

Revising Type-2 Computation and Degrees of Discontinuity

Martin Ziegler^{1,2}

*Japan Advanced Institute of Science and Technology (JAIST)
School of Information Science, 1-1 Asahidai, Nomi, Ishikawa 923-1292*

Abstract

By the sometimes so-called *Main Theorem* of Recursive Analysis, every computable real function is necessarily continuous. Weihrauch and Zheng (TCS 2000), Brattka (MLQ 2005), and Ziegler (ToCS 2006) have considered different relaxed notions of computability to cover also discontinuous functions. The present work compares and unifies these approaches. This is based on the concept of the *jump* of a representation: both a TTE-counterpart to the well known recursion-theoretic jump on Kleene's Arithmetical Hierarchy of hypercomputation; and a formalization of revising computation in the sense of Shoenfield. We also consider Markov and Banach/Mazur oracle-computation of discontinuous functions and characterize the computational power of Type-2 nondeterminism to coincide with the first level of the Analytical Hierarchy.

Keywords: Hypercomputation, Recursion Theory, Type-2 Theory of Effectivity (TTE), Kleene and Borel Hierarchies

1 Introduction

Every computable real function f is necessarily continuous!

Computability here refers to effective $(\rho \rightarrow \rho)$ -evaluation in the sense of x input to a Turing machine by means of a ρ -name, that is a fast converging sequence of rationals (q_n) ; and $y = f(x)$ output in form of a similar sequence (p_m) . Equivalently: the pre-image $f^{-1}[V]$ of an open set $V \subseteq \mathbb{R}$ is open; and the mapping $V \mapsto f^{-1}[V]$ is effective in the sense that, giving an enumeration of (the centers and radii of) open rational balls exhausting V , a Turing machine can output a similar list exhausting $f^{-1}[V]$. This amounts to $(\theta_{<} \rightarrow \theta_{<})$ -computability of $V \mapsto f^{-1}[V]$.

How can we relax this notion to include also discontinuous functions $f : X \rightarrow \mathbb{R}$? ³

¹ Supported by JSPS grant PE05501. The author wishes to express further gratitude to his Japanese host professor Hajime ISHIHARA for exuberant assistance and latitude!

² Email: mziegler@jaist.ac.jp

³ Here and in the sequel, X denotes a fixed recursively open (i.e. $\theta_{<}$ -computable) subset of \mathbb{R}^D for some $D \in \mathbb{N}$.

- i) A representation (and thus a computability notion) for L^2 -functions or distributions is devised easily and naturally [28]; but evaluation $x \mapsto f(x)$ thereon is neither effective nor mathematically well-defined.
- ii) Granting a Type-2 machine access to an oracle like, say, the Halting problem increases its recursion-theoretic power but does not lift the topological restriction to continuous real functions; see e.g. [26, Lemma 8].
- iii) Weihrauch and Zheng (2000) have considered $(\rho \rightarrow \rho_<)$ -computable functions where the output representation $\rho_<$ encodes $y = f(x) \in \mathbb{R}$ as a rational sequence (p_m) with $y = \sup_m p_m$. Such functions are in general only lower *semi*-continuous [22], that is, the pre-image $f^{-1}[V]$ is open for every $V = (y, \infty)$. As a matter of fact, f is $(\rho \rightarrow \rho_<)$ -computable if and only if $a \mapsto f^{-1}[(y, \infty)]$ is $(\rho_> \rightarrow \theta_<)$ -computable [22, Theorem 4.5].
- iv) Motivated by (a different) work of Zheng and Weihrauch (2001), [26] introduced representations $\rho', \rho'_<, \rho'', \dots, \rho^{(d)}, \rho^{(d)}_<$ weakening ρ and $\rho_<$. A real number x is ρ -computable *relative* to the Halting problem \emptyset' if and only if it is ρ' -computable [10]. More generally, x is ρ -computable *relative* to $\emptyset^{(d)}$ if and only if x is $\rho^{(d)}$ -computable [27]; similarly for $\rho^{(d)}_<$. These representations thus parallel the levels Σ_d of Kleene's Arithmetical Hierarchy.
- v) Brattka relaxes the pre-image mapping $V \mapsto f^{-1}[V]$ from being open and effectively open and instead considers Σ_d -measurability [3]. This condition requires that $f^{-1}[V]$ be a Σ_d set in Borel's topological hierarchy. For its ground level $\Sigma_1(X)$ of open subsets of X , he thus recovers classical continuity; $\Sigma_2(X)$ consists of the F_σ sets, and so on. The mapping $V \mapsto f^{-1}[V]$ must furthermore be effective in the sense that, given a $\theta_<$ -name of V , a Type-2 machine must be able to obtain a name of $f^{-1}[V]$ in terms of the natural representation $\delta_{\Sigma_d(X)}$ of $\Sigma_d(X)$; $\delta_{\Sigma_1(X)} \equiv \theta_<$.
- vi) Real nondeterminism had been introduced in [25, 26, Section 5]. A corresponding machine computing $y = f(x)$ may make a binary choice at each step, as long as any *infinite* output sequence (q_n) constitutes a ρ -name of y . This notion has been shown to include all $(\rho \rightarrow \rho^{(d)})$ -computable functions [26, Theorem 28].

Notice that proceeding from $(\rho \rightarrow \rho)$ -computability to $(\rho \rightarrow \rho^{(d)})$ -computability amounts to weakening the information to be output for the *values* (image) of the function f under consideration; whereas proceeding from effective Σ_1 -measurability (equivalent to $(\rho \rightarrow \rho)$ -computability) to, say, effective Σ_{d+1} -measurability amounts to weakening the encoding on the *pre*-image side (i.e. the domain) of f .

1.1 Overview

The present work unifies and extends approaches iii), iv), and v) above. Some main results are collected in the following

Theorem 1.1 *Fix a function $f : X \rightarrow \mathbb{R}$ and $d \in \mathbb{N}$.*

- a) *f is $(\rho \rightarrow \rho^{(d-1)})$ -computable if and only if it is effectively Σ_d -measurable.*

- b) f is $(\rho \rightarrow \rho_{<}^{(d-1)})$ -computable if and only if the mapping $\mathbb{R} \ni y \mapsto f^{-1}[(y, \infty)] \in \Sigma_d(X)$ is well-defined and $(\rho_{>} \rightarrow \delta_{\Sigma_d(X)})$ -computable.
- c) There exists a nondeterministically computable total real function which is not $(\rho \rightarrow \rho^{(d)})$ -computable for any $d \in \mathbb{N}$ whatsoever.

In particular, weakly evaluable functions (in the sense of iv) range arbitrarily high on Borel's taxonomy of discontinuity but are strictly succeeded by nondeterminism (vi). Theorem 1.1a) also gives one explanation for the dominance in [3] of the Borel classes Σ_d over the (seemingly more symmetric ones) Δ_d .

Claims a) and b) in the above theorem turn out to actually hold even uniformly in f . To this end, we introduce in Section 4 the notion of Σ_d -semimeasurability and a representation for according functions: a generalization unifying both [27] and [3]. The central concept in the present work is that of the *jump* α' of a representation α (Section 2). For the case $\alpha = \rho$, it coincides with the notion from [26] and simplifies the proofs therein.

Motivated by *revising* computation, Section 3 considers an equally natural but different kind of jump operator on representations. The power of Type-2 Nondeterminism [25,26, Section 5] is the topic of Section 5. And before concluding, we also briefly dive into oracle-supported Markov and Banach/Mazur computability (Section 6).

2 The *Jump* of a Representation

Ho has shown that a real number x is ρ -computable (that is admits effective approximations by a fast converging rational sequence) *relative* to the Halting problem \emptyset' if and only if x is the (unconditional) limit of a computable rational sequence [10, Theorem 9]. This has suggested the alternative name ρ' for the *naive* Cauchy representation encoding x as an ultimately converging rational sequence. Another example, Brattka has weakened (and extended) the representation $\theta_{<} \equiv \delta_{\Sigma_1(X)}$ for open sets to $\delta_{\Sigma_d(X)}$ mentioned above. The present section unifies these and several other notions.

