

# EFFECTIVIZING LUSIN'S THEOREM

RUSSELL MILLER

ABSTRACT. Lusin's Theorem states that, for every Borel-measurable function  $\mathbf{f}$  on  $\mathbb{R}$  and every  $\varepsilon > 0$ , there exists a continuous function  $\mathbf{g}$  on  $\mathbb{R}$  which is equal to  $\mathbf{f}$  except on a set of measure  $< \varepsilon$ . We give a proof of this result using computability theory, relating it to the near-uniformity of the Turing jump operator, and use this proof to derive several uniform computable versions. Easier results, which we prove by the same methods, include versions of Lusin's Theorem with Baire category in place of Lebesgue measure and also with Cantor space  $2^{\mathbb{N}}$  in place of  $\mathbb{R}$ . The distinct processes showing generalized lowness for generic sets and for a set of full measure are seen to explain the differences between versions of Lusin's Theorem.

## 1. INTRODUCTION

Lusin's Theorem, first proven by Nikolay Lusin (or Luzin) in 1912, states informally that every measurable function on the real numbers  $\mathbb{R}$  is “nearly continuous,” in terms of the Lebesgue measure  $\mu$  on  $\mathbb{R}$ . The standard formal statement is as follows.

**Theorem 1.1** (Lusin's Theorem, 1912). *For every Borel-measurable function  $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}$  and every  $\varepsilon > 0$ , there exists a continuous function  $\mathbf{g} : \mathbb{R} \rightarrow \mathbb{R}$  such that*

$$\mu(\{\mathbf{x} \in \mathbb{R} : \mathbf{f}(\mathbf{x}) \neq \mathbf{g}(\mathbf{x})\}) < \varepsilon.$$

Alternative versions allow  $\pm\infty$  as values of the functions in question. (Some versions only state that  $\mathbf{f}$  restricts to a continuous function on a set of comeasure  $< \varepsilon$ , but we will consider the stronger version.) A common method of proof of this result involves Egorov's Theorem, that continuity of a function on a compact set is “nearly” uniform continuity. Indeed, in the standard text [4], Lusin's Theorem is posed as an exercise, following the exposition of Egorov's Theorem. On the other hand, [5, Thm. 2.24] proves Lusin's Theorem from basic principles (and later poses Egorov's Theorem as an exercise).

The purpose of this article is to show that Lusin's Theorem is a direct consequence of the fact, well known in computability theory, that for almost all sets  $X$  of natural

---

The author was partially supported by Grant #DMS – 1362206 from the National Science Foundation, Grant #581896 from the Simons Foundation, and several grants from the PSC-CUNY Research Award Program. This work was initiated at a workshop held at the Schloß Dagstuhl in February 2017, where Vasco Brattka aided its development with useful conversations. It was subsequently improved by several suggestions from an anonymous referee.

numbers, the jump  $X'$  of  $X$  satisfies the property

$$X' \leq_T \emptyset' \oplus X,$$

i.e., the jump, which is essentially the Halting Problem relativized to  $X$ , can be computed merely from the Halting Problem  $\emptyset'$  along with  $X$  itself. (The *join*  $\emptyset' \oplus X = \{2n : n \in \emptyset'\} \cup \{2n+1 : n \in X\}$  is a least upper bound of  $\emptyset'$  and  $X$  under  $\leq_T$ .) Certain sets  $X$  fail to satisfy this property, but those sets form a meager subclass of measure 0 within the Cantor space  $2^{\mathbb{N}}$ . The computability-theoretic reason why Lusin's Theorem needs to use an  $\varepsilon > 0$  is that the Turing reduction from  $X'$  to  $\emptyset' \oplus X$  is only uniform up to a set of measure  $\varepsilon$ . In contrast, when we use Baire category theory, comeager-many  $X \in 2^{\mathbb{N}}$  satisfy  $X' \leq_T \emptyset' \oplus X$ , and here a single Turing reduction succeeds uniformly on the entire class. This accounts for the companion theorem in analysis, which is much easier to prove and requires no  $\varepsilon$ -fudging: that every Borel function  $\mathbf{f}$  restricts to a continuous function whose domain is a comeager subset of  $\mathbb{R}$ .

Our proof of Lusin's Theorem demands more background in computability and descriptive set theory than the standard proofs. The point is not to replace the original proofs, but rather to highlight the connections between Lusin's Theorem and computability theory. When given in full, our proof will require substantial attention to detail, but it can be summarized very neatly as the following sequence of steps.

- (1) As shown in [8], the continuous functions  $\mathbf{g} : \mathbb{R} \rightarrow \mathbb{R}$  are precisely those for which there is a Turing functional  $\Gamma$  and an oracle  $S \subseteq \mathbb{N}$  such that, for all Cauchy sequences  $X$  converging fast to any  $\mathbf{x}$ ,  $\Gamma^{S \oplus X}$  converges fast to  $\mathbf{g}(\mathbf{x})$ .
- (2) Borel-measurable functions  $\mathbf{f}(\mathbf{x})$  are those which can be described (in computable analysis) by the action of a Turing functional  $\Phi$  whose oracle is the  $\alpha$ -th jump of the input  $\mathbf{x}$ , for some countable ordinal  $\alpha$  that depends on  $\mathbf{f}$ , along with an oracle set  $S \subseteq \mathbb{N}$ . That is, if  $X$  is a representation (normally by a fast-converging Cauchy sequence) of  $\mathbf{x}$ , then

$$\Phi((S \oplus X)^{(\alpha)})$$

computes a Cauchy sequence which converges fast to  $\mathbf{f}(\mathbf{x})$ . The ordinal  $\alpha$  gives the level of  $\mathbf{f}$  in the Borel hierarchy (sometimes denoted by  $F_\sigma$ ,  $G_{\delta\sigma}$ , etc.; also by  $\Sigma_\alpha$  or  $\Pi_\alpha$ ). This generalizes the preceding item, where  $\alpha = 0$ ,

- (3) For each fixed ordinal  $\alpha < \omega_1$ , almost all subsets  $X \subseteq \mathbb{N}$  have the property that the  $\alpha$ -th jump  $X^{(\alpha)}$  is Turing-reducible to  $\emptyset^{(\alpha)} \oplus X$ . (For  $\alpha = 1$ , such an  $X$  is said to be *generalized low*<sub>1</sub>, or  $GL_1$ ; this situation was described above.) More generally, relativizing to a fixed  $S \subseteq \mathbb{N}$ , almost all  $X$  satisfy  $(S \oplus X)^{(\alpha)} \leq_T S^{(\alpha)} \oplus X$ . Details appear in [7].
- (4) In the preceding item, "almost all" refers to Lebesgue measure on the space of inputs  $X$ , and the Turing reducibility, while not uniform, is uniform up to a set of arbitrarily small measure, and moreover is uniform in the bound on that set's measure. That is, there is a Turing functional  $\Psi$  such that, for every

$\varepsilon > 0$  and every  $S$ ,

$$\mu(\{X : \Psi^{S^{(\alpha)} \oplus X}(\varepsilon, \cdot) \neq (S \oplus X)^{(\alpha)}\}) < \varepsilon.$$

We often write  $\Psi_\varepsilon$  for the unary functional  $\Psi(\varepsilon, \cdot)$ .  $\Psi_\varepsilon$  is not uniform in  $\alpha$ .

Taken together, these four items suggest a natural approach for proving Lusin's Theorem: if  $\mathbf{f}(\mathbf{x})$  is given by  $\Phi^{(S \oplus X)^{(\alpha)}}$  as in (2), and  $(S \oplus X)^{(\alpha)}$  is given in turn by  $\Psi_\varepsilon^{S^{(\alpha)} \oplus X}$ , then composing these two should yield a function

$$\Gamma^{S^{(\alpha)} \oplus X} = \Phi\left(\Psi_\varepsilon^{S^{(\alpha)} \oplus X}\right)$$

which will be continuous by virtue of item (1) and will equal  $\mathbf{f}$  up to a set of measure  $< \varepsilon$  by virtue of item (4).

Of course, many readers will have spotted potential flaws in this argument already, and it will require a good deal of work to address them. In particular, even for those  $X$  on which  $\Psi_\varepsilon^{S^{(\alpha)} \oplus X}$  fails to output  $X^{(\alpha)}$ ,  $\Gamma$  must still give a coherent output, i.e., a fast-converging Cauchy sequence. Moreover, the sequences computed by  $\Gamma$  for two distinct  $X_0$  and  $X_1$  must converge to the same real number whenever  $X_0$  and  $X_1$  both converge to the same input  $\mathbf{x}$ . Of course, each  $\mathbf{x}$  is the limit of continuum-many fast-converging Cauchy sequences, many of which may not be handled correctly by  $\Psi$ , so this appears to be a serious problem. Finally, Lebesgue measure on Cantor space  $2^{\mathbb{N}}$  is used for the statement that almost all  $X$  are  $GL_1$ , whereas we must use Lebesgue measure on  $\mathbb{R}$ , represented as a quotient of the space of all fast-converging Cauchy sequences, to address Lusin's Theorem.

Nevertheless, we will overcome these difficulties and give a proof of Lusin's Theorem essentially following the outline given above. This will require a reasonable background in computable analysis and also in the computability-theoretic notion of the jump operation on subsets of  $\mathbb{N}$  and its iteration through the countable ordinals. Two appendices (Sections 9 and 10) are devoted to these two topics, as few readers can be expected to be closely familiar with both of them. For more extensive presentations, we suggest [8] for the computable analysis; [3] for descriptive set theory such as the preceding characterization of the Borel functions; [6, Chapter III] for the basics of the jump; and [1, Chapters 4-5] for iterating the jump through the computable ordinals, which generalizes by relativization to all countable ordinals.

When we prove it, Lusin's Theorem will reappear as Theorem 5.2. It is normally stated as above, for the real numbers  $\mathbb{R}$ , but it also applies to Cantor space  $2^{\mathbb{N}}$ . Additionally, there are analogous theorems (for both  $\mathbb{R}$  and  $2^{\mathbb{N}}$ ) using Baire category in place of Lebesgue measure. Each of these is simpler to prove than Theorem 5.2, and we will use them as warm-ups, introducing several of the techniques required in the more complex context of Theorem 5.2 itself. Moreover, these results will demonstrate the connection between computability and a standard question about these theorems. The question is, "why does Lusin's Theorem only yield  $\mathbf{g} = \mathbf{f}$  up to a set of positive

measure, with different functions  $\mathbf{g}$  for different positive values of  $\varepsilon$ , when the Baire-category version gives a single continuous  $\mathbf{g}$  that equals  $\mathbf{f}$  everywhere except on a meager set?" This will be seen to be a direct consequence of the nature of computing  $X'$  from  $\emptyset' \oplus X$  uniformly. A uniform computation is possible on a comeager set of  $X$ -values (with no wrong answers even for the other  $X$ , but sometimes with no answer given). Under Lebesgue measure, some guessing is required, and therefore each uniform computation will be wrong on some set of positive measure (although at least it will always provide an answer, albeit occasionally an incorrect one). In some respects this can be seen to reflect the distinction between the modern notions of *generic computability* and *coarse computability*, as described for instance in [2].

Notation here is standard and follows Soare [6]. In particular, capital Greek letters such as  $\Phi$ ,  $\Gamma$ , and  $\Psi$  denote oracle Turing functionals, and  $\Phi_0, \Phi_1, \dots$  is a standard computable enumeration of all such functionals. Sometimes  $\Phi_{e,s}$  is used to denote the version of the functional  $\Phi_e$  that automatically halts after  $s$  steps, even if it has no output yet. We write  $a, b, q, r, u, v$  and sometimes  $\varepsilon$  for rational numbers;  $\mathbf{a}, \mathbf{b}, \mathbf{r}, \mathbf{x}, \mathbf{y}$  for real numbers; capitals including  $A, C, X$ , and  $Y$  for elements of Cantor space  $2^{\mathbb{N}}$  (that is, for subsets of  $\mathbb{N}$ , often used as oracles for a functional), and  $\mathbf{f}, \mathbf{g}$  for functions mapping  $2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$  or  $\mathbb{R} \rightarrow \mathbb{R}$ . Some further notation is described in the appendices (Sections 9 and 10).

## 2. COMPUTING DISCONTINUOUS FUNCTIONS

Section 9 gives background in computable analysis, including our reasons for using enumerations of Dedekind cuts to name real numbers, rather than fast-converging Cauchy sequences. Theorem 9.4 shows that it is impossible to compute a discontinuous function  $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}$  just from enumerations of Dedekind cuts for the input  $\mathbf{x} \in \mathbb{R}$ . However, if we allow ourselves more information about  $\mathbf{x}$ , then it becomes possible.

**Definition 2.1.** A function  $\mathbf{f} : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$  is *jump-computable* if there exists a Turing functional  $\Phi$  such that, for every  $X \in 2^{\mathbb{N}}$ ,  $\Phi^{(X')} : \mathbb{N} \rightarrow \{0, 1\}$  is the characteristic function of  $\mathbf{f}(X)$ . Likewise,  $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}$  is *jump-computable* if there exists a Turing functional  $\Phi$  such that, for every  $\mathbf{x} \in \mathbb{R}$  and every enumeration  $X = A \oplus B$  of the Dedekind cuts of  $\mathbf{x}$ , the function

$$\Phi^{(X')} : (\mathbb{Q} \times \mathbb{N})^2 \rightarrow \{0, 1\}$$

computes an enumeration of the Dedekind cuts of  $\mathbf{f}(\mathbf{x})$ .

More generally, for an ordinal  $\alpha < \omega_1$  and an oracle set  $S \subseteq \mathbb{N}$ ,  $f$  is  *$\alpha$ -jump  $S$ -computable* if there exists  $\Phi$  such that, for every  $\mathbf{x} \in \mathbb{R}$  and every enumeration  $X$  of the Dedekind cuts of  $\mathbf{x}$ ,

$$\Phi^{(S \oplus X)^{(\alpha)}} : (\mathbb{Q} \times \mathbb{N})^2 \rightarrow \{0, 1\}$$

computes an enumeration of the Dedekind cuts of  $\mathbf{f}(\mathbf{x})$ ; similarly for functions on  $2^{\mathbb{N}}$ .

If  $\alpha$  is a countable noncomputable ordinal, then the  $\alpha$ -th jump is not well-defined in general. Section 10 explains how we can choose an oracle  $S$  complex enough to compute the complete diagram of a presentation  $\mathcal{A}$  of  $\alpha$  (i.e., a linear order  $\mathcal{A}$  isomorphic to  $(\alpha, \epsilon)$  whose domain is  $\mathbb{N}$ ). So it is possible to discuss the situation  $\alpha \geq \omega_1^{CK}$ , although one must fix an  $S$ -decidable presentation  $\mathcal{A}$  of  $\alpha$ . Since we defined the  $\mathcal{A}$ -jump  $C^{(\mathcal{A})}$  to have  $C$  itself as its 0-th column, a functional  $\Phi$  with oracle  $(S \oplus X)^{(\mathcal{A})}$  can recover  $S$  from that column of the oracle, use it to compute the presentation  $\mathcal{A}$ , and thus make sense of the oracle  $(S \oplus X)^{(\mathcal{A})}$  uniformly.

