \bigcup_{\tau\subset\delta} Q_\tau[s]$$

where τ runs over the N-nodes and δ is the rightmost path of the tree. Hence, by Lemma 4.4 and the second clause of (17) we have, for $s \geq s_0$,

$$\mu(V^{A\restriction n}[s] - U^{\emptyset'}[s]) \leq \sum_e 2\epsilon(N_e) < \frac{1}{2}.$$

□

This concludes the proof of Theorem 4.1.

CHAPTER 5

Prompt enumerations and relative randomness

The 'dynamic' property of prompt simplicity has become an influential and important concept in the study of the c.e. Turing degrees. The equivalent property of prompt permitting is a particularly fruitful notion, arising neatly from the technique of Yates permitting. In this chapter, we will introduce an analogous notion, *prompt non-low-for-randomness.* Prompt non-low-for-randomness is a prompt form of non-low-for-random permitting, which is the natural notion of permitting in the context of relative randomness. Since non-low-for-random permitting is an analogue of Yates permitting, it is natural to ask if a prompt version of non-low-for-random permitting plays a role similar to the notion of prompt permitting, or gives a nice class of degrees analogous to the promptly simple degrees. We begin to investigate this notion by showing that the class of degrees of promptly non-low-for-random c.e. sets is a proper non-trivial subclass of the promptly simple degrees, and study some other properties of promptly non-low-for-random sets and degrees.

1. Prompt simplicity and Yates permitting

We begin with a discussion of simplicity, permitting and promptness, to establish some terminology and notation that will be useful for defining the notion of prompt non-low-for-randomness in section 2.

Let W_e ($e \in \mathbb{N}$) be a standard listing of all c.e. sets, with a uniformly computable enumeration $W_e[s]$ such that $W_e = \cup_s W_e[s]$. We call this the *canonical listing* of c.e. sets. By a suitable coding, we can consider the W_e as sets of numbers or sets of strings as appropriate. An enumeration of a c.e. set A is a computable sequence $A[s]$ of finite sets such that $A[0] = \emptyset$, $A[s] \subseteq A[s+1]$ and $A = \cup_s A[s]$. The number x *enters* A *at* s if $x \in A[s] - A[s-1]$. In the following, whenever we work with a c.e. set A, we actually work with a particular enumeration $A[s]$ of A. However we will usually suppress the enumeration: when we say 'a c.e. set A' we mean a c.e. set A along with an enumeration $A[s]$ of A. We similarly assume without further mention that c.e. operators U (considered as c.e. sets of axioms) come with a particular enumeration $U[s]$.

Recall that a c.e. set A is *simple* if it is co-infinite and it intersects with every infinite c.e. set; that is,

$$\forall e \qquad W_e \text{ infinite } \Rightarrow \exists x : x \in A \text{ and } x \in W_e. \tag{25}$$

Post (see Cooper [**15**], Theorem 6.2.3) first constructed a simple set in 1944. Dekker [**16**] showed simple sets occur in every nonzero c.e. Turing degree. In the 1980s, certain constructions in connection with structural properties of the c.e. Turing degrees aroused interest in dynamic properties of enumerations of c.e. sets. Maass [**48**] defined the notion of prompt simplicity: a c.e. set A is *promptly simple* if it is co-infinite and there is a computable function p such that

$$\forall e \qquad W_e \text{ infinite } \Rightarrow \exists x, s : x \text{ enters } W_e \text{ at stage } s \text{ and } x \in A[p(s)].$$

This notion was further studied in Maass, Shore and Stob [**49**] and in an influential paper by Ambos-Spies, Jockusch, Shore and Soare [**1**]. It has since become well-known and has been studied extensively. The standard reference is Soare [**68**] chapter XIII, in which a presentation of the following results may be found. Say that a Turing degree is promptly simple if it contains a promptly simple set. Several important results were proved in [**1**] about the promptly simple Turing degrees. In particular, they prove that the promptly simple Turing degrees form a strong filter in the c.e. Turing degrees.[1] Moreover, the non-promptly simple degrees form an ideal in the c.e. Turing degrees.[2] They also prove the following important co-incidence between the 'dynamic' property of being promptly simple and two structural properties of the c.e. Turing degrees.

THEOREM 5.1 (Ambos-Spies, Jockusch, Shore, Soare [**1**]). *Let* $\mathbf{a}$ *be a c.e. Turing degree. The following are equivalent.*

(i) $\mathbf{a}$ *is a promptly simple degree,*

(ii) $\mathbf{a}$ *is noncappable; that is, there is no nonzero c.e. Turing degree* $\mathbf{b}$ *such that* $\mathbf{a} \cap \mathbf{b} = \mathbf{0}$,

[1] Recall that a set $\mathcal{F}$ in an upper semi-lattice $\mathcal{L}$ is a filter if $\mathcal{F}$ is closed upwards and in taking greatest lower bounds (when they exist); it is a strong filter if it is closed upwards and every pair of elements in $\mathcal{F}$ bound a third element in $\mathcal{F}$.

[2] A set $\mathcal{I}$ in an upper semi-lattice $\mathcal{L}$ is an ideal if it is closed downwards and in taking least upper bounds.

(iii) $\mathbf{a}$ is low cuppable; that is, there is a c.e. Turing degree $\mathbf{b}$ such that $\mathbf{b}' = \mathbf{0}'$ and $\mathbf{a} \cup \mathbf{b} = \mathbf{0}'$.

An important equivalent condition to prompt simplicity is *prompt permitting.* We first discuss the general technique of permitting before considering prompt permitting, to establish some notation and terminology which will be useful later for defining prompt non-low-for-randomness.

The general idea behind permitting is, given a c.e. set A with some noncomputability property (such as 'being noncomputable' or 'being non-low-for-random'), to construct a c.e. set B in such a way that $B \upharpoonright n$ changes at stage s only if $A \upharpoonright n$ also changes at stage s (while also satisfying some other requirements). This guarantees that $B \leq_T A$, since given an oracle for A we can compute $B \upharpoonright n$ by finding a stage such that $A[s] \upharpoonright n = A \upharpoonright n$; after this stage, $B \upharpoonright n$ cannot change. We can use the noncomputability condition to guarantee enough changes in $A \upharpoonright n$ to fulfill our requirements. The exact method of ensuring A-changes depends on the noncomputability condition. We typically do it by enumerating 'request sets', with each enumeration being a request that A change on some initial segment. In its original form as used by Friedberg [**26**] and Yates [**73**], now known as Yates permitting, the noncomputability condition is 'being noncomputable', and the request sets are c.e. sets of numbers x, where each number x is a request that $A \upharpoonright x$ changes. Yates permitting is based on the following lemma.

LEMMA 5.2. Suppose that A is a non-computable c.e. set and W is an infinite c.e. set. Then

$$P^Y(W) := \{x \in W : x \text{ enters } W \text{ at some stage } s \text{ and } A[s] \upharpoonright x \neq A \upharpoonright x\}$$

is infinite.

PROOF. Assume that W is infinite and $P^Y(W)$ is finite; we argue that A is computable. Let s_0 be the least stage such that $P^Y(W) \subseteq W[s_0]$. To compute $A \upharpoonright x$, find a stage $s > s_0$ such that some $y > x$ enters W at y. Since $y \notin P^Y(W)$, $A[s] \upharpoonright x = A \upharpoonright x$. □

In the notation $P^Y(W)$ (and similar notations used later) there is an implicit dependence on A; however since A is usually understood to be fixed, and to avoid additional subscripts we omit explicit mention of the set A.

Thinking of W as a request set, the set $P^Y(W)$ is the *permission set* for W, consisting of those requests which are successful. (The superscript Y for 'Yates' is to distinguish this from the non-low-for-random permission set $P(W)$ we will define later.) The lemma guarantees that infinitely many requests from W will succeed.

Although Lemma 5.2 guarantees that infinitely many requests will succeed, it provides no indication of how long we might have to wait for any particular request to succeed. We can obtain a notion of prompt permitting by requiring that the successful permissions occur within a computable time interval. Fix the c.e. set A and let $p : \mathbb{N} \to \mathbb{N}$ be a computable function such that $p(s) > s$. Let

$$\text{(26)} \quad \mathrm{PP}^Y_p(W_e) = \{x \in W_e : x \text{ enters } W_e \text{ at some stage } s \text{ and } A[s] \upharpoonright x \neq A[p(s)] \upharpoonright x\}.$$

$\mathrm{PP}^Y_p(W_e)$ is the set of *prompt permissions* from W_e, with respect to the function p. Say that A is *promptly permitting* if there is a computable function p such that $p(s) > s$ and

$$\text{(27)} \qquad \forall e \qquad W_e \text{ infinite } \Rightarrow \mathrm{PP}^Y_p(W_e) \text{ infinite.}$$

This can be seen as a strong version of Lemma 5.2, ensuring that infinitely many requests will succeed *within a computable time from when the request is made.* An important characterisation of the promptly simple degrees in terms of prompt permitting is the following theorem of [**1**].

THEOREM 5.3 ([**1**]). Let A be a c.e. set. The following are equivalent:

(i) A has promptly simple degree;

(ii) there is a computable function p such that for all e,

$$W_e \text{ infinite } \Rightarrow \mathrm{PP}^Y_p(W_e) \text{ infinite ;}$$

(iii) there is a computable function q such that for all e,

$$W_e \text{ infinite } \Rightarrow \mathrm{PP}^Y_q(W_e) \neq \emptyset.$$

The degrees of promptly simple and the degrees of promptly permitting c.e. sets thus co-incide. Although the property of prompt simplicity lends its name to this important class of degrees, the property of prompt permitting is in many ways the better notion. Properties (ii) and (iii) are often the more convenient to work with, being phrased in terms of permitting that makes them applicable to permitting constructions without modification. Also, prompt permitting is degree-invariant, in the sense that if some c.e. set of degree **a** satisfies (ii) or (iii) then all c.e. sets of degree **a** satisfy (ii) and (iii), unlike prompt simplicity. Finally, simplicity can be seen as a special kind of noncomputability arising from the particular sparseness property (25), whereas the permitting property of Lemma 5.2 applies to all noncomputable sets regardless of special properties. Prompt permitting is thus descended from the more general noncomputability property. Although in the remainder of the chapter we talk of prompt simplicity and promptly simple degrees, we are really thinking of prompt permitting. When we define prompt non-low-for-randomness in section 2, we will define it in terms of prompt permitting, in analogy with property (ii) of Theorem 5.3.

Note that condition (27) only concerns c.e. sets W_e from the canonical listing. Suppose X is a c.e. set, and let e be such that $X = W_e$ (although the enumerations $X[s]$ and $W_e[s]$ may differ). Condition (27) requires that we measure promptness of permissions from X *relative to the canonical enumeration* $W_e[s]$, and not relative to some other non-canonical enumeration $X[s]$. In particular, if we enumerate a request x into a set X during some construction, we must judge promptness not relative to the stage s at which we enumerate x into X, but relative to the stage t at which x is enumerated into W_e according to the canonical enumeration. The usual way to do this is by the Slowdown Lemma.

LEMMA 5.4 (Slowdown Lemma, [**68**] XIII.1.5). *Let X_e be a sequence of c.e. sets with a uniformly computable enumeration $X_e = \cup_s X_e[s]$. There is a computable function $g : \mathbb{N} \to \mathbb{N}$ such that for all e,*

$$W_{g(e)} = X_e \text{ and } W_{g(e)}[s+1] \cap \big(X_e[s+1] - X_e[s]\big) = \emptyset.$$

That is, when we construct a sequence of c.e. sets X_e, we can computably obtain canonical indexes for c.e. sets $W_{g(e)}$ such that numbers enter $W_{g(e)}$ *strictly later* than they enter X_e.

PROOF. By the Recursion Theorem with parameters (Theorem 1.12) we can define g by

$$W_{g(e)} = \{x : \exists s\ (x \in X_e[s] - W_{g(e)}[s])\}.$$

□

With the above notation and terminology established, we can now adapt definition (26) and condition (27) from Yates permitting to non-low-for-random permitting to obtain the notion of prompt non-low-for-randomness.

2. Non-low-for-random permitting and prompt permitting

In the context of relative randomness and LR-degrees, the most natural form of permitting is *non-low-for-random permitting.* This was first used in [**8**], and further developed by Barmpalias in [**3**] and [**2**]. The idea of non-low-for-random permitting is as follows. Let A be a non-low-for-random c.e. set. By Theorem 1.5, there is an A-c.e. set of strings U^A with weight < 1 such that, for any c.e. set of strings W,

$$U^A \subseteq W \;\Rightarrow\; \text{weight } W = \infty. \tag{28}$$

We can use this fact to force changes in the c.e. set A. Suppose at stage s during some construction we wish some number $< n$ to be enumerated into A, so that the approximation to $A \upharpoonright n$ will change. We can request a change by taking a string σ from $U^A[s]$ with use $\leq n$ and enumerating σ into a c.e. set W. If we do this repeatedly, we threaten to make $U^A \subseteq W$. By (28), if we succeed in making $U^A \subseteq W$ then we must have weight $W = \infty$. Since the weight of U^A is finite, the strings in $W - U^A$ must contribute infinite weight to W. Each of these strings $\sigma \in W - U^A$ corresponds to a successful change of some initial segment of A, as we had $\sigma \in U^A[s]$ at the stage when σ was put into W. Hence we are guaranteed enough successful A-changes to ensure weight $W = \infty$. This technique was used for instance in [**3**] to show that every non-zero Δ^0_2 LR-degree has uncountably many predecessors in the LR-degrees.

Note the following point about the technique of non-low-for-random permitting as sketched above. If we enumerate a string σ into W at stage s, then we have $\sigma \in U^A[s]$. Let u be the use of $\sigma \in U^A[s]$. The request corresponding to σ succeeds as soon as $A \upharpoonright u$ changes at a stage after s. Suppose W is a given c.e. set (as opposed to one that we enumerate during a permitting construction). If a string σ is enumerated into W at stage s but $\sigma \notin U^A[s]$, then, as far as permitting is concerned, σ is irrelevant until σ appears in $U^A[t]$ at some $t > s$ (if ever). This observation motivates the following definition of W^*. Fix the c.e. sets A and W, and the universal A-c.e. set U^A. As we approximate $U^A[s]$ in a Σ^0_2 way, we can approximate the c.e. set

$$W^* = \{\sigma : \exists s\ (\sigma \in W[s] \cap U^A[s])\}$$

via the enumeration

$$W^*[s] = \{\sigma : \exists t \leq s\ (\sigma \in W[t] \cap U^A[t])\}. \tag{29}$$

That is, a string σ is enumerated into W^* at stage s if s is the first stage at which σ is in *both* $W[s]$ and $U^A[s]$. Note that if $U^A \subseteq W$ then $U^A \subseteq W^*$, and if A is non-low-for-random and $U^A \subseteq W$ then weight W^* is infinite. For the purposes of non-low-for-random permitting, W^* is equivalent to W.

With A fixed, let

$$P_{U^A}(W) = \{\sigma : \sigma \text{ enters } W^* \text{ at some stage } s,$$
$$\sigma \in U^A[s] \text{ with use } u \text{ and } A[s] \restriction u \neq A \restriction u\}.$$

This is the set of strings from W that are permitted via U^A. The 'non-low-for-random permitting principle' (28) can be expressed as

$$U^A \subseteq W \Rightarrow \text{ weight } P_{U^A}(W) = \infty.$$

We can now formulate a notion of prompt non-low-for-random permitting. Let p be a computable function such that $p(s) > s$. By analogy with (26), we can define the *prompt permitting set* for W with respect to U^A and p,

$$\mathrm{PP}_{U^A,p}(W) = \{\sigma : \sigma \text{ enters } W^* \text{ at some stage } s, \tag{30}$$
$$\sigma \in U^A[s] \text{ with use } u \text{ and } A[s] \restriction u \neq A[p(s)] \restriction u\}.$$

When the function p and/or class U^A are understood to be fixed, we can omit the subscripts. With this notation established, we can now give a definition of prompt non-low-for-randomness.

DEFINITION 5.5. Let A be a c.e. set. A is *promptly non-low-for-random* if there is an A-c.e. set U^A such that weight $U^A < 1$ and a computable function $p : \mathbb{N} \to \mathbb{N}$ such that, for all e,

$$U^A \subseteq W_e \Rightarrow \text{ weight } \mathrm{PP}_{U^A,p}(W_e) = \infty. \tag{31}$$

We say that A *is promptly non-low-for-random via* U^A, p if U^A and p satisfy (31). This definition asserts that if U^A is contained in a c.e. request set W_e, then the requests which succeed promptly (w.r.t. p) will have infinite weight.

It is clearly equivalent to require only that weight $U^A < \infty$ rather than weight $U^A < 1$ in Definition 5.5. We show in Theorem 5.6 that we can equivalently replace the condition weight $\mathrm{PP}_{U^A,p}(W_e) = \infty$ with weight $\mathrm{PP}_{U^A,p}(W_e) \geq 1$.

As with the definition of promptly (Yates) permitting, Definition 5.5 only concerns c.e. sets from the canonical enumeration. Note that in this case we measure promptness relative to the enumeration of W_e^*, which depends not only on the canonical enumeration of W_e but also on the Σ_2^0 approximation of U^A.

We now give an equivalent condition to that of Definition 5.5 which will be useful later; namely that the condition that weight $\mathrm{PP}(W) = \infty$ can be replaced with weight $\mathrm{PP}(W) \geq 1$.

THEOREM 5.6. Let A be a c.e. set and U be a c.e. operator such that weight $U^A < 1$. Then the following are equivalent:

(i) there is a computable function q such that for all e,

$$U^A \subseteq W_e \Rightarrow \text{weight } \mathrm{PP}_{U^A,q}(W_e) = \infty;$$

(ii) there is a computable function p such that for all e,

$$U^A \subseteq W_e \Rightarrow \text{weight } \mathrm{PP}_{U^A,p}(W_e) \geq 1.$$

PROOF. (i) implies (ii) is immediate, so we prove (ii) implies (i). Assume (ii). We may assume without loss of generality that p is strictly increasing and $p(s) > s$ for all s. Let W_e^* be a delayed enumeration of a subset of W_e such that σ enters W_e^* at s iff σ appears in $W_e[s] \cap U^A[s]$ for the first time at s. We will define the computable function q satisfying (i).

