MUCHNIK DEGREES AND CARDINAL CHARACTERISTICS

BENOIT MONIN AND ANDRÉ NIES

Contents

Abstract. We denote by IOE(F) is the set of functions which equals infinitely often every computable function bounded by F. We denote by AED(F) the dually notion : the set of functions which differs almost everywhere from every computable function bounded by F. We study in this paper the Medvedev and Munchnik degrees of these two classes, with respect to Coarse computability.

We also show, together with Joe miller that for any F, there exists a function $G > F$ such that IOE(F) $<_W$ IOE(G). The fact that for any F, there exists a function $G < F$ such that AED(F) $<_W$ AED(G) follows from the work of Khan and Miller on DNC functions and bushy tree forcing [10].

We finally study the connection between the Medvedev and Muchnik degrees of the classes IOE(F) and AED(F), with set theory and cardinal characteristics.

1. Introduction

It is of fundamental interest in computability theory to determine the inherent computational complexity of an object, such as an infinite bit sequence, or a function f on the natural numbers. To determine this complexity, one can introduce classes of objects that all have a similar complexity. We focus on highness properties. They specify a sense in which the object in question is computationally powerful.

The Γ parameter of an infinite bit sequence A, introduced by Andrews et al. [1], is a value between 0 and 1 that in a sense measures how well all oracle sets in its Turing degree can be approximated by computable sequences. For each $p \in (0,1]$, "$\Gamma(A) < p$" is a highness property of A. The values $0, 1/2$ and 1 occur [7, 1]. Further, $\Gamma(A) > 1/2 \Leftrightarrow \Gamma(A) = 1 \Leftrightarrow A$ is computable [7]. Andrews et al. asked whether the Γ value can be strictly between 0 and $1/2$. The precise definition of $\Gamma(A)$ will be given shortly in Subsection 1.1.

Monin [13] answered their question in the negative, and also characterised the degrees with Γ-value $< 1/2$ using such a property of functions. He built on some initial work with the second author [14] involving functions that agree with each computable function infinitely often.

Our goal is to put these concepts and methods into a wider context. We rely on an analogy between highness properties of oracles and cardinal characteristics in set theory. This analogy was first noted and studied by Rupprecht [17, 16]. Thereafter, Brooke-Taylor et al. [2] investigated it via a notation system that makes it possible to automatically transfer many highness properties of oracles into cardinal characteristics, and vice versa.

The rest of the introduction will provide more detail on the notions mentioned above, and describe the main results.

1.1. **Defining the Γ parameter of a sequence.** We recall how to define the Γ parameter of an infinite bit sequence (often simply termed "sequence"); this definition will only depend on its Turing degree. For a sequence Z, also viewed as a subset of ω, the lower density is defined by

$$\underline{\rho}(Z) = \liminf_n \frac{|Z \cap [0,n)|}{n}.$$

For sequences X, Y one denotes by $X \leftrightarrow Y$ the sequence Z such that $Z(n) = 1$ iff $X(n) = Y(n)$. To measure how closely a sequence A can be approximated by a computable sequence X, Hirschfeldt et al. [7] defined

$$\gamma(A) = \sup_{X \text{ computable}} \underline{\rho}(A \leftrightarrow X).$$

Clearly this depends on the particular sequence A, rather than its Turing complexity. Andrews et al. [1] took the infimum of the γ values over all Y in the Turing degree of A:

$$\Gamma(A) = \inf\{\gamma(Y) \colon \ Y \equiv_T A\}.$$

See [15, Section 7] for more background on the Γ parameter. In particular, $1 - \Gamma(A)$ can be seen as a Hausdorff pseudodistance between $\{Y \colon \ Y \leq_{\mathrm{T}} A\}$ and the computable sets with respect to the Besicovitch distance $\overline{\rho}(U \triangle V)$ between bit sequences U, V (where $\overline{\rho}$ is the upper density). Thus, a large value $\Gamma(A)$ literally means that A is "close to computable".

1.2. **Duality.** For functions f, g on ω, we say that g dominates f if $f(n) \leq g(n)$ for almost all $n \in \omega$. The unbounding number $\mathfrak{b}$ is the least size of a collection of functions f such that no single function dominates the entire collection. The domination number $\mathfrak{d}$ is the least size of a collection of functions so that each function is dominated by a function in the collection. In computability theory, the analog of $\mathfrak{b}$ is the usual highness $A' \geq_T \emptyset''$ of an oracle A, and its dual, the analog of $\mathfrak{d}$, is being of hyperimmune degree.

In the area of cardinal characteristics there is a persistent duality between pairs of cardinals. For instance, the unbounding number $\mathfrak{b}$ is the dual of the domination number $\mathfrak{d}$.

Brendle and Nies in the 2015 Logic Blog [3, Section 7], modifying the work in [7, 1], defined for $p \in [0, 1/2]$ highness properties $\mathcal{D}(p)$ such that $\Gamma(A) < p \Rightarrow A \in \mathcal{D}(p) \Rightarrow \Gamma(A) \le p$. They defined $\mathcal{D}(p)$ to be the set of oracles A that compute a bit sequence Y such that $\underline{\rho}(Y \leftrightarrow X) \le p$ for each computable sequence X. They then obtained via the framework in Brooke-Taylor et al. [2] cardinal characteristics $\mathfrak{d}(p)$. Dualising this both in computability and in set theory, they introduced the highness property $\mathcal{B}(p)$ for $0 \le p < 1/2$, the class of oracles A that compute a bit sequence Y such that for each computable sequence X, we have $\underline{\rho}(X \leftrightarrow Y) > p$, and the analogous cardinal characteristics $\mathfrak{b}(p)$.

Extending Monin's methods [14], we will show that all the highness properties $\mathcal{D}(p)$ coincide for $0 < p < 1/2$, and similarly for the $\mathcal{B}(p)$. Since $\Gamma(A) < p \Rightarrow A \in \mathcal{D}(p)$, we re-obtain the result that $\Gamma(A) < 1/2$ implies $\Gamma(A) = 0$. Via analogous methods within set theory, we show that ZFC proves the coincidence of all the $\mathfrak{d}(p)$, and of all the $\mathfrak{b}(p)$, for $0 < p < 1/2$. We will describe these coincidences in more detail in Subsection 1.5. To do so, we first need to discuss some more concepts.

1.3. **Medvedev and Muchnik reducibility.** A non-empty subset $\mathcal{B}$ of Baire space is often called a *mass problem.* A function $f \in \mathcal{B}$ is considered a *solution* to the problem. The easiest problem the set of all functions, and the impossible problem is the empty set.

In this paper we will phrase our highness properties in the language of mass problems (rather than upward closed sets of Turing degrees), and compare them via Medvedev and Muchnik reducibility. The advantage of this approach is that we can keep track of potential uniformities when we give reductions showing that one property is at least as strong as another.

Let $\mathcal{B}, \mathcal{C}$ be mass problems. The reducibilities below say that any solution to $\mathcal{B}$ yields a solution to $\mathcal{C}$. The first, also called strong reducibility, is the uniform version: one writes $\mathcal{B} \le_S \mathcal{C}$ (and says that $\mathcal{B}$ is Medvedev reducible to $\mathcal{C}$) if there is a Turing functional Γ with domain containing $\mathcal{C}$ such that $\forall g \in \mathcal{C} [\Gamma^g \in \mathcal{B}]$. Note that $\mathcal{B} \supseteq \mathcal{C}$ implies $\mathcal{B} \le_S \mathcal{C}$ via the identity functional. One writes $\mathcal{B} \le_W \mathcal{C}$ (and says that $\mathcal{B}$ is Muchnik reducible to $\mathcal{C}$) if $\forall g \in \mathcal{C}\ \exists f \in \mathcal{B} [f \le_T g]$. Muchnik degrees correspond to end segments in the Turing degrees via sending $\mathcal{C}$ to the collection of oracles computing a member of $\mathcal{C}$. In this way, viewing highness properties as a mass problems and comparing them via Muchnik reducibility $\le_W$ is equivalent to viewing them as end segments in the Turing degrees under reverse inclusion.

1.4. **A dual pair of mass problems for functions.** The idea behind the density-related mass problems $\mathcal{D}(p)$ and $\mathcal{B}(p)$ is to determine the computational complexity of an object by comparing it to computable objects of the same type. This idea works in more than one setting. We use it to introduce two mass problems that will be important in this paper. We say that a function f is IOE if $\exists^\infty n\, [f(n) = r(n)]$ for each computable function r. We say that f is AED if $\forall^\infty n\, [f(n) \ne r(n)]$ for each computable function r. (IOE stands for "infinitely often equal", while AED for "almost everywhere different".)

The study of the class AED can be traced back to Jockusch [8, Thm. 7], though he actually considered a stronger property of a function f he denoted by SDNR: $\forall^\infty n\,[f(n) \neq r(n)]$ for each *partial* computable function r. (Kjos-Hansen, Merkle, and Stephan [11, Thm. 5.1 (1) $\to$ (2)] showed that each non-high function in AED is SDNR.) The class IOE was only introduced much later. Kurtz [12] showed that the mass problem of weakly 1-generic sets is Muchnik equivalent to the functions not dominated by a computable function (corresponding to the hyperimmune Turing degrees). Using this fact, it is not hard to show that IOE is also Muchnik equivalent to the functions not dominated by a computable function.

An order function h is a non-decreasing, unbounded computable function. In computability theory, one often uses order functions as bounds to parameterise known classes of similar complexity. For instance, DNC(h) is the class of diagonally non-computable functions $f < h$. For another example, an oracle A is h-traceable if each A-partial computable function has a c.e. trace of size bounded by h.

We focus on versions of the classes IOE and AED parameterised by an order function h. By IOE(h) we denote the mass problem of functions f such $\exists^\infty n\,[f(n) = r(n)]$ for each computable function $r < h$. Dually, AED(h) is the mass problem of functions $f < h$ such that $\forall^\infty n\,[f(n) \neq r(n)]$ for each computable function r. Clearly $g \leq h$ implies IOE(g) $\supseteq$ IOE(h) and AED(g) $\subseteq$ AED(h). Obvious questions are then whether for each order function h that grows sufficiently much faster than an order function g, we obtain IOE(g) $<_W$ IOE(h) and AED(g) $>_W$ AED(h).

For the operator AED such a result is known by combining recent work of Khan and Miller [9] on a hierarchy for the mass problems of low DNR(h) functions with a transfer due to Khan and Nies [5] between these mass problems and the mass problems AED($\widetilde{h}$) for $\widetilde{h}$ close to h.

For he operator IOE separations for some very special cases of functions g, h were obtained in [14]. In Section 4, which consists of joint work with Joseph S. Miller that will be included here, we answer the full question in the affirmative.

Kamo and Osuga [**?**] have proved the consistency with ZFC of obtaining different cardinal characteristics $\mathfrak{b}(\neq^*, h)$ depending on the growth of the function h. These are analogous to mass problems AED(h). A similar result is not known at present for their dual characteristics $\mathfrak{d}(\neq^*, h)$.

1.5. **Density.** With Subsection 1.3 in mind, the highness properties introduced by Brendle and Nies in [3, Section 7] will now be considered as mass problems. They consist of $\{0,1\}$-valued functions on ω, i.e., infinite bit sequences. Let p be a real with $0 \leq p < 1$. $\mathcal{D}(p)$ is the set of bit sequences y such that for each computable set x, we have $\underline{\rho}(x \leftrightarrow y) \leq p$. Note that this resembles the definition of IOE. $\mathcal{B}(p)$ is the set of bit sequences y such that for each computable set x, we have $\underline{\rho}(x \leftrightarrow y) > p$. This resembles the definition of AED.

Clearly $0 \leq p < q < 1$ implies $\mathcal{D}(p) \subseteq \mathcal{D}(q)$ and $\mathcal{B}(p) \supseteq \mathcal{B}(q)$. Our first result, Theorem 3.4, shows that on the density side there actually is no proper hierarchy when the parameter is positive, and gives a characterisation by a combinatorial class, relying on agreement of functions rather than on density:

$$\mathcal{D}(p) \equiv_W \mathrm{IOE}(2^{(2^n)}) \text{ and } \mathcal{B}(p) \equiv_S \mathrm{AED}(2^{(2^n)})$$

for arbitrary $p \in (0, 1/2)$. The corresponding result for cardinal characteristics is Theorem 5.5 below. The outer exponential function in the bound simply stems from the fact that we view function values as encoded by binary numbers, which correspond to blocks in the bit sequences: if a bound h has the form $2^{\widehat{h}}$ for an order function $\widehat{h}$, then a function $f < h$ naturally corresponds to a bit sequence which is the concatenation of blocks of length $\widehat{h}(i)$ for $i \in \omega$.

As part of the proof of Theorem 3.4, we show a lemma that the parameterised classes $\mathrm{IOE}(h)$ and $\mathrm{AED}(h)$ don't depend too sensitively on the bound h: if $g(n) = h(2n)$ then $\mathrm{IOE}(g) \equiv_W \mathrm{IOE}(h)$ and $\mathrm{AED}(g) \equiv_S \mathrm{AED}(h)$. Since the first equivalence we obtain is merely Muchnik, in Theorem 3.4 we also only have Muchnik in its first equivalence. Note that by the lemma, in the above, we can replace $\mathrm{IOE}(2^{(2^n)})$ by $\mathrm{IOE}(2^{(2^{n \cdot r})})$ for any $r > 0$.