The principal theorem relevant here can be found in [3]. By Lemma 9.3, for functions on  $\mathbb{R}$ , this theorem holds with  $\mathbf{x}$  and  $\mathbf{f}(\mathbf{x})$  represented either by fast-converging Cauchy sequences or by enumerations of Dedekind cuts.

**Theorem 2.2.** *For every Borel function  $\mathbf{f} : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$ , there is a Turing functional  $\Phi$ , an oracle  $S \subseteq \mathbb{N}$ , and an  $S$ -decidable presentation  $\mathcal{A}$  of some countable ordinal such that, for every  $X \in 2^{\mathbb{N}}$ ,  $\Phi^{(S \oplus X)^{(\mathcal{A})}} : \mathbb{N} \rightarrow \{0, 1\}$  is the characteristic function of  $\mathbf{f}(X)$ .*

*Similarly, for every Borel function  $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}$ , there is a Turing functional  $\Phi$ , an oracle  $S \subseteq \mathbb{N}$ , and an  $S$ -decidable presentation  $\mathcal{A}$  of some countable ordinal such that, for every enumeration  $A \oplus B$  of the Dedekind cuts of any  $\mathbf{x} \in \mathbb{R}$ ,  $\Phi^{(S \oplus X)^{(\mathcal{A})}}$  enumerates the Dedekind cuts of  $\mathbf{f}(\mathbf{x})$ .*

Computable analysts have commonly approached jump-computability the opposite way, by taking the limit of a computable function:

$$\mathbf{f}(\mathbf{x}) = \lim_{s \rightarrow \infty} \Phi^X(\cdot, s).$$

That is, for each  $s \in \mathbb{N}$ , one computes an approximation to  $\mathbf{f}(\mathbf{x})$ , and the actual value  $\mathbf{f}(\mathbf{x})$  is the limit of these approximations. As an example, the derivative of a differentiable function  $\mathbf{h}(\mathbf{x})$  could be given by letting  $\Phi^X$  be the difference quotient  $s \cdot (\mathbf{h}(\mathbf{x} + \frac{1}{s}) - \mathbf{h}(\mathbf{x}))$ . The Turing functional  $\Phi$  is readily defined from the functional and oracle computing  $\mathbf{h}$ . (The derivative of a computable differentiable function is not in general computable, so this is often the best that can be done.) The connection between this method and ours is given by the *Transparency Lemma*.

**Lemma 2.3** (Transparency Lemma (folklore)). *A function is jump-computable if and only if it is the limit of a computable function (in the sense immediately above).*

So one could attack Lusin's Theorem by iterating the limit operation instead of the jump operation. Our choice to use the jump is dictated mainly because it allows us to apply the known results of Section 10 on near-uniform continuity of the jump, especially Theorem 10.4.

### 3. RESULTS ON CANTOR SPACE $2^{\mathbb{N}}$

Our two main theorems about Borel functions on Cantor space depend heavily on three lemmas, which are the heart of the connection between computability and

Lusin's Theorem for  $2^{\mathbb{N}}$ . These lemmas are well-known, but due to their importance, we give the proofs here, at least for the base case  $\alpha = 1$ . Understanding them will prepare the reader for the analogous theorems about functions on  $\mathbb{R}$ , which are conceptually similar but more technical. The three analogous lemmas for  $\mathbb{R}$  all appear in the appendix (Section 10).

**Lemma 3.1.** *Let  $X \in 2^{\mathbb{N}}$  be generic. Then  $X' \equiv_T \emptyset' \oplus X$ , and indeed there exists a single Turing functional  $\Psi$  such that, for every generic (or even 1-generic)  $X \in 2^{\mathbb{N}}$ ,  $\Psi^{(\emptyset' \oplus X)}$  computes the characteristic function of  $X'$ .*

*More generally, for every oracle set  $S \subseteq \mathbb{N}$  and every  $S$ -decidable presentation  $\mathcal{A}$  of a countable ordinal  $\alpha$ , there exists some  $\Psi$  such that, for all  $X \in 2^{\mathbb{N}}$  that are generic relative to  $S$ ,  $\Psi^{(S^{(\mathcal{A})} \oplus X)}$  computes the characteristic function of  $(S \oplus X)^{(\alpha)}$ . Moreover, even for non-generic  $X \in 2^{\mathbb{N}}$  (and for every  $e \in \mathbb{N}$ ),  $\Psi^{(S^{(\mathcal{A})} \oplus X)}(e)$  either diverges or computes correctly whether  $e \in (S \oplus X)^{(\alpha)}$ .*

*Proof.* We describe here the situation with  $S = \emptyset$  and  $\alpha = 1$ . Given the oracle  $(\emptyset' \oplus X)$ , the functional  $\Psi$  on input  $e \in \mathbb{N}$  asks, for  $s = 1, 2, \dots$  in turn:

- whether  $\Phi_{e,s}^{(X \upharpoonright s)}(e)$  halts (using no more of the oracle than the first  $s$  bits); and also
- whether there exists any  $\sigma \supseteq X \upharpoonright s$  and any  $t$  such that  $\Phi_{e,t}^{\sigma}(e)$  halts.

If the first answer is ever positive,  $\Psi$  concludes that  $e \in X'$ , while if the second answer is ever negative,  $\Psi$  concludes that  $e \notin X'$ . Each conclusion is clearly correct (if ever reached), and for a 1-generic  $X$ , one of these conclusions must eventually be reached. (Notice also that, even for non-generic  $X$ ,  $\Psi^{(\emptyset' \oplus X)}$  never gives a wrong answer. However, it could simply never give an answer.)  $\square$

For successor  $\alpha = \beta + 1$ , a similar procedure relativized to  $\emptyset^{(\beta)}$  succeeds. The proof of the general result uses the uniformity of the above procedure for all  $\beta < \alpha$ .

Our first Lusin-type result follows from Lemma 3.1. This is a standard theorem.

**Theorem 3.2.** *Let  $\mathbf{f} : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$  be a Borel function. Then  $\mathbf{f}$  restricts to a continuous function  $\mathbf{g}$  whose domain is the (comeager) set  $G$  containing all elements of  $2^{\mathbb{N}}$  that are generic relative to  $S$ .*

*Proof.* Theorem 2.2 yields  $\Phi$ ,  $S$ , and  $\mathcal{A}$  (as described there) such that  $\Phi^{(S \oplus X)^{(\mathcal{A})}}$  computes  $\mathbf{f}(X)$  for every  $X \in 2^{\mathbb{N}}$ . Its restriction to  $G$  is therefore computed by

$$\Phi^{\Psi^{(S^{(\mathcal{A})} \oplus X)}},$$

as witnessed by Lemma 3.1. Since this function is  $S^{(\mathcal{A})}$ -computable, it is continuous on its domain, which contains the comeager set  $G$ . (For  $X \notin G$ , either the computation will output the correct value  $\mathbf{f}(X)$ , or there will exist some input  $n$  on which the computation never halts. As noted above, the computation will never give an incorrect answer, even for a single  $n$ .)  $\square$

Now we turn to Lebesgue measure on  $2^{\mathbb{N}}$ , for which each basic open set  $\mathcal{U}_\sigma = \{X \in 2^{\mathbb{N}} : \sigma \subseteq X\}$  has measure  $2^{-|\sigma|}$ , with the measure extending to all measurable subsets of  $2^{\mathbb{N}}$  in the usual Lebesgue definition. There is an analogue of Lemma 3.1 for Lebesgue measure, but, perhaps contrary to one's expectations, it is uniform only for sets of arbitrarily small measure, not up to a set of measure zero. The following lemma appears as Theorem 2 of [7], but a proof may also be deduced from the (more complicated) Lemma 10.3 and Theorem 10.4 in the appendices, especially in concert with the proof of Lemma 3.4 below.

**Lemma 3.3.** *For every oracle set  $S \subseteq \mathbb{N}$  and every  $S$ -decidable presentation  $\mathcal{A}$  of a countable ordinal  $\alpha$ , there exists some  $\Psi$  such that, for each fixed rational  $\varepsilon > 0$ ,  $\Psi^{(S^{(\mathcal{A})} \oplus X)}(\varepsilon, \cdot)$  computes the characteristic function of  $(S \oplus X)^{(\mathcal{A})}$  for all  $X$  outside a subset of  $2^{\mathbb{N}}$  of measure  $< \varepsilon$ . Moreover, the computation  $\Psi^{(S^{(\mathcal{A})} \oplus X)}(\varepsilon, e)$  halts for every  $X \in 2^{\mathbb{N}}$  and every  $(\varepsilon, e)$ , although for  $X$  within the set of measure  $< \varepsilon$  it may output a value distinct from the "correct" answer  $\chi_{(S \oplus X)^{(\mathcal{A})}}(e)$ .*

We will write  $\Psi_\varepsilon$  for the function  $\Psi(\varepsilon, \cdot)$ , with any oracle. For Lebesgue measure, a second fact is also necessary.

**Lemma 3.4.** *In Lemma 3.3, the error set described, whose measure is  $< \varepsilon$ , is the union of an  $S$ -computably enumerable sequence of basic open sets, uniformly in both  $\mathcal{A}$  and  $\varepsilon$ .*

*Proof.* First we explain this proof when  $\alpha = 1$  and with  $S$  omitted. Now  $\Psi_\varepsilon^{\emptyset' \oplus X}$  is attempting to compute  $X'$  as described in Lemma 3.3. Whenever  $\Psi_\varepsilon^{\emptyset' \oplus X}(e) \downarrow = 1$ , it does so because it has seen the computation  $\Phi_e^X(e)$  converge already, so  $e \in X'$  and this answer is correct. All errors (if any occur for this  $X$ ) occur for values  $e$  where  $\Psi_\varepsilon^{\emptyset' \oplus X}(e) \downarrow = 0$ .

To enumerate an error set, therefore, once  $\Psi_\varepsilon$  has declared that  $\Psi_\varepsilon^{\emptyset' \oplus X}(e) \downarrow = 0$  for all  $X$  extending some particular  $\tau \in 2^{<\mathbb{N}}$ , we watch for any strings  $\rho \supseteq \tau$  and any  $t$  for which  $\Phi_e^\rho(e) \downarrow$ . Any time we find such a  $\rho$ , we enumerate the basic open set defined by  $\tau$  into our error set, knowing that this entire basic open set lies in the error set. Since all errors are errors of this type (guessing that  $e \notin X'$  and later being proven wrong), this list of basic open sets is precisely the error set and thus has measure  $< \varepsilon$ .

In the general setting, the jump  $(S \oplus X)^{(\mathcal{A})}$  consists of many columns, one for each point  $k$  in the presentation  $\mathcal{A}$  of the ordinal  $\alpha$ . If  $k$  represents a successor ordinal  $\beta + 1$  in  $\mathcal{A}$ , then we do the same procedure as above, waiting for  $\Psi_\varepsilon$  to use its own computation of  $(S \oplus X)^{(\beta)}$  to declare that  $e \notin (S \oplus X)^{(\beta+1)}$  for all  $X \supset \tau$  and then watching for  $\Phi_e^{(S \oplus \rho)^{(\beta)}}(e) \downarrow$  for each  $\rho \supseteq \tau$ . Only  $\varepsilon \cdot 2^{-k-1}$  of the measure  $\varepsilon$  of the error set is allotted to this process, and it certainly enumerates all  $X$  for which  $\Psi_\varepsilon$  gave the wrong answer about whether some  $e$  lies in  $(S \oplus X)^{(\beta)}$  but gave correct answers to these questions for all ordinals  $< \beta$ . Now if in fact  $\Psi_\varepsilon$  gives a wrong answer for some

$e$  about whether  $e \in (S \oplus X)^{(A)}$ , then there is a least element  $\beta$  of  $\alpha$  for which it did so, and for this  $\beta$ ,  $X$  will belong to a basic open set enumerated into our error set.  $\square$

**Theorem 3.5.** *Let  $\mathbf{f} : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$  be a Borel function, and fix  $\varepsilon > 0$ . Then there exists a continuous function  $\mathbf{g} : 2^{\mathbb{N}} \rightarrow 2^{\mathbb{N}}$  such that*

$$\mu(\{X \in 2^{\mathbb{N}} : \mathbf{g}(X) \neq \mathbf{f}(X)\}) < \varepsilon.$$

*Proof.* The proof is somewhat analogous to that of Theorem 3.2, using Lemmas 3.3 and 3.4. In order to ensure that  $\mathbf{g}$  is defined on all of  $2^{\mathbb{N}}$  (which was impossible in Theorem 3.2), we must provide a value for  $\mathbf{g}(X)$  even for those  $X$  such that  $\Psi_{\varepsilon}^{(S^{(A)} \oplus X)} \neq (S \oplus X)^{(A)}$ . The danger here is that for these  $X$ , the functional  $\Phi$  running with the incorrect oracle  $\Psi_{\varepsilon}^{(S^{(A)} \oplus X)}(\varepsilon, \cdot)$  may fail to compute a total function, in which case our naive version of  $\mathbf{g}$ , namely

$$\Phi^{\Psi_{\varepsilon}^{(S^{(A)} \oplus X)}},$$

will fail to map all of  $2^{\mathbb{N}}$  into  $2^{\mathbb{N}}$ . Therefore it is necessary for us to be able to realize, at some point during the computation above, that  $\Psi_{\varepsilon}^{(S^{(A)} \oplus X)}$  fails to compute  $(S \oplus X)^{(A)}$  correctly. This is exactly the point of Lemma 3.4.

The procedure for  $\Gamma$  begins by starting the naive computation  $\Phi^{\Psi_{\varepsilon}^{(S^{(A)} \oplus X)}}$ . If we ever reach a stage  $s$  at which the  $s$ -th basic open set (given by a string  $\sigma_s \in 2^{<\mathbb{N}}$  such that the basic open set is  $\{Y \in 2^{\mathbb{N}} : \sigma \subset Y\}$ ) in the enumeration above contains  $X$  – i.e.,  $\sigma_s \subset X$  – then at that stage we simply end the computation on all inputs: the output is

$$\mathbf{g}(X) = \{n \in \mathbb{N} : \Psi_{\varepsilon, s}^{(S^{(A)} \oplus X)}(n) \downarrow = 1\},$$

where  $\Psi_{\varepsilon, s}$  indicates that we only run  $\Psi_{\varepsilon}$  for  $s$  steps. If we never reach such a stage  $s$ , then  $\Gamma$  continues running each computation  $\Psi_{\varepsilon}^{(S^{(A)} \oplus X)}$  until it halts, as it must, since in this case, with  $X$  not in the error set, this function succeeds in computing  $\mathbf{f}(X)$ . Thus  $\Gamma$  does indeed compute a function  $\mathbf{g}$  defined on all of  $2^{\mathbb{N}}$ , and thus continuous, differing from  $\mathbf{f}$  only on the error set. Notice also that the computation of an index for  $\mathbf{g}$  is uniform in the positive rational  $\varepsilon$  chosen.  $\square$

Theorems 3.2 and 3.5 are not perfectly analogous. The former yields a single continuous version of  $\mathbf{f}$  that may be undefined on a small (i.e., meager) set of inputs  $X$ . The latter yields different versions  $\mathbf{g}_{\varepsilon}$  for each  $\varepsilon > 0$ , each of which is continuous on all of  $2^{\mathbb{N}}$  but may differ from  $\mathbf{f}$  on a small set (i.e., of measure  $< \varepsilon$ ). The computability-theoretic difference between working with generic sets and working with Lebesgue measure accounts for the difference in the corresponding lemmas, and thus may be seen as explaining the difference between the theorems.