Let $2^{<k}$ denote the set of all binary strings of length $< k$, and let $D_i, i \in \mathbb{N}$ be a standard computable listing of all finite sets of strings. Define the c.e. set $X_{e,k,i}$ by the enumeration

$$X_{e,k,i}[s] = D_i \cup \{\sigma : |\sigma| \geq k \ \wedge \ \sigma \in W_e^*[s]\}.$$

In the limit we have

$$X_{e,k,i} = (W_e^* - 2^{<k}) \cup D_i,$$

and strings of length $\geq k$ enter $X_{e,k,i}$ at the same stage as they enter W_e^*. Let $g(e,k,i)$ be a computable function obtained from the Slowdown Lemma 5.4 such that $W_{g(e,k,i)} = X_{e,k,i}$ and strings enter $W_{g(e,k,i)}$ strictly later than they enter $X_{e,k,i}$. Note that for every k there is some i such that

$$D_i = U^A \cap 2^{<k} = W_{g(e,k,i)} \cap 2^{<k}.$$

We now define q. Fix s, and let

$$Z = \{\langle e, \sigma \rangle : \sigma \text{ enters } W_e^* \text{ at s}\}.$$

For each pair $\langle e, \sigma \rangle \in Z$, for each $k < |\sigma|$ and for each i such that $D_i \subseteq 2^{<k}$, let $t_{e,\sigma,k,i}$ be the stage when σ enters $W_{g(e,k,i)}$. Define $q(s)$ to be the maximum of $p(t_{e,\sigma,k,i})$ over all those $t_{e,\sigma,k,i}$ just defined, or $q(s) = p(s) + 1$ if no $t_{e,\sigma,k,i}$ were defined (ie., if Z is empty). Note that $q(s) > t_{e,\sigma,k,i} > p(s)$ since $t_{e,\sigma,k,i} > s$ and p is increasing.

We claim that if $|\sigma| \geq k$ and $\sigma \in \mathrm{PP}_{U^A,p}(W_{g(e,k,i)})$ for some i then $\sigma \in \mathrm{PP}_{U^A,q}(W_e)$. Suppose that $\sigma \in \mathrm{PP}_{U^A,p}(W_{g(e,k,i)})$ and $|\sigma| \geq k$. Since $W_{g(e,k,i)} - 2^{<k} \subseteq W_e^*$, σ enters W_e^* at some stage s, and therefore enters $W_{g(e,k,i)}$ at some $t > s$. Let u be the use of $\sigma \in U^A[s]$. If $\sigma \in U^A[t]$ with use u, then we must have $A[t] \restriction u \neq A[p(t)] \restriction u$ since σ is promptly permitted, and hence $A[s] \restriction u \neq A[q(s)] \restriction u$ since $q(s) \geq p(t)$. So $\sigma \in \mathrm{PP}_{U^A,q}(W_e)$. Otherwise, either σ is not in $U^A[t]$ or it is in $U^A[t]$ with some other use; but either way A must have changed below u between s and t and hence $\sigma \in \mathrm{PP}_{U^A,q}(W_e)$ since $q(s) > t$.

Now we can prove that q satisfies (i). Let ϵ be such that $0 < \epsilon < 1-$weight U^A. Suppose that $U^A \subseteq W_e$. By (ii), $\mathrm{PP}_{U^A,p}(W_e) - U^A$ has weight $> \epsilon$, and so $\mathrm{PP}_{U^A,q}(W_e) - U^A$ has weight $> \epsilon$ also since $\mathrm{PP}_{U^A,p}(W_e) \subseteq \mathrm{PP}_{U^A,q}(W_e)$. Let k be such that

$$(\mathrm{PP}_{U^A,q}(W_e) - U^A) \cap 2^{<k} \tag{32}$$

has weight $> \epsilon$. For some i we have $D_i = U^A \cap 2^{<k}$. Then $U^A \subseteq W_{g(e,k,i)}$, and therefore

$$\mathrm{PP}_{U^A,p}(W_{g(e,k,i)}) - U^A \tag{33}$$

has weight $> \epsilon$. By the earlier claim, $\mathrm{PP}_{U^A,p}(W_{g(e,k,i)}) \subseteq \mathrm{PP}_{U^A,q}(W_e)$. Since the sets (32) and (33) are disjoint, $\mathrm{PP}_{U^A,q}(W_e)$ has weight $> 2\epsilon$. We may now repeat the argument with k such that

$$\text{weight}\left((\mathrm{PP}_{U^A,q}(W_e) - U^A) \cap 2^{<k}\right) > 2\epsilon$$

to show that weight $\mathrm{PP}_{U^A,q}(W_e) > 3\epsilon$. We can repeat this argument arbitrarily many times, each time adding ϵ to weight $\mathrm{PP}_{U^A,q}(W_e)$. Hence weight $\mathrm{PP}_{U^A,q}(W_e)$ is unbounded, establishing (i). □

3. Prompt non-low-for-random sets

One usually obtains an example of a promptly simple set by analysing the standard simple set construction. That is, Post's example of a simple set (see Cooper [**15**], Theorem 6.2.3) turns out in fact to be promptly simple via the identity function. The same thing occurs in the case of prompt non-low-for-randomness; a standard construction of a non-low-for-random c.e. set in fact yields a set which is promptly non-low-for-random. We now give a version of the non-low-for-random construction, adapted slightly to simplify the promptness verification.[3]

THEOREM 5.7. *There is a c.e. set A which is promptly non-low-for-random. Moreover, we can have $A \geq_T \emptyset'$.*

We will construct the set A, as well as a sequence of c.e. operators T_e to satisfy

$$P_e: \qquad T_e^A \subseteq W_e \Rightarrow \text{weight } \mathrm{PP}_{T_e^A,\mathrm{id}}(W_e) \geq 1$$

where id is the identity function. We ensure that weight $T_e^A \leq 2^{-e-1}$ and that the T_e^A are pairwise disjoint (as sets of strings). Hence we may set $T^A = \cup_e T_e^A$ to obtain a bounded A-c.e. set such that A is promptly non-low-for-random by Theorem 5.6 via T^A and the identity function. We will argue that $A \geq_T \emptyset'$ automatically.

[3] Usually one would obtain a non-low-for-random c.e. set indirectly, for instance by using the fact that all low-for-randoms are low and hence that any non-low c.e. set is non-low-for-random. If one wished to explicitly construct a non-low-for-random c.e. set, the construction of Theorem 5.7 would be the standard method.

The basic strategy for P_e will be to put a string σ of fixed length l into T_e^A and wait for $\sigma \in W_e[s]$. As soon as this occurs, we remove σ from T_e^A by enumerating into A below the use of $\sigma \in T_e^A[s]$, thus promptly permitting σ. Then we repeat with the next string of length l. After at most 2^l many repetitions, we will either have some string σ which never appears in W_e and is permanently in T_e^A, and hence $T_e^A \not\subseteq W_e$, or we will have promptly permitted all strings of length l, which have weight 1. At any stage, there is at most one string in T_e^A of length l; thus T_e^A has weight at most 2^{-l}. Let $W_e^*[s]$ be (as in equation (29)) a delayed enumeration of a subset of W_e, such that a string σ is enumerated into W_e^* at stage s iff s is the least such that $\sigma \in W_e[s] \cap T_e^A[s]$.

To simplify the verification, each time a higher priority requirement acts we will make P_e start over with a new length l. Let $j(e, s)$ be the number of times that any requirement $P_{e'}$ with $e' < e$ has acted by stage s, and let $l_e[s] = p_e^{j(e,s)+1}$, where p_e is the e'th prime. At stage s, P_e will use strings of length $l_e[s]$. This simplifies the verification by ensuring that no two requirements will ever use strings of the same length.

The construction consists of the P_e requirements in a finite injury setting. Initially we have A, T_e all empty. At stage 0, do nothing. At stage $s+1$, let e be the least such that $T_e^A[s] \subseteq W_e^*[s]$, or $T_e^A[s] = \emptyset$ and there is some string σ of length $l_e[s]$ not in $W_e^*[s]$.

If there is some $\sigma' \in T_e^A[s]$, then enumerate the use of $\sigma' \in T_e^A[s]$ into $A[s+1]$ (removing σ' from T_e^A). We say that P_e *acts* at $s+1$.

Let σ be the lexicographically least string of length $l_e[s]$ which is not in $W_e^*[s]$, if it exists. If σ exists, declare $\sigma \in T_e^A[s+1]$ with fresh use u. If σ does not exist, do nothing more.

End of construction.

LEMMA 5.8. For each e, P_e acts only finitely often.

PROOF. This is a standard finite injury argument. Assume inductively that the claim holds for $e' < e$, and let s_0 be the least stage such that no P_e' for $e' < e$ acts at any $s \geq s_0$. Then $l_e[s]$ is fixed after s_0. Note that if $\sigma \in T_e^A[s]$ and $\sigma' \in T_{e'}^A[s]$ for $e' > e$ then the latter has larger use than the former. Thus if P_e enumerates into A at stage s then all $T_{e'}^A[s+1]$ become empty, for $e' > e$. If some string is put into $T_{e'}^A$ after s then it will have use larger than that of any string in T_e^A. So enumerations into A by lower priority requirements will not disturb strings in T_e^A, and any strings put into T_e^A after

s_0 will remain there until removed by P_e. After s_0, P_e can act at most $2^{l_e[s_0]}$ times until every string of length $l_e[s_0]$ is in V_e. □

LEMMA 5.9. A is promptly non-low-for-random.

PROOF. Set $T = \cup_e T_e$. Note that each T_e^A contains at most one string, of length $\geq e+1$, so weight $T^A < 1$. Suppose that $T^A \subseteq W_e$. Let s_0 be the least stage such that no requirement $P_{e'}$ for $e' < e$ acts after s_0. Then $l := l_e[s]$ is fixed after s_0. At s_0, P_e will put the first string σ of length l into T^A. Since $T^A \subseteq W_e$, there will be a least $s_1 > s_0$ with $\sigma \in W_e[s_1]$. Since no higher priority requirement ever acts after s_0, P_e will act at s_1 and will enumerate into A below the use of σ. Thus $\sigma \in PP(W_e)$. P_e will then put the next string of length l into $T_e^A[s_1+1]$. Since $T_e^A \subseteq W_e^*$, $W_e^*[s_0]$ contains no strings of length l, and no other requirement uses strings of length l, this will happen 2^l many times, until every string of length l is in $PP(W_e)$. But then weight $PP(W_e) \geq 1$. Hence A is promptly non-low-for-random by Theorem 5.6, via the operator T and the identity function. □

LEMMA 5.10. $A \geq_T \emptyset'$.

PROOF. Let f be a computable function such that

$$W_{f(n)} = \begin{cases} \{0, 00, 000, 0000, \ldots\} & \text{if } n \in \emptyset'; \\ \emptyset & \text{otherwise.} \end{cases}$$

That is, $W_{f(n)}$ enumerates $\emptyset'$, and if it finds $n \in \emptyset'$ then it enumerates strings of zeros of every length. From the proof of Lemma 5.8, it is clear that using an oracle for A we can find the least stage after which $P_{f(n)}$ is never injured. Let s_0 be that stage. At s_0, $P_{f(n)}$ will put a string of zeros into $T_{f(n)}^A$ with some use u. If $n \in \emptyset'$, then P_e will enumerate into A below u. Thus it suffices to find the least stage $s_1 \geq s_0$ such that $A[s_1] \upharpoonright u = A \upharpoonright u$. Then we have $n \in \emptyset'$ iff $n \in \emptyset'[s_1]$. □

The method used above can clearly be combined with other finite-injury strategies. For instance, we could make A low by using the usual lowness strategy, or we could use Sacks restraints (as in Theorem 2.1) to avoid a non-trivial upper cone of Turing or LR-degrees. In the presence of restraints, we can no longer ensure that T_e^A contains at most one string, since a higher-priority requirement might impose a restraint on P_e and prevent it from removing a string from T_e^A. However, by making P_e use longer strings each time it is injured, we can still ensure that T_e^A contains at most one string of each length, so we can keep the weight of T^A under control.

4. Prompt non-low-for-randomness, prompt simplicity and Turing degrees

We now show that the promptly non-low-for-random Turing degrees are a subset of the promptly simple Turing degrees. In section 5 we show that the subset is proper. Although the promptly simple Turing degrees form a filter in the c.e. Turing degrees, it is not known if the promptly non-low-for-random Turing degrees form a filter. However we show that the promptly non-low-for-random degrees are closed upwards under $\leq_T$, and discuss the obstacles to establishing the remaining filter condition.

THEOREM 5.11. Let A be a c.e. set. If A is promptly non-low-for-random then A is of promptly simple degree.

PROOF. Suppose A is promptly non-low-for-random via U and p. We construct a computable function q such that

$$W_e \text{ infinite } \Rightarrow \exists x, s \;:\; x \in W_e[s] - W_e[s-1] \text{ and } A[s] \upharpoonright x \neq A[q(s)] \upharpoonright x.$$

That is, A promptly permits via q and hence has promptly simple degree by Theorem 5.3. We also construct c.e. sets V_e for $e \in \mathbb{N}$, and assume that we have a computable function g given by the Slowdown Lemma 5.4 such that $W_{g(e)} = V_e$ and strings enter $W_{g(e)}$ strictly later than they enter V_e.

Set $q(0) = 0$. At stage $s+1$, do the following for each $e \leq s$. Let x be the largest number which entered W_e at $s+1$, if any. If x exists, and if there is a string $\rho \in U^A[s]$ with use $\leq x$ and $\rho \notin V_e[s]$, then enumerate the oldest such ρ into $V_e[s+1]$. Let t_e be the stage when ρ appears in $W_{g(e)}$. If x or ρ do not exist then t_e is undefined. Finally, let $q(s+1)$ be the maximum of $p(t_e)$ for all those t_e defined at this stage (or $q(s+1) = q(s)$ if no t_e were defined).

Verification. Suppose that W_e is infinite. Then there will be infinitely many stages s when some x enters W_e and there is a $\rho \in U^A[s]$ with use $\leq x$ and $\rho \notin V_e[s]$. (To see this, observe that for every $\rho \in U^A$ there will be such a stage.) So we will enumerate infinitely many strings into V_e, and since we always choose the oldest string to enumerate, we will have $U^A \subseteq V_e$. But then weight $\mathrm{PP}(W_{g(e)}) = \infty$ since A is promptly non-low-for-random. In particular, there is at least one x, s and string ρ such that x enters W_e at s, $\rho \in U^A[s]$ with use $< x$, and $A[s] \upharpoonright x \neq A[q(x)] \upharpoonright x$. □

We show in section 5 that the converse of this does not hold.

We now show that prompt non-low-for-randomness is closed upwards under Turing reducibility. The proof is an adaptation of that of Theorem XIII.1.6 from Soare [**68**].

THEOREM 5.12. If A, B are c.e. sets, $A \leq_T B$ and A is promptly non-low-for-random, then B is promptly non-low-for-random.

PROOF. Suppose that $A = \Phi^B$ for a Turing functional Φ, and that A is promptly non-low-for-random via U, p. Assume without loss of generality that p is nondecreasing, and assume the convention that if $\Phi^B(n)[s] \downarrow$ with use u then $\Phi^B(z)[s] \downarrow$ for all $z < n$ with use $\leq u$. We define T, q so that B is promptly non-low-for-random via T, q. We also construct auxiliary c.e. sets V_e. Let g be a computable function given by the Slowdown Lemma 5.4 such that $W_{g(e)} = V_e$, and strings enter $W_{g(e)}$ strictly later than they enter V_e.

We can define T in advance: when we see a string $\rho \in U^A[s]$ with use u such that $\Phi^B[s] \upharpoonright u = A[s] \upharpoonright u$, then declare $\rho \in T^B[s]$ with use $v = \operatorname{use} \Phi^A(u)[s]$ (if it is not already in $T^B[s]$).

Initially all V_e are empty and $q(s)$ undefined for all s. At stage 0, set $q(0) = 0$. At stage $s+1$, do the following for each $e < s+1$. Let

$$X = \left(W_e[s] \cap T^B[s]\right) - V_e[s]$$

be the strings in W_e and T^B but not in V_e at $s+1$. If X is empty, then do nothing for e at this stage. Otherwise, put each string σ from X into $V_e[s+1]$. For each $\sigma \in X$, let t_σ be the stage when σ was put into T^B with the current computation (ie, the least $t < s+1$ such that $\sigma \in T^B[t]$ with use u such that $B[t] \upharpoonright u = B[s] \upharpoonright u$), and let u_σ be the use of $\sigma \in U^A[t_\sigma]$. Let t_e be the least stage such that for all $\sigma \in X$:

- σ appeared in $W_{g(e)}$ at some $t' > s+1$ and $t_e \geq p(t')$, and
- $\Phi^B[t_e] \upharpoonright u_\sigma = A[t_e] \upharpoonright u_\sigma$.

Finally, at the end of stage $s+1$ declare $q(s+1)$ to be the maximum of those t_e defined at this stage (or $q(s+1) = q(s)$ if no t_e were defined). End of construction.

Verification. First we observe that $T^B = U^A$: certainly if $\sigma \in U^A$ then there will be a stage when $\sigma \in U^A[s]$ via a permanent computation, and $\Phi^B[s]$ correctly computes A on the use. At this stage σ will be put permanently into T^B (if not already). Further, since strings are only put into T^B at s if they are already in $U^A[s]$ and $\Phi^B[s]$ agrees with $A[s]$ on the use, if a string leaves U^A after it has been put into T^B then B must change below the use of the corresponding Φ^B computation, which will remove the string from T^B also.

Suppose now that $T^B \subseteq W_e$. Then $U^A = T^B \subseteq V_e \subseteq W_e$. Since A is promptly non-low-for-random, we must have weight $\mathrm{PP}_{U^A}(W_{g(e)}) = \infty$. We claim that $\mathrm{PP}_{U^A}(W_{g(e)}) \subseteq \mathrm{PP}_{T^B}(W_e)$, and thus $\mathrm{PP}_{T^B}(W_e)$ has infinite weight also.

Suppose $\sigma \in \mathrm{PP}_{U^A}(W_{g(e)})$. Let s_0 be the stage at which we put σ into V_e. At s_0 we have $\sigma \in W_e[s_0]$ and $\sigma \in T^B[s_0]$ with some use v. Moreover, s_0 is the first stage at which σ is in both W_e and T^B (or else we would have put σ into V_e at an earlier stage). Thus it suffices to show that $B[s_0] \restriction v \neq B[q(s_0)] \restriction v$.

Let t_σ and u_σ be as in the construction, and let t' be the stage when σ enters $W_{g(e)}$. We have $A[t_\sigma] \restriction u_\sigma \neq A[p(t')] \restriction u_\sigma$; either because $A[t'] \restriction u_\sigma \neq A[t_\sigma] \restriction u_\sigma$, or if $A[t'] \restriction u_\sigma = A[t_\sigma] \restriction u_\sigma$ then because $\sigma \in \mathrm{PP}_{U^A}(W_{g(e)})$. But

$$\begin{aligned}
\Phi^B[s_0] \restriction u_\sigma &= A[t_\sigma] \restriction u_\sigma \\
&\neq A[q(s_0)] \restriction u_\sigma && \text{since } q(s_0) \geq p(t') \\
&= \Phi^B[q(s_0)] \restriction u_\sigma && \text{by choice of } q(s_0).
\end{aligned}$$

Therefore $B[s_0] \restriction v \neq B[q(s_0)] \restriction v$, and $\sigma \in \mathrm{PP}_{T^B}(W_e)$. □

The equivalent of Theorem 5.12 for LR-reducibility instead of Turing reducibility does not hold. We describe in section 6 that there is an LR-complete c.e. set B which is not promptly non-low-for-random. In particular, B is $\geq_{LR}$ all promptly non-low-for-random sets including the Turing-complete set from Theorem 5.7.