We start with Cantor space $\{0, 1\}^\omega$ which is usually and canonically represented by the identity ι [21, Definition 3.1.2.1].

Definition 2.1 Let the representation $\iota' : \subseteq \{0, 1\}^\omega \rightarrow \{0, 1\}^\omega$ encode an infinite string $\bar{\sigma} \in \{0, 1\}^\omega$ as (the pairing of) a sequence of infinite strings ultimately converging to $\bar{\sigma}$.

This amounts to the *naive* Cauchy representation of the effective metric Cantor space [2, Section 6]. An ι' -name for $(\sigma_n)_n$ is thus (an ι -name for) some $((\tau_{(n,m)})_n)_m \in \{0, 1\}^\omega$ such that, for each $n \in \mathbb{N}$, $\sigma_n = \lim_{m \rightarrow \infty} \tau_{(n,m)}$. The name ι' , reminiscent of the recursion-theoretic *jump*, is justified because Shoenfield's Limit Lemma immediately yields

Remark 2.2 Let \mathcal{O} denote an arbitrary oracle. An infinite string is $(\iota-)$ computable *relative* to \mathcal{O}' if and only if it is ι' -computable relative to \mathcal{O} .

Moreover we have

- Lemma 2.3** a) Every $((i \rightarrow i)-)$ computable string function $F : \subseteq \{0,1\}^\omega \rightarrow \{0,1\}^\omega$ is also $(i' \rightarrow i')$ -computable;
- b) more precisely the **apply operator** $(F, \bar{\sigma}) \mapsto F(\bar{\sigma})$ is $(\eta^{\omega\omega} \times i' \rightarrow i')$ -computable.
- c) Every $(i' \rightarrow i')$ -continuous string function $F : \subseteq \{0,1\}^\omega \rightarrow \{0,1\}^\omega$ is (Cantor-)continuous.
- d) Whenever $\alpha : \subseteq \{0,1\}^\omega \rightarrow A$ is a representation for A , then so is $\alpha \circ i'$.
- e) $\alpha \preceq \beta$ implies $\alpha \circ i' \preceq \beta \circ i'$.

In b), $\eta^{\omega\omega}$ denotes a natural representation for continuous string functions [21, Section 2.3].

Proof.

- a) follows from b).
- b) Let $\tilde{\tau}_m := F(\bar{\tau}_m)$ where $F : \subseteq \{0,1\}^\omega \rightarrow \{0,1\}^\omega$ is continuous. Then $\lim_m \tilde{\tau}_m = F(\lim_m \bar{\tau}_m)$.
- c) See [2, Section 6].
- d) immediate.
- e) Let F denote a computable string function converting α -names to β -names. By a), F has a computable $(i' \rightarrow i')$ -realization $G : \subseteq \{0,1\}^\omega \rightarrow \{0,1\}^\omega$. This G converts $(\alpha \circ i')$ -names to $(\beta \circ i')$ -names. \square

The rest of this section relates several known representations to ones of the form $\alpha \circ i'$ for some α .

2.1 Weak Real Representations

Recall from [25,26, Section 2] the following

Definition 2.4 Consider the representations of \mathbb{R} where a real y is encoded as

- ρ : a rational sequence (p_m) such that $|y - p_m| \leq 2^{-m}$ (i.e. fast convergence)
- $\rho_<$: a rational sequence (p_m) such that $y = \sup_m p_m$ (i.e. lower approximation)
- ρ' : a rational sequence (p_m) such that $y = \lim_m p_m$ (i.e. ultimate convergence)
- $\rho'_<$: a rational sequence (p_m) such that $y = \sup_m \inf_n p_{\langle m,n \rangle}$ (equivalently: \liminf)
- ρ'' : a rational sequence (p_m) such that $y = \lim_m \lim_n p_{\langle m,n \rangle}$
- $\rho''_<$: a rational sequence (p_m) such that $y = \sup_m \inf_n \sup_k p_{\langle m,n,k \rangle}$
- \vdots
- $\rho^{(d)}$: a rational sequence (p_m) such that $y = \lim_{n_1} \lim_{n_2} \lim_{n_3} \cdots \lim_{n_d} p_{\langle n_1, n_2, \dots, n_d \rangle}$

Since the limit (ρ' of course coincides with the well-known *naive* Cauchy-representation ρ_{Cn} .) These encodings constitute a hierarchy

$$\rho \preceq \rho_{<} \preceq \rho' \preceq \rho'_{<} \preceq \rho'' \preceq \rho''_{<} \preceq \dots \preceq \rho^{(d)} \preceq \rho^{(d)}_{<} \preceq \dots$$

of representations introduced in [26, Section 2.2]. This hierarchy correspond to—and is in particular as strict as—Kleene's Arithmetical Hierarchy of hypercomputation

$$\Delta_1 \subsetneq \Sigma_1 \subsetneq \Delta_2 \subsetneq \Sigma_2 \subseteq \Delta_3 \subseteq \Sigma_3 \subseteq \dots \subseteq \Delta_{d+1} \subseteq \Sigma_{d+1} \subseteq \dots$$

in the following way: A real number y is $\rho^{(d)}$ -computable if and only if y is $\rho^{(k)}$ -computable *relative* to $\emptyset^{(d-k)}$ for some (or, equivalently, for every) $0 \leq k \leq d$; and y is $\rho^{(d)}_{<}$ -computable if and only if y is $\rho^{(k)}_{<}$ -computable relative to $\emptyset^{(d-k)}$, see [27, Section 7]. Notice how this extends Shoenfield's Limit Lemma from discrete to the continuous realm [27, Section 4].

2.2 Jump of the Cauchy Representation

Proposition 2.5 $\rho \circ i' \equiv \rho'$.

In combination with Remark 2.2, this implies [10, Theorem 9]; and together with Lemma 2.3b) it includes [26, Scholium 17].

Proof. A $(\rho \circ i')$ -name for $x \in \mathbb{R}$ is (basically) a sequence of rational sequences eventually stabilizing (elementwise) to a fast converging Cauchy sequence $(q_{(n,\infty)})_n$; that is a double sequence $(q_{(n,m)})$ in \mathbb{Q} such that

$$\forall n \exists m_0 \forall m \geq m_0 : q_{(n,m)} = q_{(n,m_0)} \wedge |x - q_{(n,m_0)}| \leq 2^{-n}.$$

\preceq : For each m , let $(q_{(1,m)}, q_{(2,m)}, \dots, q_{(N_m,m)})$ denote the longest initial part of $(q_{(1,m)}, \dots, q_{(m,m)})$ satisfying

$$|q_{(n,m)} - q_{(n',m)}| \leq 2^{1-n} \quad \forall 1 \leq n \leq n' \leq N_m. \quad (1)$$

Since $(q_{(n,\infty)})_n$ is a ρ -name and due to the eventual stabilization, $N_m \rightarrow \infty$ as $m \rightarrow \infty$. Also, the sequence $(N_m)_m$ is computable from the above input. Consider the following algorithm, starting with empty output tape:

For each $m = 1, 2, \dots$, test whether the initial parts of $q_{(\cdot,m)}$ and $q_{(\cdot,m-1)}$ up to N_m coincide: $(q_{(1,m)}, \dots, q_{(N_m,m)}) = (q_{(1,m-1)}, \dots, q_{(N_m,m-1)})$? (For notational convenience, set $q_{n,0} := \infty$ and $N_0 := 0$.) If so, then obviously $N_m \geq N_{m-1}$; so append (the possibly empty sequence) $(q_{(N_{m-1},m)}, \dots, q_{(N_m,m)})$ to the output. Otherwise let n_m be maximal with $(q_{(1,m)}, \dots, q_{(n_m,m)}) = (q_{(1,m-1)}, \dots, q_{(n_m,m-1)})$; obviously $n_m < N_m$, so append $(q_{(n_m,m)}, \dots, q_{(N_m,m)})$ to the output in this case.