Acknowledgements. Several of the questions studied here arose in work between Jörg Brendle and the second author that is archived in [3, Section 7]. We thank Brendle for these very helpful discussions.

We thank Joseph Miller for his contribution towards Section 4 in this paper.

Nies is supported in part by the Marsden Fund of the Royal Society of New Zealand, UoA 13-184.

The work was completed while the authors visited the Institute for Mathematical Sciences at NUS during the 2017 programme "Aspects of Computation".

2. Defining mass problems dependent on relations

Towards proving our main theorems, we will need a general formalism to define mass problems based on relations, similar to [2]. We consider "spaces" X, Y, which will be effectively closed subsets of Baire space. Let the variable x range over X, and let y range over Y. Let $R \subseteq X \cdot Y$ be a relation, and let $S = \{\langle y, x\rangle \in Y \cdot X : \neg xRy\}$.

Definition 2.1. We define the dual pair of mass problems

$$\mathcal{B}(R) = \{y \in Y : \forall x \text{ computable } [xRy]\}$$

$$\mathcal{D}(R) = \mathcal{B}(S) = \{x \in X : \forall y \text{ computable } [\neg xRy]\}$$

To re-obtain the mass problems discussed in the introduction, we consider the following two types of relation.

Definition 2.2. *1.* Let $h : \omega \to \omega - \{0, 1\}$. Define for $x \in {}^{\omega}\omega$ and $y \in \prod_n \{0, \ldots, h(n) - 1\} \subseteq {}^{\omega}\omega$,

$$x \neq^*_h y \Leftrightarrow \forall^\infty n\, [x(n) \neq y(n)].$$

2. Recall that $\underline{\rho}(z) = \liminf_n |z \cap n|/n$ for a bit sequence z. Let $0 \le p < 1$. Define, for $x, y \in {}^\omega 2$

$$x \bowtie_p y \Leftrightarrow \underline{\rho}(x \leftrightarrow y) > p,$$

where $x \leftrightarrow y$ is the set of n such that $x(n) = y(n)$.

For the convenience of the reader we summarise the specific mass problems determined by these relations.

Remark 2.3. Let h be a computable function. Let p be a real with $0 \le p \le 1/2$.
$\mathcal{D}(\neq^*_h)$, which we actually denote by IOE(h), is the set of functions $y < h$ such that for each computable function x, we have $\exists^\infty n\, x(n) = y(n)$.

$\mathcal{B}(\neq^*_h)$, which we actually denote by AED(h), is the set of functions $y < h$ such that for each computable function x, we have $\forall^\infty n\, x(n) \neq y(n)$.

$\mathcal{D}(\bowtie_p)$, or $\mathcal{D}(p)$ for short, is the set of bit sequences y such that for each computable set x, we have $\underline{\rho}(x \leftrightarrow y) \le p$.

$\mathcal{B}(\bowtie_p)$, or $\mathcal{B}(p)$ for short, is the set of bit sequences y such that for each computable set x, we have $\underline{\rho}(x \leftrightarrow y) > p$.

3. Main result in the setting of computability theory

As mentioned, our goal is to show that

$$\mathcal{D}(p) \equiv_W \mathrm{IOE}(2^{(2^n)}) \text{ and } \mathcal{B}(p) \equiv_S \mathrm{AED}(2^{(2^n)})$$

for arbitrary $p \in (0, 1/2)$. We begin with some preliminary facts of independent interest. On occasion we denote a function $\lambda n.f(n)$ simply by $f(n)$.

Lemma 3.1. (i) *Let h be nondecreasing and $g(n) = h(2n)$. We have* $\mathrm{IOE}(h) \equiv_W \mathrm{IOE}(g)$ *and* $\mathrm{AED}(h) \equiv_S \mathrm{AED}(g)$*.*
(ii) *For each $a, b > 1$ we have* $\mathrm{IOE}(2^{(a^n)}) \equiv_W \mathrm{IOE}(2^{(b^n)})$ *and* $\mathrm{AED}(2^{(a^n)}) \equiv_S \mathrm{AED}(2^{(b^n)})$*.*

Note that the duality appears to be incomplete: for the statement involving the $\mathcal{D}$-type problems, we only obtain weak equivalence.

Proof. (i) Trivially, $h \le g$ implies

$$\mathrm{IOE}(h) \supseteq \mathrm{IOE}(g) \text{ and } \mathrm{AED}(h) \subseteq \mathrm{AED}(g).$$

So it suffices to show only one inequality in each case.
$\boxed{\mathrm{IOE}(h) \ge_W \mathrm{IOE}(g):}$ Let $y < h$ be a function in IOE(h). Let $\widehat{y}_1 < h(2n)$ and $\widehat{y}_2 < h(2n+1)$ be defined by $\widehat{y}_1(n) = y(2n)$ and $\widehat{y}_2(n) = y(2n+1)$. We claim that at least one function among $\widehat{y}_1, \widehat{y}_2$ belongs to IOE(g). Suppose otherwise. Then there are computable functions $x_1, x_2 < g$ which differ almost all the time from $\widehat{y}_1$ and $\widehat{y}_2$, respectively. Since h is nondecreasing, the computable function x defined by $x(2n) = x_1(n)$ and $x(2n+1) = x_2(n)$ satisfies $x < h$. It is clear that x differs almost all the time from y, which contradicts $y \in \mathrm{IOE}(h)$.
$\boxed{\mathrm{AED}(h) \le_S \mathrm{AED}(g):}$ Let $y < g$ be a function in AED(g). Let $\widehat{y}(2n+i) = y(n)$ for $i \le 1$, so that $\widehat{y} < h$. Given any computable function x, for almost

every n we have $x(2n) \neq y(n)$ and $x(2n+1) \neq y(n)$. Therefore $x(n) \neq \widehat{y}(n)$ for almost every n. Hence $\widehat{y} \in \mathrm{AED}(h)$.

(ii) is immediate from (i) by iteration, using that $a^{2^i} > b$ and $b^{2^i} > a$ for sufficiently large i. □

The following operators will be used for the rest of the section.

Definition 3.2. Let h be a function of the form $2^{\widehat{h}}$ with $\widehat{h}\colon \omega \to \omega$, and let X_h be the space of all h-bounded functions. For such a function we view $x(n)$ either as a number, or as a binary string of length $\widehat{h}(n)$ via the binary expansion with leading zeros allowed. We define $L_h\colon X_h \to {}^\omega 2$ by $L_h(x) = \prod_n x(n)$, i.e. the concatenation of these strings. We let $K_h\colon {}^\omega 2 \to X_h$ be the inverse of L_h.

Lemma 3.3. *Let $a \in \omega - \{0\}$. We have* $\mathrm{IOE}(2^{(a^n)}) \geq_S \mathcal{D}(1/a)$ *and* $\mathrm{AED}(2^{(a^n)}) \leq_S \mathcal{B}(1/a)$.

Proof. Let I_m for $m \geq 2$ be the $(m-1)$-th consecutive interval of length a^m in $\omega - \{0\}$, i.e.

$$I_m = \left[\frac{a^m - 1}{a-1}, \frac{a^{m+1}-1}{a-1}\right)$$

Let $h(m) = 2^{(a^m)}$. Let us first show that $\mathrm{IOE}(2^{(a^n)}) \geq_S \mathcal{D}(1/a)$. Let $y < 2^{(a^n)}$ be a function in $\mathrm{IOE}(2^{(a^n)})$ and let $\widehat{y} = L_h(y)$. Given a computable set x, let $x' = K_h(1-x)$. As $x'(m) = y(m)$ for infinitely many m, for infinitely many intervals m, all bits of x with location in I_m differ from all the bits of $\widehat{y}$ in this location. It follows that $\widehat{y} \in \mathcal{D}(1/a)$.

Let us now show that $\mathrm{AED}(2^{(a^n)}) \leq_S \mathcal{B}(1/a)$. Let $y \in \mathcal{B}(1/a)$, and let $\widehat{y} = K_h(y)$. Given a computable function $x < h$, let $x' = 1 - L_h(x)$. Since $\underline{\rho}(x' \leftrightarrow y) > 1/a$, for large enough n, there is $k \in I_n$ such that $x'(k) = y(k)$. Hence we cannot have $x(n) = \widehat{y}(n)$. Thus $\widehat{y} \in \mathrm{AED}(2^{(a^n)})$. □

We remark that similar argument shows

$$\mathrm{IOE}(2^{\widehat{h}(m)}) \geq_S \mathcal{D}(0) \text{ and } \mathrm{AED}(2^{\widehat{h}(m)}) \leq_S \mathcal{B}(0)$$

for any computable function $\widehat{h}$ such that $\forall a \, \forall^\infty m \, \widehat{h}(m) \geq a^m$. Now one chooses the m-th interval to have length $\widehat{h}(m)$.

Theorem 3.4. *Fix any $p \in (0, 1/2)$. We have*

$$\mathcal{D}(p) \equiv_W \mathrm{IOE}(2^{(2^n)}) \text{ and } \mathcal{B}(p) \equiv_S \mathrm{AED}(2^{(2^n)}).$$

The rest of the section is dedicated to the proof of Theorem 3.4. The two foregoing lemmas imply $\mathcal{D}(p) \leq_W \mathrm{IOE}(2^{(2^n)})$ and $\mathcal{B}(p) \geq_S \mathrm{AED}(2^{(2^n)})$. It remains to show the more difficult converse reductions $\mathcal{D}(p) \geq_W \mathrm{IOE}(2^{(2^n)})$ and $\mathcal{B}(p) \leq_S \mathrm{AED}(2^{(2^n)})$. Let us informally describe the proof of the first reduction, which is closely based on arguments in Monin's proof [13] that $\Gamma(A) < 1/2 \Leftrightarrow \Gamma(A) = 0$.

Given $A \in \mathcal{D}(p)$ we want to find a function $f \leq_T A$ that agrees with each computable function $g < 2^{(2^n)}$ infinitely often. For an appropriate k let $\widehat{h}(n) = \lfloor 2^{n/k} \rfloor$ and $h(n) = 2^{\widehat{h}(n)}$. We split the bits of A into consecutive intervals of length $\widehat{h}(n)$. The first step (Claim 3.7) makes the crucial transition from the density setting towards the setting of functions agreeing on certain arguments. We will show that for k large enough, the function $f_0 = K_h(A) < h$ has the property that for each computable function $g < h$, for infinitely many n, $f_0(n)$ and $g(n)$ disagree on a fraction of fewer than p bits when viewed as binary strings of length $\widehat{h}(n)$.

In the second step (Claim 3.11) we next use f_0 to compute a special kind of approximation s to computable functions: for each n, $s(n)$ is a set of L many values (where L is an appropriate constant) such that for every computable function $g < h$ we have $\exists^\infty n\ g(n) \in s(n)$. The function s is called a slalom (or trace). This important step uses a result from the theory of error correcting codes, which determines the constant L.

In the third step (Claim 3.12), which is non-uniform, we replace s by a slalom s' such that still $s'(n)$ has size at most L, but now all computable functions g with $g(n) < 2^{(2^{Ln})}$ are captured infinitely often. In a final, non-uniform step (Claim 3.13) we then compute from s' a function f as required: for some i, $f(n)$ is the i-th block of length 2^n of the i-th element of $s(n)$.

We now provide the detailed argument.

Definition 3.5. For strings x, y of length r, the normalized Hamming distance is defined as the proportion of bits on which x, y disagree, that is,

$$d(x, y) = \frac{1}{r}|\{i \colon x(n)(i) \neq y(n)(i)\}|$$

Definition 3.6. Let h be a function of the form $2^{\widehat{h}}$ with $\widehat{h} \colon \omega \to \omega$, and let X_h be the space of h-bounded functions. Let $q \in (0, 1/2)$. We define a relation on $X_h \cdot X_h$ by:

$$x \neq^*_{\widehat{h},q} y \Leftrightarrow \forall^\infty n\,[d(x(n), y(n)) \geq q]$$

namely for almost every n the strings $x(n)$ and $y(n)$ disagree on a proportion of at least q of the bits. We will usually write $\langle \neq^*, \widehat{h}, q\rangle$ for this relation.

Claim 3.7. *Let $q \in (0, 1/2)$. For each $c \in \omega$ such that $2/c < q$, there is $k \in \omega$ such that*

$$\mathcal{D}(q - 2/c) \geq_S \mathcal{D}\langle \neq^*, \lfloor 2^{n/k} \rfloor, q\rangle \text{ and } \mathcal{B}(q - 2/c) \leq_S \mathcal{B}\langle \neq^*, \lfloor 2^{n/k} \rfloor, q\rangle.$$

Proof. Let k be large enough so that $2^{1/k} - 1 < \frac{1}{2c}$. Let $\widehat{h}(n) = \lfloor 2^{n/k} \rfloor$ and $h = 2^{\widehat{h}}$. Write $H(n) = \sum_{r \leq n} \widehat{h}(r)$. By the sum formula for the geometric series we have $\widehat{h}(n+1) - 1 = (2^{1/k} - 1)H(n)$, and thus $\widehat{h}(n+1) - 1 \leq (1/2c)H(n)$. It follows that for sufficiently large n,

$$\widehat{h}(n+1) \leq \frac{1}{c} H(n) \tag{1}$$

We rely on the following.