The technique of enumerating the error set, used in Lemma 3.4, will reappear in Section 5, when we address Lusin's Theorem itself using Lebesgue measure on  $\mathbb{R}$ .



4. BAIRE CATEGORY ON  $\mathbb{R}$ 

Our next step is to address the real line  $\mathbb{R}$  rather than Cantor space  $2^{\mathbb{N}}$ . The difficulty here lies in representing real numbers  $\mathbf{x} \in \mathbb{R}$ , since our fundamental objects are elements  $X \in 2^{\mathbb{N}}$  and each  $\mathbf{x} \in \mathbb{R}$  will be represented by many different  $X \in 2^{\mathbb{N}}$ , each naming a Cauchy sequence in  $\mathbb{Q}$  that converges fast to  $\mathbf{x}$ . (In Section 3, this was not an issue: each  $X \in 2^{\mathbb{N}}$  simply represented itself!) Indeed, the  $X$ 's representing a single  $\mathbf{x}$  will have many distinct Turing degrees.

The first impulse, for addressing this problem, is to switch to Dedekind cuts, since each  $\mathbf{x} \in \mathbb{R}$  is represented by a single strict left Dedekind cut (i.e., a nonempty subset of  $\mathbb{Q}$ , downward-closed under  $<$ , with no greatest element). Unfortunately, these cannot be used directly as oracles: the analogue of Lemma 9.3 for this system of representation of real numbers is false. Lemma 9.3 is our compromise, converting fast-converging Cauchy sequences into enumerations of strict left and right Dedekind cuts (and back) effectively. This is not a perfect solution, since for a given  $\mathbf{x}$ , the strict left and right cuts  $L_{\mathbf{x}}$  and  $R_{\mathbf{x}}$  each have many distinct enumerations, of many Turing degrees. However, as we now describe, it enables us to use enumerations of these cuts to run oracle Turing computations in an effective way, so that the output depends only on  $L_{\mathbf{x}}$  and  $R_{\mathbf{x}}$  (and thus only on  $\mathbf{x}$  itself), not on the choice of enumerations of  $L_{\mathbf{x}}$  and  $R_{\mathbf{x}}$ . Definition 9.1 describes the “canonical” enumeration of each  $\mathbf{x}$ , which is crucial for the proof of the next theorem. Our use of Dedekind cuts stems from our lack of any similar notion of a canonical Cauchy sequence converging fast to  $\mathbf{x}$ .

The generic real numbers form a comeager subset  $G$  of  $\mathbb{R}$ , where by definition  $\mathbf{x} \in \mathbb{R}$  is generic just if the cut  $L_{\mathbf{x}}$  is generic among all downward-closed sets of rational numbers. Clearly, genericity among such sets rules out having a greatest element, being empty, or being co-empty, so these sets are all strict left cuts of irrational numbers  $\mathbf{x}$ . Moreover, no definable left cut  $\mathbf{x}$  can be generic in this sense. This set  $G$  will be the comeager set we use to prove the Baire-category version of Lusin's Theorem.

**Theorem 4.1.** *Let  $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}$  be a Borel function. Then  $\mathbf{f}$  restricts to a continuous function  $\mathbf{g}$  whose domain is the (comeager) set  $G$  containing all elements of  $\mathbb{R}$  that are generic relative to  $S$ .*

*Proof.* Once again, Theorem 2.2 yields  $\Phi$ ,  $S$ , and  $\mathcal{A}$  such that, for every  $\mathbf{x} \in \mathbb{R}$  and every enumeration  $A \oplus B$  of the Dedekind cuts of  $\mathbf{x}$ ,  $\Phi^{(S \oplus A \oplus B)^{(\mathcal{A})}}$  enumerates the Dedekind cuts of  $\mathbf{f}(\mathbf{x})$ . On an enumeration  $A \oplus B$  of  $\mathbf{x} \in G$ , one might hope for it to be computed by a function along the lines of

$$\Phi^{\Psi^{(S^{(\mathcal{A})} \oplus (A \oplus B))}},$$

as witnessed by Lemma 3.1. Since this function is  $S^{(\mathcal{A})}$ -computable, it would be continuous on its domain.

Unfortunately, this does not suffice for a proof. This function is only defined on generic enumerations of cuts, and no matter which  $\mathbf{x} \in \mathbb{R}$  one chooses, there will exist non-generic enumerations of its cuts. Thus, our program above may fail to accept certain enumerations of  $\mathbf{x}$ , whereas a computable function must accept all of them. To rectify this, we give a slightly more involved program, which will succeed in computing  $\mathbf{f}(\mathbf{x})$  below an  $S^{(A)}$  oracle for every enumeration of each  $\mathbf{x} \in \mathbb{R}$  whose left Dedekind cut  $L_{\mathbf{x}}$  forms an  $S$ -generic downward-closed subset of  $\mathbb{Q}$ . We first describe our program  $\Gamma$ , which “uniformizes” the enumerations  $A \oplus B$  given to it, and then analyze it.

$\Gamma$  is given the oracle  $A \oplus B$ , along with the fixed set  $S^{(A)}$ . At each stage  $s$ , it defines  $A_s = A \cap (\{0, \dots, s\})^2$ ,  $B_s = B \cap (\{0, \dots, s\})^2$ , and  $l_s = \min\{n \in \mathbb{N} : q_n \notin \pi_1(A_s \cup B_s)\}$ , the greatest (up to length  $s$ ) initial segment of a fixed computable listing  $\{q_0, q_1, \dots\}$  of  $\mathbb{Q}$  for which the cuts  $L_{\mathbf{x}}$  and  $R_{\mathbf{x}}$  are fully defined by stage  $s$ . Then it runs the program  $\Phi$  for  $s$  steps on the given input  $q \in \mathbb{Q}$ , using the oracle:

$$\Psi_s^{S^{(A)} \oplus (\pi_1(A_s) \times \mathbb{N}) \oplus (\pi_1(B_s) \times \mathbb{N})}$$

and outputs the same value as this program in case it halts, or else goes back and repeats the same procedure for  $s + 1$ . (If, in response to an oracle question from  $\Gamma$ ,  $\Psi$  fails to halt within  $s$  steps using its own smaller oracle, then again  $\Gamma$  starts its own program over at  $s + 1$ .) This is the entire program for  $\Gamma$ .

In  $s$  steps,  $\Psi$  can never ask an oracle question about an element  $> s$ , so all oracle values used by  $\Psi$  here are exactly as in the canonical enumeration  $(L_{\mathbf{x}} \times \mathbb{N}) \oplus (R_{\mathbf{x}} \times \mathbb{N})$  of the cuts of  $\mathbf{x}$ . Thus, if  $\Psi$  actually gives an answer (on an input  $e$ ) using its oracle here, that answer will be identical to the one given by  $\Psi$  using that canonical oracle. In case  $\mathbf{x}$  is generic relative to  $S$ , it will therefore tell whether  $e \in (S \oplus (L_{\mathbf{x}} \times \mathbb{N}) \oplus (R_{\mathbf{x}} \times \mathbb{N}))^{(A)}$  or not. Thus our uniformization to the canonical oracle will have eliminated all uncertainty introduced by the use of the enumeration  $(A \oplus B)$ . Moreover, if  $l_s$  goes to  $+\infty$  as  $s$  increases, then  $\Psi$  will ultimately have access to as much of the canonical oracle as it needs for its computation. Finally, notice that a generic  $\mathbf{x} \in \mathbb{R}$  cannot be a rational number, and therefore every  $q_n \in \mathbb{Q}$  lies in  $L_{\mathbf{x}} \cup R_{\mathbf{x}}$ , hence lies in  $\pi_1(A_s \cup B_s)$  for all sufficiently large  $s$ . Thus for generic  $\mathbf{x}$ , we automatically have  $\lim_s l_s = +\infty$ , and so  $\Psi$  ultimately computes (correctly) as much of  $(S \oplus (L_{\mathbf{x}} \times \mathbb{N}) \oplus (R_{\mathbf{x}} \times \mathbb{N}))^{(A)}$  as  $\Phi$  requires. In short, for every generic  $\mathbf{x} \in \mathbb{R}$  and every enumeration  $A \oplus B$  of the cuts of  $\mathbf{x}$ , the program  $\Gamma$  produces the exact same output as  $\Phi^{(S \oplus A \oplus B)^{(A)}}$ , namely, an enumeration of the left and right cuts of  $\mathbf{f}(\mathbf{x})$ , exactly as claimed by the theorem.

Finally, notice that  $\Psi$  always uses an oracle that describes  $(L_{\mathbf{x}} \times \mathbb{N}) \oplus (R_{\mathbf{x}} \times \mathbb{N})$  correctly: the only possible problem arises when  $\mathbf{x}$  is not  $S$ -generic, and even then,  $\Psi$  never gives a wrong answer. Therefore,  $\Gamma$  never outputs a wrong answer either: for each  $q \in \mathbb{Q}$  and each enumeration  $A \oplus B$  of any  $\mathbf{x} \in \mathbb{R}$ , if  $\Gamma$  halts and outputs 0 on input  $q$ , then indeed  $q \in L_{\mathbf{f}(\mathbf{x})}$ ; likewise the output 1 always indicates that  $q \in R_{\mathbf{f}(\mathbf{x})}$ .

For an  $\mathbf{x}$  that is not generic relative to  $S$ , the program may fail to halt on various inputs  $q \in \mathbb{Q}$ , in which case that  $\mathbf{x}$  is not in the domain of the continuous function defined by  $\Gamma$ . However, non- $S$ -generic real numbers form a meager subset of  $\mathbb{R}$ , so we have proven the theorem.  $\square$

## 5. LUSIN'S THEOREM

Now we may approach Lusin's Theorem. The key lemma, again well-known, is described in more detail in the second appendix (Section 10), as Corollary 10.5.

**Lemma 5.1.** *There exists a Turing functional  $\Psi_\varepsilon$ , uniform in the rational  $\varepsilon > 0$ , such that for every set  $S \subseteq \mathbb{N}$  and every fixed  $S$ -decidable presentation  $\mathcal{A}$  of any ordinal  $\alpha < \omega_1^S$  (with  $\alpha$ -jumps  $C^{(\mathcal{A})}$  defined using this presentation), the “error set”*

$$\mathcal{U}_{\varepsilon, S, \mathcal{A}} = \{\mathbf{x} \in \mathbb{R} : \Psi_\varepsilon^{S^{(\mathcal{A})} \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}}} \neq (S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})^{(\mathcal{A})}\}$$

has measure  $< \varepsilon$  and is an  $S^{(\mathcal{A})}$ -effective union of rational open intervals, uniformly in  $\varepsilon$ ,  $S$ , and  $\mathcal{A}$ . Moreover, for all  $\mathbf{x}$ ,  $\Psi_\varepsilon^{S^{(\mathcal{A})} \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}}}$  is total.  $\square$

**Theorem 5.2** (Lusin, 1912). *Every Borel-measurable function  $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}$  is nearly continuous.*

*Proof.* As usual, Theorem 2.2 provides  $\Phi$ ,  $S$ , and  $\mathcal{A}$  such that, for every  $\mathbf{x} \in \mathbb{R}$  and every enumeration  $A \oplus B$  of the Dedekind cuts of  $\mathbf{x}$ ,  $\Phi^{(S \oplus A \oplus B)^{(\mathcal{A})}}$  enumerates the Dedekind cuts of  $\mathbf{f}(\mathbf{x})$ . We wish to construct a  $\Gamma$ , similar to that of Theorem 4.1, that uses  $\Psi$  to compute  $(S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})^{(\mathcal{A})}$  from  $(S^{(\mathcal{A})} \oplus A \oplus B)$ , then applies  $\Phi$  to enumerate the cuts of  $\mathbf{f}(\mathbf{x})$ . The catch is that now we wish the  $\mathbf{g}$  computed by  $\Gamma$  to have domain  $\mathbb{R}$ ; on the other hand, we may allow  $\mathbf{g}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x})$  on the error set  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$  given by Corollary 10.5. We write  $\mathcal{U} = \mathcal{U}_{\varepsilon, S, \mathcal{A}}$ , having fixed the three parameters, and, using Corollary 10.5, fix an  $S^{(\mathcal{A})}$ -computable enumeration of rational open intervals whose union equals  $\mathcal{U}$ . We also fix a computable list  $q_0, q_1, \dots$  of all rational numbers, without repetitions. It will be convenient to assume that the interval of  $\mathcal{U}$  enumerated at stage  $s$  is of the form  $(q_m, q_n)$  with  $m, n \leq s$ . This is not difficult to arrange, except that we must allow there to be stages at which no interval is enumerated.

The crucial fact here is the  $S^{(\mathcal{A})}$ -computable enumerability of the error set  $\mathcal{U}$  as a union of intervals with rational end points. This will allow our  $\Gamma$  to recognize these intervals as they appear, at which stage it will abandon its hope of computing a function equal to  $\mathbf{f}$  within such an interval, and will instead do “damage control” to ensure that it does actually compute a function (necessarily continuous) on each such interval. As long as we make these “damage control” functions meet at the end points of their intervals, we will have continuity everywhere: outside the error set,  $\Gamma$  will compute  $\mathbf{f}(\mathbf{x})$  successfully, in exactly the style of Theorem 4.1, since  $\Psi$  makes no mistakes outside the error set in computing  $(S \oplus (L_{\mathbf{x}} \times \mathbb{N}) \oplus (R_{\mathbf{x}} \times \mathbb{N}))^{(\mathcal{A})}$ .

It complicates matters, but it is also liberating, to recall that every rational  $\mathbf{x}$  must lie in  $\mathcal{U}$ . (In particular,  $\Psi$  will always guess incorrectly about whether the program

index  $e$  lies in  $(S \oplus A \oplus B)^{(A)}$ , where, for every input,  $\Phi_e$  halts as soon as the rational  $\mathbf{x}$  appears in  $\pi_1(A \cup B)$ .) The complication lies in the fact that, whenever a rational interval  $(a, b)$  is enumerated into the error set, the rational  $a$  itself must lie in another interval of the error set, as must  $b$ , causing these intervals ultimately to metastasize into one long open interval with irrational endpoints, the union of infinitely many overlapping rational intervals in  $\mathcal{U}$ . The liberation results from the fact that, as in Theorem 4.1, we are again free to ignore the difficulty that, for rational  $\mathbf{x}$ ,  $L_{\mathbf{x}} \cup R_{\mathbf{x}}$  omits an element (namely  $\mathbf{x}$  itself). Given an enumeration  $A \oplus B$  of the cuts of an arbitrary  $\mathbf{x} \in \mathbb{R}$ , we know that either  $\mathbf{x}$  will eventually enter the error set (which we can enumerate!), or else  $L_{\mathbf{x}} \cup R_{\mathbf{x}} = \mathbb{Q}$ . As long as  $\mathbf{x}$  is not yet in the error set, therefore, we are safe in waiting arbitrarily long for the next rational  $q_n$  on our list to be enumerated by either  $A$  or  $B$ . If  $\mathbf{x}$  ever does appear in the error set, we switch to damage control on  $\mathbf{x}$ .