To show that the promptly non-low-for-random degrees form a filter in the c.e. Turing degrees, it would suffice to show that given any two promptly non-low-for-random sets A, B there is a promptly non-low-for-random set C which is computable in both A, B. Given A and B, one would typically use double permitting below both A and B to construct the required C. In the case of the promptly simple degrees, the argument is as follows. Suppose A and B are of promptly simple degree and we want to make $C \leq_T A, B$ promptly simple. Suppose at some stage we see a number x enter a c.e. set W_e, and we would like to make C promptly permit x by changing below x. We could enumerate x into a set V_e, and see if x is promptly permitted by A. If it is not promptly permitted, then we abandon x and try again later with some other number from W_e. If A does promptly permit x, then we enumerate x into a second set V_e' and see if B promptly permits x. If not, again we abandon x. Otherwise, we have received prompt permissions from both A and B, so we can enumerate x into C, satisfying the prompt simplicity requirement for W_e. The fact that A and B are promptly simple ensures that some x will eventually receive both prompt permissions, although arbitrarily many other x may have to be discarded first.

In the case of prompt non-low-for-randomness, we would like to perform a similar construction. Suppose we are given promptly non-low-for-random c.e. sets A, B and we want to construct C, T^C and q such that $C \leq_T A, C \leq_T B$ and C is promptly non-low-for-random via T^C, q. Suppose at some stage we see a string $\sigma \in W_e$ which we would like C to promptly permit. We put σ into T^C with some use u and then attempt to get A and B-permissions to change C below u. If the A and B permissions both succeed then we can change $C \restriction u$, promptly permitting σ. However, if one of the A or B permissions fail, then we cannot remove σ from T^C. It becomes junk and contributes unwanted weight to T^C. The risk is that weight T^C will become infinite. The fact that A and B are promptly non-low-for-random guarantees that the set of strings that do receive both A and B permissions has infinite weight, but makes no guarantee about the strings that do not receive both permissions. In particular we cannot ensure that weight $T^C < \infty$. It is unknown if the promptly non-low-for-random degrees form a filter in the c.e. Turing degrees.

5. Non-prompt non-low-for-randomness

We now present some c.e. Turing degrees that do not contain promptly non-low-for-random sets. One class of c.e. degrees that are not promptly non-low-for-random are the cappable degrees: by Theorem 5.1, the cappable c.e. Turing degrees are exactly the non-promptly simple c.e degrees, and by Theorem 5.11, every non-promptly simple Turing degree is not promptly non-low-for-random. Cappable degrees are known to occur widely, for instance, in every class $\mathbf{Low_n}$ and $\mathbf{High_n}$ [**49**].

Certainly low-for-random degrees cannot be promptly non-low-for-random. These include both promptly simple and non-promptly simple degrees: it is easy to construct a cappable low-for-random c.e. set by adapting the usual minimal pair construction, and the standard cost function construction from **[39]** of a non-computable low-for-random c.e. set yields a promptly simple low-for-random set.

Hence there are c.e. Turing degrees which are promptly simple but not promptly non-low-for-random. This example of a promptly simple low-for-random is not so interesting though, as a low-for-random c.e. set does not (non-low-for-random) permit at all, let alone permit promptly. A more interesting question is whether there is a non-low-for-random c.e. set which is promptly simple but not promptly non-low-for-random. We now give a direct construction of such a set. The strategy for making a set non-promptly non-low-for-random is very similar to that for making a c.e. set cappable in the c.e. Turing degrees (ie. the minimal pair method).

THEOREM 5.13. *There is a non-low-for-random c.e. set A which is of promptly simple degree but is not promptly non-low-for-random.*

We will construct the required c.e. set A. Let $\langle U_e, \phi_e \rangle_{e \in \mathbb{N}}$ be a listing of all pairs of a c.e. bounded operator U and a (possibly partial) computable function ϕ. We assume the convention that if $\phi(x)[s]\downarrow = z$ then $z < s$ and $\phi(y)[s]\downarrow$ for all $y < z$. To ensure that A is not promptly non-low-for-random, it will suffice to construct a c.e. set X_e (with a canonical enumeration $W_{g(e)}$ given by the Slowdown Lemma 5.4) for each pair U_e, ϕ_e such that if ϕ_e is total then $U_e^A \subseteq W_{g(e)}$ but weight $\mathrm{PP}_{U_e,\phi_e}(W_{g(e)}) < 1$. That is, we will satisfy each requirement

$$N_e : \qquad \phi_e \text{ total and } U_e^A \text{ infinite } \Rightarrow U_e^A \subseteq X_e \text{ but weight } \mathrm{PP}_{U_e,\phi_e}(W_{g(e)}) < 1.$$

To ensure that A is of promptly simple degree, it suffices by Theorem 5.1 to make A promptly permitting via the function $p(s) = s + 2$. Thus we have the promptness requirements

$$PS_e : \qquad W_e \text{ infinite } \Rightarrow \exists x, s : x \text{ enters } W_e \text{ at } s \text{ and } A[s] \upharpoonright x \neq A[s+2] \upharpoonright x$$

for each $e \in \mathbb{N}$.

Let V_e be a listing of all bounded c.e. sets; that is, c.e. sets such that weight $V_e < 1$. To make sure that A is not low-for-random we also meet the non-low-for-randomess requirements

$$P_e : \qquad T^A \not\subseteq V_e$$

for $e \in \mathbb{N}$, where T^A is an A-c.e. set with weight < 1 that we construct. By Theorem 1.5 this ensures that A is not low-for-random. The construction will take place on a tree; in fact we will uniformly construct a c.e. operator T_α for each P-node α on the tree, and we will set $T^A = \dot\cup_\alpha T^A_\alpha$.

The strategy for meeting P_e is essentially that used in Theorem 5.7. A P_e-node α will place a string σ of fixed length k into T^A_α and wait for $\sigma \in V_e[s]$. When α sees $\sigma \in V_e[s]$, it removes σ from T^A_α by enumerating into A, and repeats with the next string σ of length k. Since weight $V_e < 1$, after at most 2^k many repetitions we must have some σ such that $\sigma \notin V_e$. This σ will be permanently in T^A_α, but no other string will be permanently in T^A_α. By suitable choice of k we can ensure that weight T^A_α is as small as necessary. Unlike Theorem 5.7, the tree framework will mean that σ will not be removed from T^A_α as soon as it appears in V_e, but only when the node α is next visited. Hence the strategy will make A non-low-for-random, without making it promptly so.

Occasionally it will be necessary to impose a restraint on A; when this happens, α may be unable to enumerate into A to remove a string from T^A_α. In this case, α must abandon its old string σ (which remains permanently in T^A). To ensure that weight $T^A < 1$, α will need to use longer strings in future. It will increment k and restart its strategy. Each time α is injured, the additional amount of junk contributed to T^A halves. In a finite injury setting, α will be able to satisfy its requirement while still contributing an arbitrarily small permanent weight to T^A.

The strategy for N_e will be as follows. We will try to build a c.e. set X such that if ϕ_e is total then $U^A_e \subseteq X$ but $\mathrm{PP}_{U^A_e,\phi_e}(X)$ has finite weight. Strictly, we should actually be concerned with weight $\mathrm{PP}(W_e)$ where W_e is a canonical version of X given by the Slowdown Lemma 5.4, but we overlook this technicality in the following discussion. Suppose at stage s we have some string $\tau \in U^A_e[s]$ with use u, but τ is not yet in X. Put τ into $X[s]$, and note that $\tau \in X[s] \cap U^A_e[s]$ for the first time at s. We want to ensure τ is not promptly permitted with respect to the function ϕ_e. This will happen if A changes below u before stage $\phi_e(s)$. If $A \restriction u$ does change before stage $\phi_e(s)$ then τ will contribute to $\mathrm{PP}_{U_e,\phi_e}(X)$, which we want to keep small. So after we have enumerated τ into X, we will restrain $A \restriction u$ until a stage t such that $\phi_e(s)[t] \downarrow$ (by convention if $\phi_e(s)[t] \downarrow= x$ then $x < t$). At stage t we may drop the restraint, since any change in $A \upharpoonright u$ after t will not contribute to $\mathrm{PP}_{U_e,\phi_e}(X)$. If $\phi_e(s) \uparrow$, then we will never drop the restraint, but nor will we ever impose any additional restraint for N_e. In this case, we have a permanent finite restraint. Otherwise, when ϕ_e is total, the restraint will be dropped infinitely often, providing infinitely many windows for the positive P-requirements to enumerate into A.

This situation is reminiscent of the construction of a minimal pair of c.e. Turing degrees. As in the minimal pair strategy, the negative requirement either drops its restraint infinitely often, or eventually imposes a single finite permanent restraint. With multiple N-requirements working together, the (potential) difficulty for the P-requirements is that the different N-requirements may not drop their restraints at the same time. This is solved exactly as in the minimal pair case by performing the construction on a tree.

For the prompt simplicity requirements, we have to promptly enumerate some number into A as soon as we see a larger number enter the c.e. set W_e. This would appear to be in direct conflict with the negative requirements, which want us to delay enumerations. However, the N_e strategy outlined above would in fact construct a c.e. set X such that $U_e^A \subseteq X$ but $\mathrm{PP}_{U_e^A,\phi_e}(X) = \emptyset$. This is stronger than we need to satisfy N_e; we don't need $\mathrm{PP}(X)$ to be empty, but merely to have small weight. We can allow some weaker priority PS requirements to ignore a higher priority restraint, enumerate into A, and promptly permit some string in X, as long as the total weight that is promptly permitted is small.

The tree will consist of nodes labelled N_e and P_e for $e \in \mathbb{N}$. The prompt simplicity reqirements PS_e do not reside on the tree. Nodes of even length $2e$ are labelled N_e, and nodes of odd length $2e+1$ are labelled P_e. N-nodes have two outcomes $\infty < f$, representing, respectively, the infinitary outcome where ϕ_e is total and U_e^A is infinite, and the finitary outcome where ϕ_e is partial or U_e^A is finite. P-nodes have a single outcome 0. The ordering $\infty < f$ induces an ordering on tree nodes as usual. We denote nodes of the tree by α, β etc. A node β has lower priority than α if β extends α or is to the right of α.

We write U_α, ϕ_α to denote U_e, ϕ_e when α is an N_e-node. Each N-node α will build a c.e. set X_α. Each P-node α has parameters k_α which is a number, and σ_α which is a string of length k_α. Initially, assign each P-node α on the tree a unique parameter k_α such that $\sum_\alpha 2^{-k_\alpha} < 2^{-2}$.

Let g be a function from N-nodes to $\mathbb{N}$ given by the Slowdown Lemma 5.4 such that $W_{g(\alpha)} = X_\alpha$ and strings enter $W_{g(\alpha)}$ strictly later than they enter X_α.

During the construction we will declare some nodes *injured.* When an P-node α is injured, we increment k_α and declare $\sigma_\alpha \uparrow$. We needn't do anything when an N-node is injured except to note the fact.

We will say that an N-node is *expansionary* at a stage s if it is not waiting for a computation $\phi(t)$ to halt, so it is safe to enumerate into A without promptly permitting any strings in X_α (the formal definition is given below). Note that all nodes belonging to a single N-requirement share the same pair U, ϕ and reside on the same level of the tree. Suppose that α, β are N_e-nodes with α to the left of β. If at stage s the node α is waiting for a computation $\phi(t)$ to halt, then it appears to α at that stage that ϕ is partial. Since α has stronger priority than β, β may safely adopt α's judgement and also assume at that stage that ϕ is partial. Thus β need not act while α is waiting for a computation. We can thereby co-ordinate the nodes of each N-requirement so that *at most one node on each level is imposing restraint at any time.* This simplifies calculating the cost of enumerations for PS requirements. Note that this is the same principle used in the proofs of Theorem 2.13 and Theorem 4.1, used first by Cholak, Greenberg and Miller [**12**]. The following definition of expansionary stage captures this principle.

Let α be an N_e-node. Say that a stage s is *α-expansionary* if $X_\alpha[s] = \emptyset$, or

- t is the greatest β-expansionary stage $< s$ for any N_e-node $\beta \leq \alpha$, and some string ρ was put into $X_\beta[t+1]$,
- ρ has appeared in each $W_{g(\gamma)}$ for all N_e-nodes $\gamma \geq \beta$ by some $t', t < t' \leq s$, and t' is the least such,
- $\phi_e(t')[s]\downarrow$, and
- $U_\alpha^A[s] - X_\alpha[s] \neq \emptyset$.

The first three clauses state that α (or a higher priority β) isn't still delaying enumerations to prevent a string from being promptly permitted; the last clause states that there is a new string ready to be added to X_α.

To satisfy a promptly simple requirement PS_e, we will need to enumerate a number into A as soon as some larger number appears in W_e. Such an enumeration might cause some strings from the X_α's to be promptly permitted. We will allow PS_e to injure the lower priority N-requirements $N_{e'}$ for $e' > e$, but we need to ensure that PS_e will cause only a small weight of prompt permissions in the sets X_α belonging to higher-priority N-requirements. Let $i, x, s \in \mathbb{N}$; we define $\text{cost}(i, x, s)$ which is the weight that would be promptly permitted into the sets X_α belonging to N_i-nodes α if x were enumerated into A at stage s. Let β be the leftmost N_i-node such that s is not β-expansionary, if it exists; let ρ be the string most recently added to X_β at some $t+1 < s$, and let u be the use of $\rho \in U^A[t]$. If $x > u$ or if there is no such β then let $\text{cost}(i, x, s) = 0$. Otherwise let $\text{cost}(i, x, s) = 2^{-|\rho|}$.

To prevent the P and PS requirements from interfering with each other, we will reserve the odd numbers for satisfying PS requirements and the even numbers for P. We will assume that when a string σ is put into some T^A_α with use u it remains there until the number u is enumerated into A. In particular, σ remains in T^A_α even if numbers $< u$ enter A.

A P_e-node α *requires attention* at stage $s+1$ if $\sigma_\alpha \uparrow$, or $\sigma_\alpha \downarrow$, $\sigma_\alpha \in T^B_e[s]$ and $\sigma_\alpha \in V_e[s]$.

The construction. At even stages we will take action for N and P-requirements; at odd stages we will take action for PS requirements. At stage 0 and 1, do nothing. At stage $s+1 > 1$, we are given $A[s], \phi_e[s]$ etc and we define $A[s+1]$.

If $s+1$ is odd, then let e be the least such that PS_e is not yet satisfied and there exists z and an odd number $x \leq z$ satisfying

- $z \in W_e[s+1] - W_e[s-1]$,
- $x \notin A[s]$, and
- $\text{cost}(i, x, s) < 2^{-e-1}$ for all $i \leq e$.

If there is no such e then go to the next stage. Otherwise enumerate the least such x into $A[s+1]$ and injure all nodes α of length $|\alpha| > 2e+1$ (these are all the $N_{e'}$ nodes for $e' > e$ or $P_{e'}$-nodes for $e' \geq e$). Declare PS_e to be satisfied.

If $s+1$ is even, then perform steps 1 and 2 below in order.

Step 1. Let α be shortest P-node that requires attention at stage $s+1$ and such that if β is an N-node with $\beta \subset \alpha$ then $\beta^\frown\infty \subseteq \alpha$ iff $s+1$ is β-expansionary. (Note that previously unvisited P-nodes will always require attention, so such a node always exists). Let the current approximation of the true path $\mathrm{TP}_{s+1} = \alpha$.

- If $\sigma_\alpha[s] \uparrow$, then let $\sigma_\alpha[s+1]$ be the lexicographically least string of length $k_\alpha[s]$ which is not in $V_\alpha[s]$. Put σ_α into $T_e^B[s+1]$ with large even use. (Note that such a string must exist since weight $V_\alpha < 1$.)
- Otherwise, $\sigma_\alpha \downarrow$, $\sigma_\alpha \in T_\alpha^B[s]$ with some use u and $\sigma_\alpha \in V_\alpha[s]$. Enumerate u into $A[s+1]$ to remove σ_α from $T_\alpha^A[s+1]$, and declare $\sigma_\alpha[s+1] \uparrow$.

Step 2. For each N-node $\beta \subset \mathrm{TP}_{s+1}$ such that s is β-expansionary, in order of length, do the following. Let ρ be the oldest string in $U_\beta^A[s] - X_\beta[s]$. That is, the unique $\rho \in U_\beta^A[s] - X_\beta[s]$ such that ρ was enumerated into U_β^A at some $s' \leq s$ with use u, $A[s] \upharpoonright u = A[s'] \upharpoonright u$, and if $\rho' \in U_\beta^A[s]$ then ρ' was enumerated into U_β^A after s' or $\rho' \geq \rho$ (in the usual length/lexicographic order). Enumerate ρ into $X_\beta[s+1]$ and into $X_\gamma[s+1]$ for all N_e-nodes γ to the right of β. If ρ does not exist then do nothing for β. Injure all P-nodes of lower priority than TP_{s+1}.

End of construction.

Verification. Say that s is an α-stage if s is even and $\alpha \subseteq \mathrm{TP}_s$. α is *accessible* at s if s is an α-stage. Define the true path $\mathrm{TP} = \liminf_{\text{even } s} \mathrm{TP}_s$. We verify simultaneously that the true path TP is infinite, each node on TP is injured only finitely often, and that each P-node on TP requires attention only finitely often.

LEMMA 5.14. For each n, there is a unique node α of length n such that

(i) α is accessible infinitely often, and is the leftmost such node of length n;
(ii) α is injured only finitely often;
(iii) if α is a P-node, then α requires attention only finitely often and there is a string $\sigma \in T_\alpha^A$ permanently but $\sigma \notin V_\alpha$.

PROOF. Induction on the length n. The claim holds trivially for the root node. Assume inductively that β is the node of length n satisfying the claim, and that s_0 is a stage such that β is never injured or receives attention after s_0. If β is an N-node, then for every β-stage $s > s_0$ we have either $\beta^\frown\infty \subseteq \mathrm{TP}_s$ or $\beta^\frown 0 \subseteq \mathrm{TP}_s$. If β is a P-node, then it has only one child $\beta^\frown 0$, and since β never receives attention after s_0, we have $\beta^\frown 0 \subseteq \mathrm{TP}_s$ for all β-stages $s > s_0$. Hence some child of β is accessible infinitely often, and so (i) holds. Let α be the leftmost such.

The node α can only be injured when $\mathrm{TP}_s < \beta$ or when some requirement PS_e with $2e+1 < |\alpha|$ enumerates into A. The former occurs only finitely often by the induction hypothesis, and there are only finitely many PS requirements with $2e+1 < |\alpha|$, and each acts at most once. So α is injured only finitely often.