Fact 3.8. *Let $x, y < h$ be functions such that $\forall^\infty n\,[d(x(n), y(n)) \leq 1 - q]$. Then $\underline{\rho}(L_h(x) \leftrightarrow L_h(y)) > q - 2/c$.*

To see this, note that by hypothesis, for almost every n we have that $L_h(y) \upharpoonright_{H(n)}$ agrees with $L_h(x) \upharpoonright_{H(n)}$ on a fraction of at least q bits. Now for any n and any m with $H(n) \leq m \leq H(n+1)$, we then have that $L_h(y) \upharpoonright_m$ agrees with $L_h(x) \upharpoonright_m$ on a fraction of at least $\frac{H(n)q}{H(n)+\widehat{h}(n+1)}$ bits, which is by (1) a fraction of at least $\frac{H(n)q}{H(n)+(1/c)H(n)}$ bits. It follows that for almost every m, we have that $L_h(y) \restriction_m$ agrees with $L_h(x) \restriction_m$ on a fraction of at least $\frac{q}{1+1/c} > q - 2/c$ bits. It implies in particular that $\underline{\rho}(L_h(x) \leftrightarrow L_h(y)) > q - 2/c$. This shows the fact.

Let us now show that $\mathcal{D}(q-2/c) \geq_S \mathcal{D}\langle \neq^*, \lfloor 2^{n/k} \rfloor, q\rangle$. Let $y \in \mathcal{D}(q-2/c)$. Let $y' = K_h(y)$. By the fact above, there is no computable function $x < h$ such that $\forall^\infty n\, [d(x(n), y'(n)) \leq 1-q]$, as otherwise we would have $L_h(y') = y \notin \mathcal{D}(q-2/c)$ which is a contradiction.Therefore, for every computable function $x < h$ we have $\exists^\infty n\, [d(x(n), y'(n)) > 1-q]$. Now let $x < h$ be a computable function and let $x' = K_h(1 - L_h(x))$. As $x' < h$ is computable we must have $\exists^\infty n\, [d(x'(n), y'(n)) > 1-q]$. But then we also have $\exists^\infty n\, [d(x(n), y'(n)) < q]$. As this is true for any computable function $x < h$ we then have $y' \in \mathcal{D}\langle \neq^*, \lfloor 2^{n/k} \rfloor, q\rangle$.

Secondly we show that $\mathcal{B}(q-2/c) \leq_S \mathcal{B}\langle \neq^*, \lfloor 2^{n/k} \rfloor, q\rangle$.
Let $y \in \mathcal{B}\langle \neq^*, \lfloor 2^{n/k} \rfloor, q\rangle$. Thus, $y < h$ and $\forall^\infty n\, [d(x(n), y(n) \geq q]$ for each computable function $x < h$. Let $y' = K_h(1 - L_h(y))$. Then

$$\forall^\infty n\, [d(x(n), y'(n)) \leq 1-q]$$

for each computable function $x < h$. By the fact above, we then have that $\underline{\rho}(L_h(x) \leftrightarrow L_h(y')) > q - 2/c$ for each computable function $x < h$. It follows that $L_h(y') \in B(q - 2/c)$. $\square_{3.7}$

For $L \in \omega$, an L-slalom is a function $s\colon \omega \to \omega^{[\leq L]}$, i.e. a function that maps natural numbers to sets of natural numbers with a size of at most L.

Definition 3.9. Fix a function $u\colon \omega \to \omega$ and $L \in \omega$. Let X be the space of L-slaloms (or traces) s such that $\max s(n) < u(n)$ for each n. Thus s maps natural numbers to sets of natural numbers of size at most L, represented by strong indices. Let Y be the set of functions such that $y(n) < u(n)$ for each n. Define a relation on $X \cdot Y$ by

$$s \not\ni^*_{u,L} y \Leftrightarrow \forall^\infty n[s(n) \not\ni y(n)].$$

We will write $\langle \not\ni^*, u, L\rangle$ for this relation.

For what follows, we use the list decoding capacity theorem from the theory of error-correcting codes. Given q as above and $L \in \omega$, for each r there is a "fairly large" set C of strings of length r (the allowed code words) such that for each string, at most L strings in C have normalized Hamming distance less than q from σ. (Hence there is only a small set of strings that could be the error-corrected version of σ.) Given a string σ of length r, let $B_q(\sigma)$ denote an open ball around σ in the normalized Hamming distance, namely, $B_q(\sigma) = \{\tau \in {}^r2\colon\ \sigma, \tau$ disagree on fewer than qr bits$\}$.

Theorem 3.10 (List decoding, Elias [6])**.** *Let $q \in (0, 1/2)$. There are $\epsilon > 0$ and $L \in \omega$ such that for each r, there is a set C of $2^{\lfloor \epsilon r \rfloor}$ strings of length r as follows:*

$$\forall \sigma \in {}^r2\, [|B_q(\sigma) \cap C| \leq L].$$

The previous theorem allows us to show the following:

Claim 3.11. *Given $q < 1/2$, let L, ϵ be as in Theorem 3.10. Fix a nondecreasing computable function $\widehat{h}$, and let $u(n) = 2^{\lfloor \epsilon \widehat{h}(n) \rfloor}$. We have*

$$\mathcal{D}\langle \neq^*, \widehat{h}, q\rangle \geq_S \mathcal{D}\langle \not\ni^*, u, L\rangle \text{ and } \mathcal{B}\langle \not\ni^*, u, L\rangle \geq_S \mathcal{B}\langle \neq^*, \widehat{h}, q\rangle.$$

Proof. For each r of the form $\widehat{h}(n)$ compute a set $C = C_r$ as in Theorem 3.10. Since $|C_r| = 2^{\lfloor \epsilon r \rfloor}$ there is a uniformly computable sequence $\langle \sigma_i^r \rangle_{i<2^{\lfloor \epsilon r \rfloor}}$ listing C_r in increasing lexicographical order.

Let us first show $\mathcal{D}\langle \neq^*, \widehat{h}, q\rangle \geq_S \mathcal{D}\langle \not\ni^*, u, L\rangle$. Suppose that $y \in D\langle \neq^*, \widehat{h}, q\rangle$. Let s be the y-computable L-trace such that

$$s(n) = \{i < u(n) \colon d(\sigma_i^{\widehat{h}(n)}, y(n)) < q\}.$$

Let now $x < u$ be a computable function. As infinitely often $d(\sigma_{x(n)}^{\widehat{h}(n)}, y(n)) < q$, then also infinitely often we have $x(n) \in s(n)$. It follows that $s \in D\langle \not\ni^*, u, L\rangle$, as required.

Let us now show $\mathcal{B}\langle \not\ni^*, u, L\rangle \geq_S \mathcal{B}\langle \neq^*, \widehat{h}, q\rangle$. Suppose that $y \in \mathcal{B}\langle \not\ni^*, u, L\rangle$. Let $h = 2^{\widehat{h}}$ and $\widetilde{y} < h$ be the function given by $\widetilde{y}(n) = \sigma_{y(n)}^{\widehat{h}(n)}$. We show that $\widetilde{y} \in \mathcal{B}\langle \neq^*, \widehat{h}, q\rangle$. Let $x < h$ be a computable function. Let

$$s_x(n) = \{i < u(n) \colon d(\sigma_i^{\widehat{h}(n)}, x(n)) < q\}.$$

Note that $\langle s_x \rangle$ is an L-trace because the listing $\langle \sigma_i^r \rangle_{i<2^{\lfloor \epsilon r \rfloor}}$ has no repetitions. Since $y \in \mathcal{B}\langle \not\ni^*, u, L\rangle$, for almost every n we have $y(n) \notin s_x(n)$. Hence also for almost every n we have $d(\widetilde{y}(n), x(n) \geq q$, as required. $\square_{3.11}$

We next need an amplification tool in the context of traces. The proof is almost verbatim the one in Lemma 3.1, so we omit it.

Claim 3.12. *Let $L \in \omega$, let the computable function u be nondecreasing and let $w(n) = u(2n)$. We have*

$$\mathcal{D}\langle \not\ni^*, u, L\rangle \equiv_W \mathcal{D}\langle \not\ni^*, w, L\rangle \text{ and } \mathcal{B}\langle \not\ni^*, u, L\rangle \equiv_S \mathcal{B}\langle \not\ni^*, w, L\rangle.$$

Iterating the claim, starting with the function $\widehat{h}(n) = \lfloor 2^{n/k} \rfloor$ with k as in Claim 3.7, we obtain that $\mathcal{D}\langle \not\ni^*, 2^{\widehat{h}}, L\rangle \equiv_W \mathcal{D}\langle \not\ni^*, 2^{(L2^n)}, L\rangle$ and $\mathcal{B}\langle \not\ni^*, 2^{\widehat{h}}, L\rangle \equiv_S \mathcal{B}\langle \not\ni^*, 2^{(L2^n)}, L\rangle$. It remains to verify the following, which would work for any computable function $\widehat{h}(n)$ in place of the 2^n in the exponents.

Claim 3.13.

$$\mathcal{D}\langle \not\ni^*, 2^{(L2^n)}, L\rangle \geq_W \text{IOE}(2^{(2^n)}) \text{ and } \text{AED}(2^{(2^n)}) \geq_S \mathcal{B}\langle \not\ni^*, 2^{(L2^n)}, L\rangle.$$

Proof. Given n, we write a number $k < 2^{(L2^n)}$ in binary with leading zeros if necessary, and so can view k as a binary string of length $L2^n$. We view such a string as consisting of L consecutive blocks of length 2^n.

Let us first show $\mathcal{D}\langle \not\ni^*, 2^{(L2^n)}, L\rangle \geq_W \text{IOE}(2^{(2^n)})$. Let $s \in \mathcal{D}\langle \not\ni^*, 2^{(L2^n)}, L\rangle$. For every $i \leq L$, let y_i be the s-computable function such that $y_i(n)$ is the i-th block of the i-th element of $s(n)$. Suppose for a contradiction that for every i we have a computable function $x_i \leq 2^{(2^n)}$ which differs on almost every argument from y_i. Let $x \leq 2^{(L2^n)}$ be the computable function defined by $x(n)$ to be the concatenation of $x_i(n)$ for each $i \leq L$. Then also for almost every n we have $s(n) \not\ni x(n)$, which is a contradiction. Therefore we must have that $x_i \in \mathrm{IOE}(2^{(2^n)})$ for some $i \leq L$.

Let us now show $\mathrm{AED}(2^{(2^n)}) \geq_S \mathcal{B}\langle \not\ni^*, 2^{(L2^n)}, L\rangle$. Let $y \in \mathrm{AED}(2^{(2^n)})$. That is, $y < 2^{(2^n)}$ and $\forall^\infty n\, x(n) \neq y(n)$ for each computable function x. Let y' be the function bounded by $2^{(L2^n)}$ such that for each n, each block of $y'(n)$ equals $y(n)$. Given a computable L-trace s with $\max s(n) < 2^{(L2^n)}$, for $i < L$ let x_i be the computable function such that $x_i(n)$ is the i-th block of the i-th element of $s(n)$ (as before we may assume that each string in $s(n)$ has length $L2^n$). For sufficiently large n, we have for all $i < L$ that $y(n) \neq x_i(n)$. Hence $\forall^\infty n\, s(n) \not\ni y'(n)$ and thus $y' \in \mathcal{B}\langle \not\ni^*, 2^{(L2^n)}\rangle$. $\square_{3.13}$

We can now summarize the arguments that $\mathcal{D}(p) \geq_W \mathrm{IOE}(2^{(2^n)})$ and $\mathrm{AED}(2^{(2^n)}) \geq_S \mathcal{B}(p)$.

Proof of Theorem 3.4. Pick c large enough such that $q = p + 2/c < 1/2$. By Claim 3.7 there is k such that

$$\mathcal{D}(p) \geq_S \mathcal{D}\langle \neq^*, \lfloor 2^{n/k}\rfloor, q\rangle \text{ and } \mathcal{B}\langle \neq^*, \lfloor 2^{n/k}\rfloor, q\rangle \geq_S \mathcal{B}(p).$$

By Claim 3.11 there are L, ϵ such that where $\widehat{h}(n) = \lfloor 2^{n/k}\rfloor$ and $u(n) = 2^{\lfloor \epsilon \widehat{h}(n)\rfloor}$, we have

$$\mathcal{D}\langle \neq^*, \widehat{h}, q\rangle \geq_S \mathcal{D}\langle \not\ni^*, u, L\rangle \text{ and } \mathcal{B}\langle \not\ni^*, u, L\rangle \geq_S \mathcal{B}\langle \neq^*, \widehat{h}, q\rangle.$$

Applying Claim 3.12 sufficiently many times we have

$$\mathcal{D}\langle \not\ni^*, u, L\rangle \equiv_W \mathcal{D}\langle \not\ni^*, 2^{(L2^n)}, L\rangle \text{ and } \mathcal{B}\langle \not\ni^*, u, L\rangle \equiv_S \mathcal{B}\langle \not\ni^*, 2^{(L2^n)}, L\rangle.$$

Finally

$$\mathcal{D}\langle \not\ni^*, 2^{(L2^n)}, L\rangle \geq_W \mathrm{IOE}(2^{(2^n)}) \text{ and } \mathrm{AED}(2^{(2^n)}) \geq_S \mathcal{B}\langle \not\ni^*, 2^{(L2^n)}, L\rangle.$$

by Claim 3.13. Combining all this yields

$$\mathcal{D}(p) \geq_W \mathrm{IOE}(2^{(2^n)}) \text{ and } \mathcal{B}(p) \leq_S \mathrm{AED}(2^{(2^n)}).$$

$\square_{3.4}$

The Γ and Δ parameters of a Turing oracle. The Γ parameter of a Turing oracle A defined in the introduction has a dual, first considered by Merkle, Stephan and the second author [4], and then in [15, Section 7].