It remains to show that that yields a valid ρ' -name for x . Let $\epsilon = 2^{1-n}$. Then $|q_{(n,\infty)} - q_{(n',\infty)}| \leq \epsilon$ for all $n' \geq n$ because $q_{(n,\infty)}$ constitutes a ρ -name. Moreover due to stabilization, there exists some maximal m with $q_{(n,m)} \neq q_{(n,m-1)}$. During the phase no. m corresponding to that last change, the above algorithm will detect

$n_m < N_m$ and thus output (a finite sequence beginning with) $q_{(n,m)}$. Moreover as $q_{(n,\cdot)}$ afterwards does not change anymore, all elements $q_{(n',m')}$ appended subsequently will have $n' \geq n$ and $m' \geq m$; in fact $N_{m'} \geq n' \geq n_{m'} \geq n$, hence $|q_{(n',m')} - q_{(n,m)}| \leq \epsilon$ because $q_{(n,m')} = q_{(n,m)}$ and due to Equation (1). Therefore the output constitutes a (naive) Cauchy sequence converging to x .

$\underline{\Sigma}$: Let $(q_n)_n$ be a sequence in \mathbb{Q} ultimately converging to x . There exists an increasing sequence $(n_m)_m$ in \mathbb{N} such that

$$\forall k \geq n_m : |q_{n_m} - q_k| \leq 2^{-m-1} . \quad (2)$$

The subsequence $(q_{n_m})_m$ constitutes a ρ -name for x . For each single m , Condition (2) can be falsified (formally: is co-r.e. in the input). A Turing machine is therefore able to iteratively try for n_m all integer values from n_{m-1} on and fail only finitely often for each m .

Trial no. ℓ thus yields a sequence $(n'_{(\ell,m)})_{m \leq \ell}$ of length ℓ such that, for each m , $n'_{(\cdot,m)}$ eventually stabilizes to n_m satisfying (2). By artificially extending each finite sequence to an infinite one, we obtain a $\rho \circ \iota'$ -name for x . \square

2.3 Jump of Lower Real Representation

Our next result includes, in view of Lemma 2.3a+c), [26, Theorem 11a+b) and Theorem 15b+c)] because $(\rho \rightarrow \rho_{<})$ -continuity implies lower-semicontinuity and $(\rho_{<} \rightarrow \rho_{<})$ -continuity requires monotonicity [22].

Proposition 2.6 $\rho_{<} \circ \iota' \equiv \rho'_{<}$.

Proof. A $(\rho_{<} \circ \iota')$ -name for $x \in \mathbb{R}$ amounts to a sequence of rational sequences eventually stabilizing (elementwise) to a sequence approaching x from below, that is a double sequence $(q_{(n,m)})$ in \mathbb{Q} such that

$$\forall n \exists m_0 = m_0(n) \forall m \geq m_0 : q_{(n,m)} = q_{(n,m_0)} \wedge x = \sup_n q_{(n,m_0(n))} .$$

$\underline{\Sigma}$: Since the limit (which exists) coincides with the least accumulation point, we have

$$x = \sup_n \lim_m q_{(n,m)} = \sup_n \sup_j \inf_{m \geq j} q_{(n,m)} = \sup_{\langle n,j \rangle} \inf_m \begin{cases} q_{(n,m)} : m \geq j \\ \infty : m < j \end{cases}$$

deduced a $\rho'_{<}$ -name for x .

$\underline{\Sigma}$: Let $(q_{(n,m)})$ be the given double sequence in \mathbb{Q} with $x = \sup_n \inf_m q_{(n,m)}$. We may suppose that all single sequences $q_{(n,\cdot)}$, $n \in \mathbb{N}$, are monotonically nonincreasing; and that the single sequence $(\inf_m q_{(n,m)})_n$ is nondecreasing: by proceeding (in either order!) from $q_{(n,m)}$ to $\min_{k \leq m} q_{(n,k)}$ and to $\max_{\ell \leq n} q_{(\ell,m)}$, respectively. Moreover one can assert each single sequence $q_{(n,\cdot)}$ to eventually stabilize, thus yielding a $\rho_{<} \circ \iota'$ -name of x : Consider for $m \in \mathbb{N}$ the function $\lfloor \cdot \rfloor_m : \mathbb{Q} \rightarrow \mathbb{Q}$

mapping every rational to the next lower dyadic rational having denominator 2^{-m} ; formally: $a/b \mapsto \frac{\lfloor a \cdot 2^m / b \rfloor}{2^m}$ where $\lfloor \cdot \rfloor = \lfloor \cdot \rfloor_0$ denotes the usual floor function on integers. Then proceeding from $q_{(n,m)}$ to $\lfloor q_{(n,m)} \rfloor_m$ satisfies this requirement without affecting $x = \sup_n \inf_m q_{(n,m)}$. □

2.4 Jump of the Weierstraß Representation

The limit of a uniformly converging sequence of polynomials is of course continuous again. Weierstraß has shown that the converse holds as well: Any continuous function on a compact set is the uniform limit of a sequence of polynomials. This leads to the Weierstraß Representation $[\rho \rightarrow \rho]$ of the class $C(K)$ of continuous functions $f : K \rightarrow \mathbb{R}$ for compact $K := [0, 1]^D$: a name of $f \in C(K)$ is (an encoding of the degrees and coefficients of) a sequence of polynomials $P_n \in \mathbb{Q}[X]$ with

$$\sup_{x \in K} |f(x) - P_n(x)| =: \|f - P_n\| \stackrel{!}{\leq} 2^{-n}. \quad (3)$$

By the famous **Effective Weierstraß Theorem**, it is equivalent to several other natural representations of $C(K)$ [21, Section 6.1]. [25, Lemma 12b] and [26, Lemma 22] employ a representation $[\rho \rightarrow \rho]'$ for $C(K)$ where the required *fast* uniform convergence bound 2^{-n} in Equation (3) is weakened to ‘ultimate’ uniform convergence $\|f - P_n\| \rightarrow 0$. This kind of *naive* Weierstraß representation, too, results from a jump:

Proposition 2.7 $[\rho \rightarrow \rho] \circ \iota' \equiv [\rho \rightarrow \rho]'$.

This result includes [10, Theorem 16]. The proof proceeds similarly to that of Proposition 2.5 because Equations (1) and (2) are still decidable and co-r.e. when replacing rational numbers q with rational polynomials Q and absolute value $|q|$ with maximum norm $\|Q\|$:

Fact 2.8 *Given $q_0, \dots, q_m, b \in \mathbb{Q}$ (in binary encoding, say), $\sup_{0 \leq x \leq 1} |q_0 + q_1x + \dots + q_mx^m| = b$ is decidable: by virtue of constructive root bounds, see e.g. [11].*

2.5 Iterated Jumps

Climbing up in Kleene’s Arithmetical Hierarchy corresponds to iterated jumps of the Halting problem. We proceed similarly with our hierarchy of representations:

Definition 2.9 Let $\iota^{(d+1)} := \iota^{(d)} \circ \iota' = \iota' \circ \iota^{(d)}$.

Straight forward inductive application of Remark 2.2 shows that $\iota^{(d)}$ -computability is equivalent to ι -computability relative to $\emptyset^{(d)}$. If F and G are partial $(\iota \rightarrow \iota')$ -computable string functions, then their composition $G \circ F$ is $(\iota \rightarrow \iota'')$ -computable by Lemma 2.3a).