The bulk of our description of the procedure for the functional  $\Gamma$  deals with the rational intervals (finitely many of them, at each stage  $s$ ) that currently constitute the error set. Once we have settled what to do with them, the procedure on the remainder of  $\mathbb{R}$  is the same one already applied several times. So suppose that as of stage  $s$ , our enumeration of  $\mathcal{U}$  so far consists of rational intervals  $(a_1, b_1), (a_2, b_2), \dots, (a_{s-1}, b_{s-1})$ , with each  $b_i < a_{i+1}$ . By induction we assume that we have *assigned values* (as defined below) to all points in the closure  $\overline{\mathcal{U}}_s$ . This means that we have already determined the finitely many values  $\mathbf{g}(a_i)$  and  $\mathbf{g}(b_i)$ , possibly along with values  $\mathbf{g}(c)$  at finitely many other rational  $c \in \mathcal{U}_s$ , and that we intend  $\mathbf{g}$  on  $\overline{\mathcal{U}}_s$  to be the piecewise linear function that “connects these dots” within each connected component of  $\overline{\mathcal{U}}_s$ .

Before we describe the action of  $\Psi$  on a particular  $(A \oplus B)$ , we explain how it sets the scene at each stage  $s + 1$ . We have a fixed computable enumeration  $\{q_0, q_1, \dots\}$  of  $\mathbb{Q}$ , so suppose that  $\{q_0, \dots, q_s\}$  is ordered as  $q_{j_0} < \dots < q_{j_s}$ . These define  $s + 1$  open intervals  $(-\infty, q_{i_0}), (q_{j_0}, q_{j_1}),$  etc., each giving an initial piece of a Dedekind cut: write  $L_{i,s} = \{q_{j_0}, \dots, q_{j_{i-1}}\}$  and  $R_{i,s} = \{q_{j_i}, \dots, q_{j_s}\}$ .  $\Psi$  runs the programs

$$\Phi_s^{S^{(A)} \oplus (L_{i,s} \times \mathbb{N}) \oplus (R_{i,s} \times \mathbb{N})}(q_k)$$

for each  $i \leq s$  and each  $k \leq s$ , giving up after  $s$  steps if there is not yet any convergence. This gives us general constraints on the possible values of  $\mathbf{f}$  in each interval  $(q_{j_{i-1}}, q_{j_i})$ : an upper bound (possibly  $+\infty$ )

$$u_{i,s} = \min\{q_k : k \leq s \ \& \ \Phi_s^{S^{(A)} \oplus (L_{i,s} \times \mathbb{N}) \oplus (R_{i,s} \times \mathbb{N})}(q_k) \downarrow = 1\}$$

and a lower bound (possibly  $-\infty$ )

$$v_{i,s} = \max\{q_k : k \leq s \ \& \ \Phi_s^{S^{(A)} \oplus (L_{i,s} \times \mathbb{N}) \oplus (R_{i,s} \times \mathbb{N})}(q_k) \downarrow = 0\}.$$

If the program in question actually computes  $\mathbf{f}$  on  $(q_{j_{i-1}}, q_{j_i})$  correctly, then the values of  $\mathbf{f}$  there must all lie in  $(v_{i,s}, u_{i,s})$ . Thus the set of boxes  $(q_{j_{i-1}}, q_{j_i}) \times (v_{i,s}, u_{i,s})$  may be viewed as a tentative approximation of the graph of  $\mathbf{f}$  in the  $xy$ -plane. Of course,

this does not take the error set  $\mathcal{U}$  into account. It could happen that  $u_{i,s} \leq v_{i,s}$  under this definition, if the interval  $(q_{j_{i-1}}, q_{j_i})$  is contained within  $\mathcal{U}$ . Also, two adjacent boxes (as described above) might have nonintersecting closures, which again would indicate interference by the error set. Therefore we have only defined  $u_{i,s}$  and  $v_{i,s}$  here, without using them as yet.

We now continue stage  $s+1$  by considering the next interval  $I_s = (a_s, b_s)$  to appear in the enumeration of intervals comprising  $\mathcal{U}$ . We wish to assign  $\mathbf{g}$ -values to all points  $\mathbf{x} \in [a_s, b_s]$ . This is the damage-control operation: we have given up on computing  $\mathbf{f}(\mathbf{x})$  correctly for these points, and merely hope to make  $\mathbf{g}$  continuous. Within  $I_s$  this will be easy, but we also need to ensure that our chosen values allow  $\mathbf{g}$  to be continuous at the boundary of  $\mathcal{U}$ , where it interfaces with error-free computations of  $\mathbf{f}$ . (The boundary of  $\mathcal{U}$  is in fact all the rest of  $\mathbb{R}$ , since  $\mathcal{U}$  is dense.)

Let  $\mathcal{U}_{s+1} = \cup_{i < s} [a_i, b_i]$ . If  $I_s \subseteq \mathcal{U}_s$  (which is decidable from the end points), we do nothing, since values will have already been assigned to all points in  $\mathcal{U}_s$  at preceding stages. Otherwise  $(I_s - \mathcal{U}_s)$  is a finite union of closed nonempty intervals, and we treat each of its connected components  $[c, d]$  separately. First, if any of the points  $q_{j_0}, \dots, q_{j_s}$  defined earlier lies in  $[c, d]$ , then for each such  $q_{j_i}$  satisfying  $\min(u_{i,s}, u_{i,s+1}) \geq \max(v_{i,s}, v_{i+1,s})$ , we define

$$\mathbf{g}(q_{j_i}) = \frac{\min(u_{i,s}, u_{i+1,s}) + \max(v_{i,s}, v_{i+1,s})}{2}.$$

(In case  $\min(u_{i,s}, u_{i+1,s}) = +\infty$ ,  $\mathbf{g}(q_{j_i}) = \max(v_{i,s}, v_{i+1,s}) + 1$ . In case  $\max(v_{i,s}, v_{i+1,s}) = -\infty$ ,  $\mathbf{g}(q_{j_i}) = \min(u_{i,s}, u_{i+1,s}) - 1$ . If both occur, then  $\mathbf{g}(q_{j_i}) = 0$ .)

If  $q_{j_i}$  lies in  $[c, d]$  but  $\min(u_{i,s}, u_{i,s+1}) < \max(v_{i,s}, v_{i+1,s})$ , then these bounds create an inconsistency at  $q_{j_i}$ . In this situation we continue to enumerate  $\mathcal{U}$  until a rational interval  $(r_0, r_1)$  containing  $q_{j_i}$  is enumerated into  $\mathcal{U}$ . (This must occur eventually, as all rational numbers lie in  $\mathcal{U}$ .) When this first happens, we define

$$\mathbf{g}\left(\frac{\max(r_0, c, q_{j_{i-1}}) + q_{j_i}}{2}\right) = \frac{u_{i,s} + v_{i,s}}{2} \quad \mathbf{g}\left(\frac{q_{j_i} + \min(r_1, d, q_{j_{i+1}})}{2}\right) = \frac{u_{i+1,s} + v_{i+1,s}}{2}.$$

The argument on the left lies below  $q_{j_i}$  but above each of  $r_0$ ,  $c$ , and  $q_{j_{i-1}}$ , hence within  $[r_0, r_1]$ , within the interval where  $u_{i,s}$  and  $v_{i,s}$  are the upper and lower bounds, and within  $[c, d]$ , so it is safe to assign to it this  $\mathbf{g}$ -value between those bounds; similarly for the argument on the right. (In case  $q_{j_i} = c \in \overline{\mathcal{U}_s}$ , only the equation on the right applies, as in this case the left equation has argument  $c$  and  $\mathbf{g}(c)$  is already defined. Similarly, if  $q_{j_i} = d \in \overline{\mathcal{U}_s}$ , only the equation on the left applies.)

Next, we attend to the end points. If either  $c$  or  $d$  lies in the closure  $\overline{\mathcal{U}_s}$ , then it already has been assigned a  $\mathbf{g}$ -value. If  $c \notin \overline{\mathcal{U}_s}$ , then find the interval  $(q_{j_{i-1}}, q_{j_i})$  to

which it belongs and define

$$\mathbf{g}(c) = \begin{cases} \frac{u_{i,s} + v_{i,s}}{2}, & \text{if this is defined;} \\ u_{i,s} - 1, & \text{if } v_{i,s} = -\infty \ \& \ u_{i,s} \neq +\infty; \\ v_{i,s} + 1, & \text{if } u_{i,s} = +\infty \ \& \ v_{i,s} \neq -\infty; \\ 0, & \text{if } v_{i,s} = -\infty \ \& \ u_{i,s} = +\infty. \end{cases}$$

Likewise if  $d \notin \overline{\mathcal{U}}_s$  and  $d \in (q_{j_{k-1}}, q_{j_k})$ , then define  $\mathbf{g}(d)$  exactly the same way, using  $u_{k,s}$  and  $v_{k,s}$ . The case where  $c$  or  $d$  actually equals  $q_{j_i}$  for some  $i \leq s$  was already covered in our first step.

Now we have defined  $\mathbf{g}$  on  $c$ , on  $d$ , and on every  $q_{j_i}$  in between. We finish by “connecting the dots”: simply define  $\mathbf{g}$  between each pair of consecutive points among these by linear interpolation between the  $\mathbf{g}$ -values of the two points.

It now remains to describe the procedure followed by  $\Gamma$  with a specific enumeration  $A \oplus B$  of strict Dedekind cuts (of an arbitrary  $\mathbf{x} \in \mathbb{R}$ ) in its oracle. We use sets  $A_s$  and  $B_s$  and the length  $l_s$  just as defined in the proof of Theorem 4.1. At stage  $s + 1$ ,  $\Gamma$  searches for the least  $t \in \mathbb{N}$  such that either

- (1)  $t > s$  and  $l_t \geq s$ ; or
- (2) there exist  $m, n \leq t + 1$  such that  $q_m \in A_{t+1}$  and  $q_n \in B_{t+1}$  and the open interval  $(q_m, q_n)$  is enumerated into  $\mathcal{U}$  by stage  $t + 1$ .

If condition (2) holds, then by running the instructions above all the way up to the stage  $t$  at which  $(q_m, q_n)$  is enumerated into  $\mathcal{U}_t$ , we can determine the  $\mathbf{g}$ -values assigned at that stage to points in this interval. Of course, at this stage we still only have a finite approximation to  $\mathbf{x}$ , but, knowing the finitely many linear pieces of  $\mathbf{g}$  on this interval, we can use the approximation given by  $A_s$  and  $B_s$  to approximate  $\mathbf{g}(\mathbf{x})$  in the obvious way. (Notice that this works even if  $\mathbf{x}$  itself is rational: the fact that some single rational never appears in either Dedekind cut does not stop us from computing  $\mathbf{g}(\mathbf{x})$ .)

If condition (1) holds, then from  $A_t$  and  $B_t$  we can decide which of the intervals  $(q_{j_{i-1}}, q_{j_i})$  defined at stage  $s + 1$  contains  $\mathbf{x}$ . If  $u_{i,s} \leq v_{i,s}$ , then we do nothing at this stage, since then this entire interval must eventually enter  $\mathcal{U}$ . Assuming  $u_{i,s} < v_{i,s}$ , we know that  $\mathbf{x}$  itself does not lie in  $\mathcal{U}_{s+1}$ : condition (1) requires  $t > s$ , so if an interval  $(q_m, q_n)$  containing  $\mathbf{x}$  had been enumerated into  $\mathcal{U}_{s+1}$ , it would have  $m, n \leq s$  (by our convention for this enumeration), and thus stage  $s$  would have satisfied condition (2). However, we do not want to deal with an  $\mathbf{x}$  that is too close to either  $q_m$  or  $q_n$ . So we search again, for a stage  $t'$  such that  $\mathcal{U}_{t'}$  contains both an interval  $(a'_m, b'_m)$  that contains  $q_m$  and also an interval  $(a'_n, b'_n)$  that contains  $q_n$ , and such that either

- (1a)  $A_{t'}$  contains a rational number  $\geq b'_m$ , and  $B_{t'}$  contains a rational number  $\leq a'_n$  (making  $\mathbf{x} \in (b'_m, a'_n)$ ); or
- (1b)  $A_{t'}$  contains a rational number  $\geq a'_n$  or  $B_{t'}$  contains a rational number  $\leq b'_m$  (making  $\mathbf{x} \in \mathcal{U}_{t'}$ ).

If (1b) holds,  $\Gamma$  does not converge on any inputs here, because we know that our  $\mathbf{x}$  will eventually enter  $\mathcal{U}$ . If (1a) holds, then  $\Gamma$  halts and outputs 1 for all inputs  $\geq u_{i,s}$ , and halts and outputs 0 for all inputs  $\leq v_{i,s}$ . (Recall that we have checked that  $v_{i,s} < u_{i,s}$ .) This means that these values will be upper and lower bounds (respectively) for the real number  $\mathbf{g}(\mathbf{x})$  ultimately determined by  $\Gamma$ . This completes our description of the program executed by  $\Gamma$ .

It remains to show that  $\Gamma^{S^{(A)} \oplus A \oplus B}$  succeeds in computing a function  $\mathbf{g} : \mathbb{R} \rightarrow \mathbb{R}$ , as  $A \oplus B$  varies over all enumerations of Dedekind cuts, and then that  $\mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathbf{x})$  outside of the error set  $\mathcal{U}$  (which we know has measure  $< \varepsilon$ ). Being computed by  $\Gamma$  this way,  $\mathbf{g}$  will necessarily be continuous.