Suppose that α is a P-node. Let s_1 be the least α-stage such that α is never injured at any $s \geq s_1$. Then k_α is fixed after s_1, and α receives attention at s_1. Since weight V_α is bounded, it cannot contain all strings of length k_α. Hence there is a lexicographically least string $\sigma \notin V_\alpha$ of length k_α. After receiving attention finitely many times after s_1, α will set $\sigma_\alpha = \sigma$ and will put σ into $T^A_\alpha[s]$. After this, σ is in T^A_α permanently and α will never require attention again. $\square$

We can now verify that the P and N requirements are satisfied by the nodes on the true path.

LEMMA 5.15. Each requirement P_e is satisfied. Therefore A is not low-for-random.

PROOF. By Lemma 5.14, $T^A = \cup_\alpha T^A_\alpha \not\subseteq V_e$ for all e. We just need to verify that weight $T^A < 1$. Let α be a P-node and q_α be the initial value of k_α. Since k_α is increased each time α is injured, and at most one string is left in T^A_α with each injury, we have weight $T^A_\alpha \leq \sum_n 2^{-q_\alpha - n - 1}$. By the choice of the q_α, weight $T^A \leq \sum_\alpha 2^{-q_\alpha} < 1$. $\square$

LEMMA 5.16. Each requirement N_e is satisfied. Therefore A is not promptly non-low-for-random.

PROOF. Let α be the N_e-node on TP, and let s_0 be the least α-stage such that α is never injured at any $s \geq s_0$. If ϕ_α is partial or U^A_α is finite, then N_e is satisfied and there are only finitely many α-expansionary stages after s_0. Suppose that ϕ_α is total and U^A_α is infinite. Then there are infinitely many α-expansionary stages after s_0, and at each such stage s we put some string from $U^A_\alpha[s]$ into X_α. Since we always choose the oldest string from $U^A_\alpha[s] - X_\alpha[s]$, if $\rho \in U^A_\alpha$ permanently then eventually we will put ρ into X_α. Hence $U^A_\alpha \subseteq X_\alpha = W_{g(\alpha)}$. We argue that weight $\mathrm{PP}_{U_\alpha, \phi_\alpha}(W_{g(\alpha)}) < \infty$.

Each time α is injured, some strings may be promptly permitted into $\mathrm{PP}(W_{g(\alpha)})$. However α is injured only finitely often so this contributes only finite weight to $\mathrm{PP}(W_{g(\alpha)})$. The tree layout ensures that lower-priority P-requirements will not contribute to $\mathrm{PP}(W_{g(\alpha)})$, and after s_0 nor will higher priority P or PS requirements. So the only contributions to $\mathrm{PP}(W_{g(\alpha)})$ after s_0 can come from requirements PS_i with $i > e$. But each of these acts at most once, and contributes at most $\mathrm{cost}(e, x, s) < 2^{-i-2}$ to $\mathrm{PP}(W_{g(\alpha)})$. Thus the total contribution to $\mathrm{PP}(W_{g(\alpha)})$ after s_0 is at most $\sum_{i>e} 2^{-i-2} = 2^{-e-2}$, which establishes the claim. □

LEMMA 5.17. Each requirement PS_e is satisfied. Hence, A is of promptly simple degree.

PROOF. Let e be such that W_e is infinite. We show that eventually the cost conditions $\mathrm{cost}(i, x, s) < 2^{-e-2}$ hold for each $i < e$ and all sufficiently large x and s. Fix $i < e$, and let α be the N_i-node on the true path. Because an N-node must wait for the leftward nodes before it can have an expansionary stage, we have the following fact. Either (i) α has infinitely many expansionary stages, or (ii) there are only finitely many stages when any N_i-node is expansionary. If (ii) holds, let s_0 be a stage such that no N_i-node is expansionary after s_0. Then $\mathrm{cost}(i, x, s) = 0$ for all $x, s > s_0$. If (i) holds, let s_0 be a stage such that no string shorter than $e + 2$ is added to W_β after s_0 for any N_i-node β. Then $\mathrm{cost}(i, x, s) < 2^{-e-2}$ for all $x, s > s_0$.

Let s_1 be such that $\mathrm{cost}(i, x, s) < 2^{-e-2}$ for all $i < e$ and all $x, s > s_1$, and such that no requirement PS_j for $j < i$ acts after s_1. Let $z, s > s_1 + 1$ be such that z is enumerated into W_e at s. Then some $x \leq z$ will be enumerated into A at the first odd stage $\geq s$ and PS_e will be satisfied, if it is not already satisfied. This establishes Lemma 5.17 and Theorem 5.13. □

As noted earlier, the nodes belonging to each requirement N_i co-operate in such a way that there is at most one node on each level imposing restraint at any time. This is exactly the condition used in Theorem 4.1 for constructing an LR-complete c.e. set. We can in fact modify the above construction to make the set A be LR-complete, by replacing the P-requirements with the LR-completeness strategy exactly as in Theorem 4.1. Since the N-strategy already satisfies the condition that at most one node on each level imposes restraint at a time, the modifications needed to make A LR-complete are straightforward and we omit the details.

6. Prompt non-low-for-randomness and LR-degrees

So far we have been investigating prompt non-low-for-randomness within the Turing degrees. We might ask whether the LR-degrees of promptly non-low-for-randoms form a nice class within the LR-degrees also. Although it is possible that the class of LR-degrees of promptly non-low-for-randoms might have some nice properties, the LR-degrees do not seem as natural a setting for the study of prompt non-low-for-randomness as the Turing degrees. Theorem 5.12 shows that prompt non-low-for-randomness is Turing degree-invariant: if A is promptly non-low-for-random and $A \equiv_T B$ for c.e. sets A, B, then B is also promptly non-low-for-random. This does not hold for LR-degrees however; an LR-degree can contain both prompt and non-prompt non-low-for-randoms. By Theorem 5.7 there is a Turing complete, and therefore LR-complete, promptly non-low-for-random c.e. set. By Theorem 4.1 and the implications

$$\begin{aligned}\text{non-cuppable} \Rightarrow \text{cappable} &\Rightarrow \text{not promptly simple}\\ &\Rightarrow \text{not promptly non-low-for-random}\end{aligned}$$

there is an LR-complete c.e. set that is not promptly non-low-for-random. Hence the LR-degree of $\emptyset'$ contains promptly and non-promptly non-low-for-random c.e. sets. Moreover, by the remarks at the end of section 5, the LR-degree of $\emptyset'$ contains a promptly simple but not promptly non-low-for-random c.e. set. Thus the LR-degree of $\emptyset'$ contains c.e. sets of all possibilities: promptly non-low-for-random, promptly simple but not promptly non-low-for-random, and not promptly simple.

It is not known whether there is a nonzero c.e. LR-degree that contains no promptly non-low-for-random c.e. sets, or whether there is a c.e. LR-degree in which all the c.e. sets are promptly non-low-for-random. One possible approach to the latter is via jump inversion. If $\mathbf{a}$ is a Turing degree which is $\geq_T \emptyset'$ and c.e. in $\emptyset'$, the *atomic jump class* of $\mathbf{a}$ is the set of those c.e. Turing degrees $\mathbf{b}$ such that $\mathbf{b}' = \mathbf{a}$. The Sacks jump inversion theorem (see Soare [**68**] VIII.3.1) states that every atomic jump class is nonempty. Cooper [**14**] showed that there is an atomic jump class that contains only noncappable c.e. Turing degrees: there is a degree $\mathbf{a}$ c.e. in and $\geq_T \emptyset'$ such that if a c.e. degree $\mathbf{b}$ has $\mathbf{b}' = \mathbf{a}$ then $\mathbf{b}$ is noncappable. Since $B \equiv_{LR} C$ implies $B' \equiv_{tt} C'$, the LR-degree of such a $\mathbf{b}$ contains only noncappable, and hence promptly simple, c.e. sets. If Cooper's theorem could be strengthened to produce an atomic jump class of promptly non-low-for-randoms, ie $\mathbf{b}' = \mathbf{a}$ implies $\mathbf{b}$ is promptly non-low-for-random, then this would give a c.e. LR-degree in which all the c.e. sets are promptly non-low-for-random.

A similar jump inversion argument cannot produce a nontrivial c.e. LR-degree without promptly non-low-for-randoms however, because promptly non-low-for-randoms occur in every atomic jump class. Robinson [**63**] proved that the Sacks jump inversion theorem can be done above any low c.e. Turing degree. That is, given a low c.e. Turing degree $\mathbf{d}$ and a Turing degree $\mathbf{a}$ c.e. in and $\geq_T \emptyset'$, there is a c.e. Turing degree $\mathbf{b} \geq_T \mathbf{d}$ with $\mathbf{b}' = \mathbf{a}$. By the comments after Theorem 5.7, there is a low promptly non-low-for-random c.e. set D. By Robinson's theorem, every atomic jump class has a representative $\geq_T D$, and hence promptly non-low-for-random by Theorem 5.12. (The same argument works for showing that promptly simples occur in every atomic jump class.)

Bibliography

[1] K. Ambos-Spies, C. G. Jockusch Jr., R. A. Shore and R. I. Soare. An algebraic decomposition of the recursively enumerable degrees and the coincidence of several degree classes with the promptly simple degrees. *Transactions of the American Mathematical Society* **281** (1984), 109–128.

[2] G. Barmpalias. Elementary differences between the degrees of unsolvability and the degrees of compressibility. To appear.

[3] G. Barmpalias. Relative randomness and cardinality. To appear.

[4] G. Barmpalias. Tracing and domination in the Turing degrees. To appear.

[5] G. Barmpalias, A. E. M. Lewis and M. Soskova. Randomness, lowness and degrees. *Journal of Symbolic Logic* **73** (2008), 559–577.

[6] G. Barmpalias, A. E. M. Lewis and F. Stephan. Π^0_1 classes, LR degrees and Turing degrees. *Annals of Pure and Applied Logic* **156** (2008), 21–38.

[7] G. Barmpalias and A. Montalbán. A cappable almost everywhere dominating computably enumerable degree. *Electronic Notes in Theoretical Computer Science* **167** (2007).

[8] G. Barmpalias and A. Morphett. Non-cupping, measure and computably enumerable splittings. *Mathematical Structures in Computer Science* **19** (2009), 25–43.

[9] S. Binns, B. Kjos-Hanssen, M. Lerman and R. Solomon. On a conjecture of Dobrinen and Simpson concerning almost everywhere domination. *Journal of Symbolic Logic* **71** (2006), 119–136.

[10] C. Calude, P. Hertling, B. Khoussainov and Y. Wang. Recursively enumerable reals and Chaitin Ω numbers. *Theoretical Computer Science* **255** (2001), 125–149.

[11] G. Chaitin. A theory of program size formally identical to information theory. *Journal of the Association for Computing Machinery* **22** (1975), 329–340.

[12] P. Cholak, N. Greenberg and J. S. Miller. Uniform almost everywhere domination. *Journal of Symbolic Logic* **71** (2006), 1057–1072.

[13] S. B. Cooper. On a theorem of C. E. M. Yates. Handwritten notes (1974).

[14] S. B. Cooper. A jump class of noncappable degrees. *Journal of Symbolic Logic* **54** (1989), 324–353.

[15] S. B. Cooper. *Computability Theory.* Chapman & Hall/CRC Press, Boca Raton, New York, London, Washington D.C., 2004.

[16] J. C. E. Dekker. A theorem on hypersimple sets. *Proceedings of the American Mathematical Society* **5** (1954), 791–796.

[17] N. L. Dobrinen and S. G. Simpson. Almost everywhere domination. *Journal of Symbolic Logic* **69** (2004), 914–922.

[18] R. G. Downey. Array nonrecursive degrees and lattice embeddings of the diamond. *Illinois Journal of Mathematics* **37** (1993), 349-374.

[19] R. G. Downey, E. Griffiths and G. Laforte. On Schnorr and computable randomness, martingales, and machines. *Mathematical Logic Quarterly* **50** (2004), 613–627.

[20] R. G. Downey and D. R. Hirschfeldt. *Algorithmic Randomness and Complexity.* Springer, to appear.

[21] R. G. Downey, D. R. Hirschfeldt, J. S. Miller and A. Nies. Relativizing Chaitin's halting probability. *Journal of Mathematical Logic* **5** (2005), 167–192.

[22] R. G. Downey, D. R. Hirschfeldt, A. Nies and F. Stephan. Trivial reals. In *Proceedings of the 7th and 8th Asian Logic Conferences*, Singapore University Press, Singapore, 2003, 103–131.

[23] R. G. Downey, D. R. Hirschfeldt, A. Nies and S. Terwijn. Calibrating randomness. *Bulletin of Symbolic Logic* **12** (2006), 411–491.

[24] R.G. Downey and T.A. Slaman. Completely mitotic r.e. degrees. *Annals of Pure and Applied Logic* **41** (1989), 119–152.

[25] R. G. Downey and M. Stob. Splitting theorems in recursion theory. *Annals of Pure and Applied Logic* **65** (1993), 1–106.

[26] R. M. Friedberg. The fine structure of degrees of unsolvability of recursively enumerable sets. *Summaries of Cornell University Summer Institute for Symbolic Logic*, Communications Research Division, Inst. for Def. Anal., Princeton, 1957, 404–406.

[27] P. Gács. Every sequence is reducible to a random one. *Information and Control* **70** (1986), 186–192.

[28] D. R. Hirschfeldt, A. Nies and F. Stephan. Using random sets as oracles. *Journal of the London Mathematical Society* **75** (2007), 610–622.

[29] M. Ingrassia. P-genericity for recursively enumerable sets. Ph.D. dissertation, University of Illinois at Urbana-Champaign (1981).

[30] B. Kjos-Hanssen. Low for random reals and positive-measure domination. *Proceedings of the American Mathematical Society* **135** (2007), 3703–3709.

[31] B. Kjos-Hanssen, J. S. Miller and D. R. Solomon. Lowness notions, measure and domination. To appear.

[32] S. C. Kleene. On notations for ordinal numbers. *Journal of Symbolic Logic* **3** (1938), 150–155.

[33] A. N. Kolmogorov. On tables of random numbers. *Sankhya, Ser. A* **25** (1963), 369–376.

[34] A. N. Kolmogorov. Three approaches to the definition of the concept "quantity of information". *Problemy Pederači Informacii* **1** (1965), 3–11.

[35] A. Kučera. Measure, Π^0_1-classes and complete extensions of PA. In *Recursion Theory Week (Oberwolfach, 1984)*, Springer, Berlin, 1985.

[36] A. Kučera. On relative randomness. *Annals of Pure and Applied Logic* **63** (1993), 61–67.

[37] A. Kucera and T. A. Slaman. Randomness and recursive enumerability. *SIAM Journal of Computing* **31** (2001), 199–211.

[38] A. Kučera and T. A. Slaman. Low upper bounds of ideals. *Journal of Symbolic Logic* **74** (2009), 517–534.

[39] A. Kučera and S. A. Terwijn. Lowness for the class of random sets. *Journal of Symbolic Logic* **64** (1999), 1396–1402.

[40] A. H. Lachlan. Lower bounds for pairs of recursively enumerable degrees. *Proceedings of the London Mathematical Society* **3** (1966), 537–569.

[41] A. H. Lachlan. The priority method I. *Z. Math. Logik Grundlag. Math* **13** (1967), 1–10.

[42] R.E. Ladner. A completely mitotic nonrecursive recursively enumerable degree. *Transactions of the American Mathematical Society* **184** (1973), 479–507.

[43] R.E. Ladner. Mitotic recursively enumerable sets. *Journal of Symbolic Logic* **38** (1973), 199-211.

[44] A. Li, T. A. Slaman and Y. Yang. A nonlow$_2$ c.e. degree which bounds no diamond bases. Unpublished draft (2001).

[45] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications.* Springer-Verlag, New York, London, 1993.

[46] D. Loveland. A new interpretation of the von Mises' concept of random sequence. *Z. Math. Logik Grundlagen Math.* **12** (1966), 279–294.

[47] J. Lutz. Dimension in complexity classes. *Proceedings of the Fifteenth Annual IEEE Conference on Computational Complexity*, IEEE Computer Society, 2000, 158–169.

[48] W. Maass. Recursively enumerable generic sets. *Journal of Symbolic Logic* **47** (1982), 809–823.

[49] W. Maass, R. A. Shore and M. Stob. Splitting properties and jump classes. *Israel Journal of Mathematics* **39** (1981), 210–224.

[50] P. Martin-Löf. The definition of random sequences. *Information and Control* **9** (1966), 602–619.

[51] E. Mayordomo. A Kolmogorov complexity characterization of constructive Hausdorff dimension. Tech. Report TR01-059, Electronic Colloquium on Computational Complexity, 2001.

[52] W. Merkle, J. S. Miller, A. Nies, J. Reimann and F. Stephan. Kolmogorov-Loveland randomness and stochasticity. *Annals of Pure and Applied Logic* **138** (2006), 183–210.

[53] D. P. Miller. High recursively enumerable degrees and the anti-cupping property. In *Logic Year 1979-80* (M. Lerman et al., editors), Lecture Notes in Mathematics, vol. 859, Springer-Verlag, Berlin, 1981.

[54] J. S. Miller. The K-degrees, low for K degrees, and weakly low for K sets. To appear.

[55] J. S. Miller and L. Yu. Unpublished notes (2006).

[56] A. A. Muchnik. Seminar at Moscow State University, Moscow, Russia, 1999.

[57] A. A. Muchnik, A. L. Semenov and V. A. Uspensky. Mathematical metaphysics of randomness. *Theoretical Computer Science* **207** (1998), 263–317.

[58] A. Nies. Reals which compute little. In *Proceedings of Logic Colloquium 2002*, Chatizdais, Z, Koepke, P. and Pohlers, W., editors, Lecture Notes in Logic 27 (2002), 261–275.

[59] A. Nies. Lowness properties and randomness. *Advances in Mathematics* **197** (2005), 274–305.

[60] A. Nies. *Computability and Randomness.* Oxford University Press, Oxford, 2009.

[61] A. Nies, F. Stephan and S. Terwijn. Randomness, relativization and Turing degrees. *Journal of Symbolic Logic* **70** (2005), 515–535.

[62] P. G. Odifreddi. *Classical Recursion Theory Volume II.* North-Holland, Amsterdam, Oxford, 1999.

[63] R. W. Robinson. Jump restricted interpolation in the recursively enumerable degrees. *The Annals of Mathematics* **93** (1971), 586–596.

[64] G. E. Sacks. A minimal degree below $\mathbf{0}'$. *Bulletin of the American Mathematical Society* **67** (1961), 416–419.

[65] G. E. Sacks. On the degrees less than $\mathbf{0}'$. *Annals of Mathematics* **77** (1963), 221–231.

[66] C. P. Schnorr. Process complexity and effective random tests. *Journal of Computer and System Sciences* **7** (1973), 376–388.

[67] S. G. Simpson. Almost everywhere domination and superhighness. *Mathematical Logic Quarterly* **53** (2007), 462–482.

[68] R. I. Soare. *Recursively Enumerable Sets and Degrees.* Springer-Verlag, Berlin, London, 1987.