Theorem 2.10 *For each $d \in \mathbb{N}$, it holds $\rho^{(d)} \equiv \rho \circ \iota^{(d)}$ and $\rho_{<}^{(d)} \equiv \rho_{<} \circ \iota^{(d)}$.*

Proof. The induction start $d = 1$ has been treated in Propositions 2.5 and 2.6, respectively. Since a $\rho^{(d+1)}$ -name of $x \in \mathbb{R}$ is the join of $\rho^{(d)}$ -names of elements x_n with $x = \lim_n x_n$, Proposition 2.5 together with Lemma 2.3e) also provides the induction step; similarly for $\rho_{<}^{(d+1)}$. \square \square

As a consequence, we obtain the following extension of [26, Theorems 11 and 15]:

Corollary 2.11 *Fix $f : X \rightarrow \mathbb{R}$.*

- a) *If f is $(\rho^{(d)} \rightarrow \rho^{(d)})$ -continuous, then it is continuous. If f is $(\rho^{(d)} \rightarrow \rho^{(d)})$ -computable, then it is also $(\rho^{(d+1)} \rightarrow \rho^{(d+1)})$ -computable,*
- b) *If f is $(\rho^{(d)} \rightarrow \rho_{<}^{(d)})$ -continuous, then it is lower semi-continuous. If f is $(\rho^{(d)} \rightarrow \rho_{<}^{(d)})$ -computable, then it is also $(\rho^{(d+1)} \rightarrow \rho_{<}^{(d+1)})$ -computable.*
- c) *If f is $(\rho_{<}^{(d)} \rightarrow \rho_{<}^{(d)})$ -continuous, then it is monotonically nondecreasing. If f is $(\rho_{<}^{(d)} \rightarrow \rho_{<}^{(d)})$ -computable, then it is also $(\rho_{<}^{(d+1)} \rightarrow \rho_{<}^{(d+1)})$ -computable.*

The proof of [26, Theorem 11] covers as many as five pages of text and treated only very small values of d . Now it boils down to a mere application of Lemma 2.3a+c) inductively in d .

2.6 Borel Set Representations

The representation $\theta_{<}$ encodes an open subset U of X as a list of (centers and radii) of open rational balls exhausting U . For a topological space X , the Borel Hierarchy starts with the class $\Sigma_1(X)$ of open subsets U of X and proceeds inductively from $\Sigma_d(X)$ to the class $\Sigma_{d+1}(X)$ of countable unions $\bigcup_m (X \setminus S_m)$ over complements of sets S_m from $\Sigma_d(X)$. Brattka has renamed $\theta_{<}$ to $\delta_{\Sigma_1(X)}$ and generalized it to higher order Borel sets:

Definition 2.12 Consider the following representations of Borel subsets of X :

$\delta_{\Sigma_1(X)}$ encodes $U \in \Sigma_1(X)$ as a list B_m of open rational balls s.t. $U = \bigcup_m B_m$

$\delta_{\Sigma_2(X)}$ encodes $S \in \Sigma_2(X)$ as a list B_m of open rational balls such that

$$S = \bigcup_m (X \setminus \bigcup_n B_{\langle m, n \rangle})$$

\vdots

$\delta_{\Sigma_d(X)}$ encodes $S \in \Sigma_d(X)$ as (the join of) Σ_{d-1} -names of sets $S_m \in \Sigma_{d-1}(X)$ such that $S = \bigcup_m (X \setminus S_m)$.

It turns out that these natural representations are related to jumps, too:

Proposition 2.13 $\theta_{<} \circ \iota' \preceq \delta_{\Sigma_2(X)} \Big|_{\Sigma_1(X)}.$

Recall that $\Sigma_1(X)$ denotes the class of open subsets of X which $\theta_{<} \equiv \delta_{\Sigma_1(X)}$ is a representation for. Of course the restriction of $\delta_{\Sigma_2(X)}$ is thus necessary for the equivalence to make sense.

Proof. A $(\theta_{<} \circ \iota')$ -name for $U \in \mathcal{O}(X)$ consists of two rational double sequences $(c_{(n,m)})$ and $(r_{(n,m)})$ such that each single sequence $c_{(n,\cdot)}$ and $r_{(n,\cdot)}$, $n \in \mathbb{N}$, eventually stabilizes to some $c_{(\cdot,\infty)}$ and $r_{(\cdot,\infty)}$ where $U = \bigcup_{n \in \mathbb{N}} B(c_{(n,\infty)}, r_{(n,\infty)})$ and $B(c, r)$ denotes the open ball with center c and radius r .

Both representations admit effective countable unions: apply Lemma 2.3a) to [21, Example 5.1.19.1] and see [3, Proposition 3.2(5)], respectively. It therefore suffices to show them equivalent on open rational balls, that is we may suppose w.l.o.g. $U = B(c_m, r_m)$ for all $m \geq m_0$.

So let

$$A_n := \begin{cases} \overline{B}(c_m, r_m \cdot (1 - 2^{-n})) : (c_k, r_k) = (c_{k+1}, r_{k+1}) \forall k \geq m \\ \emptyset : \text{otherwise} \end{cases},$$

so $U = \bigcup_n A_n$. Moreover the closed set A_n can be $\psi_{>}$ -computed, uniformly in n and the given sequences (c_m) and (r_m) : start generating $\overline{B}(\cdots)$; if the co-r.e. condition “ $\forall k \geq m$ ” eventually turns out to fail, the machine may still revert to a $\psi_{>}$ -name for \emptyset by adding further negative information to the output. Hence we obtain a $\delta_{\Sigma_2(X)}$ -name for U . \square

The following extends [21, Example 5.1.17.2] and [27, Corollary 6.6a)]:

Example 2.14 For reals $a < b$, the open interval $U = (a, b) \subseteq \mathbb{R}$ is Σ_d -computable if and only if $+b$ and $-a$ are both $\rho_{<}^{(d-1)}$ -computable.

Conjecture 2.15 *The converse of Proposition 2.13 holds as well: $\delta_{\Sigma_2(X)} \big|_{\Sigma_1(X)}^{\perp} \preceq \theta_{<} \circ \iota'$.*

Problem 2.16 *Recalling the weak representations of regular sets $\overline{\psi}_{<}^{\mathcal{R}}$ and $\overline{\theta}_{>}$ from [24, Definition 3.3], characterize them in terms of ι' and some known representations!*

3 Revising Computation

This section provides some motivation and related background to the jump α' of a representation α as well as for a different kind of jump $\widehat{\alpha}$ to be introduced in Section 3.3 below.

An important (though somewhat hidden) point in the definition of a Type-2 machine is that its output tape be *one-way*; compare e.g. [21, top of p.15]. This condition allows to abort a real number computation as soon as the desired precision is reached, knowing that this preliminary approximation will *not be reverted*. It also is crucial for the Main Theorem to hold.

In the Type-1 setting, revising computations have been studied well. Here a machine writes only a finite string, but it does not terminate and may *revert* its output an arbitrary finite number of times. The model with this semantics goes under such names as Limiting [6], Trial-and-Error [16], Inductive [4], or General [17]

Turing Machines. It is motivated by the capabilities of early display terminals (see Section 3.2 below) as well as by Shoenfield’s Limit Lemma.

A sequence $(\sigma_n)_n$ of finite strings (Type-1) converges (to a finite string) if and only if the sequence $\sigma_{n,i}$ of i -th symbols eventually stabilizes for each i . For infinite strings (Type-2 setting) however, one has to carefully distinguish both conditions: symbol-wise convergence underlies Definition 2.1 whereas overall stabilization will be required in Definition 3.6.

Both appear naturally when formalizing the output displayed by a (not necessarily terminating) program to a terminal as explicated in Section 3.2. They also arise as input fed to a streaming algorithm:

3.1 Revising Input: Streams

Many practical applications are desired to run ‘forever’: a scheduler, a router, a monitor all are not supposed to terminate but to continue processing the stream of data presented to them. This has led to the prospering field of **Data Stream Algorithms**⁴. It distinguishes various ways in which the input can be presented to the program [12, Section 4.1]:

- In the *Time Series Model*, all data items (binary digits, say) are to be enumerated in order; in particular, they must not later be reverted.