To see this, fix any  $\mathbf{x} \in \mathbb{R}$  and any enumeration  $A \oplus B$  of the cuts  $L_{\mathbf{x}}$  and  $R_{\mathbf{x}}$  of  $\mathbf{x}$ . Suppose first that  $\mathbf{x} \in \mathcal{U}$ . Here we do not claim that  $\mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathbf{x})$ , but we do show that  $\Gamma^{S^{(A)} \oplus A \oplus B}$  computes the same real-number output  $\mathbf{g}(\mathbf{x})$  regardless of the enumeration of  $\mathbf{x}$ . Indeed, since  $\mathbf{x} \in \mathcal{U}$ , there is some interval  $I_{s_0} = (a_{s_0}, b_{s_0})$  with  $\mathbf{x} \in I_{s_0}$  enumerated into  $\mathcal{U}$  at some stage  $s_0$ . Notice that at every subsequent stage  $s > s_0$ , condition (2) will hold with  $t = s_0 - 1$ , while condition (1) can only be applied at stage  $s + 1$  if  $t > s$ ; thus the instructions for condition (1) will never be followed again. Moreover, regardless of the choice of enumeration  $A \oplus B$  of  $L_{\mathbf{x}}$  and  $R_{\mathbf{x}}$ , we will recognize from stage  $s_0 + 1$  onwards that the enumeration of  $I_{s_0}$  into  $\mathcal{U}$  has placed us in condition (2). Therefore, from stage  $s_0 + 1$  on,  $\Gamma$  will always proceed with the computation of the assigned value  $\mathbf{g}(\mathbf{x})$ , based on the linear functions chosen at stage  $s + 1$  for the interval of  $\mathcal{U}_s$  containing  $\mathbf{x}$ . Of course, the computation of these linear functions (with rational coefficients, no less!) is effective regardless of the enumeration  $A \oplus B$ , and the end points of the linear function were selected specifically so that no upper or lower bounds enumerated by  $\Gamma$  at preceding stages could possibly contradict the value  $\mathbf{g}(\mathbf{x})$  assigned to any  $\mathbf{x}$  in the interval, including our specific  $\mathbf{x}$ . (It is also important to note here that we never assigned contradictory information in the computation of  $\mathbf{g}(\mathbf{x})$ , even when using condition (1). All rationals enumerated into the lower cut of  $\mathbf{g}(\mathbf{x})$  were checked to be  $<$  all rationals enumerated into the upper cut. Moreover, as long as condition (1) applied, the same rationals were enumerated into each cut independently of the specific enumeration  $A \oplus B$ . The choice of  $A \oplus B$  only affects the stage at which the computation permanently switches over to condition (2), which occurs at or before stage  $s_0$ , and the scene-setting procedure included a check that all upper and lower bounds enumerated for values  $\mathbf{x}$  in  $I_{s_0}$  are consistent with the choice of  $\mathbf{g}(\mathbf{x})$ . Therefore  $\Gamma$  does compute the same value  $\mathbf{g}(\mathbf{x})$  independent of the specific enumeration  $A \oplus B$  of the cuts of  $\mathbf{x}$ . This is all that we require of  $\Gamma$  on an  $\mathbf{x} \in \mathcal{U}$ : the damage-control procedure succeeded.

The other case occurs when  $\mathbf{x} \notin \mathcal{U}$ . Now the procedure above never acts on condition (2). Since every rational number lies in  $\mathcal{U}$ , it follows that  $\mathbf{x} \notin \mathbb{Q}$ , so every  $q_m$  lies in  $L_{\mathbf{x}} \cup R_{\mathbf{x}}$ , and thus the length  $l_s$  of the approximation  $A_s \oplus B_s$  goes to  $+\infty$  as  $s$  increases. Consequently, for every stage  $s$ , there does exist some  $t > s$  with  $l_t \geq s$ .

Now since  $\mathbf{x} \notin \mathcal{U}$ ,  $\Phi$  does enumerate the cuts of  $\mathbf{f}(\mathbf{x})$  when it runs with the oracle set  $\Psi^{S^{(A)} \oplus (L_{\mathbf{x}} \times \mathbb{N}) \oplus (R_{\mathbf{x}} \times \mathbb{N})}$ . Therefore, for any specific rational  $u > \mathbf{f}(\mathbf{x})$  and  $v < \mathbf{f}(\mathbf{x})$ , there exists some  $s_0$  such that  $\Phi$  with this oracle enumerates  $u$  into the upper cut and  $v$  into the lower cut, in at most  $s_0$  steps, using only the initial portion  $((L_{\mathbf{x}} \upharpoonright s_0) \times \mathbb{N}) \oplus ((R_{\mathbf{x}} \upharpoonright s_0) \times \mathbb{N})$  of the oracle of  $\Psi$ . But  $l_t \rightarrow +\infty$  as  $t$  increases, so at every stage  $\geq s_0$ , running  $\Gamma$  with the oracle  $A \oplus B$  will produce this much of  $L_{\mathbf{x}}$  and  $R_{\mathbf{x}}$ . Now fix some  $m, n > s_0$  such that  $\mathbf{x} \in (q_m, q_n)$  but no  $q_k$  with  $k < \max(m, n)$  lies in  $(q_m, q_n)$ . When we reach stage  $s + 1$  with  $s = \max(m, n)$ , this interval will be the  $i$ -th interval in the partition of  $\mathbb{R}$  (that is,  $m = j_{i-1}$  and  $n = j_i$  at this stage), and so the bounds  $u_{i,s}$  and  $v_{i,s}$  determined at that stage will have  $\mathbf{f}(\mathbf{x}) < u_{i,s} \leq u$  and  $\mathbf{f}(\mathbf{x}) > v_{i,s} \geq v$ . Moreover, since  $\mathbf{x} \notin \mathcal{U}$ , condition (1b) must hold (at this and all other stages), and so  $u_{i,s}$  and  $v_{i,s}$  are enumerated into the upper and lower cuts of  $\mathbf{g}(x)$  at that stage if not before, no matter what input  $A \oplus B$  enumerating the cuts  $L_{\mathbf{x}}$  and  $R_{\mathbf{x}}$  is used. With  $v < v_{i,s}$  and  $u_{i,s} < u$ , this makes it clear that  $\Gamma$  enumerates our original (arbitrary)  $u$  and  $v$  into the correct cuts. Therefore, on all inputs  $A \oplus B$  enumerating  $L_{\mathbf{x}}$  and  $R_{\mathbf{x}}$ ,  $\Gamma$  does compute  $\mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathbf{x})$  as required.  $\square$

## 6. UNIFORMITY

It was not the original purpose of this article to prove anything new. The intention was to present a new proof of Lusin's Theorem in real analysis, using known facts from computability theory and descriptive set theory, and thus to illustrate and illuminate a connection between the principles used in standard proofs of Lusin's Theorem and the principles from computability which make our proof here work. Nevertheless, certain uniformities and computability results became apparent during the creation of the proof in Section 5, and in the end we have an effective version of Lusin's Theorem. Sometimes new ideas entail new results, even when not intended to do so.

The substantial uniformity in the creation of  $\mathbf{g}$  from  $\mathbf{f}$  in our proof of Theorem 5.2 yields the function  $h$  that we describe here. Similar uniform versions hold for our simpler results in Theorems 3.2, 3.5, and 4.1.

**Theorem 6.1.** *There is a computable total function  $h : \mathbb{Q} \times \mathbb{N} \rightarrow \mathbb{N}$  such that, for each fixed  $\alpha$  and  $S$  and each  $S$ -decidable presentation  $\mathcal{A}$  of  $\alpha$ , whenever an  $\alpha$ -jump-computable function  $\mathbf{f}$  is given by the oracle computation*

$$\Phi_e^{(S \oplus A \oplus B)^{(A)}}$$

for all enumerations  $A \oplus B$  of the cuts of each  $\mathbf{x} \in \mathbb{R}$ , the function  $\mathbf{g}(\mathbf{x})$  defined by

$$\Phi_{h(\varepsilon, \varepsilon)}^{E(\mathcal{A}) \oplus S^{(\mathcal{A})} \oplus A \oplus B}$$

realizes Lusin's Theorem 5.2 for this  $\mathbf{f}$  and an arbitrary rational  $\varepsilon > 0$ . Here  $E(\mathcal{A})$  is the elementary diagram of  $\mathcal{A}$ , given as a subset of  $\mathbb{N}$  by a Gödel coding.



*That is, the Turing functional for computing  $\mathbf{g}$  can be determined uniformly from that for  $\mathbf{f}$ , uniformly in  $\varepsilon$  and independently of the level of  $\mathbf{f}$  in the Borel hierarchy up to  $\alpha$ .*  $\square$

By the Padding Lemma (see e.g. [6, Lemma I.3.2]),  $h$  may also be assumed injective. Of course,  $h$  only describes how to determine the program for computing this  $\mathbf{g}$ . In order to compute  $\mathbf{g}$ , one also needs the oracle  $S^{(\mathcal{A})}$  and the elementary diagram  $E(\mathcal{A})$  of the linear order  $\mathcal{A}$ . However, the choice of program for  $\mathbf{g}$  depends only on  $\varepsilon$  and the program given for computing  $\mathbf{f}$ , not on which  $S$  and  $\mathcal{A}$  are used to compute  $\mathbf{f}$ . In fact, not all of the atomic diagram  $E(\mathcal{A})$  is required as an oracle; it suffices to know which elements of  $(\mathcal{A}, <)$  are limit points from the left, which is the zero element, which pairs  $(m, n)$  are adjacencies (with  $m < n$  and no elements between them), and whether  $\mathcal{A}$  itself is a limit ordinal, a successor, or zero. For uniformity, though, even the finite information, such as knowing which element is the left end point of  $\mathcal{A}$ , must be given. Finally, it is not necessary to be given this diagram itself: since an  $S$ -oracle is given, one only needs to know an index for computing this information about  $\mathcal{A}$  from  $S$ .

Nowhere in the proof of Theorem 5.2 did we use the fact that the values  $\mathbf{f}(\mathbf{x})$  were finite real numbers (as opposed to  $\pm\infty$ ). Indeed, the same proof would work even if enumerations of improper Dedekind cuts were allowed as outputs. Moreover, the construction in Theorem 5.2 ensures that the points  $\mathbf{x}$  where  $\mathbf{g}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x})$  all have  $\mathbf{g}(\mathbf{x})$  finite, as they all lie in  $\mathcal{U}$ , which is the union of intervals on which  $\mathbf{g}$  is defined to be finite.

**Theorem 6.2.** *For every Borel-measurable function  $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R} \cup \{\pm\infty\}$  and every  $\varepsilon > 0$ , there exists a continuous function  $\mathbf{g} : \mathbb{R} \rightarrow \mathbb{R} \cup \{\pm\infty\}$  such that  $\mu(\{\mathbf{x} \in \mathbb{R} : \mathbf{f}(\mathbf{x}) \neq \mathbf{g}(\mathbf{x})\}) < \varepsilon$ . Moreover, the value  $\mathbf{g}(\mathbf{x})$  is infinite only when  $\mathbf{f}(\mathbf{x})$  is, and all uniformities described in Theorem 6.1 still hold here.*  $\square$

## 7. THE CAUSES OF DISCONTINUITY

Our proof of Theorem 5.2 emphasizes a remarkable fact about Lusin's Theorem. Once the parameters  $\varepsilon$ ,  $S$ , and  $\mathcal{A}$  are fixed, the proof uses the exact same error set  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$  for every function  $\mathbf{f}$  it is given. So one may legitimately argue that the non-continuity of functions  $\mathbf{f}$  at this level of the Borel hierarchy is the "fault" of the real numbers  $\mathbf{x}$  in

$$\mathcal{U}_{S, \mathcal{A}} = \bigcap_{\varepsilon > 0} \mathcal{U}_{\varepsilon, S, \mathcal{A}},$$

namely, the set of those  $\mathbf{x} \in \mathbb{R}$  that are not generalized- $\alpha$ -low relative to  $S$  (using the presentation  $\mathcal{A}$  of  $\alpha$ ). In this sense, the functions themselves are not the obstacle: their non-continuity was caused by our inability to approximate  $(S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})^{(\mathcal{A})}$  for those  $\mathbf{x}$  in the error intervals.

**Proposition 7.1.** *Fix  $\alpha$ ,  $S$ , an  $S$ -decidable presentation of  $\alpha$ , and a rational  $\varepsilon > 0$ . Then for every  $\alpha$ -jump  $S$ -computable function  $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}$ , the procedure in Theorem 6.1 produces a continuous  $\mathbf{g}$  such that the set  $\{\mathbf{x} \in \mathbb{R} : \mathbf{f}(\mathbf{x}) \neq \mathbf{g}(\mathbf{x})\}$  is always contained within the same open set  $\mathcal{U}_{\varepsilon, S, \mathcal{A}} \subseteq \mathbb{R}$  of measure  $< \varepsilon$ , independent of the choice of  $\mathbf{f}$ . Indeed, for  $\varepsilon_0 \leq \varepsilon_1$ , we have  $\mathcal{U}_{\varepsilon_0, S, \mathcal{A}} \subseteq \mathcal{U}_{\varepsilon_1, S, \mathcal{A}}$ .  $\square$*

The immediate objection to this proposition is that, just by translating  $\mathbf{f}$  by a certain fixed parameter  $\mathbf{c}$ , one could define a function  $\mathbf{f}_{\mathbf{c}}(\mathbf{x}) = \mathbf{f}(\mathbf{x} - \mathbf{c})$  for which most of the discontinuities of  $\mathbf{f}$  move out of  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$ . This is true, but it requires  $\mathbf{c}$  to be noncomputable, indeed not  $S$ -computable, and so  $\mathbf{f}_{\mathbf{c}}$  does not belong to the class of functions considered in Proposition 7.1. In fact,  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$  is closed under translation by  $S^{(\mathcal{A})}$ -computable parameters, and under other similar gambits one might concoct.

The more informed objection to the proposition is that it is obvious: there are only countably many  $\alpha$ -jump  $S$ -computable functions, so by applying Lusin's Theorem to the  $n$ -th such function with tolerance  $\frac{\varepsilon}{2^{n+1}}$ , we immediately prove the proposition. This is correct, but the spirit of the proposition is that it was not necessary to slice up the  $\varepsilon$ -amount of measure this way: our proof of Theorem 5.2 defined  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$  using basic computability theory, and then uniformly constructed some continuous  $\mathbf{g}$  for each  $\mathbf{f}$  such that they differed only within  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$ . Probably the best way to express this is to note that the restriction of every such  $\mathbf{f}$  to the complement of each  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$  is itself  $S^{(\mathcal{A})}$ -computable and hence continuous on this domain, and that each such domain is simply a  $\Pi_1^{S^{(\mathcal{A})}}$  set of real numbers.

In contrast, however, the restriction of such an  $\mathbf{f}$  to the complement of  $\mathcal{U}_{S, \mathcal{A}}$  (defined just above) need not be continuous. Analogously, while only measure-0-many real numbers fail to be generalized- $\alpha$ -low relative to  $S$ , no single Turing functional can compute  $(S \oplus X)^{(\alpha)}$  from  $S^{(\alpha)} \oplus X$  for all but measure-0-many  $X$ . As an example of a 2-jump-computable function  $\mathbf{f}$  such that no restriction of  $\mathbf{f}$  to a set  $\mathcal{D} \subseteq \mathbb{R}$  of full measure is continuous on the domain  $\mathcal{D}$ , consider

$$\mathbf{f}(\mathbf{x}) = \sum_{e \in (L_{\mathbf{x}} \oplus R_{\mathbf{x}})'} \frac{1}{2^e}.$$

The set  $(L_{\mathbf{x}} \oplus R_{\mathbf{x}})'$  is c.e. relative to  $(L_{\mathbf{x}} \oplus R_{\mathbf{x}})$ , hence uniformly decidable using the oracle  $(A \oplus B)''$  whenever  $(A \oplus B)$  enumerates the cuts of  $\mathbf{x}$ . Thus  $\mathbf{f}$  is 2-jump-computable. However, from any enumeration of the cuts of  $\mathbf{f}(\mathbf{x})$ , one can compute  $(L_{\mathbf{x}} \oplus R_{\mathbf{x}})'$ , uniformly in  $\mathbf{x}$  and in the choice of enumeration. Therefore, by the last part of Lemma 10.3, the restriction of  $\mathbf{f}$  to an arbitrary set of full measure cannot possibly be an  $S$ -computable function, no matter what oracle set  $S$  one chooses. Then one adapts Theorem 9.4 to functions on subdomains within  $\mathbb{R}$  to show that this restriction cannot be continuous.