[69] R. J. Solomonoff. A formal theory of inductive inference. *Information and Control* **7** (1964), 1–22.

[70] R. Solovay. Draft of a paper (or series of papers) on Chaitin's work. Unpublished handwritten notes, 215 pages, 1975.

[71] M. van Lambalgen. *Random sequences.* Ph.D. dissertation, University of Amsterdam, 1987.

[72] R. von Mises. Grundlagen der Wahrscheinlichkeitsrechnung. *Math. Zeitschrift* **5** (1919), 52–99.

[73] C. E. M. Yates. Three theorems on the degree of recursively enumerable sets. *Duke Mathematical Journal* **32** (1965), 461–468.

[74] C. E. M. Yates. A minimal pair of recursively enumerable degrees. *Journal of Symbolic Logic* **31** (1966), 159–168.

[75] C. E. M. Yates. Recursively enumerable degrees and the degrees less than $0'$. In *Sets, Models and Recursion Theory (Proc. Summer School Math. Logic and Tenth Logic Colloq., Leicester, 1965),* North-Holland, Amsterdam, 1967, 264–271.

[76] L. Yu and Y. Yang. On the definable ideal generated by nonbounding c.e. degrees. *Journal of Symbolic Logic* **20** (2005), 252–270.

[77] D. Zambella. On sequences with simple initial segments. IILC technical report ML 1990-05, University of Amsterdam, 1990.

APPENDIX A

Non-cupping, measure and computably enumerable splittings

Title: Non-cupping, measure and computably enumerable splittings

Authors: George Barmpalias and Anthony Morphett

Barmpalias was supported by EPSRC Research Grant No. EP/C001389/1. Morphett was supported by MEST-CT-2004-504029 MATHLOGAPS Marie Curie Host Fellowship. We would like to thank André Nies for motivating the results of section 3.

Status: Published in *Mathematical Structures in Computer Science* **19**, Issue 1, February 2009, pp 25–43

Key words and phrases: Almost-everywhere domination, computably enumerable Turing degrees, incompleteness

Abstract: We show that there is a computably enumerable function f (i.e. computably approximable from below) which dominates almost all functions and $f \oplus W$ is incomplete, for all incomplete computably enumerable sets W. Our main methodology is the LR equivalence relation on reals: $A \equiv_{LR} B$ iff the notions of A-randomness and B-randomness coincide. We also show that there are c.e. sets which cannot be split into two c.e. sets of the same LR degree. Moreover a c.e. set is low for random iff it computes no c.e. set with this property.

1. Introduction

Computability theory studies the real line from the point of view of relative computation. Interactions with measure theory were explored from fairly early on, see for example [**7, 18, 22**]. The study of algorithmic randomness has produced a large body of work on measure and computability; good references for this are Downey and Hirshfeldt [**9**] and Nies [**21**]. More recently, Dobrinen and Simpson [**8**] introduced the notion of almost everywhere domination which was investigated more deeply in several follow-up papers [**3**, **13**, **5**, **12**]. A function f is *almost-everywhere dominating* if $\mu\{X \in 2^\omega : f \text{ dominates all total } g \leq_T X\} = 1$ where μ is the Lebesgue measure, and a set is almost-everywhere dominating if it computes such a function f. This notion is degree-theoretic and we can also talk about almost everywhere dominating degrees. In this paper we are interested in the computably enumerable almost everywhere dominating degrees. Nies [**20**] noticed that these degrees $\mathbf{a}$ are high, i.e. $\mathbf{a}' \geq \mathbf{0}''$, and Binns, Kjos-Hanssen, Lerman and Solomon showed that there are high c.e. degrees which are not almost everywhere dominating. Cholak, Greenberg and Miller [**5**] established the existence of incomplete c.e. almost everywhere dominating degrees, and Barmpalias and Montalban [**3**] showed that some of them are halves of minimal pairs. In section 4 we show that some of these c.e. degrees are non-cuppable, i.e. their join with any incomplete c.e. degree is incomplete.

THEOREM 1.1. There is a c.e. almost everywhere dominating set A such that $A \oplus W \not\equiv_T \emptyset'$ for all c.e. $W <_T \emptyset'$.

Theorem 1.1 has the very interesting corollary that if a set is computed by all almost everywhere dominating c.e. degrees, then it must be non-cuppable (the existence of noncomputable such sets is still open). Also, it can be viewed as a generalization of a theorem of Harrington (see [**19**]) which asserts that there is a function of c.e. degree which dominates all computable functions and has incomplete join with all incomplete c.e. sets. A fundamental question, which also served as motivation for Theorem 1.1 is whether almost everywhere dominating sets have degree-theoretic properties which are not shared by all the high degrees. More precisely, is there a formula ϕ in the language of $(\mathcal{R}, <)$ (where $\mathcal{R}$ denotes the c.e. Turing degrees) such that for all $\mathbf{a}$ c.e. almost everywhere dominating $\phi(\mathbf{a})$ holds but there is a high c.e. degree $\mathbf{b}$ such that $\phi(\mathbf{b})$ fails?

In section 3 we consider splittings of c.e. sets in relation with relative randomness. It is a natural question whether a c.e. set is the disjoint union of two c.e. sets B, C which induce the same notion of randomness (i.e. the class of random numbers relative to A is the same as the class of random numbers relative to B). We show that this is not always the case and that a set is low for random iff it can compute such a counterexample.

2. Preliminaries

In the following, we use c.e. sets of strings to generate subclasses of the Cantor space. In particular, we never use the relations $\subset$, $\subseteq$, $\supset$ and $\supseteq$, the measure μ and the operations $\cap$ and $\cup$ for sets U of strings; these relations and operations always refer to the class $S(U) = \{A \in 2^\omega \mid \exists n(A \upharpoonright n \in U)\}$. In other words, $\mu(U)$ is $\mu(S(U))$, $U \subseteq V$ iff $S(U) \subseteq S(V)$ and $U \cap V$ denotes actually $S(U) \cap S(V)$, not $S(U \cap V)$. For union, $S(U \cup V)$ and $S(U) \cup S(V)$ would, for both interpretations of $\cup$, anyway be the same. We recall some basic notions of relative randomness. An oracle Martin-Löf test (U_e) is a uniform sequence of oracle machines which output finite binary strings such that if U_e^B is the range of the e-th machine with oracle $B \in 2^\omega$ then for all $B \in 2^\omega$, $e \in \mathbb{N}$, $\mu(U_e^B) < 2^{-(e+1)}$ and $U_e^B \supseteq U_{e+1}^B$. A real A is called B-random if for every oracle Martin-Löf test (U_e) we have $A \notin \cap_e U_e^B$. A universal oracle Martin-Löf test is an oracle Martin-Löf test (U_e) such that for every $A, B \in 2^\omega$, A is B-random iff $A \notin \cap_e U_e^B$. Given any oracle Martin-Löf test (U_e), each U_e can be thought of as a c.e. set of axioms $\langle \tau, \sigma \rangle$. If $B \in 2^\omega$ then $U_e^B = \{\sigma \mid \exists \tau(\tau \subset B \ \wedge \ \langle \tau, \sigma \rangle \in U_e)\}$. The suffix $[s]$ indicates the value of a parameter at the beginning of stage s. The notion of almost everywhere domination turned out to be very related with the so-called LR reduciblity, defined by Nies [**20**]. We say that a set A is LR reducible to set B (and write $A \leq_{LR} B$) if all B-random reals are also A-random. Kjos-Hanssen, Miller and Solomon [**13**] (also see [**23**]) showed that A is almost everywhere dominating iff $\emptyset' \leq_{LR} A$. Kjos-Hanssen [**12**] showed that $A \leq_{LR} B$ iff for some member U of a universal oracle Martin-Löf test, there is a $\Sigma_1^0(A)$ class V^A with $U^B \subseteq V^A$ and $\mu(V^A) < 1$.

3. Splittings of computably enumerable sets inside their LR-degree

Given a c.e. set it is natural to ask if it can be expressed as the disjoint union of two c.e. sets of the same degree as itself. In the context of Turing degrees this notion has been widely studied. Lachlan [**14**] showed that not every c.e. set has this property. The c.e. sets which can be split into two (disjoint) c.e. sets of the same degree are known as *mitotic*. Ladner [**15, 16**] studied further this notion, showing that every noncomputable c.e. set computes a non-mitotic set and that there is a non-zero Turing degree whose c.e. sets are all mitotic. More results about this notion were shown in [**10**], and the reader can find a comprehensive survey on the general theme of splittings of c.e. sets in [**11**].

It is interesting to carry such notions in the context of the LR reducibility. If a c.e. set is low for random, then obviously it can be split into two c.e. sets of the same LR degree. However we show that there is a c.e. set (even a complete one) which does not have this property. That is, there is a c.e. set which cannot be expressed as a disjoint union of two c.e. sets B, C such that the class of B-random numbers is the same as the class of C-random numbers. Moreover, we show that every c.e. set which is not low for random computes a c.e. set which cannot be split into two c.e. sets of the same LR degree. The latter construction is interesting as it demonstrates a notion of "non-low for random permitting": c.e. sets which are not low for random permit certain properties to occur in the Turing degrees below them, as this happens with noncomputable, array noncomputable, non-low$_2$ sets etc.

THEOREM 3.1. There is a c.e. set A that cannot be split into two c.e. sets X, Y such that $A \equiv_{LR} X \equiv_{LR} Y$. Moreover A can be such that $A \equiv_T \emptyset'$.

Proof. Let (X_i, Y_i, V_i, q_i) be an effective list of all quadruples (X, Y, V, q) of c.e. sets X, Y with $X \cap Y = \emptyset$ and pairs V, q where V is a c.e. operator such that $\mu(V^\beta) < q$ for all $\beta \in 2^\omega$, and $q < 1$. It suffices to construct a c.e. set A and a uniform sequence (T_e^A) of $\Sigma_1^0(A)$ classes such that $\mu(T_e^A) < 2^{-e-1}$ and the following requirements are satisfied:

$$R_i : X_i \dot{\cup} Y_i = A \Rightarrow T_i^A \not\subseteq V_i^{X_i} \text{ or } T_i^A \not\subseteq V_i^{Y_i}$$

(the construction will automatically satisfy $A \equiv_T \emptyset'$). Then if $T^A = \cup_i T_i^A$ we have $\mu(T^A) < 1$ and for every $i \in \mathbb{N}$, if $X_i \cup Y_i = A$ then either $T^A \not\subseteq V_i^{X_i}$ or $T^A \not\subseteq V_i^{Y_i}$, which is what we wanted. For each i we define the quota $p_i := (1-q_i) \cdot 2^{-i-2}$ for R_i. The idea for the satisfaction of R_i is to put a clopen set $B_i \subseteq 2^\omega - V_i^{X_i}$ of measure $p_i/2$ and a clopen set $D_i \subseteq 2^\omega - V_i^{Y_i}$ of measure $p_i/2$ into T_i^A (with use u_i), and wait until $C_i \subseteq V_i^{X_i}$ and $C_i \subseteq V_i^{Y_i}$ with use w, where $C_i := B_i \cup D_i$. Then we remove C_i from T_i^A (by enumerating into A) and restrain $A \upharpoonright w$. Note that since $\mu(V_i^{X_i}) < q_i$ and $\mu(V_i^{Y_i}) < q_i$ the procedure is well defined. Since $X_i \cap Y_i = \emptyset$, either $A \upharpoonright w \neq (X_i \cup Y_i) \upharpoonright w$ or B_i permanently stays in $V_i^{X_i}$ or D_i permanently stays in $V_i^{Y_i}$. If we repeat this procedure $\frac{4}{p_i}$ times then either some round has the first outcome, or one of $T_i^A \not\subseteq V_i^{X_i}$, $T_i^A \not\subseteq V_i^{Y_i}$ holds. In any case R_i is satisfied in a Σ_1^0 way. We say that R_i *requires attention at* *stage* s if either $u_i[s], B_i[s], D_i[s]$ are undefined, or they are defined and $B_i[s] \subseteq V_i^{X_i}[s]$, $D_i[s] \subseteq V_i^{Y_i}[s]$ and $(X_i[s] \cup Y_i[s]) \upharpoonright w = A[s] \upharpoonright w$ where w is the least number greater than the use of $B_i[s]$ in $V_i^{X_i}$ and $D_i[s]$ in $V_i^{Y_i}$. When we say *the leftmost clopen subset of Q of measure q* for a clopen set Q such that $\mu(Q) > q$ we mean the unique subset P of Q which has measure q and the property that if $\beta \in P$ then all reals in Q which are lexicographically smaller than β belong to P.

Construction. At stage s let i be the least number $< s$ such that R_i requires attention at s (if there is no such i, go to the next stage). If $u_i[s] \uparrow, B_i[s] \uparrow, D_i[s] \uparrow$, choose the leftmost clopen set $B_i[s+1] \subseteq 2^\omega - V_i^{X_i}[s]$ of measure $p_i/2$ and the leftmost clopen set $D_i[s+1] \subseteq 2^\omega - V_i^{Y_i}[s]$ of measure $p_i/2$, and put them into T_i^A with *big* use $u_i[s+1]$. Otherwise put $u_i[s]$ into A and set $u_j[s+1] \uparrow, B_j[s+1] \uparrow, D_j[s+1] \uparrow$ for all $j \geq i$.

Verification. By the above discussion the construction is well defined, i.e. when it chooses B_i, D_i, suitable such sets exist. Also note that if $u_i[s] \downarrow, u_j[s] \downarrow$ and $i < j$ then $u_i[s] < u_j[s]$. In particular, as long as $u_i[s] \downarrow$, no requirement R_j with $j > i$ can change $A \upharpoonright (u_i + 1)$. Note that $T_i^A[s] = B_i[s] \cup D_i[s]$ (or $\emptyset$ if $B_i[s], D_i[s] \uparrow$); and if B_i, D_i are set $\uparrow$ at s then A changes below $u_i[s]$. So $T^A = \cup_i T_i^A$ is a $\Sigma_1^0(A)$ class. We have $\mu(T_i^A) < 2^{-i-2}$ since at any time $B_i[t] \cup D_i[t]$ has measure at most p_i, which is $< 2^{-i-2}$.

Next we show by induction that all R_i require attention finitely often and are satisfied. Suppose that this holds for all R_j, with $j < i$ and s_0 is the least stage after all stages where one of these R_j requires attention. At (the beginning of) s_0 we must have $u_i[s_0] \uparrow, B_i[s_0] \uparrow, D_i[s_0] \uparrow$ and so R_i will receive attention at s_0. In the following stages R_i can only redefine its parameters at most $\frac{4}{p_i}$ times, since $\mu(V_i^{X_i}) < q_i$ and $\mu(V_i^{Y_i}) < q_i$. When this stops at some stage s_1, we will have $u_j = u_j[s] \downarrow, B_j = B_j[s] \downarrow, D_j = D_j[s] \downarrow$ for all $s > s_1$ and $j \leq i$, and either $A \upharpoonright w \neq (X_i \cup Y_i) \upharpoonright w$ for some w or $B_i \not\subseteq V_i^{X_i}$ or $D_i \not\subseteq V_i^{Y_i}$, and so R_i is satisfied.

Finally we show that $A \equiv_T \emptyset'$. Let f be a computable function such that $X_{f(i)} = Y_{f(i)} = \emptyset$, $q_{f(i)} = 2^{-1}$ and for all i,

$$V_{f(i)}^{X_{f(i)}} = V_{f(i)}^{Y_{f(i)}} = \begin{cases} \{00\} & \text{if } i \in \emptyset' \\ \emptyset & \text{if } i \notin \emptyset' \end{cases}$$

where '00' is a string representing the leftmost quarter of 2^ω. According to the argument above and the construction, $u_i := \lim_s u_i[s]$ exists for all i and A can compute a modulus of convergence for this function, i.e. there is an A-recursive function Φ^A such that for each i and all $t \geq \Phi^A(i)$ we have $u_i = u_i[t]$. Then according to the construction and since we always choose the leftmost suitable clopen set to enumerate into T^A we have that $i \in \emptyset'$ iff $i \in \emptyset'[\Phi^A(f(i))]$ and so $\emptyset' \leq_T A$. □

THEOREM 3.2. If B is c.e. and $B \not\leq_{LR} \emptyset$ then there is a c.e. set $A \leq_T B$ which cannot be split into two c.e. sets X, Y such that $A \equiv_{LR} X \equiv_{LR} Y$.

Proof. We use the ideas and some of the notation in the proof of Theorem 3.1 in a more refined form. Let (U_i) be universal oracle Martin-Löf test and let t_i be the least such that $2^{-t_i} < 2^{-i-2} \cdot (1 - q_i)$ (so that, in particular, $\mu(U_{t_i}^B) < 1 - \mu(V_i)$). Without loss of generality we can assume that

$$\mu(U_{t_i}^B[s]) < 2^{-t_i} < 2^{-i-2} \cdot (1 - q_i) \text{ for all } s. \tag{1}$$

Since $B \not\leq_{LR} \emptyset$ for all Σ_1^0 classes E such that $U_{t_i}^B \subseteq E$ we have $\mu(E) = 1$. To satisfy R_i we will enumerate clopen sets into T_i^A (as before) as well as a Σ_1^0 class E_i such that $U_{t_i}^B \subseteq E_i$. The idea is that, roughly speaking, for any amount that is put into T_i^A (and so $V_i^{X_i}$ or $V_i^{Y_i}$), the same amount is put into E_i. Eventually, the measure in $E_i - U_{t_i}^B$ will translate into measure in $V_i^X - T_i^A$ or $V_i^Y - T_i^A$ (B changes will allow A changes) and by making $\mu(E_i)$ large enough, we have that either $V_i^{X_i}$ or $V_i^{Y_i}$ will stop covering T_i^A; also the measure that permanently stays in T_i^A will be at most $\mu(U_{t_i}^B) < 2^{-i-2}$. Since $\mu(E_i)$ can be at most 1, we will remove $E_i - U_{t_i}^A$ (i.e. the useless measure) from E_i a finite number of times, provided that $\mu(E_i) > 2^{-1} + 2^{-t_i}$. Each time we do this, we ensure that $\mu(V_i^{X_i})$ and $\mu(V_j^{Y_i})$ have increased by a total of 2^{-2}, so that after finitely many times requirement R_i is satisfied.