This corresponds in TTE to the identity presentation ι of an infinite string by itself.

- The *Turnstile Model* on the other hand permits (finitely many) later updates to previously enumerated items.

This corresponds to the presentation ι' from Definition 2.1.

3.2 Revising Output: Terminals

Recall the two most basic **ascii** control characters understood already by the earliest text display consoles [23]: BS and CR. The first, called “backspace”, moves the cursor left by one position, thus allowing the last printed symbol to be overwritten; whereas the second, “carriage return”, commands to restart output from the beginning (of the present line).

Example 3.1 The character sequence

G o o d _ b y e CR H e l l o , _ M r s BS BS BS w o r l d
will display as: Hello, world.

So consider a program generating an infinite sequence of characters *including* BS and CR; how do they appear on an (infinitely long, one-line) display? Let us require that each character position does settle down eventually, leading ultimately to the display of a truly infinite string (without BS and CR).

⁴ which usually focuses on the (space) complexity of randomized approximations of discrete problems, though

Definition 3.2 A \boxtimes -name of $\bar{\sigma} \in \{0, 1\}^\omega$ is an infinite string over $\{0, 1, \boxed{\text{CR}}, \boxed{\text{BS}}\}$ which leads to the display of $\bar{\sigma}$ in the above sense.

Now this is exactly what we had already considered in Definition 2.1:

Remark 3.3 $\boxtimes \equiv \iota'$.

Each occurrence of the control character $\boxed{\text{CR}}$ leads to the entire display being purged. In order for already the first character to eventually stabilize, a valid \boxtimes -name may thus contain at most finitely many $\boxed{\text{CR}}$'s. Let us now consider a terminal incapable of processing $\boxed{\text{BS}}$, that is, restrict \boxtimes to $\{0, 1, \boxed{\text{CR}}\}^\omega$. Then any valid name will make the displayed text settle down not only character-wise but globally. This motivates a different jump operator $\alpha \mapsto \hat{\alpha}$ formally introduced in the sequel:

Remark 3.4 $\boxtimes|_{\{0, 1, \boxed{\text{CR}}\}^\omega} \equiv \hat{\iota}$.

Hopefully you, most valued reader, are now indeed curious enough to read on and learn about the computational power induced by this

3.3 Other Kind of Jump

[26, Section 5.1] characterizes the computational power of Chadzelek and Hotz' quasi-strongly δ - \mathbb{Q} -analytic machines in terms of Type-2 machines by introducing the representation ρ_H as follows:

Definition 3.5 A ρ_H -name for $x \in \mathbb{R}$ is a sequence $(q_n)_n$ in \mathbb{Q} such that

$$\exists N \in \mathbb{N} \forall n \geq N : |q_n - x| \leq 2^{-n}$$

This representation is *non*-uniformly equivalent to ρ yet uniformly (in terms of reducibility that is) lies strictly between ρ and ρ' .

Similarly to Section 2, we now generalize this particular construction into a generic way:

Definition 3.6 For a representation $\alpha : \subseteq \{0, 1\}^\omega \rightarrow A$, write $\hat{\alpha} := \alpha \circ \hat{\iota}$.

The representation $\hat{\iota} : \subseteq \{0, 1\}^\omega \rightarrow \{0, 1\}^\omega$ in turn encodes an infinite string $\bar{\sigma} = (\sigma_n)_n \in \{0, 1\}^\omega$ as a sequence of infinite strings $\bar{\tau}_m = (\tau_{(n,m)})_n \in \{0, 1\}^\omega$, $m \in \mathbb{N}$, such that there is some $M \in \mathbb{N}$ with $\bar{\tau}_m = \bar{\sigma}$ for all $m \geq M$.

In contrast to Definition 2.1, the sequence $(\bar{\tau}_m)$ is thus required to ultimately stabilize *uniformly* in the position index n .

In view of Claim f) of the following lemma, Claims a) to c) generalize [26, Lemma 31]; and Claims d)+e) generalize [26, Proposition 32b+a].

Lemma 3.7 Fix representations α of A and β of B .

- a) An element $a \in A$ is α -computable if and only if it is $\hat{\alpha}$ -computable.
- b) It holds $\alpha \preceq \hat{\alpha} \preceq \alpha'$. The converse reductions are in general discontinuous.

- c) For any function $f : \subseteq A \rightarrow B$, $(\alpha \rightarrow \hat{\beta})$ -computability is equivalent to $(\hat{\alpha} \rightarrow \hat{\beta})$ -computability.
- d) Every $(\alpha \rightarrow \beta)$ -computable function f is also $(\hat{\alpha} \rightarrow \hat{\beta})$ -computable; even uniformly in f .
- e) An $(\hat{\alpha} \rightarrow \hat{\beta})$ -computable function need not be $(\alpha \rightarrow \beta)$ -continuous.
- f) $\hat{\rho} \equiv \rho_H$.

Proof. It suffices to treat the case $(A, \alpha) = (B, \beta) = (\{0, 1\}^\omega, \iota)$ —except for f) of course.

- a) Encode the M from Definition 3.6 into the machine computing $(\tau_{(n,M)})_{(n,M)}$ and make it output $(\tau_{(n,M)})_n$.
- b) The positive claims are immediate, the negative ones are straight-forward discontinuity arguments.
- c) By a), every $(\iota \rightarrow \hat{\iota})$ -computable function is $(\hat{\iota} \rightarrow \hat{\iota})$ -computable, too. For the converse implication, take the Type-2 Machine \mathcal{M} converting ι -names for $x \in \mathbb{R}$ to $\hat{\iota}$ -names for $y = f(x)$. Let $(\bar{\sigma}_m)$ be given with $\bar{\sigma}_m = \bar{\sigma}_M$ for all $m \geq M$, $M \in \mathbb{N}$ unknown.

Now simulate \mathcal{M} on $\bar{\sigma}_1$ (implicitly supposing $M = 1$) and simultaneously check that $\bar{\sigma}_1 = \bar{\sigma}_m$ for all $m \geq 1$. If (or, rather, when) the latter turns out to fail, restart under the presumption $M = 2$ and so on. The check will however succeed after finitely many tries (after reaching the ‘true’ M used in the input). We thus obtain a finite sequence of output strings, that is a valid $\hat{\iota}$ -name for $f(\bar{\sigma})$.

- d) The apply operator $(F, \bar{\sigma}) \rightarrow F(\bar{\sigma})$ is $(\eta^{\omega\omega} \times \hat{\iota} \rightarrow \hat{\iota})$ -computable: if $\bar{\tau}_m = \bar{\sigma}$ for all $m \geq M$, then also $F(\bar{\tau}_m) = F(\bar{\sigma})$ for all $m \geq M$.
- e) Consider the discontinuous function $F(1^\omega) := 1^\omega$, $F(\bar{\sigma}) := 0^\omega$ for $\bar{\sigma} \neq 1^\omega$. We assert it to be $(\iota \rightarrow \hat{\iota})$ -computable; the claim the follows by c).

Given $\bar{\sigma} = (\sigma_n)_n$, for each $n = 1, 2, \dots$ test $\sigma_n = 1$ and, as long as this holds, append 1 to the output. Otherwise restart the output to 0^ω . Since this restart takes place (if at all) after finite time, we obtain in either case a valid $\hat{\iota}$ -name.