Definition 3.14.

$$\delta(A) = \inf_{X \text{ computable}} \underline{\rho}(A \leftrightarrow X)$$
$$\Delta(A) = \sup\{\delta(Y)\colon\, Y \leq_T A\}.$$

Intuitively, $\Gamma(A)$ measures how well computable sets can approximate the sets that A computes, counting the asymptotically worst case (the infimum over all $Y \leq_T A$). In contrast, $\Delta(A)$ measures how well the sets that A computes can approximate the computable sets, counting the asymptotically best case (the supremum over all $Y \leq_T A$). Clearly $\Delta(A) \leq 1/2$ for each A.

Corollary 3.15. *(i)* $\Gamma(A) < 1/2$ *implies* $\Gamma(A) = 0$.
(ii) $\Delta(A) > 0$ *implies* $\Delta(A) = 1/2$.

Proof. By the definitions, for each $p \in (0, 1/2)$, we have

$$\Gamma(A) < p \Rightarrow \exists Y \leq_{\mathrm{T}} A[Y \in \mathcal{D}(p)] \Rightarrow \Gamma(A) \leq p$$

and

$$\Delta(A) > p \Rightarrow \exists Y \leq_{\mathrm{T}} A[Y \in \mathcal{B}(p)] \Rightarrow \Delta(A) \geq p.$$

Now apply Theorem 3.4. □

The Δ values 0 and $1/2$ can be realized by the following two facts.

Proposition 3.16. *Let A compute a Schnorr random Y. Then $\Delta(A) = 1/2$.*

Proof. If Y is Schnorr random, then we must have $\underline{\rho}(A \leftrightarrow X) = 1/2$ for every computable set A. □

Proposition 3.17. *Suppose A is 2-generic. Then $\Delta(A) = 0$.*

Proof. A is neither high nor d.n.c., so A is not in $\mathcal{B}(\neq^*)$ = high or d.n.c., the class where no bound is imposed on the witness function A computes. So A is not in $\mathrm{AED}(2^{(2^n)})$, hence $\Delta(A) = 0$ by the second equivalence in Theorem 3.4. □

4. A proper hierarchy of problems IOE(h) in the weak degrees

Recall that by $\mathrm{IOE}(h)$ we denote the mass problem of functions f such $\exists^\infty n\, [f(n) = r(n)]$ for each computable function $r < h$. In this section we study how the Muchnik degree of $\mathrm{IOE}(h)$ depends on the function h. In [14] the authors obtained the following two results:

Theorem 4.1 ([14])**.** *Let $c \geq 2$ be any integer, which we view as a constant function.*

$$\mathrm{IOE}(2) \equiv_S \mathrm{IOE}(c) \equiv_S \{X \ : \ X \text{ is not computable}\}$$

The difficult part of the theorem is to show that $\mathrm{IOE}(2) \geq_S \mathrm{IOE}(c)$ for $c > 2$. This can be done using error-correcting codes.

Theorem 4.2 ([14])**.** *For any functions $F_1 < F_2$ such that $\sum_n 1/F_1(n) = \infty$ and $\sum_n 1/F_2(n) < \infty$, we have:*

$$\mathrm{IOE}(F_1) <_W \mathrm{IOE}(F_2)$$

We now show that given any function F, one can find a function $G > F$ such that:

$$\mathrm{IOE}(F) <_W \mathrm{IOE}(G)$$

In what follows, given a function F we let $w_F(n)$ be the number of possible combinations of n first values for functions $f \leq F$, that is,

$$w_F(n) = \Pi_{0 \leq i \leq n} F(i).$$

To improve the readability of expressions with iterated exponentiation, we will mostly write $\exp(x)$ for 2^x.

Theorem 4.3 (with Joe Miller)**.** *Let $F \in {}^\omega\omega$ be an order function. Let $G \in {}^\omega\omega$ be an order function with $G(n) \geq 2$ for every n and such that:*

$$\forall k\ \forall^\infty n\ \ \exp(w_F(\exp(n \cdot k))) < G(n)$$

There exists a function $f \in \mathrm{IOE}(F)$ and such that $g \notin \mathrm{IOE}(G)$ for every $g \leq_T f$.

For instance, if $F(n) = n$ we can let $G(n) = \exp\exp\exp(n^2)$. The rest of the section is dedicated to the proof of Theorem 4.3. Let us first introduce some terminology.

Definition 4.4. By a *tree* we mean a set of strings closed under prefixes. Let $H : \omega \to \omega$. We denote by ${}^{<\omega}H$ the tree whose paths are the functions less than or equal to H. Let $T \subseteq {}^{<\omega}H$ be a tree. We say that T is *H-full-branching* if for every $f < H$ we have $f \in [T]$. For a string $\sigma \in {}^{<\omega}\omega$ and $n > |\sigma|$, we say that T is *$H\restriction_n$-full-branching above σ* if for every $f < H$ with $\sigma \prec f$ we have $f\restriction_n \in T$.

Given a node σ of length m and a $H\restriction_{m+n}$-full-branching tree T above σ, we sometimes say that n is the *height* of the full-branching part of T. Let us continue with one of these lemma whose statement is more complicated than the proof:

Lemma 4.5. Let $H : \omega \to \omega$. Let $\sigma \in {}^{<\omega}\omega$ with $|\sigma| = m$. Let $n \in \omega$ and let $T \subseteq {}^{<\omega}H$ be a finite $H\restriction_{m+2n}$-full-branching tree above σ. Let $\sigma_0, \dots, \sigma_k$ be all the leaves of T. Consider a partition C_1, C_2 of these leaves. Then one of the following holds:

(1) If we keep only the nodes compatible with some element of C_1 and discard the rest, the remaining tree is $H\restriction_{m+n}$-full-branching above σ.
(2) If we keep only the nodes compatible with some element of C_2 and discard the rest, there exists a node $\tau \succ \sigma$ of length $m+n$ such that the remaining tree is $H\restriction_{m+2n}$-full-branching above τ.

In particular, in both cases, the full-branching part of the remaining tree has height n.

Proof. Suppose (1) fails. Then there is a string $\tau \succ \sigma$, $\tau \in T$ of length $m+n$ such that all the extensions in T of length $m+2n$ of τ are leaves of T which are not in C_1. Then these leaves are in C_2. So (2) holds. □

The idea of the proof is that given any functional Φ, we are able to compute an infinite tree $T \subseteq {}^{<\omega}F$ such that:

(1) For every path $X \in [T]$ we have that $\Phi(X) \notin \mathrm{IOE}(G)$.
(2) For every path $X \in [T]$, there are infinitely many m such that T is $F\restriction_{m+1}$-full-branching above $X\restriction_m$.
(3) T has no dead ends.

Note that (3) ensures that the tree is computable in a strong sense : if a node σ si in T, then there exists an infinite path $X \in [T]$ with $X \succ \sigma$. By combining (1) with (2) we actually know that the set of infinite paths extending σ is perfect. While (1) ensures that no path of T computes an element of $\mathrm{IOE}(G)$ via Φ, (2) ensures that the tree T still contains an element of $\mathrm{IOE}(F)$. Also, starting from the tree ${}^{<\omega}F$, one can compute a sub-tree T which satisfies (1) and (2) using Lemma 4.5.

In order to help the reader understand the full proof, we sketch here a construction to obtain a computable tree T that satisfies both (1) and (2) given some functional Φ, under the assumption that G grows sufficiently faster than F. Of course this allows us to defeat only one functional Φ. To defeat more than one functional Φ we would need not only to obtain (2), but to obtain a computable tree for which we have infinitely many large full-branching blocks. In this case we can repeat the construction in the tree we end up with, so as to defeat yet another functional. This will be achieved by the upcoming Lemma 4.7, elaborating on the ideas already present in the construction we present now.

Sketch of a construction to obtain (1), (2) and (3). We work here under the assumptions of Theorem 4.3. Note however that in the simpler case of defeating only one functional, the assumption on how fast G grows compare to F can be relaxed somewhat: we merely need that

$$\forall k\ \forall^\infty n\ \exp(k \cdot \exp(n)) < G(n).$$

In the following all strings will be chosen from the F-full-branching tree. We can suppose without loss of generality that given any σ and any n there exists an extension τ of σ such that $\Phi(\tau, n)$ is defined. Otherwise there is a string σ and some n such that $\Phi(X, n)$ is undefined for every path X extending σ and the desired tree T is given by all the nodes compatible with σ.

The construction inductively defines finite trees $T_0 \subset T_1 \subset \dots$ together with integers $n_0 < n_1 < \dots$ such that :

(a) For every k, every leaf of T_k has a full-branching extensions in T_{k+1}.
(b) For every k, every leaf $\rho \in T_k$ and every $t < n_k$, every value $\Phi(\rho, t)$ is defined.
(c) For every k, every $t \leq n_k$ one value smaller than G(t) is different from $\Phi(\rho, t)$ for every leaf $\rho \in T_k$.
(d) For every k we have $\exp(c) < G(n_k)$ where c is the number of leaves in T_k

Note that unlike (a) (b) and (c), (d) does not achieve by itself anything we want, but it will be necessary at each step to continue the induction, in particular in order to show (c).

To begin the inductive definitions, let n_0 be least such that

$$\exp(F(\exp(n_0))) < G(n_0).$$

Consider the $F\restriction_{\exp(n_0)}$-full-branching tree above the empty string. Let $\sigma_0, \dots, \sigma_c$ be an enumeration of the leaves of this $F\restriction_{\exp(n_0)}$-full-branching tree. For each $i \leq c$, we look for an extension τ_i of σ_i such that $\Phi(\tau_i, t)$ is defined for every $t \leq n_0$. We can assume without loss of generality that every node τ_i has the same length m_0 (presumably much larger than n_0). We now partition the set of nodes τ_i into those such that $\Phi(\tau_i, 0) = 0$ and those such that $\Phi(\tau_i, 0) \neq 0$. By Lemma 4.5, we can either remove all nodes of length m_0 forcing $\Phi(0) = 0$, or all nodes of length m_0 forcing another value (and everything compatible with these nodes), in such a way that we have a node σ above which the tree consisting of the nodes we keep is $F\restriction_{|\sigma|+\exp(n_0-1)}$-full branching. Note that σ can be either the root of the tree or a string of length $\exp(n_0 - 1)$.

We continue inductively the previous operation for each of the n_0 first values of Φ. At the end, we have a node σ above which there is a $F\restriction_{|\sigma|+1}$-full-branching tree, and such that given any $t \leq n_0$, the remaining nodes τ_i of length m_0 are altogether such that $\Phi(\tau_i, t) = 0$ or such that $\Phi(\tau_i, t) \neq 0$.

Let T_0 be the tree consisting of the remaining nodes τ_i and everything below them. For every $t \leq n_0$, in the first case we define $g(t) = 1$ and in the second $g(t) = 0$. Note that as $\exp(F(\exp(n_0))) < G(n_0)$, then also we must have $\exp(c) < G(n_0)$ where c is the number of nodes of length m_0 in T_0.

Suppose now by induction that we have a finite tree T_k with leaves $\tau_0, \dots, \tau_c \in T_k$ each of length m_k, and a value n_k such that (a), (b), (c) and (d) are verified. In particular we have $\exp(c) < G(n_k)$. Let $n_{k+1} > n_k$ be the smallest such that

$$\exp(c \cdot F(m_k + \exp(n_{k+1}))) < G(n_{k+1})$$

Let us show that for any a with $0 \leq a \leq c$, we can computably find a finite tree T^a whose nodes are all compatible with τ_a and such that:

- T^a is $F\lceil_{|\sigma|+1}$-full branching above some $\sigma \succ \tau_a$.
- Each leaf ρ of T^a is such that $\Phi(\rho, t)$ is defined for $n_k < t \leq n_{k+1}$.
- For every $n_k < t \leq n_{k+1}$, there is at least one value smaller than $G(t)$ which is different from every value $\Phi(\rho, t)$ for leaves ρ of T^a.

For any $a \leq c$ we do the following: consider the finite $F\lceil_{|\tau_a|+\exp(n_{k+1})}$)-full branching tree above τ_a. Let $\sigma_0, \dots, \sigma_k$ be an enumeration of the leaves of this finite tree. For each of these nodes σ_i, look for an extension τ'_i such that $\Phi(\tau'_i, t)$ is defined for every $n_k < t < n_{n+1}$. Let $T^{a\prime}$ be the finite tree consisting of these extensions τ'_i and everything below them.

We now partition the set of leaves of $T^{a\prime}$ into two sets C_1 and C_2 such that the leaves ρ in C_1 are these for which the a-th bit of $\Phi(\rho, n_k + 1)$ is 0 and the leaves in C_2 are these for which the a-th bit of $\Phi(\rho, n_k + 1)$ is 1. By the Lemma 4.5, we can either remove all nodes of C_1 or all nodes of C_2 (and everything compatible with these nodes), in such a way that we have a node $\sigma \in T^{a\prime}$ such that the tree consisting of the nodes we keep, is $F\lceil_{|\sigma|+\exp(n_{k+1}-1})$-full branching above σ.