## 8. COMPUTING CONTINUOUS FUNCTIONS

When Lusin's Theorem is applied to a function  $\mathbf{f}$  that is already continuous, it holds trivially: just take  $\mathbf{g} = \mathbf{f}$ . One might ask whether the procedure given in Section 5 reflects this. The immediate answer is that it does not: if  $\mathbf{f}$  is continuous but is presented to us as an  $\alpha$ -jump-computable function, applying the procedure there will often produce a  $\mathbf{g}$  that, while satisfying the requirements of Lusin's Theorem, is not in fact equal to  $\mathbf{f}$ , not even up to a set of measure 0. For future investigation, we conjecture that this is inherent: no uniform procedure (as in Theorem 6.1) instantiating Lusin's Theorem can also succeed in making  $\mathbf{g} = \mathbf{f}$  when  $\mathbf{f}$  is itself continuous.

However, if we ask the same question restricted entirely to continuous functions  $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}$ , then it is possible to produce a procedure for computing the function from a procedure for  $\alpha$ -jump-computing it. (In general a stronger oracle is required, though.) This situation could plausibly arise: for example, perhaps we can only determine a jump-computation for a solution  $\mathbf{f}$  to some differential equation under certain initial conditions, although such an  $\mathbf{f}$ , being differentiable, must be continuous.

**Theorem 8.1.** *Let  $\alpha$  be a countable ordinal and  $\mathcal{A}$  an  $S$ -decidable presentation of  $\alpha$ . Then there exists a computable total injective function  $h : \mathbb{N} \rightarrow \mathbb{N}$  such that, whenever*

$$\mathbf{f} = \Phi_e^{((S \oplus A \oplus B)^{(\mathcal{A})})} : \mathbb{R} \rightarrow \mathbb{R}$$

*is an  $\alpha$ -jump  $S$ -computation of a continuous  $\mathbf{f}$ , we have a 0-jump  $S^{(\mathcal{A}+1)}$ -computation*

$$\Phi_{h(e)}^{(S^{(\mathcal{A}+1)} \oplus A \oplus B)} = \mathbf{f}.$$

Here  $\mathcal{A}+1$  is the presentation of the ordinal  $\alpha+1$  with  $\text{dom}(\mathcal{A}+1) = \text{dom}(\mathcal{A}) \cup \{k\}$ , where the number  $k = \min(\mathbb{N} - \text{dom}(\mathcal{A}))$  is adjoined to  $\mathcal{A}$  as a new greatest element.

*Proof.* Where in Theorem 5.2, the rational numbers were a hindrance to be handled by error sets, here instead they serve as our guide. For every  $q \in \mathbb{Q}$ , the left and right cuts  $L_q$  and  $R_q$  are computable uniformly in  $\mathbb{Q}$ , so  $(S \oplus L_q \oplus R_q)^{(\mathcal{A})}$  is  $S^{(\mathcal{A})}$ -computable, uniformly in  $q$ , and an  $(S \oplus L_q \oplus R_q)^{(\mathcal{A}+1)}$  oracle can decide the set

$$D = \{(a, b, u, v) \in \mathbb{Q}^4 : (\forall q \in [a, b]) v < \mathbf{f}(q) < u\}.$$

The elements of  $D$  are “boxes”  $(a, b) \times (v, u)$  in  $\mathbb{R}^2$  within which the graph of  $\mathbf{f}$  (restricted to  $(a, b)$ ) must lie. Now for any  $\mathbf{x} \in \mathbb{R}$  and any enumeration  $A \oplus B$  of the cut of  $\mathbf{x}$ , we get an  $S^{(\mathcal{A}+1)}$ -computable enumeration of

$$E_{\mathbf{x}} = \{(u, v) \in \mathbb{Q}^2 : (\exists a \in p_1(A))(\exists b \in p_1(B)) (a, b, u, v) \in D\}.$$

By continuity there are boxes in  $D$  with  $u - v$  arbitrarily small, and so the projections  $p_3$  and  $p_4$  of  $E_{\mathbf{x}}$  are the right and left cuts  $R_{\mathbf{f}(\mathbf{x})}$  and  $L_{\mathbf{f}(\mathbf{x})}$ . Thus we have a computation of  $\mathbf{f}$  below an  $S^{(\mathcal{A}+1)}$ -oracle, whose program is uniform in the index  $e$ .  $\square$

9. APPENDIX: COMPUTABLE FUNCTIONS ON  $\mathbb{R}$ 

Turing computability normally applies to functions from  $\mathbb{N}$ , the set of all nonnegative integers, into itself. By fixing a computable bijection between  $\mathbb{N}$  and  $\mathbb{Q}$ , we may equally well consider functions  $\mathbb{Q} \rightarrow \mathbb{Q}$ , or  $\mathbb{Q} \rightarrow \mathbb{N}$ , or  $\mathbb{N} \rightarrow \mathbb{Q}$ . We also use a standard computable bijection  $\langle m, n \rangle$  mapping  $\mathbb{N} \times \mathbb{N}$  onto  $\mathbb{N}$ , and similarly for  $\mathbb{Q}$ .

We write  $a, b, q, r, u, v$  and sometimes  $\varepsilon$  for rational numbers, and  $\mathbf{a}, \mathbf{b}, \mathbf{r}, \mathbf{x}, \mathbf{y}$  for real numbers. Real numbers correspond bijectively to Dedekind cuts under their usual definition: nonempty downward-closed proper subsets of  $\mathbb{Q}$  with no greatest element. For our purposes, this definition must be adapted slightly.

**Definition 9.1.** The *Dedekind cut* of  $\mathbf{x} \in \mathbb{R}$  is the pair  $(L, R)$ , where  $L = \{q \in \mathbb{Q} : q < \mathbf{x}\}$  and  $R = \{q \in \mathbb{Q} : q > \mathbf{x}\}$ . In particular, if  $\mathbf{x} \in \mathbb{Q}$ , then  $\mathbf{x} \notin L \cup R$ . We also define *generalized Dedekind cuts* to include the pairs  $(\emptyset, \mathbb{Q})$  and  $(\mathbb{Q}, \emptyset)$ , corresponding to  $-\infty$  and  $+\infty$ , in addition to the *proper* Dedekind cuts defined above.

An *enumeration* of a generalized Dedekind cut  $(L, R)$  is a set that, when expressed as a join  $A \oplus B$ , satisfies  $\pi_1(A) = L$  and  $\pi_1(B) = R$ , where  $\pi_1(\langle q, n \rangle) = q$  is the projection map. The *canonical enumeration* of  $(L, R)$  is the join  $(L \oplus \mathbb{N}) \oplus (R \oplus \mathbb{N})$ .

It is natural to regard  $A$  and  $B$  as subsets of  $\mathbb{Q} \times \mathbb{N}$ , so that an oracle for  $A \oplus B$  allows us to list out the elements in the projections of  $A$  and  $B$ , and thus to enumerate both the lower cut and the upper cut of the real  $\mathbf{x}$ .

We can now give a definition of computability for functions on  $\mathbb{R}$  using Dedekind cuts, instead of the usual fast-converging Cauchy sequences, to represent real numbers.

**Definition 9.2.** For each subset  $S \subseteq \mathbb{N}$ , a function  $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}$  is *S-computable* if there exists a Turing functional  $\Phi$  such that, whenever  $X$  is an enumeration of the Dedekind cut of any  $\mathbf{x} \in \mathbb{R}$ ,  $\Phi^{S \oplus X}$  is the characteristic function of an enumeration of the Dedekind cut of  $\mathbf{f}(\mathbf{x})$ .

It is equivalent, and often simplifies matters, to have  $\Phi^{S \oplus X}$  be a partial function from  $\mathbb{Q}$  into  $\{0, 1\}$ , understanding  $\{q : \Phi^{S \oplus X}(q) \downarrow = 0\}$  and  $\{q : \Phi^{S \oplus X}(q) \downarrow = 1\}$  to be the left and right Dedekind cuts of  $\mathbf{f}(\mathbf{x})$ , respectively. In this case, if  $\mathbf{f}(\mathbf{x})$  itself is rational, then  $\Phi^{S \oplus X}(\mathbf{f}(\mathbf{x}))$  never halts.

The possibility that  $L \cup R$  omits an element of  $\mathbb{Q}$  is the reason for considering enumerations of cuts. If we had simply taken  $L$  as an oracle, rather than an enumeration of  $(L, R)$ , then for rational  $q$ , the characteristic function of the interval  $[q, +\infty)$  would have been computable; similarly with  $R$  and the interval  $(-\infty, q]$ . On the other hand, if we had required the cut of a rational  $\mathbf{y}$  to include  $\mathbf{y}$  itself on one side or the other, then functions such as  $\mathbf{f}(\mathbf{x}) = \mathbf{x}^2 - 2$  would not be computable. (For that  $\mathbf{f}$ , given an enumeration of the cuts of  $\mathbf{x} = \sqrt{2}$ ,  $\Phi$  would never be able to place the rational 0 with certainty on either side of the cut of  $\mathbf{f}(\sqrt{2})$ .)

Classically it is standard to express Definition 9.2 using *fast-converging Cauchy sequences* instead of Dedekind cuts. (A Cauchy sequence  $\langle q_n \rangle_{n \in \mathbb{N}}$  *converges fast* to  $\mathbf{x}$

=  $\lim_n q_n$  if, for every  $n$ ,  $|q_n - \mathbf{x}| < 2^{-n}$ .) Readers will understandably be baffled at first by our choice to use enumerations of Dedekind cuts as inputs and outputs of functions, rather than following tradition. We request forbearance: in our view, this is the simplest way to prove Theorems 4.1 and 5.2, because (as suggested in Definition 9.1) it affords a canonical enumeration of the Dedekind cuts of each  $\mathbf{x}$ , whereas in general no Cauchy sequence converging fast to  $\mathbf{x}$  is identified as the canonical such sequence. Were it not for the simplification of the proofs of these theorems, we would gladly use the traditional definition, which has proven appropriate for all work so far in computable analysis.

Definition 9.1 makes Definition 9.2 equivalent to the usual definition of computable functions on  $\mathbb{R}$ . The next lemma proves this, by showing that we can pass effectively between the different methods of representing real numbers.

**Lemma 9.3.** *There exist Turing functionals converting each of the following representations of real numbers  $\mathbf{x}$  into the other:*

- An arbitrary enumeration of a Dedekind cut for  $\mathbf{x}$  in  $\mathbb{Q}$ .
- An arbitrary Cauchy sequence that converges fast to  $\mathbf{x}$ .

*Proof.* The lemma states that, for example, there exists a Turing functional  $\Upsilon$  such that, whenever  $A \oplus B$  is an enumeration of the Dedekind cuts in  $\mathbb{Q}$  for some real number  $\mathbf{x}$ ,  $\Upsilon^{A \oplus B}$  will be a Cauchy sequence converging fast to the same  $\mathbf{x}$ . Indeed, given an enumeration  $A \oplus B$  of the cut of some  $\mathbf{x}$ , on input  $n$ ,  $\Upsilon$  searches for  $\langle a, s \rangle \in A$  and  $\langle b, t \rangle \in B$  with  $b - a < \frac{1}{2^{n-1}}$ . For the first such pair of pairs to be found, it outputs  $q_n = \frac{a+b}{2}$  as the  $n$ -th term. Since  $\mathbf{x} \in (a, b)$ , this  $q_n$  is strictly within  $\frac{1}{2^n}$  of  $\mathbf{x}$ , so  $\langle q_n \rangle_{n \in \mathbb{N}}$  converges fast to  $\mathbf{x}$ . (This is the definition of *fast convergence*:  $|q_n - \mathbf{x}| < \frac{1}{2^n}$ .) Conversely, given any Cauchy sequence  $\langle q_n \rangle_{n \in \mathbb{N}}$  converging fast to  $\mathbf{x}$ , the set

$$\left\{ \langle a, m \rangle \in \mathbb{Q} \times \mathbb{N} : a < q_m - \frac{1}{2^m} \right\} \oplus \left\{ \langle b, n \rangle \in \mathbb{Q} \times \mathbb{N} : b > q_n + \frac{1}{2^n} \right\}$$

enumerates the Dedekind cuts of  $\mathbf{x}$ . □

With Lemma 9.3 it is clear that the well-known theorem of Weihrauch holds for the functions of Definition 9.2.

**Theorem 9.4** (Weihrauch; see [8]). *A function  $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}$  is continuous if and only if there exists a set  $S \subseteq \mathbb{N}$  such that  $\mathbf{f}$  is  $S$ -computable in the sense of Definition 9.2.*

## 10. APPENDIX: APPROXIMATING THE ITERATED JUMP

In this appendix we describe known concepts and results that may be unfamiliar to readers outside computability theory, and also describe the precise version of the iterated jump used in this article. By definition, the *jump*, or *Turing jump*,  $A'$  of a set  $A \subseteq \mathbb{N}$  is the relativization of the Halting Problem to the set  $A$ :

$$A' = \{e \in \mathbb{N} : \Phi_e^A(e) \text{ halts}\}.$$

Here  $\Phi_0, \Phi_1, \dots$  is the standard enumeration of all Turing functionals, that is, of all programs for Turing machines endowed with an arbitrary “oracle set” of natural numbers. One writes  $\Phi_e^A$  for the partial function from  $\mathbb{N}$  into  $\mathbb{N}$  computed by the  $e$ -th such program when using the set  $A$  as its oracle. We refer the reader to [6, Chap. III] for full definitions. The set  $\emptyset'$  is essentially the Halting Problem itself, and just as  $\emptyset'$  is not computable, so likewise  $A$  is always strictly below  $A'$  in Turing reducibility:  $A <_T A'$ , by which we mean  $A \leq_T A'$  but  $A' \not\leq_T A$ . (It is not completely obvious that one can compute  $A$  using an  $A'$ -oracle, but the proof is not difficult.)

The *jump operator* is simply the function  $A \mapsto A'$ , sending each subset of  $\mathbb{N}$  to its jump, and thus mapping the power set  $\mathcal{P}(\mathbb{N})$  into itself. This map is injective but not surjective, and it preserves Turing reductions but can fail to preserve non-reductions: the implication

$$A \leq_T B \implies A' \leq_T B'$$

always holds, but its converse can fail. Since the jump operator maps  $\mathcal{P}(\mathbb{N})$  into itself, it is natural to iterate it. We write  $A^{(k+1)}$  for the jump of  $A^{(k)}$ , with  $A^{(0)} = A$ , thus defining all finite jumps of each set  $A$ .