- f) Given $(\bar{\tau}_m)_m$ with $\bar{\tau}_m = \bar{\sigma}$ for all $m \geq M$, consider for each m the longest initial segment of τ_m constituting the beginning of a valid ρ -name. This is computable because $\text{dom}(\rho)$ is r.e.; and it yields a ρ_H -name for $\rho(\bar{\sigma})$, i.e. we have “ $\hat{\rho} \preceq \rho_{\text{Cn}}$ ”. The converse reduction proceeds similarly. \square

4 Real Hypercomputation and Degrees of Discontinuity

Computability of a function $f : X \rightarrow \mathbb{R}$ in Recursive Analysis means $(\rho \rightarrow \rho)$ -computability; equivalently [21, Lemma 6.1.7]: the pre-image $f^{-1}[V] = \{x : f(x) \in V\}$ of any open $V \subseteq \mathbb{R}$ is again open (that is in $\Sigma_1(X)$) and the pre-image mapping $V \mapsto f^{-1}[V]$ is $(\theta_{<} \rightarrow \delta_{\Sigma_1(X)})$ -computable. In particular, every computable real

function is necessarily continuous.

How can we extend the notion of computability to incorporate also (at least some) discontinuous functions?

Recalling the introduction, one may

iii) consider $(\rho \rightarrow \rho_{<})$ -computable functions.

A function $f : X \rightarrow \mathbb{R}$ is $(\rho \rightarrow \rho_{<})$ -continuous if and only if it is *lowersemicontinuous*, i.e., $f^{-1}[V]$ is open for any $V = (y, \infty)$. It is $(\rho \rightarrow \rho_{<})$ -computable if and only if the mapping $\mathbb{R} \ni y \mapsto f^{-1}[(y, \infty)] \in \Sigma_1(X)$ is well-defined and $(\rho_{>} \rightarrow \delta_{\Sigma_1(X)})$ -computable [22, Theorem 4.5(1) and Corollary 5.1(2)]. A natural representation (here denoted by $[\rho \rightarrow \rho_{<}]$) of lower-semicontinuous functions on X encodes f as the join of the $\theta_{<}$ -names of the open sets $f^{-1}[(y, \infty)]$, $y \in \mathbb{Q}$; cf. [22, Definition 3.2].

Another approach due to Brattka, it is equally natural to

v) consider functions $f : X \rightarrow \mathbb{R}$ for which the pre-image $f^{-1}[V]$ of any open $V \subseteq \mathbb{R}$ belongs to the Borel class $\Sigma_d(X)$ (is Σ_d -measurable) and the mapping $V \mapsto f^{-1}[V]$ is $(\theta_{<} \rightarrow \delta_{\Sigma_d(X)})$ -computable (called *effectively Σ_d -measurable*).

The comprehensive paper [3] thoroughly studies this notion and its consequences. It is as general as to include also partial and multi-valued functions on arbitrary computable metric spaces but in that respect goes beyond our purpose. [3] also introduces a natural representation $\delta_{\Sigma_d(X \rightarrow \mathbb{R})}$ for Σ_d -measurable functions as the join of $\delta_{\Sigma_d(X)}$ -names of the sets $f^{-1}[V]$, V running through all open rational balls. For reasons which will be come clear soon, the present work prefers to write $[\rho \rightarrow \rho^{(d-1)}]$ for $\delta_{\Sigma_d(X \rightarrow \mathbb{R})}$.

Let us unify these two Approaches iii) and v):

Definition 4.1 Call $f : X \rightarrow \mathbb{R}$ be Σ_d -lowersemimeasurable if $f^{-1}[(y, \infty)] \in \Sigma_d(X)$ for all $y \in \mathbb{R}$. It is *effectively Σ_d -lowersemimeasurable* if $\mathbb{R} \ni y \mapsto f^{-1}[(y, \infty)]$ is in addition $(\rho_{>} \rightarrow \delta_{\Sigma_d(X)})$ -computable. The representation $[\rho \rightarrow \rho_{<}^{(d)}]$ of all Σ_{d+1} -lowersemimeasurable functions is defined to encode $f : X \rightarrow \mathbb{R}$ as the join of $\delta_{\Sigma_d(X)}$ -names of $f^{-1}[(y, \infty)]$ for all $y \in \mathbb{Q}$.

Obviously $[\rho \rightarrow \rho^{(d)}] \equiv [\rho \rightarrow \rho_{<}^{(d)}] \wedge [\rho \rightarrow \rho_{>}^{(d)}]$, exploiting $f^{-1}[U \cap V] = f^{-1}[U] \cap f^{-1}[V]$ and [3, Proposition 3.2(4)] as well as $\rho^{(d)} \equiv \rho_{<}^{(d)} \wedge \rho_{>}^{(d)}$ by Lemma 2.3a) and Theorem 2.10.

The main result of the present section connects these notions to weak function evaluation $(\rho \rightarrow \rho^{(d)})$ and $(\rho \rightarrow \rho_{<}^{(d)})$ —recall Section 1, Approach iv)—with the representations from Section 2.1. In fact, justifying the above names for representations of (lowersemi)measurable functions, we show

Theorem 4.2 a) The uniformly characteristic function $\mathbf{1} : \Sigma_d(X) \times X \rightarrow \{0, 1\}$, defined by $(S, \mathbf{x}) \mapsto \mathbf{1}_S(\mathbf{x}) := 1$ if $\mathbf{x} \in S$ and $\mathbf{1}_S(\mathbf{x}) := 0$ if $\mathbf{x} \notin S$, is $(\delta_{\Sigma_d(X)} \times \rho \rightarrow \rho_{<}^{(d-1)})$ -computable.

- b) The **apply operator** $(f, x) \mapsto f(x)$ of Σ_{d+1} -lowersemimeasurable functions on X is $([\rho \rightarrow \rho_{<}^{(d)}] \times \rho \rightarrow \rho_{<}^{(d)})$ -computable.
- c) Every $(\rho \rightarrow \rho_{<}^{(d)})$ -continuous function $f : X \rightarrow \mathbb{R}$ is Σ_{d+1} -lowersemimeasurable; every $(\rho \rightarrow \rho_{<}^{(d)})$ -computable one is effectively Σ_{d+1} -lowersemimeasurable, uniformly in f given by an $\eta^{\omega\omega}$ -name of a realization.

Claims b) and c) together immediately establish the non-uniform Theorem 1.1b) which in turn yields Theorem 1.1a).

An alternative proof of Theorem 1.1a), however only for $d \geq 3$, could proceed by induction [3, Corollary 9.6] and exploit that the pointwise limit f of a sequence f_n of $(\rho \rightarrow \rho^{(d-2)})$ -computable functions is $(\rho \rightarrow \rho^{(d-1)})$ -computable.

Proof (Theorem 4.2)

- a) By induction on d , starting with $d = 1$: Given a ρ -name of $\mathbf{x} \in X$ and a $\theta_{<}$ -name of an open $U \subseteq U$, membership “ $\mathbf{x} \in U$ ” is semi-decidable; so output 0s while uncertain and start writing 1s as soon as membership has been established: this yields a $\rho_{<}$ -name of $\mathbf{1}_U(\mathbf{x})$.

Now let $S = \bigcup_n (X \setminus S_n) \in \Sigma_{d+1}(X)$ be given by the joint $\delta_{\Sigma_d(X)}$ -names of $S_n \in \Sigma_d(X)$, $n \in \mathbb{N}$. By induction hypothesis, $\rho_{<}^{(d-1)}$ -compute the respective values $y_n := \mathbf{1}_{S_n}(\mathbf{x})$. Since $\mathbf{x} \in S \Leftrightarrow \exists n : \mathbf{x} \notin S_n$, we have $\mathbf{1}_S(\mathbf{x}) = \sup_n (1 - y_n)$.