We continue inductively the previous operation for each of the next values of Φ up to n_{k+1}. At the end, we have a node $\sigma \in T^{a\prime}$ above which there is a $F\lceil_{|\sigma|+1}$-full-branching tree, such that for each $n_k < t \leq n_{k+1}$, there exists $i \in \{0, 1\}$ such that for all the remaining leaves τ'_i of our $F\lceil_{|\sigma|+1}$-full-branching tree, the a-th bit of $\Phi(\tau'_i, t)$ is the same. We define the tree T^a to be this set of remaining leaves τ'_i and everything below them.

Once every tree T^a has been defined, we define each value of $g(t)$ for $n_k < t \leq n_{k+1}$, as follow : If the leaves ρ of T^a are such that the a-th bit of $\Phi(\rho, t)$ equals 0, then the a-th bit of $g(n)$ is defined to be 1, and vice-versa. Remember that we have $\exp(c) < G(n_k)$. In particular any number coded on at most c bits is smaller than $G(t)$ for any $n_k < t \leq n_{k+1}$. It follows that $g(t) < G(t)$ for any $n_k < t \leq n_{k+1}$. Also we necessarily have that $g(t)$ is different from every possible value $\Phi(\rho, t)$ for every leaf $\rho \in \bigcup_{a \leq c} T^a$. Let $T_{k+1} = \bigcup_{a \leq c} T^a$. Note that by the choice of n_{k+1} we have that $\exp(d) < G(n_{k+1})$ where d is the number of leaves in T_{k+1}.

By continuing the induction, we define a computable subtree $T = \bigcup_k T_k$ of the F-full-branching tree as well as a computable function $g < G$, such that along any path of T, infinitely many nodes are full-branching, and such that for any $f \in [T]$ we have that $\Phi(f, n) \neq g(n)$ for any n. □

As mentioned before, the previous proof works only to defeat one functional Φ. Suppose now that we want to defeat every functional. Let $\Phi_0, \Phi_1, \Phi_2, \dots$ be a list of all functionals. The previous proof gives us a tree T_0 which defeats Φ_0. To defeat Φ_1, we have to perform a similar construction, but starting now from the computable tree T_0 in place of the F-full-branching tree ${}^{<\omega}F$. In this way we obtain a computable tree $T_1 \subseteq T_0$ which defeats both Φ_0 and Φ_1. The main problem is that to use Lemma 4.5 we need to work in a tree that has large full-branching blocks (which is the case of ${}^{<\omega}F$). Also if T_0 itself does not have large full-branching blocks, it is not necessarily possible to defeat Φ_1 starting from T_0 in place of ${}^{<\omega}F$. To overcome this problem, it is not sufficient to merely ensure (2) for T_0: we actually need to ensure that for every path $X \in T_0$, there are infinitely many m such that T_0 is $F\upharpoonright_{m+n_m}$-full-branching above $X\upharpoonright_m$ for n_m sufficiently large. This leads to the following definition:

Definition 4.6. Let $F, G \in {}^{\omega}\omega$ be order functions. Let $T \subseteq {}^{<\omega}F$ be a finite tree. Let $n_1 < n_2 < \cdots < n_k$. We say that T is *G-fat for* $(n_1, n_2, \dots, n_k)$ if for every leaf $\sigma \in T$, there exists $m_1 < m_2 < \cdots < m_k < |\sigma|$ such that for every $1 \leq t \leq k$:

(1) The tree T is $F\upharpoonright_{m_t+\exp(n_t \cdot t)}$-full-branching above $\sigma\upharpoonright_{m_t}$.
(2) $\exp(w_F(m_t + \exp(n_t \cdot t))) < G(n_t)$.

We say that $T \subseteq {}^{<\omega}F$ is *infinitely often G-fat* if there exists an infinite sequence $n_1 < n_2 < \dots$ such that for every k, there exists m such that T restricted to its node of length m, is G-fat for $(n_1, \dots, n_k)$.

The following lemma is the heart of the proof. It says that for any computable infinitely often G-fat tree T and any functional Φ, there is a computable infinitely often G-fat tree $T' \subseteq T$ such that no path of T' computes an element of $\mathrm{IOE}(G)$ via Φ.

Lemma 4.7. *Let $F \in {}^{\omega}\omega$ be an order function. Let $G \in {}^{\omega}\omega$ be an order function such that $G(n) \geq 2$ for every n. Let $T \subseteq {}^{<\omega}F$ be a computable infinitely often G-fat tree with no dead ends. Let Φ be a functional. Then there exists a computable infinitely often G-fat tree $T' \subseteq T$ with no dead ends, and a computable function $g < G$ such that for every path $X \in [T']$ for which $\Phi(X)$ is total, we have $\Phi(X, n) \neq g(n)$ for every n.*

Before giving the proof of the Lemma, we show how to use it in order to obtain the proof of Theorem 4.3, using simple forcing machinery.

Proof of Theorem 4.3. Let $F \in {}^{\omega}\omega$ be an order function. Let $G \in {}^{\omega}\omega$ be an order function with $G(n) \geq 2$ for every n and such that:

$$\forall k\ \forall^{\infty} n\ \ \exp(w_F(\exp(n \cdot k)) < G(n)$$

Let us show that there exists a function $f \in \mathrm{IOE}(F)$ and such that $g \notin \mathrm{IOE}(G)$ for every $g \leq_T f$. The proof is done by forcing, using Lemma 4.7. We first need to argue that under the above hypothesis, the tree $T = {}^{<\omega}F$ is infinitely often G-fat. In what follows, the notation $T\upharpoonright_n$ refers to the finite tree consisting of the nodes of T of length smaller than or equal to n. Let n_1 be the smallest such that $\exp(w_F(\exp(n_1))) < G(n_1)$. The tree $T\upharpoonright_{\exp(n_1)}$ is $F\upharpoonright_{\exp(n_1)}$-full-branching above the empty string and in particular the tree

$T\upharpoonright_{\exp(n_1)}$ is G-fat for (n_1). Suppose now that we have defined $n_1 < \cdots < n_k$ such that $T\upharpoonright_{\exp(n_k)}$ is G-fat for $(n_1, \ldots, n_k)$. Let n_{k+1} be the smallest such that

$$\begin{array}{rrcl} & \exp(n_k) + \exp(n_{k+1} \cdot (k+1)) & < & \exp(n_{k+1} \cdot (k+2)) \\ \text{and} & \exp(w_F(\exp(n_{k+1} \cdot (k+2)))) & < & G(n_{k+1}) \end{array}$$

Then in particular we have

$$\exp(w_F(\exp(n_k) + \exp(n_{k+1} \cdot (k+1)))) < G(n_{k+1})$$

It follows that the tree $T\upharpoonright_{\exp(n_{k+1})}$ is G-fat for $(n_1, \ldots, n_k, n_{k+1})$. Therefore the tree T is infinitely often G-fat for the infinite sequence $\{n_k\}_{1 \le k < \omega}$.

So we start the forcing with the tree $T = {}^{<\omega}F$. Let $\mathcal{P}$ be the set of forcing conditions consisting of all the computable infinitely often G-fat subtrees of T. For two forcing conditions $P_1, P_2 \in \mathcal{P}$, the partial order $P_2 \preceq P_1$ is defined by $P_2 \subseteq P_1$. Let Φ be a functional. By Lemma 4.5, the set of infinitely often G-fat trees $P \in \mathcal{P}$ such that for every path X of $[P]$ we have $\Phi(X) \notin \mathrm{IOE}(G)$, is dense in $\mathcal{P}$.

We simply have to argue that for any computable function $f < F$, the set of infinitely often G-fat trees $P \in \mathcal{P}$ such that every path X of $[P]$ equals at least once to f, is dense in $\mathcal{P}$. It is clear, because given a tree $P \in \mathcal{P}$, consider any node $\tau \in P$ of length m such that P is $F\upharpoonright_{m+1}$-full-branching above τ. Let τ' equals $\tau^\frown f(m+1)$. Note that $\tau' \in P$. Now let P' to be the nodes of P which are compatible with τ'. It is clear that $P' \in \mathcal{P}$ and that $P' \preceq P$. Thus the set of infinitely often G-fat trees $P \in \mathcal{P}$ such that every path X of P equals at least once to f, is dense in $\mathcal{P}$.

Consider now any sufficiently generic set of conditions $\{P_n\}_{n\in\omega}$ with $P_1 \succ P_2 \succ \ldots$. We have that $\bigcap_n P_n$ contains at least one infinite path X. Also this path necessarily equals at least once every computable function bounded by F, and thus equals infinitely often every computable function bounded by F. It follows that $X \in \mathrm{IOE}(F)$. Furthermore for any function Φ we have that $\Phi(X) \notin \mathrm{IOE}(G)$. This shows the theorem. □

We now turn to the proof of Lemma 4.7, which we restate here for convenience: *Let $F \in {}^{\omega}\omega$ be an order function. Let $G \in {}^{\omega}\omega$ be an order function such that $G(n) \ge 2$ for every n. Let $T \subseteq {}^{<\omega}F$ be a computable infinitely often G-fat tree with no dead ends. Let Φ be a functional. Then there exists a computable infinitely often G-fat tree $T' \subseteq T$ with no dead ends, and a computable function $g < G$ such that for every path $X \in [T']$ for which $\Phi(X)$ is total, we have $\Phi(X, n) \neq g(n)$ for every n.*

The reader can refer to Figure 1 which illustrate a part of the proof.

Proof of Lemma 4.7. Suppose first that there exists a node $\sigma \in T$ such that for every $X \succ \sigma$ with $X \in [T]$, we have that $\Phi(X)$ is partial. Then we define the computable tree T' to be the nodes of T compatible with σ. It is clear that T' is infinitely often G-fat. Also as $\Phi(X)$ is partial for every $X \in [T']$ the lemma is verified.

So we can now suppose without loss of generality that for every node $\sigma \in T$ and every n, there exists an extension $\tau \succeq \sigma$ such that $\Phi(\tau, n)$ is defined.

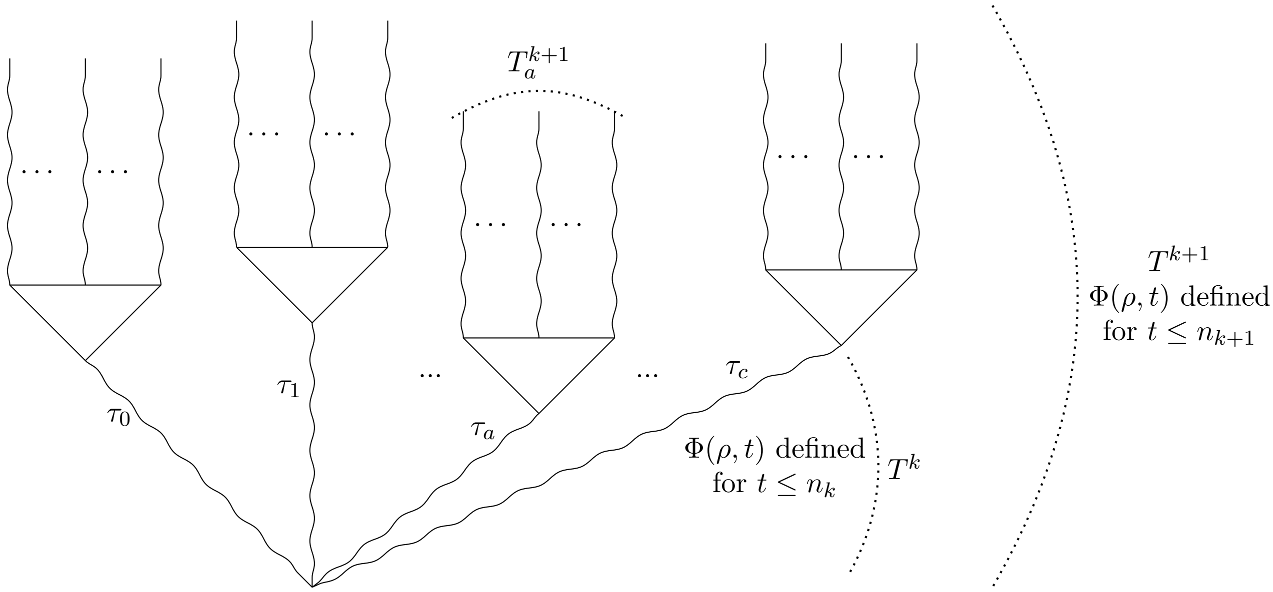

Figure 1. Construction of T_{k+1} from T_k in Lemma 4.7. We have $2^c < G(n_k)$ and for every $n_k < t \le n_{k+1}$, the a-th bit of $\Phi(\rho, t)$ is the same for every $\rho \in T^{k+1}_a$
.

Now from T we want to find $T' \subset T$ as in the lemma. This is done step by step. At each step k we find values $n_1 < \cdots < n_k$ and a finite tree $T_k \supseteq T_{k-1}$ such that T_k is G-fat for $(n_1 < n_2 < \cdots < n_k)$ and such that for leaves ρ of T_k, the values $\Phi(\rho, e)$ are all different from something smaller than $G(e)$ for every $e \le n_k$. However, we do not show right away that the values $\Phi(\rho, e)$ are all different from something smaller than $G(e)$. We first show that we can make large group of leaves that all agrees on a specific bit. The fact that we can use that to have the values $\Phi(\rho, e)$ all different from something smaller than $G(e)$ will be made clear later. Here is a claim which says how one step is done : building the tree T_k from the tree T_{k-1}.