However, this is not the end of the iteration. For  $\omega$ , the first infinite ordinal, the  $\omega$ -th jump  $A^{(\omega)}$  of a set  $A$  is a sort of union of all the preceding jumps of  $A$ :

$$A^{(\omega)} = \{\langle n, k \rangle \in \mathbb{N}^2 : n \in A^{(k)}\}.$$

One views the  $k$ -th jump of  $A$  (for each  $k$ ) as being coded into  $A^{(\omega)}$  as the  $k$ -th column, under the usual computable bijection from  $\mathbb{N}^2$  onto  $\mathbb{N}$  mapping the ordered pair  $(n, k) \in \mathbb{N}^2$  to its code number  $\langle n, k \rangle \in \mathbb{N}$ . Clearly no finite jump  $A^{(k)}$  can compute  $A^{(\omega)}$ , since if it could, it would then compute the  $(k+1)$ -st column  $A^{(k+1)}$ , contradicting the fundamental property of the jump operator. We view  $A^{(\omega)}$  as a natural “next” jump, in the sense of ordinals, after all finite jumps have been built. (To be clear: the set  $\omega$  is actually just the set  $\mathbb{N}$ , but now viewed as an ordinal.)

Nor yet is this the end of the process. One now continues through successor ordinals as before, with  $A^{(\omega+1)} = (A^{(\omega)})'$  and so on. At subsequent limit ordinals  $\lambda$ , it is not always as obvious as for  $\omega$  exactly how to define the  $\lambda$ -th jump, but we can do so if given a computable presentation of the ordinal  $\lambda$  – that is, a computable linear ordering  $<$  of the domain  $\mathbb{N}$  such that  $(\mathbb{N}, <)$  is isomorphic to  $\lambda$  as a linear order. (Later we will assume that in the order  $<$ , the successor and limit relations are computable as well.) Then we can define the  $k$ -th column of  $A^{(\lambda)}$  to represent the  $\alpha$ -th jump  $A^{(\alpha)}$ , where  $k \in \mathbb{N}$  is mapped to  $\alpha \in \lambda$  by the isomorphism from  $(\mathbb{N}, <)$  onto  $\lambda$ . This gives a reasonable notion of  $A^{(\lambda)}$ , except that it depends on the choice of the presentation  $<$  of  $\lambda$ . With proper use of ordinal notations, one can now define the set  $A^{(\alpha)}$  for all ordinals  $\alpha$  with computable presentations, and, although the actual set depends on the notation chosen, its Turing degree does not.

Church and Kleene knew that there must be a countable ordinal with no computable presentation, and the least such ordinal is now known as  $\omega_1^{\text{CK}}$ . Iterating the

jump to  $A^{(\omega_1^{\text{CK}})}$  and beyond requires presentations of noncomputable ordinals. In the work in this article, we are generally able to use as oracle a (noncomputable) set  $S$  capable of giving presentations of noncomputable countable ordinals. Of course, for each  $S$  there is a least countable ordinal  $\omega_1^S$  which has no  $S$ -computable presentation (and it then follows that no ordinal  $> \omega_1^S$  has any  $S$ -computable presentation either). On the other hand, every countable ordinal is isomorphic to some linear order on  $\mathbb{N}$ , just by the definition of countability, and so, for every countable  $\alpha$ , there is some oracle set  $S$  such that  $\alpha < \omega_1^S$ , i.e., such that  $\alpha$  has an  $S$ -computable presentation. (Again, we will need a slightly stronger presentation of  $\alpha$ , with successors and limits known, but with the right oracle this can also be assumed. In fact, Spector showed that  $S$  itself can give such a presentation.)

Thus we have (sketchily) described the iterated notion of the jump operator. Having done so, we will now give formal definitions for use in this article.

**Definition 10.1.** A *presentation*  $\mathcal{A}$  of a nonzero ordinal  $\alpha < \omega_1$  is a linear order  $\mathcal{A} = (D, <)$  isomorphic to  $(\alpha, \epsilon)$ , whose domain  $D$  is a coinfinite subset of  $\mathbb{N}$  and whose least element is the number 0 in  $\mathbb{N}$ . The presentation  $\mathcal{A}$  is  *$S$ -decidable* if  $S$  can compute the complete diagram of  $\mathcal{A}$ .

A presentation is normally called  *$S$ -computable* if  $S$  can compute its atomic diagram. We will need more than just that here, and we could be more precise (as above) about the exact information we require  $S$  to compute: the successor function on  $\mathcal{A}$ , the existence of a greatest element in  $\mathcal{A}$ , and the unary relations on  $\mathcal{A}$  of having no immediate predecessor and of being the least or the greatest element of  $\mathcal{A}$ . Demanding that  $S$  compute the complete diagram is overkill, but keeps the definition simple. The requirement that  $D$  be coinfinite is unusual in computable structure theory, but helpful for our purposes here: we will need it when  $\alpha$  is a successor, in order to have a location in which to code one more jump.

It is important to notice that, for every nonzero  $k \in D$ , the substructure  $\mathcal{B}_k$  with domain  $\{j \in D : j < k\}$  is a presentation of an ordinal  $\beta_k < \alpha$ , and that each  $\beta < \alpha$  is isomorphic to  $\beta_k$  for some unique  $k \in D$ . Moreover,  $S$  can compute the complete diagram of each  $\mathcal{B}_k$ , uniformly in  $k$ .

**Definition 10.2.** For a presentation  $\mathcal{A}$  of a nonzero ordinal  $\alpha < \omega_1$ , the  *$\mathcal{A}$ -jump*  $C^{(\mathcal{A})}$  of a set  $C \subseteq \mathbb{N}$  is the subset of  $\mathbb{N}$  containing those codes  $\langle k, n \rangle$  for pairs  $(k, n)$  such that either:

- $k \in \text{dom}(\mathcal{A})$  and  $n \in C^{(\mathcal{B}_k)}$ ; or
- $\mathcal{A}$  has a  $<$ -greatest element  $j$  and  $k$  is the least number  $> j$  in  $\overline{D}$  and

$$(\forall n \in \mathbb{N}) [\langle k, n \rangle \in C^{(\mathcal{A})} \iff \Phi_e^{(C^{(\mathcal{B}_j)})}(e) \downarrow].$$

The ordinal  $\alpha = 0$  has only one presentation  $\mathcal{A}_0$ , with empty domain, and we define  $C^{(\mathcal{A}_0)} = \mathbb{N} \times C$  for every set  $C \subseteq \mathbb{N}$ .

So, for  $k \in D$ , the  $k$ -th column of  $C^{(\mathcal{A})}$  is simply the set  $C^{(\mathcal{B}_k)}$ , meaning that for every  $\beta < \alpha$ , the  $\beta$ -th jump of  $C$  (under the presentation  $\mathcal{B}_k$ ) appears as the  $k$ -th column. If  $\alpha$  is a limit ordinal, all other columns are empty, but for a successor ordinal  $\alpha = \beta + 1$ , with  $j$  the greatest element of  $\mathcal{A}$ , the  $k$ -th column of  $C^{(\mathcal{A})}$  (for to the least number  $k > j$  in  $\overline{D}$ ) presents the  $\alpha$ -th jump of  $C$  (under the presentation  $\mathcal{A}$ ). With  $D$  coinfinite, such a column does exist and is found using the diagram of  $\mathcal{A}$ .

Notice that the 0-th column  $\{n : \langle 0, n \rangle \in C^{(\mathcal{A})}\}$  of  $C^{(\mathcal{A})}$  is just the set  $C$  itself, for every presentation  $\mathcal{A}$  of any ordinal, because 0 is always the least element of the presentation. This gives us a uniform way to recover  $C$  from  $C^{(\mathcal{A})}$ , independent of the presentation.

When  $\alpha = m$  is finite,  $C^{(\mathcal{A})}$  is not literally the same set as the jump  $C^{(m)}$  discussed above, but for our purposes in computability they are equivalent: one column of  $C^{(\mathcal{A})}$  actually is the set  $C^{(m)}$ , all others are computable from  $C^{(m)}$ , and we have a 1-reduction from the  $k$ -th column to  $C^{(m)}$  uniformly in  $k$ . More generally, for a presentation  $\mathcal{A}$  of a successor  $\alpha = \beta + 1$ , one can view  $C^{(\mathcal{A})}$  as the jump of  $C^{(\mathcal{B}_j)}$ , where  $j$  is the greatest element in  $\mathcal{A}$ : this essentially says that the  $\alpha$ -th jump is the jump of the  $\beta$ -th jump. ( $C^{(\mathcal{B}_j)}$  itself also appears inside  $C^{(\mathcal{A})}$ , just as the  $\beta$ -th jump is 1-reducible to the  $\alpha$ -th jump.) The uniformity of Definition 10.2 across finite and infinite ordinals will simplify our arguments below.

The key property used in this article is that, for a fixed oracle set  $S$ , “most” sets  $A$  satisfy  $(S \oplus A)' \leq_T S' \oplus A$ , and that this reduction is uniform for most of those sets  $A$ . It is impossible for this to hold for all  $A$  (and in particular for  $A = S'$ ), but it does hold for all 1-generic sets  $A$ , as seen the proof of Lemma 3.1 above. For Lebesgue measure on  $2^{\mathbb{N}}$ , Theorem 2 of [7] proves that a Turing reduction  $A' \leq_T \emptyset' \oplus A$  exists for measure-1-many sets  $A$ . It is not uniform on any set of full measure. (That is, no single functional  $\Phi$  suffices, even up to a set of measure 0.)

Since we wish to address functions on  $\mathbb{R}$  as well as on  $2^{\mathbb{N}}$ , we need a more specific theorem, using only (strict) Dedekind cuts  $L \oplus R$  of real numbers as our oracles. Since the set of all such cuts has measure 0 under Lebesgue measure on Cantor space (viewed here as  $2^{(\mathbb{Q} \oplus \mathbb{Q})}$ ), we must re-prove the result of [7] for our own measure, namely Lebesgue measure on  $\mathbb{R}$ .

The first lemma is a warm-up for the main theorem, demonstrating the basic technique. Fix a computable enumeration  $q_0, q_1, \dots$  of all rationals in the interval  $(0, 1)$ . We write  $L_{\mathbf{x}}$  for the strict left Dedekind cut of a real number  $\mathbf{x}$ , and  $R_{\mathbf{x}}$  for its strict right cut. For each  $a = q_j$  and  $b = q_k$  in  $\mathbb{Q}$  such that  $0 \leq a < b \leq 1$  and every  $q_i \in (a, b)$  has  $i > \max\{j, k\}$ , we define the binary strings  $\lambda_{a,b}$  and  $\rho_{a,b}$  each to have length  $l$ , where  $l$  is least with  $q_l \in (a, b)$ , and set

$$\lambda_{a,b}(i) = \begin{cases} 1, & \text{if } q_i \leq a \\ 0, & \text{if } q_i \geq b \end{cases} \quad \text{and} \quad \rho_{a,b}(i) = 1 - \lambda_{a,b}(i) = \begin{cases} 1, & \text{if } q_i \geq b \\ 0, & \text{if } q_i \leq a. \end{cases}$$



Thus the real numbers  $\mathbf{x}$  in the open interval  $(a, b)$  are precisely those  $\mathbf{x}$  such that  $\lambda_{a,b}$  is an initial segment of  $L_{\mathbf{x}}$  (viewed as an infinite binary string) and  $\rho_{a,b}$  is an initial segment of  $R_{\mathbf{x}}$ . (These strings are examined further at the end of the proof of Theorem 10.4.) For intervals  $(a, b)$  where some  $q_i \in (a, b)$  has  $i < j$  or  $i < k$ , it is possible to divide the interval into finitely many subintervals with the property required above, effectively, and to consider a string  $\lambda \oplus \rho$  for each subinterval.

**Lemma 10.3.** *For every rational  $\varepsilon > 0$ , there exists a Turing functional  $\Psi_\varepsilon$  such that, for every  $S \subseteq \mathbb{N}$ ,*

$$\mu(\{\mathbf{x} \in (0, 1) : \Psi_\varepsilon^{S' \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}}} = (S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})'\}) > 1 - \varepsilon.$$

Moreover, there is a computable function  $h$  mapping each  $\varepsilon > 0$  to an index  $h(\varepsilon) \in \mathbb{N}$  such that  $\Phi_{h(\varepsilon)} = \Psi_\varepsilon$ . So this process is uniform in  $\varepsilon$ , although when  $\varepsilon = 0$ , no  $\Psi$  suffices.

*Proof.* Given an  $e \in \mathbb{N}$ ,  $\Psi_\varepsilon^{S' \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}}}$  searches for a rational number  $r \in [0, 1]$ , a finite initial segment  $\sigma \subset S$ , and a finite collection  $(a_0, b_0), \dots, (a_m, b_m)$  of disjoint open rational subintervals of  $(0, 1)$  and a stage  $s$  such that:

- $(\forall i \leq m) \Phi_{e,s}^{\sigma \oplus \lambda_{a_i,b_i} \oplus \rho_{a_i,b_i}}(e) \downarrow$ ; and
- $\sum_{i \leq m} (b_i - a_i) > r$ ; and
- there do *not* exist a number  $t$ , a  $\tau \subset S$ , and finitely many disjoint rational intervals  $(c_0, d_0), \dots, (c_n, d_n)$  within  $(0, 1)$  such that  $\sum_{i \leq n} (d_i - c_i) \geq r + \frac{\varepsilon}{2^{e+1}}$  and  $(\forall i \leq n) \Phi_{e,t}^{\tau \oplus \lambda_{a_i,b_i} \oplus \rho_{a_i,b_i}}(e) \downarrow$ .

The  $S'$ -oracle allows  $\Psi_\varepsilon$  to recognize the truth or falseness of the final statement for any specific  $r$ , while the first two statements are decidable. For the  $r$  that is found, we have  $r < \mu(\{\mathbf{y} \in (0, 1) : \Phi_e^{S \oplus L_{\mathbf{y}} \oplus R_{\mathbf{y}}}(e) \downarrow\}) \leq r + \frac{\varepsilon}{2^{e+1}}$ . (This also makes it clear why such an  $r$  must exist: arbitrarily much of the measure of this set can be covered by finitely many initial segments of oracles  $L_{\mathbf{x}} \oplus R_{\mathbf{x}}$ .) Now  $\Psi_\varepsilon$  examines the  $L_{\mathbf{x}} \oplus R_{\mathbf{x}}$ -portion of its oracle. If, for some  $i \leq m$ ,  $\lambda_{a_i,b_i} \oplus \rho_{a_i,b_i} \subseteq L_{\mathbf{x}} \oplus R_{\mathbf{x}}$ , then it outputs 1, since such an  $\mathbf{x}$  will lie in one of the intervals  $(a_i, b_i)$ . For all other  $\mathbf{x}$ , it outputs 0, meaning that it thinks that  $\Phi_e^{S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}}}(e) \uparrow$ . This output 0 may be wrong for certain values  $\mathbf{x}$ , but only for  $\frac{\varepsilon}{2^{e+1}}$ -many. Since this holds for every  $e$ , incorrect outputs can only occur for at most  $\varepsilon$ -much of the interval  $(0, 1)$ . Moreover, it is clear that this procedure is uniform in  $\varepsilon > 0$ .  $\square$

Lemma 10.3 clearly can be repeated for all intervals  $(n, n+1)$  with precision  $\frac{\varepsilon}{4^{|n|+1}}$ , uniformly in  $n \in \mathbb{Z}$ , to make it hold on all of  $\mathbb{R}$ . Finally, we will need to know that this lemma holds not only for the jump operator, but for all iterations and relativizations of it. Here is the full result.