- b) Given (a ρ -name of) $x \in X$, compute for all $y \in \mathbb{Q}$ a $\delta_{\Sigma_{d+1}(X)}$ -name of $S_y := f^{-1}[(y, \infty)]$. Claim a) yields from that a $\rho_{<}^{(d)}$ -name of $z_y := \mathbf{1}_{S_y}(x)$, that is $z_y = 1$ in case $x \in S_y$ and $z_y = 0$ in case $x \notin S_y$. Easy scaling converts that to $z'_y = a$ in case $f(x) > y$ and to $z'_y = -\infty$ in case $f(x) \leq y$. We finally obtain a $\rho_{<}^{(d)}$ -name of $\sup_y z'_y = f(x)$ because $\bar{\mathbb{R}}^{\mathbb{N}} \ni (x_n)_n \mapsto \sup_n x_n \in \bar{\mathbb{R}}$ is obviously $((\rho_{<}^{(d)})^{\mathbb{N}} \rightarrow \rho_{<}^{(d)})$ -computable.

- c) To start with, recall the proof of [22, Theorem 3.7] the classical case

$d = 0$: Evaluate f simultaneously on all $x \in X$ to obtain rational sequences $p_{x,n}$ with $f(x) = \sup_n p_{x,n}$. More precisely, using feasible countable (as opposed to infeasible uncountable) dove-tailing, simulate the machine evaluating f on all initial parts of ρ -names of $x \in X$, that is on all finite rational sequences $\bar{q} = (q_1, q_2, \dots, q_N)$ with $N \in \mathbb{N}$ and $|q_n - q_k| \leq 2^{-n} \forall n \leq k \leq N$. For each \bar{q} , we obtain as output a finite rational sequence $(p_{\bar{q},m})_{m \leq M}$. Observe that \bar{q} is initial segment of a ρ -name to any $x \in \overline{B_{\bar{q}}} := \bigcup_{n=1}^{N(\bar{q})} \overline{B}(q_n, 2^{-n})$, $\overline{B_{\bar{q}}}$ having non-empty interior. Hence

$$\exists m : p_{\bar{q},m} > a \quad \Leftrightarrow \quad \forall x \in \overline{B_{\bar{q}}} : f(x) > a \quad \Leftrightarrow \quad \exists x \in \overset{\circ}{\overline{B_{\bar{q}}}} : f(x) > a$$

of regular languages over infinite strings [20] as well as by the famous Immerman–Szelepcsenyi concept of nondeterministic *function* computation [15, Theorem 7.6] and by fair nondeterminism [19], we introduced in [25,26, Section 5] the *nondeterministic Type-2 Model*:

Definition 5.1 Let A and B be sets with respective representations $\alpha : \subseteq \{0, 1\}^\omega \rightarrow A$ and $\beta : \subseteq \{0, 1\}^\omega \rightarrow B$. A function $f : \subseteq A \rightarrow B$ is called *nondeterministically* $(\alpha \rightarrow \beta)$ -computable if some nondeterministic one-way Turing Machine \mathcal{M} ,

- upon input of any α -name $\bar{\sigma} \in \{0, 1\}^\omega$ for some $a \in \text{dom}(f)$,
- has a computation which outputs a β -name for $b = f(a)$ and
- every infinite computation of \mathcal{M} on $\bar{\sigma}$ outputs a β -name for $b = f(a)$.

A subset L of A is *nondeterministically decidable* if the characteristic function $\mathbf{1}_L : A \rightarrow \{0, 1\} \times \{-\}^\omega$ is nondeterministically $(\alpha \rightarrow \iota)$ -computable.

While admittedly even less realistic than a classical \mathcal{NP} -machine, its capabilities have turned out to exhibit (in addition to closure under composition) particular structural elegance: All presentations $\rho^{(d)}$, $d \in \mathbb{N}$, can nondeterministically be converted to and from each other. Hence we may simply speak of nondeterministic computability and observe that this notion includes all functions $(\rho \rightarrow \rho^{(d)})$ -computable for *any* d , that is by Theorem 1.1a) the entirety of Brattka’s hierarchy of effective measurability.

Remark 5.2 In [25, Definition 14], we had defined nondeterministic computability in a way with the third condition in Definition 5.1 requiring that any infinite *output* of \mathcal{M} on $\bar{\sigma}$ constitutes a β -name for $b = f(a)$. Since any infinite output requires infinite computation but not vice versa, this may seem to lead to a different notion. However both do coincide: \mathcal{M} may additionally guess and verify a function $F : \mathbb{N} \rightarrow \mathbb{N}$ such that the n -th symbol is output after $F(n)$ steps. If F has been guessed incorrectly (and in particular if, for the given input $\bar{\sigma}$, no such F exists at all), then this can be detected within finite time and abort the computation, thus complying with the (only seemingly stronger) Definition 5.1.

The question of exactly characterizing the power of these machines, left open in [26, Section 5], is now answered in terms of the Analytical Hierarchy:

Theorem 5.3 For $L \subseteq \mathbb{N}$, the characteristic function $\mathbf{1}_L : \mathbb{N} \rightarrow \{0, 1\} \times \{-\}^\omega$ is nondeterministically computable if and only if $L \in \Delta_1^1$.

In particular, the power of Type-2 nondeterminism goes *strictly beyond* effective measurability; see Corollary 5.6 below.

The following notion turns out as both natural and useful in the proof of Theorem 5.3:

Definition 5.4 A set $L \subseteq \mathbb{N}$ is *nondeterministically semi-decidable* if there exists a nondeterministic Turing machine \mathcal{M} which, upon input of $x \in \mathbb{N}$,

- has a computational path which outputs an infinite string in case $x \in L$;

- in case $x \notin L$, aborts after finite time on all computational paths.

L is *nondeterministically enumerable* if a nondeterministic Turing machine \mathcal{M} without input

- has a computational path which outputs a list $(x_n)_n$ of integers with $L = \{x_n : n \in \mathbb{N}\}$;
- every infinite computation of \mathcal{M} prints a list $(x_n)_n$ of integers with $L = \{x_n : n \in \mathbb{N}\}$.

Nondeterministic enumerability thus amounts to nondeterministic computability of an En-name, cf. [21, Definition 3.1.2.5]. Surprisingly, it turns out as equivalent not to nondeterministic *semi*-decidability but to nondeterministic decidability:

Proposition 5.5 *With respect to Type-2 nondeterminism, it holds:*

- $\text{En} \equiv \text{Cf}$, where the latter refers to the representation of the powerset of \mathbb{N} enumerating a set's members in order [21, Definition 3.1.2.6].
- $L \subseteq \mathbb{N}$ is decidable if and only if it has a computable Cf-name; equivalently: both L and its complement are semi-decidable.
- $L \subseteq \mathbb{N}$ is semi-decidable if and only if $L \in \Sigma_1^1$.

Proof.

- “Cf \preceq En” holds already deterministically. For the converse we are given a list $(x_n)_n$ of integers enumerating L . Guess a function $F : \mathbb{N} \rightarrow \mathbb{N}$ with $x_n \geq m \forall n \geq F(m)$: Such obviously F exists; and an incorrect guess can be detected within finite time. Knowing F , we can determine and sort all restrictions $L \cap [1, m]$, $m \in \mathbb{N}$.
- Immediate.
- Let $L \in \Sigma_1^1$. By the Normal Form Theorem—see e.g. [13, Proposition IV.2.5]—

$$L = \{x \in \mathbb{N} \mid \exists \bar{b} = (b_n)_n \in \{0, 1\}^\omega \ \forall n \in \mathbb{N} : P(x, n, \langle b_1, \dots, b_n \rangle)\} \quad (4)$$

for some decidable predicate P . A nondeterministic Type-2 machine \mathcal{M} , given x , may therefore guess \bar{b} , check $P(x, n, \bar{b}|_{\leq n})$ to hold (and output a dummy symbol) for each $n \in \mathbb{N}$ and, when it fails, abort within finite time: This yields nondeterministic semi-decision of L .