Claim 4.8. *Let $T \subseteq {}^{<\omega}F$ be a computable infinitely often G-fat tree. Let $n_1 < n_2 < \cdots < n_{k-1}$. Suppose that a finite tree $T_{k-1} \subseteq T$ is G-fat for $(n_1 < n_2 < \cdots < n_{k-1})$. Let $\sigma_0, \ldots, \sigma_c$ be the leaves of T_{k-1}. Then there exists n_k such that above each node σ_a for $0 \le a \le c$, we can find an extension $\tau_a \succeq \sigma_a$ of length m_a and a finite tree $T^a \subseteq T$ whose nodes are all comparable with τ_a and such that:*

1. *T^a is $F\restriction_{m_a+\exp(n_k \cdot k)}$-full-branching above τ_a.*
2. *For every e with $n_{k-1} < e \le n_k$ and every leaf $\rho \in T^a$, the value $\Phi(\rho, e)$ is defined.*
3. *For every e with $n_{k-1} < e \le n_k$, there exists $i \in \{0, 1\}$ such that for every leaf $\rho \in T^a$, the a-th bit of $\Phi(\rho, e)$ equals i.*
4. *$\exp(w_F(m_a + \exp(n_k \cdot k))) < G(n_k)$.*

In particular, letting $T_k = \bigcup_{a<c} T^a$, we have that $T_k \subseteq T$ is G-fat for $(n_1 < n_2 < \cdots < n_k)$.

We first show how to use this claim in order to build the tree T' and the computable function g of the lemma. At step 1 we apply the claim starting from the empty tree, with the empty string as the only leaf. The claim gives us some $n_1 > 0$ and a finite subtree $T_1 \subseteq T$ which is G-fat for (n_1) and such that for every $e \leq n_1$, the first bit of $\Phi(e, \rho)$ is the same for every leaf ρ of T_1. We define in the mean time the computable function $g(e)$ for $e \leq n_1$ so that its first bit is different from the one forced on leaves of T_1. Note that $g(e) \in \{0, 1\}$ and that as $G(e) \geq 2$ we necessarily have $g(e) < G(e)$ for $e \leq n_1$. We now deal with a crucial point for the rest of the induction, corresponding to the point (d), in the proof that defeats only one functional. As T_1 is G-fat for (n_1), there exists a node $\tau_1 \in T_1$ of length m_1 such that T_1 is $F{\upharpoonright}_{m_1+\exp(n_1)}$-full-branching above τ_1 and such that $\exp(w_F(m_1 + (\exp(n_1)))) < G(n_1)$ (using (4) of he claim). Let c be the number of leaves of T_1. Note that $w_F(m_1 + (\exp(n_1))$ is the number of nodes in the $F{\upharpoonright}_{m_1+\exp(n_1)}$-full-branching tree above the empty string. As c is the number of nodes in the $F{\upharpoonright}_{m_1+\exp(n_1)}$-full-branching tree above τ_1, it follows that $c \leq w_F(m_1 + \exp(n_1))$ and then that $\exp(c) < G(n_1)$. Just as in the proof that defeats only one functional, this will allow us to continue the induction and in particular to have values smaller than $G(n_1 + t)$ for which we can continue to define g.

Suppose now by induction that at step k we have a sequence $n_1 < \cdots < n_k$ and a finite tree $T_k \subseteq T$ which is G-fat for $(n_1, \ldots, n_k)$. Let $\sigma_0, \ldots, \sigma_c$ be the leaves of T_k and suppose also that c is such that $\exp(c) < G(n_k)$. Let us define $n_{k+1} > n_k$ and a finite tree T_{k+1}, G-fat for $(n_1, \ldots, n_k, n_{k+1})$, with $T_k \subset T_{k+1} \subset T$, together with values $g(e)$ for $n_k < e \leq n_{k+1}$ such that $g(e) < G(e)$ and such that $g(e)$ is different from $\Phi(e, \rho)$ for every leaf ρ of T_{k+1}. Using the above claim, we find $n_{k+1} > n_k$ and above each node σ_a for $a \leq c$ we find an extension $\tau_a \succeq \sigma_a$ of length m_a and a finite tree $T^a_{k+1} \subseteq T$ such that T^a_{k+1} is $F{\upharpoonright}_{m_a+\exp(n_{k+1}\cdot(k+1))}$-full-branching above τ_a. Also for every e with $n_k < e \leq n_{k+1}$, the a-th bit of $\Phi(e, \rho)$ is the same for every leaf ρ of T^a_{k+1}. We can use that to define the values of $g(e)$ for $n_k < e \leq n_{k+1}$ the following way: if the a-th bit of $\Phi(e, \rho)$ is 0 for every leaf ρ of T^a_{k+1}, then the a-th bit of g is set to 1, and vice-versa. This is here that we need to use the induction hypothesis $\exp(c) < G(n_k)$. It implies in particular that $\exp(c) < G(e)$ for $n_k < e \leq n_{k+1}$ (as G is an order function). Also at most c bits of $g(e)$ are set to something, which implies $g(e) \leq \exp(c)$ and thus $g(e) < G(e)$.

Let now $T_{k+1} = \bigcup_{a<c} T^a_{k+1}$. It is clear that T_{k+1} is G-fat for $(n_0, \ldots, n_{k+1})$. Let d be the number of leaves of T_{k+1}. All we need to show now to continue the induction is that $\exp(d) < G(n_{k+1})$. To see this, let $b \leq c$ be such that $m_b \geq m_a$ for $a \leq c$. We know by (4) of the claim that $\exp(w_F(m_b + \exp(n_{k+1} \cdot (k+1)))) < G(n_{k+1})$. Also $w_F(m_b + \exp(n_{k+1} \cdot (k+1)))$ is the number of nodes in the $F{\upharpoonright}_{m_b+\exp(n_{k+1}\cdot(k+1))}$-full-branching tree above the empty string. And by the choice of m_b, for every $a \leq c$ we have that the tree T^a_{k+1} is included in the $F{\upharpoonright}_{m_b+(\exp(n_{k+1}\cdot(k+1))}$-full-branching tree above the empty string. It follows that for d the number of leaves in T_{k+1}, we must have $d \leq w_F(m_b + \exp(n_{k+1} \cdot (k+1)))$ and thus that we must have $\exp(d) < G(n_{k+1})$.

The tree T' is then defined to be $\bigcup_k T_k$. It is clear that by construction, the tree T' is computable with no dead ends, infinitely often G-fat, and that for every path $X \in [T']$, we have $\Phi(X, n) \neq g(n)$ for every e.

Let us now give the proof of the claim. The reader can refer to Figure 2 which illustrate a part of the proof. By hypothesis T is infinitely often G-fat. In particular, there exists $n_k > n_{k-1}$ such that above every σ_a, we have $m'_a \in \omega$ and an extension $\tau'_a \succ \sigma_a$ of length m'_a, such that T is $F\restriction_{m'_a+\exp(n_k\cdot(k+1))}$-full-branching above τ'_a with

$$\exp(w_F(m'_a + \exp(n_k \cdot (k+1))) < G(n_k)$$

Note that here, we truly mean $\exp(n_k \cdot (k+1))$ and not $\exp(n_k \cdot k)$. Given τ'_a of length m'_a, for each node of T of length $m'_a + \exp(n_k \cdot (k+1))$ extending τ'_a, we find an extension $\rho \in T$ of this node such that the values $\Phi(\rho, t)$ are defined for every t with $n_{k-1} < t \leq n_k$. We define the tree $T^{a\prime}$ to be all these nodes ρ and their prefixes. We now inductively apply Lemma 4.5 to the tree $T^{a\prime}$, so that for every t with $n_{k-1} < t \leq n_k$, the a-th bit of $\Phi(t, \rho)$ is the same on every leaf ρ of $T^{a\prime}$. Let us explain the first step. Given $T^{a\prime}$, we partition its leaves into these for which the a-th bit of $\Phi(n_{k-1}+1, \rho)$ is 0, and these for which the a-th bit of $\Phi(n_{k-1}+1, \rho)$ is 1. We then thin the tree $T^{a\prime}$ as described in Lemma 4.5, so that the height of the full-branching part of $T^{a\prime}$ is divided by 2, and the a-th bit of $\Phi(n_{k-1}+1, \rho)$ is the same for all the remaining leaves. We then inductively apply Lemma 4.5 on the successive resulting trees, to deal with the a-th bit of all the values $\Phi(t, \rho)$ for $n_{k-1} < t \leq n_k$. Let T^a be the tree resulting of the successive applications of Lemma 4.5.

It is clear by design that (2) and (3) are verified. Let us show that (1) is verified. Each each time we applied Lemma 4.5, it divided per 2 the height of the full-branching part of $T^{a\prime}$. We applied Lemma 4.5 at most n_k times. Also $T^{a\prime}$ is $F\restriction_{m'_a+\exp(n_k\cdot(k+1))}$-full-branching above τ'_a. It means in particular that its full-branching part has height $\exp(n_k \cdot (k+1))$. It follows that the full-branching part of T^a has height at least $\exp(n_k \cdot (k+1))\exp(-n_k) = \exp(n_k \cdot k)$. Thus we have that T^a is $F\restriction_{m_a+\exp(n_k\cdot k)}$-full branching above some node $\tau_a \succeq \tau'_a$ of length m_a. Thus also (1) is verified.

It remains to verify (4). Recall that n_k and (for every a) the string τ'_a of length m'_a were picked such that

$$\exp(w_F(m'_a + (\exp(n_k \cdot (k+1)))) < G(n_k)$$

In order to verify (4), we now want to show for every a that:

$$\exp(w_F(m_a + (\exp(n_k \cdot k))) < G(n_k)$$

It suffices to show for every a that $m_a + \exp(n_k \cdot k) \leq m'_a + \exp(n_k \cdot (k+1))$. Recall that m_a is the length of the string τ_a extending τ'_a, resulting of the successive applications of Lemma 4.5 to the full-branching part of $T^{a\prime}$. In particular we have $\tau_a \in T^{a\prime}$. Also the quantities $\exp(n_k \cdot (k+1))$ and $\exp(n_k \cdot k)$ are respectively the height of the full-branching part of $T^{a\prime}$ and the height of the full-branching part of $T^a \subseteq T^{a\prime}$. It easily follows that $m_a + \exp(n_k \cdot k) \leq m'_a + \exp(n_k \cdot (k+1))$.

□

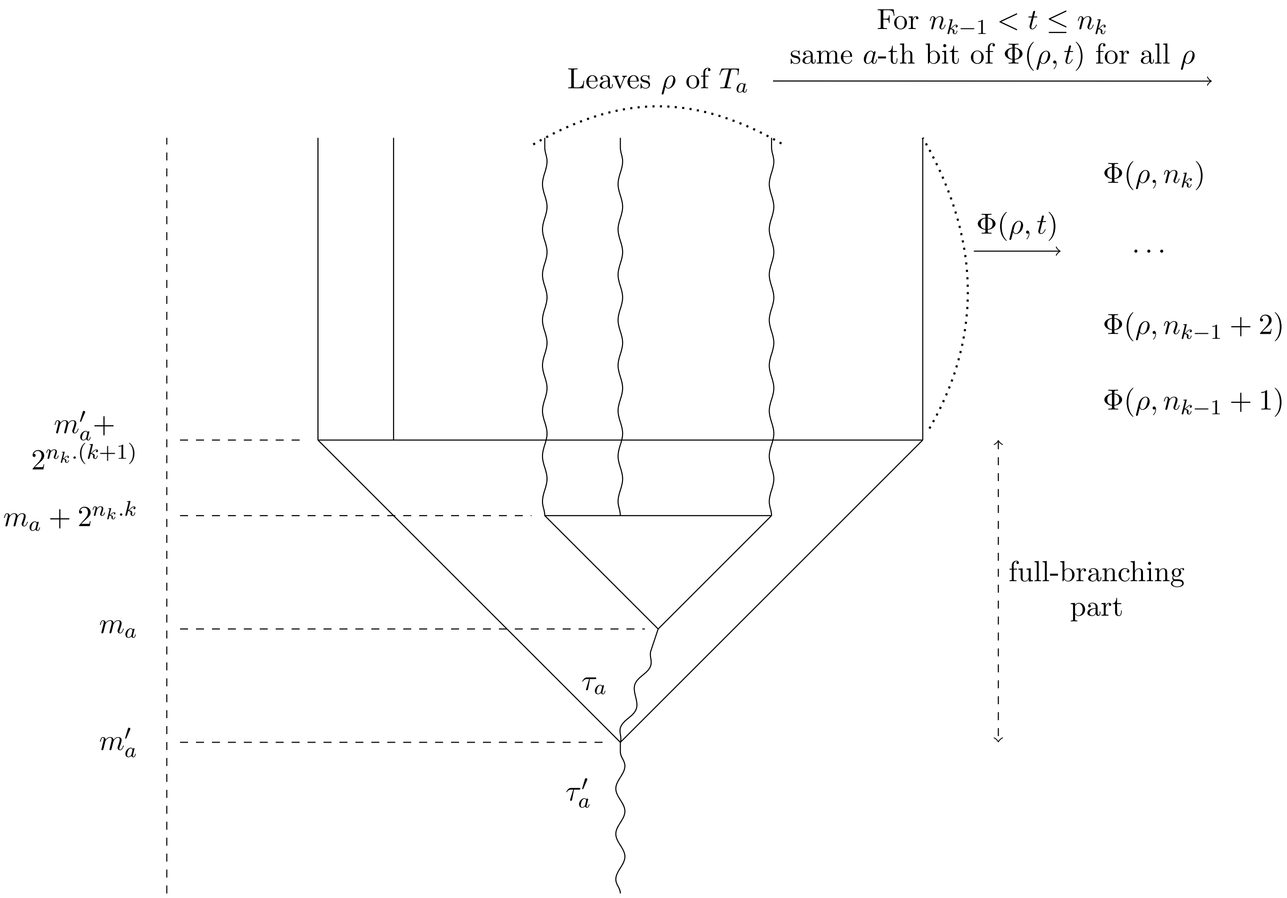

FIGURE 2. Construction of T_a in the claim of Lemma 4.7. T_a is the result of thinning the full-branching part above τ'_a into a full-branching part above a extension $\tau_a \succeq \tau'_a$
.