Order the strings as usual, first by length and then lexicographically. For each stage s and string $\rho \in U_{t_i}^B[s]$ let $p_i(\rho)[s]$ be the stage where ρ was enumerated into $U_{t_i}^B[s]$ with the current computation. At any stage s let $\rho_i[s]$ be the least string in $U_{t_i}^B[s] - E_i[s]$ such that $p_i(\rho_i[s])[s] \leq p_i(\sigma)[s]$ for all $\sigma \in U_{t_i}^B[s] - E_i[s]$ (and $\rho_i[s] \uparrow$ if such string does not exist). Also let $u_i[s]$ be the use of the computation $\rho_i[s] \in U_{t_i}^B[s]$. To *schedule an i-attack at stage s* means to pick a clopen set $C_i^x[s+1] \subseteq 2^\omega - V_i^{X_i}[s]$ of measure $2^{-|\rho_i[s]|-1}$ and a clopen set $C_i^y[s+1] \subseteq 2^\omega - V_i^{Y_i}[s]$ of measure $2^{-|\rho_i[s]|-1}$, and enumerate $C_i[s+1] := C_i^x[s+1] \cup C_i^y[s+1]$ into T_i^A with *big* use $v_i[s+1]$. An attack which was scheduled at stage s is *cancelled* at stage $t > s$ if t is the least stage with $B \restriction u_i[s] \neq B \restriction u_i[t]$. An attack scheduled at stage s is *implemented* at stage $t > s$ if $\rho_i[s]$ is enumerated into E_i at t. If an attack was implemented at stage s and $B[s] \restriction u_i[s] \neq B[t] \restriction u_i[s]$ for some $t > s$ then for the least such stage t we say that *the attack succeeds at stage t*. If an implemented attack succeeds at some stage, we say that it is *successful*; otherwise we say that it is *unsuccessful*. In the following construction when a parameter is not explicitly redefined it retains its previous value and if a string is not explicitly extracted from T_i^A it remains in it (perhaps with a different computation, but then surely with the same A-use). As usual, we assume that $U_i^B[s]$ is prefix-free for all s, as a set of strings.

We say that R_i *requires attention at stage s* if either $v_i[s] \uparrow$, or $v_i[s] \downarrow$, $((X_i \cup Y_i) \restriction v_i)[s] = (A \restriction v_i)[s]$ and one of the following holds:

(I) An i-attack is cancelled at s.

(II) An i-attack was implemented at some stage $t < s$ and it succeeds at stage s.

(III) An i-attack was scheduled at some stage $t < s$, it has not been cancelled or implemented by stage s, $C_i[t] \subseteq V_i^{X_i}[s]$, $C_i[t] \subseteq V_i^{Y_i}[s]$, and $(X_i \cup Y_i)[s] \restriction w = A[s] \restriction w$ for some w greater than the use of $C_i[t]$ in $V_i^{X_i}[s]$ and in $V_i^{Y_i}[s]$.

(IV) All previous attacks have been either implemented or cancelled.

(V) $\mu(E_i[s]) > 2^{-1} + 2^{-t_i}$.

To *initialize* R_i means to empty E_i and T_i^A.

Construction. At stage s pick the least $i < s$ such that R_i requires attention at s (if such i does not exist, go to the next stage) and do the following.

- If an i-attack is cancelled at s, enumerate $v_i[s]$ into A, remove $C_i[s]$ from T_i^A and initialize R_j for all $j > i$.
- If an i-attack was implemented at some stage $t < s$ and it succeeds at stage s, put $v_i[t]$ into A, remove $C_i[t]$ from T_i^A and initialize R_j for all $j > i$.
- If (IV) applies, *schedule an attack* at stage s.
- If (III) applies then enumerate $\rho_i[t]$ into E_i and say that this attack was *implemented* at stage s.
- If $\mu(E_i[s]) > 2^{-1} + 2^{-t_i}$, remove from E_i the set $E_i - U_{t_i}^A[s]$.

Verification. Note that at any stage an attack is scheduled only if all previous attacks are either cancelled or implemented. If an attack is implemented at stage s and another attack is scheduled at $t > s$ (and R_i is not initialized in $[s,t]$) we have $\rho_i[s] | \rho_i[t]$ (so that $[\rho_i[s]] \cap [\rho_i[t]] = \emptyset$). Also, in that case, if $A = X_i \cup Y_i$ we have $C_i^j[s] \cap C_i^j[t+1] = \emptyset$ for some $j \in \{x, y\}$, depending on whether we have $C_i^x[s] \subseteq V_i^{X_i}[t]$ or $C_i^y[s] \subseteq V_i^{Y_i}[t]$ (one of the two must occur since at s an i-attack was implemented and only one of X_i, Y_i may change below the relevant use). In particular, E_i is prefix free and $T_i^A[s]$ is prefix free (as a set of strings) for all s.

By induction on the stages we have that if $i < j$ and $v_j[t] \downarrow$ then $v_i[t] \downarrow$ and $v_i[t] < v_j[t]$. This means that if during initialization the set T_i^A is emptied at stage s, then A changes below the smallest use of existing computations of the form $\sigma \in T_i^A$ for strings σ. So (T_i^A) is indeed a uniform sequence of $\Sigma_1^0(A)$ classes, hence T^A is a $\Sigma_1^0(A)$ class. Moreover by the choice of $v_i[s]$, if $A \upharpoonright n$ changes at stage t then $B \upharpoonright n$ changes at stage t. So $A \leq_{ibT} B$ (where *ibT* indicates a Turing reduction with the use function being the identity); in particular $A \leq_T B$.

Next we show that each R_i is satisfied and stops requiring attention after some stage. For a contradiction suppose that there is a least i such that either R_i is not satisfied or it requires attention infinitely often. Suppose that s_0 is the least stage such that R_j, $j < i$ do not require attention at any stage $s \geq s_0$. In any case we have $T_i^A \subseteq V_i^{X_i}$ and $T_i^A \subseteq V_i^{Y_i}$ because otherwise some i-attack would never be implemented or cancelled. This means that i-attacks will be scheduled at infinitely many stages (by the choice of t_i and the fact that $\mu(V_i^{X_i}), \mu(V_i^{Y_i})$ are < 1 there will always be a suitable clopen set for scheduling a new attack) and by the definition of $\rho_i[s]$, infinitely many of them will not be cancelled. In fact, if η is a string in $U^B_{t_i}[s]$ with correct B-use, then for some stage t we will have $\rho_i[t] = \eta$ and if t_0 is the least such stage, the attack scheduled at t_0 will be implemented (and will be unsuccessful). This means that if we never removed measure from E_i after stage s_0 (under the fifth condition for R_i to require attention) then $U^B_{t_i} \subseteq E_i$ and since $B >_{LR} \emptyset$ we have $\mu(E_i) = 1$. In particular $\mu(E_i) > 2^{-1} + 2^{-t_i}$ which means that we will remove useless measure from E_i after s_0. The same argument shows that there will be infinitely many stages $s_1, s_2, \dots$ at which we R_i requires and receives attention under the fifth condition. If we let

$$D_i = \{\rho_i[s] \mid \text{at } s \geq s_0 \text{ an unsuccesful } i\text{-attack was implemented}\}$$

then since

$$E_i[s_j] = \{\rho_i[s] \mid \text{at stage } s \in [s_{j-1}, s_j) \text{ an } i\text{-attack was implemented}\}$$

for each $j \in \mathbb{N}$ we have $E_i[s_j] \subseteq D_i \cup H_i[s_j]$, where

$$H_i[s_j] = \{\rho_i[s] \mid \text{at } s \in [s_{j-1}, s_j) \text{ a successful } i\text{-attack was implemented}\}.$$

If at stage s an unsuccessful attack was implemented we must have $\rho_i[s] \in U^B_{t_i}$, so that $D_i \subseteq U^B_{t_i}$ and $\mu(D_i) < 2^{-t_i}$. Since $D_i \cap H_i = \emptyset$ and $\mu(E_i[s_j]) > 2^{-1} + 2^{-t_i}$ we have $\mu(H_i[s_j]) > 2^{-1}$. But every string $\eta \in \cup_j H_i[s_j]$ corresponds to a pair of clopen sets $C^x_i(\eta), C^y_i(\eta)$ such that

- $\mu(C^x_i(\eta)) = \mu(C^y_i(\eta)) = \frac{\mu([\eta])}{2}$
- $C^x_i(\eta)$ stays permanently in $V_i^{X_i}$ or $C^y_i(\eta)$ stays permanently in $V_i^{Y_i}$
- if $\eta_1 = \rho_i[t_1], \eta_2 = \rho_i[t_2]$ for $s_0 \leq t_1 < t_2$ and $C^x_i(\eta_1)$ stays permanently in $V_i^{X_i}$ then $C^x_i(\eta_1) \cap C^x_i(\eta_2) = \emptyset$; if $C^y_i(\eta_1)$ stays permanently in $V_i^{Y_i}$ then $C^y_i(\eta_1) \cap C^y_i(\eta_2) = \emptyset$

which means that at each s_j either $\mu(V_i^{X_i})$ or $\mu(V_i^{Y_i})$ has increased by at least 2^{-2} since s_{j-1}. Since the sequence (s_j) is infinite and $\mu(V_i^{X_i}), \mu(V_i^{Y_i})$ are less than $q_i < 1$, this is a contradiction.

Finally we need to show that $\mu(T^A) < 1$. Let s_0 be as before, and let W be the set of stages $t \geq s_0$ at which an unsuccessful i-attack was implemented. We have $T^A_i = \cup\{C_i[t] \mid t \in W\}$ and $\mu(C_i[t]) = \mu([\rho_i[t]])$ for all $t \in W$. Hence $\mu(T^A_i) = \mu(D_i)$ and since $D_i \subseteq U^B_{t_i}$ we have $\mu(T^A_i) < 2^{-t_i} < 2^{-i-2}$, which shows that $\mu(T^A) < 1$. $\square$

An obvious question which is left unanswered here is whether the c.e. sets of Theorem 3.1 occur in every non-zero LR degree. We conjecture that this is not the case.

Some properties of c.e. LR degrees can be derived from a combination of known properties of the structure of Turing degrees inside an LR degree and properties of the Turing degrees. As an example we demonstrate the following.

THEOREM 3.3. Let $n \in \mathbb{N}$. If A is c.e. then there exists B of properly n-c.e. Turing degree such that $A \equiv_{LR} B$.

Proof. Since every c.e. LR degree contains noncomputable c.e. sets, we can assume that A is noncomputable. By a result in [**2**] we have that there exists a c.e. set C such that $C <_T A$ and $C \equiv_{LR} A$. Then by the density theorem in [**6**] there is a set B of properly n-c.e. Turing degree such that $C <_T B <_T A$ and so $B \equiv_{LR} A$. □

4. Proof of Theorem 1.1

In the following we fix U to be the second member of a universal oracle Martin-Löf test, so that $\mu(U^X) \leq 2^{-1}$ for all $X \in 2^\omega$. To show Theorem 1.1 it suffices to construct a non-cuppable set A such that $U^{\emptyset'} \subseteq V^A$ for some $\Sigma^0_1(A)$ class V^A of measure < 1.

We adopt the usual assumptions that, for a Turing functional Γ, $\Gamma^X(z)[s]\downarrow$ only if $\Gamma^X(y)[s]\downarrow$ for all $y < z$, and that $\text{use}\,\Gamma^X(y)[s] \leq \text{use}\,\Gamma^X(z)[s] \leq s$ if $\Gamma^X(z)\downarrow$ and $y \leq z$. A Turing functional Γ may be considered as a c.e. set of axioms $\langle z, y, \sigma\rangle$ (asserting that $\Gamma^X(z) = y$ for all $X \in 2^\omega$ with $\sigma \subset X$), which are consistent in the sense that if $\langle z, y, \sigma\rangle$ and $\langle z, y', \sigma'\rangle$ are both in the set, for $y' \neq y$, then σ and σ' are incomparable. We will abbreviate $\Gamma^{X \oplus Y}$ as Γ^{XY}.

4.1. Making A non-cuppable. We describe the basic strategies for a non-cuppable degree, based on [**17, 25**]. We will construct Turing functionals Δ_e to ensure that the following holds for all $e \in \omega$:

$$N_e: \qquad \Gamma_e^{AW_e} = K \Rightarrow \Delta_e^{W_e} = \emptyset' \tag{2}$$

where $\langle \Gamma_e, W_e\rangle$ ranges over all pairs of Turing functionals and c.e. sets; assuming that $\emptyset' \subseteq 2\mathbb{N}$ we let $K = D \cup \emptyset'$ where $D \subseteq 2\mathbb{N}+1$ is an auxiliary that we enumerate. In the following discussion we omit the index e. The idea is to let Δ^W copy Γ^{AW} by monitoring the reduction Γ^{AW} and restraining A to preserve the agreement of the two reductions.

The problem with this approach is that the restraint on A may well have limit ∞, in which case very little can be done to make A nontrivial, let alone LR-above $\emptyset'$. The solution is to split N into infinitely many subrequirements M_p which are responsible just for the definition of $\Delta^W(p)$, thus splitting an infinite restraint into infinitely many finite restraints. The strategies for the subrequirements M_p will be coordinated by a master N strategy which will make sure that Δ is consistent and this coordination will be implemented on a tree of strategies.

We can think of N having two outcomes $\infty \prec f$ (i.e. ∞ is to the left of f) corresponding to whether there are infinitely many expansionary stages in $\Gamma^{AW} = K$ or not, and M_p outcomes $\infty \prec f$ according to whether $\Gamma^{AW}(p) \uparrow$ or equivalently, $\Delta^W(p) \downarrow$. This induces a uniformly labelled tree of strategies where each level is occupied by either some N or some M_p. For the consistency of Δ we make sure that at any M_p-level (i.e. occupied by an M requirement) and at any stage at most one node α will be responsible for $\Delta^W(p) \downarrow$ (by preserving A in $\Gamma^{AW}(p) \downarrow$). Any nodes to the right of α may adopt that Δ-definition but if a node to the left of α wishes to define $\Delta(p)$ it must first cancel the Δ computation that α holds. This happens by enumerating something into the auxiliary set D which in turn causes a W-change (provided that the Γ reduction is valid). Eventually, if $\Gamma^{AW} = K$, at each M_p level there will be exactly one node *on or to the left of the true path* which permanently preserves $\Delta^W(p) \downarrow = \emptyset'(p)$. Otherwise some node will witness partiality. As in any $\emptyset''$ priority argument the restraints imposed on a node on the true path will be finite.

Each M_p-node α has a *flip-point* d, which is the number enumerated into D when we wish to cancel the computation $\Delta(p) \downarrow$. When α is visited, it checks if the computation $\Gamma^{AW}(d)$ has changed since the last time it was visited and if so, it plays outcome ∞. Otherwise we may define $\Delta^W(p) = \Gamma^{AW}(p)$, with W-use $u = \text{use}\,\Gamma^{AW}(d)$ and restrain $A \upharpoonright u$. If we later want to visit a node β to the left of α, we enumerate the flip-point d into D whilst maintaining α's A-restraint. This enumeration should force a W-change below u, and so α will not hold a Δ-computation anymore (if this does not happen then N will be satisfied by a finite outcome). Then we can drop the restraint of α and β can take action. This must happen immediately upon seeing the N-expansionary stage, otherwise some other node α' to the right of β may act first and define another Δ-computation which prevents β from being visited. For this reason when we enumerate

d into D we create a link (τ, β) from the N-node τ to β and when τ is next visited at an expansionary stage we will follow the link straight to β.

4.2. Measure-guessing nodes and LR-completeness. To make A LR-complete, it suffices to construct a $\Sigma_1^0(A)$ class V^A with $U^{\emptyset'} \subseteq V^A$ and $\mu(V^A) < 1$. Without loss of generality we assume that if $\langle \sigma, \tau \rangle$ is enumerated into U at stage s then $|\sigma| = |\tau| = s$. We will also use the *hat-trick* for $U^{\emptyset'}$: let $k_s = \min\{x : x \in \emptyset'[s] - \emptyset'[s-1]\}$, or $k = s$ if there are no such x and define $\widehat{\emptyset'}[s] = \emptyset'[s] \upharpoonright k$. Then $\widehat{U^{\emptyset'}}[s] = \{\sigma : \langle \sigma, \tau \rangle \in U_s \text{ for some } \tau \subseteq \widehat{\emptyset'}[s]\}$. In the following we assume that $U^{\emptyset'}[s]$ and $\emptyset'[s]$ refer to $\widehat{U^{\emptyset'}}[s]$ and $\widehat{\emptyset'}[s]$ respectively. Then infinitely often we have *true stages* s at which $U^{\emptyset'}[s] = U^{\emptyset' \upharpoonright n} \subset U^{\emptyset'}$ for some n, and thus $\mu(U^{\emptyset'}[s]) < \mu(U^{\emptyset'})$.

Whenever an interval σ appears in $U^{\emptyset'}$, we add it to V^A with large A-use u. If a $\emptyset'$-change later removes σ from $U^{\emptyset'}$, we could remove it from V^A by enumerating u into A, provided that u is not restrained by some requirement. The A-change may also remove some legitimate intervals from V^A, but we add these again with the same use as before. This clearly gives $U^{\emptyset'} \subseteq V^A$. The main conflict is that the A-restraints will prevent us from removing some superfluous 'junk' intervals σ from V^A. For the argument to succeed, we must ensure that the total measure of junk intervals $\mu(V^A - U^{\emptyset'}) < \frac{1}{2}$. We assign each requirement (each level of the tree) a quota ϵ, which is the amount of junk measure that requirement is allowed to capture. We implement the negative strategies in such a way that we have at most one node imposing restraint at each level of the tree. A restraint may only be imposed on A if the (current) junk measure that it captures is less than the quota. To ensure that strategies will eventually be able to impose restraints under this restriction, we choose the quota $\epsilon(k)$ of level k of the tree so that $\sum_{j>k} \epsilon(j) < \epsilon(k)$ (in this way the lower priority requirements will not capture more than $\epsilon(k)$ of junk).

To ensure that the strategies do not exceed their junk quota, the predecessor of each N and M node will be a node with a strategy G which measures $\mu(U^{\emptyset'})$ in a Π_2^0 way. The backup nodes G successively subdivide the interval $[0, 1)$, assigning each of its outcomes an interval $[q, r)$ which corresponds to a guess that $\mu(U^{\emptyset'}) \in [q, r)$. The construction will make sure that if the backing node of a strategy predicts the right interval $[q, r)$ of $\mu(U^{\emptyset'}[s])$ then the junk measure that it captures will increase by no more than $r - q$ after it acts. If we choose $r - q = \epsilon$, then α will capture at most 2ϵ of junk, which is acceptable if we choose the quotas $\epsilon(k)$ such that $\sum_{k\in\omega} 2\epsilon(k) < \frac{1}{2}$. An analysis of the permanent restraints and the timing of the enumerations into A in the construction will verify that $\mu(V^A - U^{\emptyset'}) < \frac{1}{2}$.

4.3. Combining the strategies. The difficulty in combining the non-cupping and LR-completeness strategies stems from the fact that the non-cupping subrequirements are not independent of each other or of the parent N-node. In previous constructions of LR-complete c.e. sets (see [**3, 5**]) when a node holds a restraint under a measure guess which proves wrong, we initialise that strategy and all lower-priority nodes. However here we can only initialise non-cupping parent N-nodes since by initialising an M-node we may make Δ inconsistent. Once a $\Delta^W(p)$ axiom has been enumerated, we must retain the A-restraint until the axiom is invalidated by a W-change or the parent N-node is initialised.