**Theorem 10.4.** *There exists a Turing functional  $\Psi_\varepsilon$ , uniform in the rational  $\varepsilon > 0$ , such that for every set  $S \subseteq \mathbb{N}$  and every fixed  $S$ -decidable presentation  $\mathcal{A}$  of any ordinal*

$\alpha < \omega_1^S$  (with  $\alpha$ -jumps  $C^{(\mathcal{A})}$  defined using this presentation), the “error set”

$$\mathcal{U}_{\varepsilon, S, \mathcal{A}} = \{\mathbf{x} \in [0, 1] : \Psi_\varepsilon^{S^{(\mathcal{A})} \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}}} \neq (S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})^{(\mathcal{A})}\}$$

has measure  $< \varepsilon$  and is an  $S^{(\mathcal{A})}$ -effective union of rational open intervals, uniformly in  $\varepsilon$ ,  $S$ ,  $\alpha$ , and  $\mathcal{A}$ . Moreover, for all  $\mathbf{x}$ ,  $\Psi_\varepsilon^{S^{(\mathcal{A})} \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}}}$  is total.

*Proof.* We give the procedure of  $\Psi_\varepsilon$  on an input  $\langle k, e \rangle$ , using the presentation  $\mathcal{A}$  of  $\alpha$ , whose complete diagram  $\Psi_\varepsilon$  can compute using its  $S$ -oracle.

For elements  $k \in D = \text{dom}(\mathcal{A})$ , the program runs in a highly recursive manner, computing the  $k$ -th column of its output using (finitely much information from) those columns whose numbers  $i$  satisfy  $i < k$  (according to the diagram of  $\mathcal{A}$ ). Since  $\mathcal{A}$  is well-ordered by  $<$ , this procedure is well-founded and will eventually halt. On input  $\langle k, e \rangle$  with  $k \in D$ , the program checks whether  $k$  is 0 or a limit point in  $\mathcal{A}$ . For  $k = 0$ , it uses its oracle to decide whether  $e \in S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}}$  and outputs the answer. For a limit point  $k$ , it decodes  $e = \langle k', e' \rangle$  and runs itself on this pair, since  $\langle \langle k', e' \rangle, k \rangle \in (S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})^{(\mathcal{A})}$  just if  $\langle k', e' \rangle \in (S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})^{(\mathcal{A})}$ . If  $k$  is a successor, then the program finds the immediate predecessor  $i$  of  $k$  under  $<$ , using the diagram of  $\mathcal{A}$ , and attempts to determine whether  $\Phi_e^I \downarrow$ , where  $I$  is the  $i$ -th column of the program’s own output. This requires running the program itself many times, recursively, but only on finitely many inputs (and only on pairs  $\langle i', e' \rangle$  with  $i' \leq i$ ). The procedure is the same as in Lemma 10.3: the program runs until it has found a finite set of some measure  $r$  of initial segments  $\lambda_{a,b} \oplus \rho_{a,b}$  that will cause  $\Phi_e(e)$  with oracle  $\Psi_\varepsilon^{S^{(\mathcal{B}_i)} \oplus \lambda_{a,b} \oplus \rho_{a,b}}$  to halt, and has been told by the oracle  $S^{(\mathcal{A})}$  that the oracles of this form that cause  $\Phi_e(e)$  to halt have total measure at most  $r + \frac{\varepsilon}{2^{k+e+2}}$ . Here we are using  $\Psi_\varepsilon^{S^{(\mathcal{B}_i)} \oplus \lambda_{a,b} \oplus \rho_{a,b}}$  as an approximation to  $(S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})^{(\mathcal{B}_i)}$ , which is the actual content of the  $i$ -th column  $I$ . The approximation is not always correct; below we will consider the measure of the set on which it is incorrect, but for now the important point is that it does always give an output, instead of diverging. Finally,  $\Psi_\varepsilon$  determines whether  $L_{\mathbf{x}} \oplus R_{\mathbf{x}}$  begins with any of the finitely many strings  $\lambda_{a,b} \oplus \rho_{a,b}$  that were found to make  $\Phi_e^{\Psi_\varepsilon^{S^{(\mathcal{B}_i)} \oplus \lambda_{a,b} \oplus \rho_{a,b}}}(e)$  halt. If so, it outputs 1, while if not, it outputs 0.

For elements  $k \notin D$ , the program checks whether any  $j < k$  is the greatest element of  $\mathcal{A}$ , and if so, whether  $k$  is the  $<$ -least number in  $\overline{D}$  greater than that  $j$ . If not, then it immediately outputs 0. If so, then it runs in a similar manner to the program in Lemma 10.3, recursively using the column  $J = \{\langle j, n \rangle : n \in \mathbb{N}\}$  of its own output. For definiteness we specify that on input  $\langle k, e \rangle$  (with  $k \notin D$ ) it uses a tolerance of  $\frac{\varepsilon}{2^{k+e+2}}$  to approximate  $J'$ .

Since  $<$  well-orders  $\mathcal{A}$ , it is readily seen (by induction on columns whose numbers lie in  $D$ , ordered according to  $<$ ) that this program halts on every input  $\langle k, n \rangle$  with  $k \in D$ . If  $\alpha$  is a successor, the same proof then applies to the  $k \notin D$  determined above, and for all other  $k \notin D$  it halted immediately. So the program  $\Psi_\varepsilon$  with arbitrary oracle  $S^{(\mathcal{A})} \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}}$  always computes a total function. Next we consider the set  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$  of

those  $L_{\mathbf{x}}$  on which it fails to compute  $(S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})^{(A)}$ . This can happen in many ways. For the very first jump, when  $k_1$  is the second-to-left point of  $\mathcal{A}$ , the computation on input  $\langle k, e \rangle$  will be incorrect on a set of  $\mathbf{x}$  of measure  $< \frac{\varepsilon}{2^{2+k_1+e}}$ , and so the set of those  $\mathbf{x}$  for which there is an error anywhere in this column has measure  $< \frac{\varepsilon}{2^{k_1+1}}$ . For the second jump, in column number  $k_2$ , there are two reasons the computation could be incorrect: either  $\mathbf{x}$  lies in the set of measure  $< \sum_e \frac{\varepsilon}{2^{2+k_2+e}} = \frac{\varepsilon}{2^{1+k_2}}$  on which the approximation goes wrong, or else the approximation was using an incorrect version of  $(S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})^{(B_{k_1})}$  from column  $k_1$ . However, we already counted those  $\mathbf{x}$  for which the  $k_1$  column was incorrect, so the reals  $\mathbf{x}$  added to the set  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$  on account of column  $k_2$  have total measure  $< \frac{\varepsilon}{2^{1+k_2}}$ . Similarly, for every  $\mathbf{x}$  in  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$ , either there is some  $k \in D$  such that the computation for  $\mathbf{x}$  goes wrong in column number  $k$ , or else  $\alpha$  is a successor and the computation went wrong in column  $k = \min(\bar{D})$ . Therefore, the total measure of  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$  is at most

$$\left( \sum_{k \in D} \frac{\varepsilon}{2^{1+k}} \right) + \frac{\varepsilon}{2^{1+\min(\bar{D})}} \leq \sum_{k \in \mathbb{N}} \frac{\varepsilon}{2^{1+k}} = \varepsilon.$$

The foregoing paragraph already essentially explained how we can uniformly enumerate the open set  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$  from an  $S^{(A)}$ -oracle. Those  $\mathbf{x}$  in  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$  for which the first column  $k_1$  was incorrect all have  $\Psi_{\varepsilon}^{S^{(A)} \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}}}(e) = 0$  for some  $e$  such that eventually  $\Phi_e^{(S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})}(e)$  halted. With the  $S^{(A)}$ -oracle we can run both of these computations with arbitrary strings of the form  $\lambda_{a,b} \oplus \rho_{a,b}$  in place of  $L_{\mathbf{x}}$ . When we find any  $\lambda_{a,b} \oplus \rho_{a,b}$  and  $e$  for which  $\Psi_{\varepsilon}(e)$  gave 0 but  $\Phi_e^{S \oplus \lambda_{a,b} \oplus \rho_{a,b}}(e) \downarrow$ , we enumerate the open interval  $(a, b)$  of  $\mathbb{R}$  into  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$  (since all  $\mathbf{x}$  there have  $\lambda_{a,b} \oplus \rho_{a,b} \subseteq L_{\mathbf{x}} \oplus R_{\mathbf{x}}$ ). For column  $k_2$ , we do the same, using  $\Psi_{\varepsilon}^{S' \oplus L_{\mathbf{x}}}$  to compute the oracle for the computation  $\Phi_e^{(S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})^{(B_{k_1})}}(e)$ ; if it does so incorrectly, then this  $\mathbf{x}$  was already enumerated at the previous step, while if it does so correctly, then we will enumerate  $\mathbf{x}$  into  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$  just if  $\Phi_e^{(S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})^{(B_{k_1})}}(e)$  halts and  $\Psi_{\varepsilon}^{(S^{(A)} \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})^{(B_{k_1})}}(e) = 0$ . Likewise, every  $\mathbf{x} \in \mathcal{U}_{\varepsilon, S, \mathcal{A}}$  will eventually be enumerated by this process, because some finite initial segment  $\lambda \oplus \rho \subseteq L_{\mathbf{x}} \oplus R_{\mathbf{x}}$  must have been adequate to cause both of these events to occur, and we will eventually find that segment (which must be of the form  $\lambda_{a,b} \oplus \rho_{a,b}$ , with  $a = \max(\lambda^{-1}(1) \cup \{0\})$  and  $b = \min(\lambda^{-1}(0) \cup \{1\})$ ) and enumerate  $\mathbf{x}$  into  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$ .

Finally we discuss the situation of isolated points in  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$ . The strings  $\lambda_{a,b} \oplus \rho_{a,b}$  (and their substrings) are not the only possible initial segments of oracles  $L_{\mathbf{x}} \oplus R_{\mathbf{x}}$ : the other possibility occurs when  $\mathbf{x}$  itself is equal to the rational number  $q_j$ , in which case  $q_j \notin L_{\mathbf{x}}$  and  $q_j \notin R_{\mathbf{x}}$ . Initial segments  $\lambda \oplus \rho$  of such strings still satisfy the property  $\min(\lambda^{-1}(0)) \leq \max(\rho^{-1}(0))$ , but they are allowed to have at most one  $j < |\lambda|$  for which  $\lambda(q_j) = \rho(q_j) = 0$  (with  $\rho(i) = 1 - \lambda(i)$  for all other  $i$ ).

Intuitively (and by definition), all rational numbers  $q_j$  should lie in the error set  $\mathcal{U}_{\varepsilon, S, \mathcal{A}}$ . However, our enumeration so far could have omitted some rationals. To

include them (thus showing that  $\mathcal{U}_{\varepsilon,S,\mathcal{A}}$  is indeed open), notice that for each  $q_j$ , we can effectively find the indices  $c$  and  $d$  of two other relevant functionals:

- $\Phi_c^{S^{(\mathcal{A})} \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}}}$  halts just if  $L_{\mathbf{x}}$  contains some rational  $> q_j$  (that is, just if  $\mathbf{x} > q_j$ );
- and
- $\Phi_d^{S^{(\mathcal{A})} \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}}}$  halts just if  $R_{\mathbf{x}}$  contains some rational  $< q_j$  (that is, just if  $\mathbf{x} < q_j$ ).

$\Phi_c$  will contribute an error interval to  $\mathcal{U}_{\varepsilon,S,\mathcal{A}}$  of the form  $(q_j, q)$  for some rational  $q > q_j$ , and  $\Phi_d$  will contribute one of the form  $(r, q_j)$ . Therefore, including  $q_j$  itself in  $\mathcal{U}_{\varepsilon,S,\mathcal{A}}$  keeps it open, as now the entire interval  $(r, q)$  is contained in  $\mathcal{U}_{\varepsilon,S,\mathcal{A}}$ . So, along with the error intervals previously enumerated into  $\mathcal{U}_{\varepsilon,S,\mathcal{A}}$ , we also enumerate the interval  $(r, q)$  defined here for this  $q_j$ , noting that  $r$  and  $q$  were computed effectively from  $j$ . We do the same for every other rational  $q_j$  in  $\mathbb{Q}$  as well. Thus  $\mathcal{U}_{\varepsilon,S,\mathcal{A}}$  is still  $S^{(\mathcal{A})}$ -effectively open, and the countably many new points do not change its measure.  $\square$

As mentioned above, it is easy to repeat this process for all intervals  $[n, n + 1]$ . We record this as our final corollary.

**Corollary 10.5.** *There exists a Turing functional  $\Psi_\varepsilon$ , uniform in the rational  $\varepsilon > 0$ , such that for every set  $S \subseteq \mathbb{N}$  and every fixed  $S$ -decidable presentation  $\mathcal{A}$  of any ordinal  $\alpha < \omega_1^S$  (with  $\alpha$ -jumps  $C^{(\mathcal{A})}$  defined using this presentation), the “error set”*

$$\mathcal{U}_{\varepsilon,S,\mathcal{A}} = \{\mathbf{x} \in \mathbb{R} : \Psi_\varepsilon^{S^{(\mathcal{A})} \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}}} \neq (S \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}})^{(\mathcal{A})}\}$$

*has measure  $< \varepsilon$  and is an  $S^{(\mathcal{A})}$ -effective union of rational open intervals, uniformly in  $\varepsilon$ ,  $S$ ,  $\alpha$ , and  $\mathcal{A}$ . Moreover, for all  $\mathbf{x}$ ,  $\Psi_\varepsilon^{S^{(\mathcal{A})} \oplus L_{\mathbf{x}} \oplus R_{\mathbf{x}}}$  is total.*  $\square$

## REFERENCES

- [1] C.J. Ash & J. Knight; *Computable structures and the hyperarithmetical hierarchy* (Amsterdam: North-Holland Publishing Co., 2000).
- [2] C.G. Jockusch Jr. & P.E. Schupp; Generic computability, Turing degrees, and asymptotic density. *Journal of the London Mathematical Society* **85** (2) 2012, 472–490.
- [3] A. Kechris; *Classical Descriptive Set Theory* (New York: Springer-Verlag, 1995).
- [4] H. Royden; *Real Analysis, third edition* (Englewood Cliffs, NJ: Pearson, 1988).
- [5] W. Rudin; *Real and Complex Analysis* (New York: McGraw Hill Book Company, 1987).
- [6] R.I. Soare; *Recursively Enumerable Sets and Degrees* (New York: Springer-Verlag, 1987).
- [7] J. Stillwell; Decidability of the “almost all” theory of degrees. *Journal of Symbolic Logic* **37** (3) 1972, 501–506.
- [8] K. Weihrauch; *Computable Analysis: An Introduction* (Berlin: Springer, 2000).

DEPT. OF MATHEMATICS, QUEENS COLLEGE, & PH.D. PROGRAMS IN MATHEMATICS & COMPUTER SCIENCE, GRADUATE CENTER, CITY UNIVERSITY OF NEW YORK, USA

*E-mail address:* Russell.Miller@qc.cuny.edu

*URL:* <http://qcpages.qc.cuny.edu/~rmiller/>