5. Analog of Theorem 3.4 for cardinal characteristics

As before let $R \subseteq X \cdot Y$ be a relation between spaces X, Y; we also assume now that $\forall x \, \exists y \, (xRy)$ and $\forall y \, \exists x \, \neg(xRy)$. Let $S = \{\langle y, x\rangle \in Y \cdot X : \neg xRy\}$.

Definition 5.1. We define dual pairs of cardinal characteristics by

$$\mathfrak{d}(R) = \min\{|G| : G \subseteq Y \wedge \forall x \in X \, \exists y \in G \, xRy\}.$$

$$\mathfrak{b}(R) = \mathfrak{d}(S) = \min\{|F| : F \subseteq X \wedge \forall y \in Y \exists x \in F \, \neg xRy\}.$$

Note that compared to Definition 2.1, the defining properties are negated. For a discussion of this, see the beginning of Section 3 of [2].

We obtain the characteristics discussed in the introduction as $\mathfrak{d}(R)$ and $\mathfrak{b}(R)$ for the two types of relations R introduced in Def. 2.2, which we summarise briefly:

For $x \in {}^{\omega}\omega$ and $y \in \Pi_n\{0, \dots, h(n)-1\}$, let $x \neq^*_h y \Leftrightarrow \forall^\infty n \, [x(n) \neq y(n)]$.

For $0 \le p \le 1/2$, for $x, y \in {}^{\omega}2$, let $x \bowtie_p y \Leftrightarrow \underline{\rho}(x \leftrightarrow y) > p$.

It will be convenient for the reader to express Definition 5.1 for these relations in words.

Remark 5.2. $\mathfrak{d}(\neq^*_h)$ is the least size of a set G of h-bounded functions so that for each function x there is a function y in G such that $\forall^\infty n[x(n) \neq y(n)]$. (Of course it suffices to require this for h-bounded x.)

$\mathfrak{b}(\neq^*_h)$ is the least size of a set F of functions such that for each h-bounded function y, there is a function x in F such that $\exists^\infty n\, x(n) = y(n)$. (We can require that each function in F is h-bounded.)

$\mathfrak{d}(\bowtie_p)$, or $\mathfrak{d}(p)$ for short, is the least size of a set G of bit sequences so that for each bit sequence x there is a bit sequence y in G so that $\underline{\rho}(x \leftrightarrow y) > p$.

$\mathfrak{b}(\bowtie_p)$, or $\mathfrak{b}(p)$ for short, is the least size of a set F of bit sequences such that for each bit sequence y, there is a bit sequence x in F such that $\underline{\rho}(x \leftrightarrow y) \leq p$.

Our main goal is to show that $\mathfrak{d}(p) = \mathfrak{d}(\neq^*, 2^{(2^n)})$ and $\mathfrak{b}(p) = \mathfrak{b}(\neq^*, 2^{(2^n)})$ for each $p \in (0, 1/2)$. We begin with some preliminary facts of independent interest. The first lemma amplifies bounds without changing the cardinal characteristics.

Lemma 5.3. (i) *Let h be nondecreasing and $g(n) = h(2n)$. We have $\mathfrak{d}(\neq^*, h) = \mathfrak{d}(\neq^*, g)$ and $\mathfrak{b}(\neq^*, h) = \mathfrak{b}(\neq^*, g)$.*
(ii) *For each $a, b > 1$ we have $\mathfrak{d}(\neq^*, 2^{(a^n)}) = \mathfrak{d}(\neq^*, 2^{(b^n)})$ and $\mathfrak{b}(\neq^*, 2^{(a^n)}) = \mathfrak{b}(\neq^*, 2^{(b^n)})$.*

Proof. (i) Trivially, $h \leq g$ implies that $\mathfrak{d}(\neq^*, h) \geq \mathfrak{d}(\neq^*, g)$ and $\mathfrak{b}(\neq^*, h) \leq \mathfrak{b}(\neq^*, g)$. So it suffices to show two inequalities.

$\boxed{\mathfrak{d}(\neq^*, h) \leq \mathfrak{d}(\neq^*, g):}$ Let G be a witness set for $\mathfrak{d}(\neq^*, g)$. Note that G is also a witness set for $\mathfrak{d}(\neq^*, h(2n+1))$. Let $\widehat{G} = \{p_0 \oplus p_1 \colon p_0, p_1 \in G\}$, where $(p_0 \oplus p_1)(2m + i) = p_i(m)$ for $i = 0, 1$. Each function in $\widehat{G}$ is bounded by h. Since G is infinite, $|\widehat{G}| = |G|$. Clearly $\widehat{G}$ is a witness set for $\mathfrak{d}(\neq^*, h)$.

$\boxed{\mathfrak{b}(\neq^*, h) \geq \mathfrak{b}(\neq^*, g):}$ Let F be a witness set for $\mathfrak{b}(\neq^*, h)$. Let $\widehat{F}$ consist of the functions of the form $n \to p(2n)$, or of the form $n \to p(2n + 1)$, where $p \in F$. Then $|\widehat{F}| = |F|$, and each function in $\widehat{F}$ is g bounded.

Clearly, $\widehat{F}$ is a witness set for $\mathfrak{b}(\neq^*, g)$: if q is g-bounded, then $\widehat{q}$ is h bounded where $\widehat{q}(2n + i) = q(n)$ for $i = 0, 1$. Let $p \in F$ be such that $\exists^\infty k\, p(k) = \widehat{q}(k)$. Let $i \leq 1$ be such that infinitely many such k have parity i. Then the function $n \to p(2n + i)$ which is in $\widehat{F}$ is as required.

(ii) is immediate from (i) by iteration using that $a^{2^i} > b$ and $b^{2^i} > a$ for sufficiently large i. ☐

Lemma 5.4. *Let $a \in \omega - \{0\}$. We have $\mathfrak{d}(\neq^*, 2^{(a^n)}) \leq \mathfrak{d}(1/a)$ and $\mathfrak{b}(\neq^*, 2^{(a^n)}) \geq \mathfrak{b}(1/a)$.*

Proof. As above let I_m for $m \geq 2$ be the $(m-1)$-th consecutive interval of length a^m in $\omega - \{0\}$. First let G be a witness set for $\mathfrak{d}(1/a)$. Let $h(n) = 2^{(a^n)}$. We show that $\widehat{G} = \{K_h(y) \colon y \in G\}$ is a witness set for

$\mathfrak{d}(\neq^*, 2^{(a^n)})$. Otherwise there is a sequence $x \in {}^\omega 2$ such that for each $y \in \widehat{G}$ there are infinitely many m with $x(m) = K_h(y)(m)$. Let $x' = 1 - x$, that is 0s and 1s are interchanged. Then for each $y \in G$, for infinitely many m, $L_h(x')(i) \neq y(i)$ for each $i \in I_m$. If we let $n = 1 + \max I_m$, the proportion of $i < n$ such that $L_h(x)(i) = y'(i)$ is therefore at most $(a^m - 1)/(a^{m+1} - 1)$, which converges to $1/a$ as $m \to \infty$. This contradicts the choice of G.

Now let F be a witness set for $\mathfrak{b}(\neq^*, h)$. Let $\widehat{F} = \{1 - L_h(x) \colon x \in F\}$. For each $y \in {}^\omega 2$ there is $x \in F$ such that $\exists^\infty n\, K_h(y)(n) = x(n)$. This implies $\underline{\rho}(y \leftrightarrow x') \leq 1/a$ where $x' = 1 - L_h(x) \in \widehat{F}$. Hence $\widehat{F}$ is a witness set for $\mathfrak{b}(1/a)$. □

Theorem 5.5. *Fix any $p \in (0, 1/2)$. We have $\mathfrak{d}(p) = \mathfrak{d}(\neq^*, 2^{(2^n)})$ and $\mathfrak{b}(p) = \mathfrak{b}(\neq^*, 2^{(2^n)})$.*

Proof. By the two foregoing lemmas we have $\mathfrak{d}(p) \geq \mathfrak{d}(\neq^*, 2^{(2^n)})$ and $\mathfrak{b}(p) \leq \mathfrak{b}(\neq^*, 2^{(2^n)})$. It remains to show the converse inequalities:
$\mathfrak{d}(p) \leq \mathfrak{d}(\neq^*, 2^{(2^n)})$ and $\mathfrak{b}(p) \geq \mathfrak{b}(\neq^*, 2^{(2^n)})$.

Recall from Definitions 3.5 and 3.6 that for strings x, y of length r,

$$d(x, y) = \frac{1}{r} |\{i \colon x(n)(i) \neq y(n)(i)\}|$$

If h is a function of the form $2^{\widehat{h}}$ with $\widehat{h} \colon \omega \to \omega$, $X = Y = X_h$ denotes the space of h-bounded functions, and $q \in (0, 1/2)$, we defined a relation on $X \cdot Y$ by

$$x \neq^*_{\widehat{h},q} y \Leftrightarrow \forall^\infty n\, [d(x(n), y(n)) \geq q].$$

Claim 5.6. *For each $c \in \omega$ there is $k \in \omega$ such that*

$$\mathfrak{d}(q - 2/c) \leq \mathfrak{d}\langle \neq^*, \lfloor 2^{n/k} \rfloor, q\rangle, \text{ and}$$

$$\mathfrak{b}(q - 2/c) \geq \mathfrak{b}\langle \neq^*, \lfloor 2^{n/k} \rfloor, q\rangle.$$

Proof. Let k be large enough so that $2^{1/k} - 1 < \frac{1}{2c}$. Let $\widehat{h}(n) = \lfloor 2^{n/k} \rfloor$ and $h = 2^{\widehat{h}}$. Write $H(n) = \sum_{r \leq n} \widehat{h}(r)$. Recall from the proof of Claim 3.7 that for sufficiently large n

$$\widehat{h}(n + 1) \leq \frac{1}{c} H(n).$$

For the inequality involving $\mathfrak{d}$, let G be a witness set for $\mathfrak{d}\langle \neq^*, \widehat{h}, q\rangle$. Thus, for each function $x < h$ there is a function $y \in G$ such that for almost all n, $L_h(x), L_h(y)$ disagree on a proportion of q bits of Block n. Let z be the complement of $L_h(y)$. Given m, let n be such that $H(n) \leq m < H(n + 1)$. Since $m - H(n) \leq \frac{1}{c} H(n)$, for large enough m, $L_h(x)$ and z agree up to m on a proportion of at least $q - 1.5/c$ bits. So the set of complements of the $L_h(y)$, $y \in G$, forms a witness set for $\mathfrak{d}(q - 2/c)$ as required.

For the inequality involving $\mathfrak{b}$, let F be a witness set for $\mathfrak{b}(q - 2/c)$. Thus, for each $y \in {}^\omega 2$ there is $x \in F$ such that $\underline{\rho}(y \leftrightarrow x) \leq q - 2/c$. Let $\widehat{F} = \{K_h(1 - x) \colon x \in F\}$. We show that $\widehat{F}$ is a witness set for $\mathfrak{b}\langle \neq^*, \lfloor 2^{n/k} \rfloor, q\rangle$.

Give a function $y < h$, let $y' = L_h(y)$. There is $x \in F$ such that $\underline{\rho}(y' \leftrightarrow x) \leq q - 2/c$, and hence $\overline{\rho}(y' \leftrightarrow x') \geq 1 - q + 2/c$ where $x' = 1 - x$ is the complement and $\overline{\rho}$ denotes the upper density. Then there are infinitely many m such that the strings $y' \restriction_m$ and $x' \restriction_m$ agree on a proportion of $> 1-q+1/c$ bits. Suppose that $H(n) \le m < H(n+1)$, then the contribution of disagreement of Block n is at most $1/c$. So there are infinitely many k so that in Block k, y' and x' agree on a proportion of more than $1-q$ bits, and hence disagree on a proportion of fewer than q bits. $\square_{5.6}$

In the following recall Definition 3.9, and in particular that for $L \in \omega$ and a function u, for any L-slalom s and function $y < u$,

$$s \not\ni^*_{u,L} y \Leftrightarrow \forall^\infty n[s(n) \not\ni y(n)].$$

We also write $\langle \not\ni^*, u, L\rangle$ for this relation.