Thus whenever some M-node holds a restraint under a wrong assumption about $\mu(U^{\emptyset'})$ we just try to invalidate the corresponding Δ axiom by enumerating the flip point and waiting for a suitable W-change. The construction will make sure that if this does not happen and N is not reset, the junk measure from the subrequirements of N will be less than the quota of N, even though the junk measure of some M may turn out to be larger than its quota. Overall this satisfies N trivially and with small enough cost. The trick which allows the above quota-junk relation is in enumeration of $U^{\emptyset'}$: it is prefix-free and if some interval σ leaves $U^{\emptyset'}$ then all intervals which were enumerated after σ leave as well, at the same time.

4.4. Priority Tree and Definitions. The priority tree is a finite branching tree which consists of the parent nodes labelled N_e, the subrequirement nodes labelled $M_{e,p}$, and the measure-guessing backup nodes labelled G. We adopt the convention that the root node is at the top and the tree branches downwards; thus we may say that a node α is above a node β if α is an ancestor of β. Let $\langle\cdot,\cdot\rangle$ be a monotone 1–1 computable function from $\mathbb{N}\times\mathbb{N}$ onto $\mathbb{N}$. Requirement N_e has code $\langle e, 0\rangle$ and $M_{e,2p}$ has code $\langle e, p+1\rangle$ (by assumption $\emptyset' \subset 2\omega$ and so only even $\Delta^W(p)$ arguments need to be considered). We say that requirement R_1 has higher priority than R_2 (writing $R_1 < R_2$) if the code of R_1 is smaller than the one of R_2. We define the tree based on this priority ordering. If $|\alpha| = 2\langle e, 0\rangle + 1$ then α is labelled N_e and if $|\alpha| = 2\langle e, p+1\rangle + 1$ then it is labelled $M_{e,2p}$. If $|\alpha| = 2e$ then α is labelled G.

The N_e-nodes τ have outcomes $\infty \prec f$ and are associated with a functional Δ_τ that is built by the $M_{e,p}$-nodes below τ and is occasionally cleared and started afresh when τ is reset. The $M_{e,p}$ nodes have outcomes $\infty \prec f$ and are associated with a *flip-point* d_α which may change in the course of the construction. A measure-guessing G-node γ has outcomes $q_0 \prec q_1 \prec q_2 \prec q_3$ which correspond to guesses about an interval in which $\mu(U^{\emptyset'})$ may lie. Inductively we start with the root node λ, divide $[0, 2^{-1})$ (since $\mu(U^{\emptyset'}) \leq \frac{1}{2}$) into four equal intervals and assign them in increasing order to outcomes $q_0 \prec q_1 \prec q_2 \prec q_3$ respectively, which we think of as edges from λ. If $|\gamma| = 2e$ and is below interval-outcome I of $\gamma \restriction 2e-2$, divide I into four equal intervals and assign them in increasing order to outcomes $q_0 \prec q_1 \prec q_2 \prec q_3$ respectively, which we think of as edges from γ.

For an M or N-node α, with $\alpha = \gamma^\frown x$ for a G-node γ, let $I_\alpha = [q, r)$ be the interval assigned to outcome x of γ. We write $q(\alpha)$ for the lower endpoint q of I_α, and $\epsilon(\alpha)$ for $r - q$, the width of I_α. We refer to $\epsilon(\alpha)$ as α's *resolution* and $q(\alpha)$ as its *measure guess.* Since all nodes of the same label have the same length, we may write $\epsilon(N_e)$ or $\epsilon(M_{e,p})$ to denote $\epsilon(\alpha)$ for any node α labelled N_e or $M_{e,p}$, respectively. For each N or M requirement R we have

$$\sum_{R'>R} 2\epsilon(R') < \epsilon(R) \quad \text{and} \quad \sum_{e\in\omega} 2\epsilon(N_e) < \frac{1}{2} \tag{3}$$

where R' is an N or M requirement. The ordering $\prec$ on the outcomes is extended to the nodes of the tree lexicographically: $\alpha \prec \beta$ if for the longest common initial segment γ of those nodes, $\gamma^\frown x \subseteq \alpha$ and $\gamma^\frown y \subseteq \beta$ for $x \prec y$. We say that α has *higher priority than* β if either $\alpha \subset \beta$ or $\alpha \prec \beta$. We write r_α for the restraint imposed on A by node α, and α^- for the predecessor of α. Also let $R_\alpha = \max\{r_\beta : \beta \prec \alpha \text{ or } \beta \subset \alpha\}$. All parameters have a current value each time they are mentioned in the construction and their value at the beginning of stage s is indicated by the suffix $[s]$. For an $M_{e,p}$-node α, we write $\tau(\alpha)$ for the unique N_e-node $\tau \subset \alpha$. We refer to τ as α's *parent*, or say that α is *working for* τ. An $M_{e,p}$-node α with parent τ is *enabled* if $\tau^\frown\infty \subset \alpha$ and for every $M_{e,p'}$-node α' with $\tau \subset \alpha' \subset \alpha$, we have $\alpha'^\frown f \subset \alpha$. Otherwise, α is *disabled* (which means that it regards Γ^{AW_e} as partial and no further action is needed for N_e).

4.5. Construction. Set $A[0] = \emptyset, \Delta_\tau = \emptyset$ for all N-nodes τ, and $d_\alpha \uparrow, r_\alpha = 0$ for all M-nodes α. When a parameter is assigned a value, it retains that value until explicitly given a new value. To *reset* an N-node τ means to empty Δ_τ, set $r_\beta = 0$ and $d_\beta \uparrow$ for any M-nodes β working for τ, and remove any links to or from τ or any M-node β working for τ. To *reset* an M-node α means to remove any links to it and if $r_\alpha \neq 0$ and $d_\alpha \downarrow$, enumerate d_α into D, setting $d_\alpha \uparrow$. To *reset* a G-node means to remove any links to it. The construction will explicitly declare certain nodes α to be *accessible* at each stage, which does not merely mean that $\alpha \subset \delta_s$. If α is an N-node, it will also declare certain stages to be α-*expansionary*. We give the enumeration of V^A during the stages s of the construction in advance:

(4) **Enumeration of V^A.** For each $\langle\sigma,\rho\rangle \in U[s]$ with $\rho \subset \emptyset'[s]$ but $\sigma \notin V^A[s]$, if $\sigma \in V^A[t]$ with use u for some $t < s$ take the largest such t and if $\langle\sigma,\rho'\rangle \in U[t]$, $\rho' \subset \emptyset'[s]$, then enumerate σ into $V^A[s+1]$ with use u. Otherwise, put σ into $V^A[s+1]$ with fresh use.

The construction will occasionally call the following routine, which is needed in order to access certain outcomes x of nodes α.

(5) **Routine $L(\alpha, x, s)$.** Reset all N-nodes which are on the left of $\alpha^\frown x$. Then consider the longest node $\tau \subset \alpha$ which has label N_e for some $e \in \mathbb{N}$ and there is some $M_{e,p}$-node $\beta \supset \tau$ with $\beta \succ \alpha^\frown x$, $r_\beta[s] \neq 0$. If τ exists let β be the shortest node as above, enumerate d_β into D (if $d_\beta \downarrow$), set $d_\beta \uparrow$, create a link (τ,α) associated with outcome x and go to step 4. Otherwise let $\delta_{s,t+1} = \alpha^\frown x$ and go to step 3.

At stage s, we perform the following steps in order.

Step 1. *(Reset some nodes)* Look for the highest priority node α such that some $\beta \supseteq \alpha$ has been accessed since α was last reset and $\mu(U^{\emptyset'}[s]) < q(\alpha)$. If there is such, reset α and all nodes of lower priority than α.

Step 2. *(Drop some restraints)* For each M-node α with $r_\alpha \neq 0$ and $W \upharpoonright r_\alpha[s] \neq W \upharpoonright r_\alpha[t]$, where t is the stage for which the restraint r_α was last set, set $r_\alpha = 0$ and reset $\alpha^\frown f$ and all nodes of lower priority than $\alpha^\frown f$.

Step 3. *(Define δ_s in substages)* Let $\delta_{s,0} = \lambda$. Let t be the largest number such that $\delta_{s,t} \downarrow$. If $|\delta_{s,t}| \geq s$ then go to step 4. Otherwise let $\alpha = \delta_{s,t}$ and check if

(6) there is an M-node $\beta \preceq \alpha$ with $\tau(\beta)^\frown\infty \subset \alpha, r_\beta \neq 0$ and $d_\beta \uparrow$.

If so, go to step 4; otherwise declare α accessible and go to the relevant clause below.

- **α is a G-node.** Let $[a_0, a_1), \ldots [a_3, a_4)$ be the intervals corresponding to the outcomes of α and $\epsilon = a_1 - a_0$ be the resolution of α. Let $g_\alpha(s)$ be the largest $t < s$ such that $\alpha \subset \delta_t$, or 0 if such t does not exist. Let

$$\nu = \nu(\alpha, s) = \min\{\mu(U^{\emptyset'}[k]) : g_\alpha(s) < k \leq s \ \& \ \mu(U^{\emptyset'}[k]) \in [a_0, a_4)\} \tag{7}$$

(Lemma 4.2 verifies that ν always exists) and let i be such that $\nu \in [a_i, a_{i+1})$, and run routine $L(\alpha, q_i, s)$.

- **α is an $M_{e,p}$-node.** If it is a disabled $M_{e,p}$-node, let $\delta_{s,t+1} = \alpha^\frown\infty$ and go to step 3. Otherwise do as follows. Let $d = d_\alpha, \tau = \tau(\alpha), W = W_e, \Gamma = \Gamma_e$, $u =$ use $\Gamma^{AW}(d)[s]$ (if defined) and

$$h_\alpha(s) = \max\{t \leq s : \nu(\alpha^-, s) = \mu(U^{\emptyset'}[t])\} \tag{8}$$

where α^- is the predecessor of α. *$h_\alpha(s)$ is the stage for which the measure-guessing G-node of α gave its outcome.* If $d \uparrow$ choose a fresh value for d.

M1. If $\Delta_\tau^W(p)[s] \downarrow$ let $\delta_{s,t+1} = \alpha^\frown f$ and go to step 3; if $\Delta_{\tau'}^W(p)[s] \downarrow$ for some N_e-node $\tau' \prec \alpha$ then define $\Delta_\tau^W(p) = \Delta_{\tau'}^W(p)$ with the same use, let $\delta_{s,t+1} = \alpha^\frown f$ and go to step 3.

M2. Otherwise if $\Gamma^{AW}(d)[s] \uparrow$ or if $A \upharpoonright u[s] \neq A \upharpoonright u[t]$ or $W \upharpoonright u[s] \neq W \upharpoonright u[t]$ for the last stage t when α was accessible, or if α has never been accessible before, then run routine $L(\alpha, \infty, s)$.

M3. Otherwise, if

$$\mu\Big(V^{A\upharpoonright u}[s] - V^{A\upharpoonright R_\alpha}[s] - U^{\emptyset'}[h_\alpha(s)]\Big) < \epsilon(\alpha) \tag{9}$$

we define $\Delta_\tau^W(p) = \Gamma^{AW}(p)[s]$ with use u, impose restraint $r_\alpha[s+1] = u$, and go to step 4.

M4. In any other case go to step 4.

- **α is an N_e-node.** Let $l(\alpha, s) = \min\{n : \Gamma_e^{AW_e}(n)[s] \neq K(n)[s]\} \cup \{d : d$ was enumerated into D in step 1 or 2$\}$, and say that stage s is *α-expansionary* if $l(\alpha, s) > l(\beta, t)$ for all N_e-nodes $\beta \preceq \alpha$ and all $t < s$ such that β was accessible at t. If s is not α-expansionary, then let $\delta_{s,t+1} = \alpha^\frown f$ and go to step 3. Otherwise, if there is a link (α, β) associated with outcome x of β which was created at stage $t < s$, remove it and run routine $L(\beta, x, s)$. Otherwise run routine $L(\alpha, \infty, s)$.

Step 4. Set $\delta_s = \alpha$ for the longest α which was declared accessible in step 3. Reset all nodes $\succ \delta_s$ and enumerate into A the least number which is not in A and is greater than all $r_\beta[s+1]$ for all M-nodes β.

4.6. Verification. In the following, whenever we say 'M-node' we mean an enabled M-node, as disabled M-nodes have no effect on the construction. A basic fact which stems from the the hat-trick in the enumeration of $U^{\emptyset'}$ and will be used repeatedly in the verification is the following: if $s_0 < t \leq s_1$ are stages and $\mu(U^{\emptyset'})$ takes its minimum value in $(s_0, s_1]$ at t, then $U^{\emptyset'}[t] \subseteq U^{\emptyset'}[s]$ for all $s \in (s_0, s_1]$.

LEMMA 4.1. Links can never be nested or crossing. That is, if (τ, α) and (τ', α') are two distinct links both present at stage s, with $\tau \subset \alpha \subset \beta$ and $\tau' \subset \alpha' \subset \beta$ for some node β, then $\alpha \subset \tau'$ or $\alpha' \subset \tau$. Furthermore, at the end of any stage s, there is at most one link (τ, α) with $\tau \subset \alpha \subseteq \delta_s$, and such a link was created at stage s.

Proof. By induction on the stages. Note that initially there are no links and at any stage at most one link is created. Suppose that the claim holds at stage s and a link (τ, α) is created at stage $s+1$. Then α is accessible at stage $s+1$ or a link was travelled to α, and any links (τ', α') with $\tau' \subset \alpha' \subsetneq \alpha$ present at the start of stage $s+1$ have been travelled and removed. If there was a link (τ'', α'') at the start of stage $s+1$ for some $\tau'' \subset \alpha \subset \alpha''$, then that link would have been travelled and α would not be accessible. Thus the new link cannot be crossing or nested within an existing link. Finally any links (τ, α) with $\tau \subset \alpha \subset \delta_{s+1}$ which are present at the start of stage $s+1$, would be travelled and removed during the definition of δ_{s+1} in step 3. Since at most one link is created under routine (5), the last claim of the lemma holds. ☐

For a G-node γ, let $I_\gamma = [a_0, a_4)$ be the interval being subdivided by γ. The following lemma verifies that a G-node will always have a valid outcome to play when it is accessible.

LEMMA 4.2. Suppose a G-node γ is accessible at stage s_0 and let $s_1 = g_\gamma(s_0)$ be the greatest stage $< s_0$ such that $\gamma \subset \delta_{s_1}$ (or 0 if such stage does not exist). Then there is some t with $s_1 < t \leq s_0$ and $\mu(U^{\emptyset'}[t]) \in I_\gamma$. Thus, when γ is accessible in step 3, ν (as in (7)) will exist.

Proof. Let γ, s_0 and s_1 be as in the lemma. The proof is by simultaneous induction on the length of γ and the stage s_0. For the root node the claim is trivial, so let $|\gamma| > 1$ and suppose that the claim is true for all G-nodes shorter than γ and at all stages $\leq s_0$. Let $\gamma' = \gamma \restriction |\gamma| - 2$ be the last G-node above γ and note that if γ has never been accessed before, a suitable t must exist or else γ' would not have chosen the outcome leading to γ. Suppose then that γ has been accessed before. If γ' is also accessible at s_0, since $\gamma' \subset \gamma$ we have $g_{\gamma'}(s_0) \geq s_1$ and by hypothesis there is a suitable t with $g_{\gamma'}(s_0) < t \leq s_0$ and $\mu(U^{\emptyset'}[t]) \in I_\gamma$.

If γ' is not accessible at s_0, then there must be a link (τ, β) at s_0, with $\tau \subset \gamma' \subseteq \beta \subset \gamma$. Also by induction hypothesis there must be a stage $t_0 < s_0$ such that γ' is accessible at t_0 and $\mu(U^{\emptyset'}[t]) \in I_\gamma$ for some t with $g_{\gamma'}(t_0) < t \leq t_0$. We can assume that t_0 is the greatest stage $< s_0$ with the above property. If t_2 is the stage at which the link (τ, β) was created we have $t_2 \geq t_0$. Now $\delta_s \not\supseteq \gamma$ for $t_0 \leq s \leq t_2$, as otherwise t_0 would not be the greatest with the above property. Also $\delta_s \not\supseteq \gamma$ for $t_2 < s < s_0$ as otherwise the link would be travelled and removed before s_0, because by Lemma 4.1 links cannot be nested. So $s_1 < t_0$ and $s_1 \leq g_{\gamma'}(t_0)$ since $\gamma' \subset \gamma$, which means that $s_1 < t \leq s_0$. □

By the construction, if an $M_{e,p}$-node α has $r_\alpha[s] \neq 0$ and $d_\alpha \downarrow$, then d_α has not been enumerated into D via resetting or routine (5). Conversely, $r_\alpha[s] \neq 0$ and $d_\alpha \uparrow$ indicates that the construction has attempted to invalidate α's $\Delta^W(p)$ computation. The definition of τ-expansionary stage and the check for (6) in step 3 ensures that no M_e-node of lower priority than α will be accessible again until the $\Delta^W(p)$ computation is invalidated.

A restraint r_α is called *permanent at stage* s if $r_\alpha[s] = r_\alpha[t] \neq 0$ for all $t \geq s$; it is called *permanent* if it is permanent at some stage. Let P be the set of nodes with permanent restraints.

For an M-node α, let $J_\alpha[s] = \{\sigma \in V^A[s+1] - U^{\emptyset'}[s] : R_\alpha[s+1] \leq \operatorname{use} \sigma < r_\alpha[s+1]\}$, which is the junk intervals that are restrained at stage s by α but not by any higher-priority node at the *end* of stage s. For an N_e-node τ, let $Q_\tau[s] = \bigcup J_\alpha[s]$, where the union is taken over all M_e-nodes α which are either $\supset \tau$ or $\prec \tau$. The following lemma shows that if the junk captured by an M-node becomes greater than the node's quota 2ϵ then the node is reset; and although an M-node may sometimes capture more than its quota of junk (if the junk is never released via step 2), the total junk captured by nodes belonging to an N-node remains within the N-node's quota.

LEMMA 4.3. Let β be an M-node and s a stage such that $r_\beta[s+1] \neq 0$ and $d_\beta[s+1]\downarrow$ (so β has not been reset since r_β was set $\neq 0$). Then $\mu(J_\beta[s]) < 2\epsilon(\beta)$. Let τ be an N-node. Then $\mu(Q_\tau[s]) < 2\epsilon(\tau)$ for all s.