Claim 5.7. *Given $q < 1/2$, let L, ϵ be as in Theorem 3.10. Fix a nondecreasing function $\widehat{h}$, and let $u(n) = 2^{\lfloor \epsilon \widehat{h}(n)\rfloor}$. We have*

$$\mathfrak{d}\langle \neq^*, \widehat{h}, q\rangle \le \mathfrak{d}\langle \not\ni^*, u, L\rangle \text{ and } \mathfrak{b}\langle \neq^*, \widehat{h}, q\rangle \ge \mathfrak{b}\langle \not\ni^*, u, L\rangle.$$

Proof. For the inequality involving $\mathfrak{d}$, let G be a set of functions bounded by u such that $|G| < \mathfrak{d}\langle \neq^*, \widehat{h}, q\rangle$. We show that G is not a witness set for the right hand side $\mathfrak{d}\langle \not\ni^*, u, L\rangle$.

For each r of the form $\widehat{h}(n)$ choose a set $C = C_r$ as in Theorem 3.10. Since $|C_r| = 2^{\lfloor \epsilon r\rfloor}$ we may choose a sequence $\langle \sigma_i^r\rangle_{i<2^{\lfloor \epsilon r\rfloor}}$ listing C_r without repetitions. For a function $y < u$ let $\widetilde{y}$ be the function given by $\widetilde{y}(n) = \sigma^{\widehat{h}(n)}_{y(n)}$. (Thus, $\widetilde{y}(n)$ is a binary string of length $\widehat{h}(n)$.) Let $\widetilde{G} = \{\widetilde{y}\colon\ y \in G\}$. Then $|\widetilde{G}| \le |G| < \mathfrak{d}\langle \neq^*, \widehat{h}, q\rangle$. So there is a function x with $x(n) \in {}^{\widehat{h}(n)}2$ for each n such that for each $\widetilde{y} \in \widetilde{G}$ we have $\exists^\infty n\,[d(x(n), \widetilde{y}(n)) < q]$. Let s be the slalom given by

$$s(n) = \{i\colon\ d(x(n), \sigma_i^{\widehat{h}(n)}) < q\}.$$

Note that by the choice of the C_r according to Theorem 3.10 and since the listing of C_r has no repetitions, s is an L-slalom. By definition, $\max s(n) < u(n)$. So, for each $y \in G$ we have $\exists^\infty n\,[s(n) \ni y(n)]$. Hence G is not a witness set for $\mathfrak{d}\langle \not\ni^*, u, L\rangle$.

For the inequality involving $\mathfrak{b}$, suppose F is a witness set for $\mathfrak{b}\langle \neq^*, \widehat{h}, q\rangle$. That is, for each $h = 2^{\widehat{h}}$-bounded function y, there is $x \in F$ such that

$$\exists^\infty n\,[d(x(n), y(n)) < q]$$

(as usual we view $x(n), y(n)$ as binary strings of length $\widehat{h}(n)$). For $x \in F$ let s_x be the L-slalom such that

$$s_x(n) = \{i < u(n)\colon\ d(\sigma_i^{\widehat{h}(n)}, x(n)) < q\}.$$

Let $\widehat{F} = \{s_x\colon\ x \in F\}$. Given an u-bounded function y, let $\widetilde{y}(n) = \sigma^{\widehat{h}(n)}_{y(n)}$. There is $x \in F$ such that $d(x(n), \widetilde{y}(n)) < q$ for infinitely many n. This means that $y(n) \in s_x(n)$. Hence $\widehat{F}$ is a witness set for $\mathfrak{b}\langle \not\ni^*, u, L\rangle$. $\square_{5.7}$

We next need an amplification tool in the context of slaloms. The proof is almost verbatim the one in Lemma 5.3(i), so we omit it.

Claim 5.8. *Let $L \in \omega$, let the function u be nondecreasing and let $w(n) = u(2n)$. We have $\mathfrak{d}(\langle \not\ni^*, u, L\rangle = \mathfrak{d}\langle \not\ni^*, w, L\rangle$ and $\mathfrak{b}(\langle \not\ni^*, u, L\rangle = \mathfrak{b}\langle \not\ni^*, w, L\rangle$.*

Iterating the claim, starting with the function $\widehat{h}(n) = \lfloor 2^{n/k} \rfloor$ with k as in Claim 5.6, we obtain that $\mathfrak{d}\langle \not\ni^*, 2^{\widehat{h}}, L\rangle = \mathfrak{d}\langle \not\ni^*, 2^{(L2^n)}, L\rangle$, and similarly $\mathfrak{b}\langle \not\ni^*, 2^{\widehat{h}}, L\rangle = \mathfrak{b}\langle \not\ni^*, 2^{(L2^n)}, L\rangle$. It remains to verify the following.

Claim 5.9.

$$\mathfrak{d}\langle \not\ni^*, 2^{(L2^n)}, L\rangle \leq \mathfrak{d}(\neq^*, 2^{(2^n)}) \text{ and } \mathfrak{b}\langle \not\ni^*, 2^{(L2^n)}, L\rangle \geq \mathfrak{b}(\neq^*, 2^{(2^n)}).$$

Proof. Given n, we write a number $k < 2^{(L2^n)}$ in binary with leading zeros if necessary, and so can view k as a binary string of length $L2^n$. We view such a string as consisting of L consecutive blocks of length 2^n.

For the inequality involving $\mathfrak{d}$, let G be a witness set for $\mathfrak{d}(\neq^*, 2^{(2^n)})$. For functions $y_1, \ldots, y_L$ such that $y_i(n) < 2^{(2^n)}$ for each n, let $(y_1, \ldots, y_L)$ denote the function y with $y(n) < 2^{(L2^n)}$ for each n such that the i-th block of $y(n)$ equals $y_i(n)$ for each i with $1 \leq i \leq L$. Let

$$\widehat{G} = \{(y_1, \ldots, y_n) \colon y_1, \ldots, y_L \in G\}.$$

Since G is infinite we have $|\widehat{G}| = |G|$. We check that $\widehat{G}$ is a witness set for the left hand side $\mathfrak{d}\langle \not\ni^*, 2^{(L2^n)}\rangle$. Given an L-slalom s bounded by $2^{(L2^n)}$ we may assume that $s(n)$ has exactly L members, and they are binary strings of length $L2^n$. For $i \leq L$ let $x_i(n)$ be the i-th block of the i-th string in $s(n)$, so that $|x_i(n)| = 2^n$. Viewing the x_i as functions bounded by $2^{(2^n)}$, we can choose $y_1, \ldots, y_L \in G$ such that $\forall^\infty n\, x_i(n) \neq y_i(n)$. Let $y = (y_1, \ldots, y_n) \in \widehat{G}$. Then $\forall^\infty n\, [s(n) \not\ni y(n)]$, as required.

For the inequality involving $\mathfrak{b}$ let F be a witness set for $\mathfrak{b}\langle \not\ni^*, 2^{(L2^n)}$. That is, F is a set of L-slaloms s such that for each function y with $y(n) < 2^{(L2^n)}$, there is $s \in F$ such that $s(n) \ni y(n)$ for infinitely many n.

Let $\widehat{F}$ be the set of functions s_i, for $s \in F$ and $i < L$, such that $s_i(n)$ is the i-th block of the i-th element of $s(n)$ (as before we may assume that each string in $s(n)$ has length $L2^n$). Now let y be a given function bounded by $2^{(2^n)}$. Let y' be the function bounded by $2^{(L2^n)}$ such that for each n, each block of $y'(n)$ equals $y(n)$. There is $s \in F$ such that $s(n) \ni y'(n)$ for infinitely many n. There is $i < L$ such that $y'(n)$ is the i-th string in $s(n)$ for infinitely many of these n, and hence $y(n) = s_i(n)$. Thus $\widehat{F}$ is a witness set for $\mathfrak{b}(\neq^*, 2^{(2^n)})$. $\square_{5.9}$

We can now summarise the argument that $\mathfrak{d}(p) \leq \mathfrak{d}(\neq^*, 2^{(2^n)})$. Pick c large enough such that $q = p + 2/c < 1/2$.

By Claim 5.6 there is k such that

$$\mathfrak{d}(p) \leq \mathfrak{d}\langle \neq^*, \lfloor 2^{n/k} \rfloor, q\rangle.$$

By Claim 5.7 there are L, ϵ such that where $\widehat{h}(n) = \lfloor 2^{n/k} \rfloor$, we have

$$\mathfrak{d}\langle \neq^*, \widehat{h}, q\rangle \leq \mathfrak{d}\langle \not\ni^*, u, L\rangle,$$

where $u(n) = 2^{\lfloor \epsilon \widehat{h}(n) \rfloor}$.

Applying Claim 5.8 sufficiently many times we have

$$\mathfrak{d}\langle \not\ni^*, u, L\rangle \leq \mathfrak{d}\langle \not\ni^*, 2^{(L2^n)}, L\rangle.$$

Finally, $\mathfrak{d}\langle\not\ni^*, 2^{(L2^n)}, L\rangle \leq \mathfrak{d}(\neq^*, 2^{(2^n)})$ by Claim 5.9.

The argument for $\mathfrak{b}(p) \geq \mathfrak{b}(\neq^*, 2^{(2^n)})$ is dual to the above. □

6. Future directions

There are still questions regarding the possible Γ values. Theorem 3.4 implies that there are no Γ values between 0 and 1/2. But given that $\Gamma(X) < 1/2$, it does not provide a single element $Y \leq_T X$ such that $\Gamma(Y) = 0$.

Question 6.1. *Let X be a set. Suppose $\Gamma(X) = 0$. Is there always an element $Y \leq_T X$ such that $\gamma(Y) = 0$?*

This question is actually connected to other questions regarding the hierarchy of classes $\mathrm{IOE}(h)$ in the Munchnik degrees. We showed with Theorem 4.3 that this hierarchy is proper, but given f, the function $g > f$ such that $\mathrm{IOE}(f) <_W \mathrm{IOE}(g)$ is rather large. We do not know for instance if given any f we have $\mathrm{IOE}(f) <_W \mathrm{IOE}(n \mapsto f(n^2))$. So we ask here the following question:

Question 6.2. *Does there existe a computable function f such that $\forall a \in \omega\ \forall^\infty n\ f(n) > 2^{(a^n)}$ and such that:*

$$\mathrm{IOE}(2^{(2^n)}) \geq_W \mathrm{IOE}(f)$$

A positive answer to this question would also provide a positive answer to Question 6.1.

There are also questions regarding the sets X such that $\Gamma(X) = 1/2$.

Question 6.3. *Let X be a set. Suppose $\Gamma(X) = 1/2$. Let $Y \leq_T X$. Is there always a computable set A such that $\underline{\rho}(Y \leftrightarrow A) = 1/2$?*

Again, the proof of Theorem 3.4 does not help answering this question. All we know is that we have an affirmative answer to the question for all the known examples of sets X with a Γ value of 1/2.

References

[1] Uri Andrews, Mingzhong Cai, David Diamondstone, Carl Jockusch, and Steffen Lempp. Asymptotic density, computable traceability, and 1-randomness. *Fundamenta Mathematicae*, 234(1):41–53, 2016.
[2] J. Brendle, A. Brooke-Taylor, Keng Meng Ng, and A. Nies. An analogy between cardinal characteristics and highness properties of oracles. In *Proceedings of the 13th Asian Logic Conference: Guangzhou, China*, pages 1–28. World Scientific, 2013. http://arxiv.org/abs/1404.2839.
[3] A. Nies (editor). Logic Blog 2015. Available at http://arxiv.org/abs/1602.04432, 2015.
[4] A. Nies (editor). Logic Blog 2016. Available at cs.auckland.ac.nz/~nies, 2016.
[5] A. Nies (editor). Logic Forum 2017. Available at cs.auckland.ac.nz/~nies, 2017.
[6] Peter Elias. Error-correcting codes for list decoding. *Information Theory, IEEE Transactions on*, 37(1):5–12, 1991.
[7] Denis R Hirschfeldt, Carl G Jockusch Jr, Timothy H McNicholl, and Paul E Schupp. Asymptotic density and the coarse computability bound. *Computability*, 5(1):13–27, 2016.

[8] C. Jockusch, Jr. Degrees of functions with no fixed points. In *Logic, methodology and philosophy of science, VIII (Moscow, 1987)*, volume 126 of *Stud. Logic Found. Math.*, pages 191–201. North-Holland, Amsterdam, 1989.

[9] M. Khan and J. Miller. Forcing with bushy trees. *Bulletin of Symbolic Logic*, 23(2):160–180, 2017.

[10] Mushfeq Khan and Joseph S Miller. Forcing with bushy trees. *Bulletin of Symbolic Logic*, 23(2):160–180, 2017.

[11] B. Kjos-Hanssen, W. Merkle, and F. Stephan. Kolmogorov complexity and the Recursion Theorem. *Transactions of the American Mathematical Society*, 363(10):5465–5480, 2011.

[12] S. Kurtz. *Randomness and genericity in the degrees of unsolvability*. Ph.D. Dissertation, University of Illinois, Urbana, 1981.

[13] Benoit Monin. Asymptotic density and error-correcting codes.

[14] Benoit Monin and André Nies. A unifying approach to the gamma question. In *Logic in Computer Science (LICS), 2015 30th Annual ACM/IEEE Symposium on*, pages 585–596. IEEE, 2015.

[15] A. Nies. Lowness, randomness, and computable analysis. In *Computability and Complexity - Essays Dedicated to Rodney G. Downey on the Occasion of His 60th Birthday*, pages 738–754, 2017.

[16] N. Rupprecht. Relativized Schnorr tests with universal behavior. *Arch. Math. Logic*, 49(5):555–570, 2010.

[17] Nicholas Rupprecht. *Effective correspondents to cardinal characteristics in Cichoń's diagram*. PhD thesis, University of Michigan, 2010.