Proof. Suppose β and s are as in the first claim. Let t be the stage when $r_\beta[s+1]$ was set. At t, $V^{A\restriction r}[t] = V^{A\restriction r}[s+1]$ for $r = r_\beta[s+1]$ as new intervals in V^A have use chosen fresh. So,

$$\begin{aligned} \mu(J_\beta[s]) &= \mu\big(V^{A\restriction r_\beta}[s+1] - V^{A\restriction R_\beta}[s+1] - U^{\emptyset'}[s]\big) \\ &\leq \mu\big(V^{A\restriction r_\beta}[t] - V^{A\restriction R_\beta}[t] - U^{\emptyset'}[h_\beta(t)]\big) \\ &\quad + \mu\big(U^{\emptyset'}[h_\beta(t)] - U^{\emptyset'}[s]\big) \end{aligned} \tag{10}$$

where the first term of (10) is the junk that β captured when it imposed its restraint $r_\beta[s+1]$, and the second is the measure which appears to be in $U^{\emptyset'}$ at $h_\beta(t)$ but later is removed from $U^{\emptyset'}$. By (9) the first term is less than $\epsilon(\beta)$. Suppose that $\mu(U^{\emptyset'}[h_\beta(t)] - U^{\emptyset'}[s]) \geq \epsilon(\beta)$. We have $U^{\emptyset'}[h_\beta(t)] - U^{\emptyset'}[t] = \emptyset$, as otherwise (by the canonical enumeration of $U^{\emptyset'}$) there would be a stage $t', h_\beta(t) < t' \leq t$ with $\mu(U^{\emptyset'}[t']) < \mu(U^{\emptyset'}[h_\beta(t)])$, which contradicts (8). So we must have $\mu\big(U^{\emptyset'}[t] - U^{\emptyset'}[s]\big) \geq \epsilon(\beta)$. But then, again by the canonical enumeration of $U^{\emptyset'}$ there would be a stage $t', t < t' \leq s$ such that $\mu(U^{\emptyset'}[t']) \leq \mu(U^{\emptyset'}[h_\beta(t)]) - \epsilon(\beta)$, and β would be reset at t' by step 1 of the construction. So $\mu(U^{\emptyset'}[h_\beta(t)] - U^{\emptyset'}[s]) < \epsilon(\beta)$, and $\mu(J_\beta[s]) < 2\epsilon(\beta)$.

Next, let τ be an N_e-node; we need only consider the case where there is some M_e-node $\beta \supset \tau$ with $J_\beta[s] \neq \emptyset$. Let Z denote the set of M_e-nodes $\beta' \supset \tau$ or $\prec \tau$ with $r_{\beta'}[s+1] \neq 0$, and let β be the longest; by assumption $\beta \supset \tau$. Let t be the stage when $r_\beta[s+1]$ was set $\neq 0$. At t, $d_{\beta'}[t+1]\downarrow$ for all $\beta' \in Z$, as otherwise β would not be accessible at t. Also $\mu(J_\beta[t]) < \epsilon(\beta)$ by (9). So by the first part of the lemma and (3), $\mu(Q_\tau[t]) < 2\epsilon(\tau)$. Also, $d_{\beta'}[t'+1]\downarrow$ for all $t < t' \leq s$ and $\beta' \in Z, \beta' \prec \tau$, as otherwise τ would be reset, contradicting the definition of t. So if $\mu(Q_\tau[t']) \geq 2\epsilon(\tau)$ at some $t < t' \leq s$ it must be because $\sum_{\tau\subset\beta'\in Z} \mu(J_{\beta'}[t']) > \epsilon(\tau)$. But then by the canonical enumeration of $U^{\emptyset'}$ there would be a stage t'' such that $t < t'' \leq t'$ and

$\mu(U^{\emptyset'}[t'']) < \mu(U^{\emptyset'}[h_\beta(t)]) - \epsilon(\tau)$. In such a case τ would be reset at step 1, again contradicting the definition of t. So $\mu(Q_\tau[s]) < 2\epsilon(\tau)$. □

In the following lemma we prove simultaneously that the true path $TP = \liminf_s \delta_s$ is infinite, that every node on it has infinitely many chances to act, and that eventually the measure condition (9) will be satisfied for each M-node on TP.

LEMMA 4.4. If α is the leftmost node of length $|\alpha|$ such that $\alpha \subseteq \delta_s$ for infinitely many s, then

(1) α is reset only finitely often; if it is an M-node then eventually the flip-point d_α is fixed;
(2) α is accessible infinitely often;
(3) there is some extension $\beta \supset \alpha$ with $\beta \subseteq \delta_s$ for infinitely many s.

Thus $TP = \liminf_s \delta_s$ is infinite.

Proof. First of all, if $|\alpha| = 0$ then $\alpha \subseteq \delta_s$ for all s so 1-3 of the lemma implies that TP is infinite. Then it remains to assume that α is the leftmost node of length $|\alpha|$ such that $\alpha \subseteq \delta_s$ infinitely often and (inductively) that the lemma holds for all $\beta \subset \alpha$, and show claims 1-3.

For the first claim note that there are four places in the construction where α may be reset: in step 1, step 2, step 3 (through the routine L) and step 4. Let s_0 be the second stage such that $\alpha \subseteq \delta_{s_0}$, $\delta_s \not\prec \alpha\ \forall s > s_0$, any computations $\Delta^W_{\tau(\beta)}(p)\downarrow$ of nodes $\beta \prec \alpha$ that exist at s_0 are permanent and no nodes above or to the left of α are reset after s_0. After s_0, α will not be reset in step 4. If α was reset after s_0 at step 3 then it would be because routine $L(\beta, x, s)$ was run for some $\beta \subset \alpha$ such that $\beta^\frown x \prec \alpha$. But this would mean that either $\delta_s \prec \alpha$ for some $s > s_0$ or α is not $\subseteq \delta_s$ infinitely often, a contradiction.

If α was reset by step 2, by the choice of s_0 there must be some M-node β such that $\beta^\frown f \subset \alpha$ which had a computation $\Delta^W_{\tau(\beta)}(p)\downarrow$ and this was spoilt after s_0. But then the corresponding Γ computation (which has larger use) would be spoilt and the construction would define δ_s to the left of α at M2, a contradiction. Suppose that α was reset in step 1 after stage s_0. By the choice of s_0 there must be a node $\beta \subset \alpha$ and a stage $s_1 > s_0$ such that $\mu(U^{\emptyset'}[s_1]) < q(\beta)$. But in that case after stage s_1 the construction would define δ_s to the left of α, before it defines it below α, a contradiction.

Finally suppose that α is an M-node and d_α was changed after stage s_0. Since α is not reset after s_0 there must be some $\beta \subset \alpha$ which ran routine $L(\beta, x, s_1)$ for $s_1 > s_0$ and $\beta^\frown x \prec \alpha$. But in that case the construction would define δ_s to the left of α, before it defines it below α, a contradiction.

For claim 2, notice that since by hypothesis $\alpha \subseteq \delta_s$ for infinitely many s, the only way that α may stop being accessible after some stage is that for all sufficiently large stages there is a link (τ, β) with $\tau \subset \alpha \subset \beta$. Suppose, for a contradiction, that this is the case and after stage s_0 α is never accessible again. Let $Y[s]$ be the finite set of Δ-computations that are held by M-nodes below α at $s \geq s_0$. Note that if $\delta_t \supseteq \alpha$ for $t \geq s_0$ then by Lemma 4.1 a link must be created at t as otherwise the next time $\alpha \subseteq \delta_s$, α would not be covered by a link and would be accessible. Thus no new computations can be added to Y after s_0 as if a Δ-definition is made then no link is created at that stage. Also, by the construction there are no Δ-computations held by nodes $\succ \alpha$ at the end of a stage s when $\alpha \subseteq \delta_s$. Finally a link is only travelled if the Δ-computation for which it was created has been invalidated. So any link covering α at $s \geq s_0$ is created because of a computation in Y, which is removed from Y when the link is travelled. Since Y is finite and non-increasing, after finitely many stages Y will be empty and α will be accessible when next $\delta_s \supseteq \alpha$.

For claim 3, since α is accessible infinitely often the only way the claim could fail is if, whenever α is accessible after some finite stage $s_0 > |\alpha|$, step 3 is ended without any $\alpha^\frown x$ being declared accessible. Suppose this is the case. Then whenever α is accessible after s_0, step 3 is ended by routine L, or by M3 or M4 if α is an M-node, or because of (6).

At s_0 there are only finitely many $\Delta(p)$ definitions held by nodes β below α. If (6) holds at $s > s_0$ for some $\alpha^\frown x$, it is because one such β was reset while $\tau(\beta)$ was covered by a link. But the link is removed after being travelled, and the next time $\tau(\beta)^\frown\infty \subset \alpha$ is accessible, β's $\Delta(p)$ definition will have been set to 0 at step 2. Since no β below α is accessible after s_0, this can happen only finitely often for the finitely many $\Delta(p)$ computations below α. So it will not happen after some stage s_1.

If step 3 is ended after s_1 due to a routine $L(\alpha, x, s)$ for some outcome x of α, according to the induction hypothesis for α the routine will eventually define $\delta_{s,t} = \alpha^\frown x$ and so $\delta_s \supseteq \alpha^\frown x$ at some stage s. If step 3 is ended because of $M3$ applied to α, then either the Δ-definition made there is permanent (in which case $\alpha^\frown f \subseteq \delta_s$ at some later stage s) or it is not, in which case routine $L(\alpha, \infty, s)$ will be called and the previous argument applies.

Finally, suppose that whenever an $M_{e,p}$-node α is accessible after some s_1, case M4 applies and step 3 is ended at α. We show that eventually the measure condition (9) is satisfied and M3 will apply, a contradiction. At s_1, there are only finitely many nodes $\supset \alpha$ with restraints, and no nodes below α are accessible after s_1. Let s_2 be the second stage after s_1 such that

- any non-permanent restraints below α have been dropped;
- all nodes β above or left of α have settled; ie β is not reset after s_2 and if $r_\beta[s_2] \neq 0$ then $r_\beta[s_2]$ is permanent;
- $\Gamma^{AW}(d_\alpha)\downarrow$ and the use is correct;
- $V^{A\restriction u}[s_2] - V^{A\restriction R_\alpha}[s_2] - U^{\emptyset'}[s_2] = V^{A\restriction u} - V^{A\restriction R_\alpha} - U^{\emptyset'}$;
- α is accessible at s_2.

Such stage exists by the induction hypothesis and the fact that new intervals in V^A have use chosen fresh. Every interval in $V^{A\restriction u}[s_2] - V^{A\restriction R_\alpha}[s_2] - U^{\emptyset'}[s_2]$ is in $J_\beta[s_2]$ for some $\beta \supset \alpha$, as otherwise it would be removed in step 4 contradicting the choice of s_2. Letting $E = \{\beta : \beta \supset \alpha \text{ and } r_\beta[s_2] \neq 0\}$, we have

$$\mu(V^{A\restriction u}[s_2] - V^{A\restriction R_\alpha}[s_2] - U^{\emptyset'}[s_2]) = \sum_{\beta \in E} \mu(J_\beta[s_2]).$$

Write $E = F \cup G$ where

$$F = \{\beta \in E : \tau(\beta) \subset \alpha\}; \quad G = \{\beta \in E : \alpha \subset \tau(\beta)\}.$$

Note that at s_2, every node β in F has $d_\beta[s_2 + 1]\downarrow$; as otherwise β has been reset at some t, $s_0 \leq t \leq s_2$, and by choice of s_2 r_β is never set to 0 and β's Δ-definition is never invalidated. But then $\tau(\beta)$ has only finitely many expansionary stages, contradicting that $\tau(\beta)^\frown\infty \subset \alpha$ is accessible infinitely often by induction hypothesis.

Observe that the first clause of Lemma 4.3 holds for any $\beta \in F$ and $s = s_2$, and the second for $\tau = \tau(\beta)$ for any $\beta \in G$ and $s = s_2$. So by (3),

$$\begin{aligned}\mu(V^{A\restriction u}[s_2] - V^{A\restriction R_\alpha}[s_2] - U^{\emptyset'}[s_2]) &= \sum_{\beta\in F}\mu(J_\beta[s_2]) + \sum_{\tau\in\{\tau(\beta):\beta\in G\}}\mu(Q_\tau[s_2]) \\ &< \sum_{\beta\in F} 2\epsilon(\beta) + \sum_{\tau\in\{\tau(\beta):\beta\in G\}} 2\epsilon(\tau) \\ &< \epsilon(\alpha).\end{aligned}$$

Thus (9) will hold at s_2, α will make a $\Delta(p)$ definition which will be permanent, and $\alpha^\frown f$ will be accessible at some stage after s_2. □

LEMMA 4.5. All non-cupping requirements N_e are satisfied.

Proof. Let τ be the N_e-node on TP. It is clear from the construction that $\tau^\frown\infty \subset TP$ iff there are infinitely many τ-expansionary stages. By Lemma 4.4 and the construction, if α is an M_e-node with $\tau^\frown\infty \subset \alpha \subset TP$ then

- $\alpha^\frown\infty \subset TP \Rightarrow \Gamma^{AW}(d_\alpha)\uparrow$, and
- $\alpha^\frown f \subset TP \Rightarrow \Delta_\tau^{W_e}(p)\downarrow$.

To show that for each e the requirement N_e is satisfied assume that $\Gamma_e^{AW_e} = K$ and let τ be the N_e-node on TP. Since $\Gamma_e^{AW_e} = K$ there are infinitely many τ-expansionary stages. First note that by the construction, Δ_τ is consistent, i.e. at each stage s if $\langle\sigma, n, x\rangle, \langle\rho, n, y\rangle \in \Delta_\tau[s]$ and $\sigma \subseteq \rho$ then $x = y$. Also by Lemma 4.4 and the fact that all strategies appear along the true path, the function Δ_τ^W is total and the restraints imposed by each M_e-node below τ when it makes a definition ensure that $\Delta_\tau^W(p) = \Gamma_e^{AW_e}(p) = \emptyset'(p)$ for each $p \in \mathbb{N}$. Thus $W \geq_T \emptyset'$ and N_e is satisfied. □

LEMMA 4.6. $\emptyset' \leq_{LR} A$.

Proof. We must verify that $U^{\emptyset'} \subseteq V^A$ and $\mu(V^A) < 1$. Once an interval σ appears in $U^{\emptyset'}$ with correct $\emptyset'$-use, according to (4) in any later stage it will be in V^A with the same A-use. Thus eventually it will permanently belong to V^A and $U^{\emptyset'} \subseteq V^A$.

To verify $\mu(V^A) < 1$, since $\mu(U^{\emptyset'}) < \frac{1}{2}$ it suffices to show that $\mu(V^{A\restriction n}[s] - U^{\emptyset'}[s]) < \frac{1}{2}$ for all $n \in \mathbb{N}$ and all $s \geq$ some s_0. Fix n and let s_0 be a stage such that $A \restriction n[s_0] = A \restriction n$

and $V^{A\upharpoonright n}[s_0] - U^{\emptyset'}[s_0] = V^{A\upharpoonright n} - U^{\emptyset'}$. Then for all $s \geq s_0$ we have

$$V^{A\upharpoonright n}[s] - U^{\emptyset'}[s] \subseteq \bigcup_{\tau \subset \delta} Q_\tau[s]$$

where τ runs over the N-nodes and δ is the rightmost path of the tree. Hence, by Lemma 4.3 and the second clause of (3) we have, for $s \geq s_0$,

$$\mu(V^{A\upharpoonright n}[s] - U^{\emptyset'}[s]) \leq \sum_e 2\epsilon(N_e) < \frac{1}{2}.$$

□

This concludes the proof of Theorem 1.1.

References

[1] George Barmpalias, Andrew E. M. Lewis and Mariya Soskova, Randomness, lowness and degrees, *J. Symbolic Logic* **73** (2008), no. 2, 559–577

[2] George Barmpalias, Andrew E. M. Lewis and Frank Stephan, Π^0_1 classes, LR degrees and Turing degrees, to appear in *Annals of Pure and Applied Logic*

[3] George Barmpalias and Antonio Montalbán, A cappable almost everywhere dominating computably enumerable degree, *Electronic Notes in Theoretical Computer Science*, Vol. 167 (2007)

[4] Stephen Binns, Bjørn Kjos-Hanssen, Manuel Lerman and Reed Solomon, On a conjecture of Dobrinen and Simpson concerning almost everywhere domination, *J. Symbolic Logic* **71** (2006), no. 1, 119–136

[5] Peter Cholak, Noam Greenberg and Joseph S. Miller, Uniform almost everywhere domination, *Journal of Symbolic Logic*, Vol. 71 (2006)

[6] S. Barry Cooper, Steffen Lempp and Philip Watson, Weak density and cupping in the d-r.e. degrees, *Israel Journal of Mathematics*, **67**, 137–152, 1989

[7] K. de Leeuw, E. F. Moore, C. E. Shannon, and N. Shapiro, Computability by probabilistic machines, *Automata studies, pp. 183212. Annals of mathematics studies*, no. **34**. Princeton University Press, Princeton, N. J., 1956

[8] Natasha L. Dobrinen and Stephen G. Simpson, Almost everywhere domination, *J. Symbolic Logic* **69** (2004), no. 3, 914–922

[9] R. G. Downey and D. Hirschfeldt, *Algorithmic Randomness and Complexity*, Springer, *in preparation*

[10] R. G. Downey, T. A. Slaman, Completely mitotic r.e. degrees, *Ann. Pure Appl. Logic* **41** (1989), no. 2, 119–152

[11] R. G. Downey, M. Stob, Splitting theorems in recursion theory, *Ann. Pure Appl. Logic* **65** (1993), no. 1, 1–106

[12] Bjørn Kjos-Hanssen, Low for random reals and positive-measure domination, *Proceedings of the American Mathematical Society* **135** (2007), 3703–3709

[13] Bjørn Kjos-Hanssen, Joseph S. Miller and David Reed Solomon, Lowness notions, measure and domination, *unpublished draft*

[14] A. H. Lachlan, The priority method I, *Z. Math. Logik Grundlagen Math.* **13** (1967) 1–10

[15] R. E. Ladner, Mitotic Recursively Enumerable Sets, *Journal of Symbolic Logic*, vol. **38** (1973), no. 2, 199–211

[16] R. E. Ladner, A Completely Mitotic Nonrecursive R.E. Degree, *Transactions of the AMS*, vol. **184** (1973), 479–507

[17] Angsheng Li, Theodore A. Slaman and Yue Yang, A nonlow$_2$ c.e. degree which bounds no diamond bases, *unpublished draft*

[18] D. Martin, Measure, Category, and Degrees of Unsolvability. Unpublished manuscript, dating from the late 60's

[19] David P. Miller, High recursively enumerable degrees and the anti-cupping property, *Logic Year 1979-80* (M. Lerman et al., editors), Lecture Notes in Mathematics, vol. 859, Springer-Verlag, Berlin, 1981

[20] André Nies, Lowness properties and randomness. *Advances in Mathematics*, **197** (2005) 274–305

[21] André Nies, *Computability and Randomness*, Oxford University Press, *in preparation*

[22] Gerald Sacks, *Degrees of Unsolvability*, Princeton University Press, 1963

[23] Stephen G. Simpson, Almost everywhere domination and superhighness, *Mathematical Logic Quarterly*, **53** (2007), 462–482

[24] Robert I. Soare, *Recursively Enumerable Sets and Degrees*, Springer-Verlag; Berlin, London; 1987

[25] Liang Yu and Yue Yang, On the definable ideal generated by nonbounding c.e. degrees, *Journal of Symbolic Logic*, Vol. 20 (2005)