Conversely let L be semi-decided by \mathcal{M} . Then $x \in \mathbb{N}$ belongs to L if and only if there exists a sequence $(b_n)_n$ of guesses $b_n \in \{0, 1\}^\omega$ such that \mathcal{M} makes at last n steps on x and \bar{b} . The latter predicate $P(x, n, \langle b_1, \dots, b_n \rangle)$ being decidable, L is of the form (4). \square

Claims b) and c) together yield Theorem 5.3. Moreover we have

Corollary 5.6 *There is nondeterministically computable real c which does not be-*

long to (any finite⁵ level of) Weihrauch and Zheng's Arithmetical Hierarchy of real numbers.

The constant function $f(x) \equiv c$ establishes Theorem 1.1c).

Proof (Corollary 5.6) Take some hyperarithmetical but not arithmetical $L \subseteq \mathbb{N}$, that is, $L \in \Delta_1^1 \setminus \Sigma_0^1$; see e.g. [14, Theorem §16.1.XI] or [13, Corollary IV.2.23]. Since L is nondeterministically decidable, it leads to a nondeterministically $\rho_{b,2}$ -computable real $c := \sum_{n \in L} 2^{-n} \in \mathbb{R}$; compare [21, Theorem 4.1.13]. Were c $\rho^{(d)}$ -computable for some $d \in \mathbb{N}$, its (unique!) binary expansion would be decidable relative to $\emptyset^{(d)}$ [27, Theorem 7.8], that is in $\Sigma_{d+1} \subseteq \Sigma_0^1$, contradiction. \square

6 Markov Oracle–Computation

Returning to Approach ii) in Section 1, oracle access to the, say, Halting problem does not permit computational evaluation $x \mapsto f(x)$ of any discontinuous real function f in the sense of Recursive Analysis, that is with respect to input x and output $f(x)$ by means of fast convergent rational sequences. Other notions of effectivity due to A.A. Markov, Jr. [21, Section 9.6] and S. Mazur [21, Section 9.1] restrict real functions to computable arguments $x \in \mathbb{R}_c$.

Definition 6.1 A function $f : \subseteq \mathbb{R}_c \rightarrow \mathbb{R}$ is *Markov-computable* if it admits a (classically, i.e. discretely) computable Markov realization, that is a function $F : \subseteq \mathbb{N} \rightarrow \mathbb{N}$ such that, whenever e is Gödel index of a Turing machine \mathcal{M}_e ρ -computing $x \in \text{dom}(f)$, then $F(e)$ is defined and index of a machine ρ -computing $f(x)$. Call f *BM-computable* if $(f(x_n))_n$ is a computable real sequence whenever $(x_n)_n \in \text{dom}(f)$ is.

A $(\rho \rightarrow \rho)$ -computable function is obviously Markov-computable which in turn implies BM-computability. Moreover Mazur's theorem asserts every *total* BM-computable function to be continuous; and Markov-computability of a *total* real function requires $(\rho \rightarrow \rho)$ -computability according to Tseitin [21, Theorem 9.6.6]. See [7,8] for a thorough comparison of all these notions.

Now, as opposed to $(\rho \rightarrow \rho)$ -computability, Markov-computability does benefit even topologically from oracle access:

Example 6.2 The discontinuous sign function $\text{sgn} : \mathbb{R}_c \rightarrow \{-1, 0, +1\}$ is, relative to the Halting problem \emptyset' , both Markov-computable and BM-computable.

Observe that in accordance with Definition 6.1, sgn is considered on the computable reals only.

Proof. Given a Gödel index e of some machine \mathcal{M}_e computing x , modify \mathcal{M}_e slightly to abort in case $x \neq 0$. Feed this new machine's index \tilde{e} into Halting oracle. A negative answer implies $x = 0$; the remaining cases $x < 0$ and $x > 0$ are trivial. Similarly for BM-computability. \square

⁵ It may however belong to a transfinite one [1].

We are currently working the following generalizations of Mazur’s and Tseitin’s Theorems:

Problem 6.3 Fix a total function $f : \mathbb{R}_c \rightarrow \mathbb{R}$.

- a) If f is Markov-computable relative to \emptyset' , then it is Σ_2 -measurable?
- b) If f maps every ρ' -computable sequence to a ρ' -computable one, then it is continuous?
- c) Characterize the class of total functions Markov-computable relative to \emptyset' !
- d) How about higher degrees?

7 Conclusion

We have characterized $(\rho \rightarrow \rho^{(d)})$ -computable functions $f : X \rightarrow \mathbb{R}$ to coincide with Brattka’s condition of effective Σ_{d+1} -measurability; and shown his representation $\delta_{\Sigma_{d+1}(X \rightarrow \mathbb{R})}$ to be natural for the class of $(\rho \rightarrow \rho^{(d)})$ -continuous functions. We furthermore have characterized $(\rho \rightarrow \rho_{<}^{(d)})$ -computable functions and, extending work of Weihrauch and Zheng, found a natural representation for the class of $(\rho \rightarrow \rho_{<}^{(d)})$ -continuous ones.

Problem 7.1 Find a simple characterization of the respective classes of $(\rho^{(k)} \rightarrow \rho^{(d)})$ -continuous, $(\rho^{(k)} \rightarrow \rho_{<}^{(d)})$ -continuous, and $(\rho_{<}^{(k)} \rightarrow \rho_{<}^{(d)})$ -continuous functions with $1 \leq k \leq d$ arbitrary but fixed; and devise natural representations for them.

If α is an *admissible* representation, then $\alpha^{(d)}$ is usually not for $d \geq 1$, at least not in the strict sense. This seems to call for Schröder’s theory of generalized admissibility [18]. On the other hand, Corollary 2.11 succeeded well without this notion.

Acknowledgments:

The author is grateful to K. Weihrauch, V. Brattka, P. Hertling, and X. Zheng for constant support and seminal discussions. Section 5 results from research initiated by U. Kohlenbach and D. Norman during CiE’05. I would also like to repeat the many thanks expressed in Footnote 1.

References

- [1] G. Barmpalias: “A Transfinite Hierarchy of Reals”, pp.163–172 in *Mathematical Logic Quarterly* vol.49(2) (2003).
- [2] V. Brattka, P. Hertling: “Topological Properties of Real Number Representations”, pp.241–257 in *Theoretical Computer Science*
- [3] V. Brattka: “Effective Borel measurability and reducibility of functions”, pp.19–44 in *Mathematical Logic Quarterly* vol.51 (2005).
- [4] M. Burgin: “Algorithmic Complexity of Recursive and Inductive Algorithms”, pp.31–60 in *Theoretical Computer Science* vol.317 (2004).

- [5] T. Chadzelek, G. Hotz: “Analytic Machines”, pp.151–167 in *Theoretical Computer Science* vol.**219** (1999).
- [6] E.M. Gold: “Limiting Recursion”, pp.28–48 in *Journal of Symbolic Logic* vol.**30:1** (1965).
- [7] P. Hertling: “A Sequentially Computable Function that is not Effectively Continuous at any Point”, pp.131–148 in *Proc. 2nd International Conference on Computability and Complexity in Analysis* (CCA’05).
- [8] P. Hertling: “A Banach–Mazur computable but not Markov computable function on the computable real numbers”, pp.227–246 in *Ann. Pure Appl. Logic* vol.**132** (2005).
- [9] P.G. Hinman: “*Recursion–Theoretic Hierarchies*”, Springer Perspectives in Mathematical Logic (1978).
- [10] C.-K. Ho: “Relatively Recursive Real Numbers and Real Functions”, pp.99–120 in *Theoretical Computer Science* vol.**210** (1999).
- [11] M. Mignotte: “Identification of Algebraic Numbers”, pp.197–204 in *Journal of Algorithms* vol.**3** (1982).
- [12] S. Muthukrishnan: “Data Streams: Algorithms and Applications”, vol.**1:2** of *Foundations and Trends[®] in Theoretical Computer Science* (Editor: Madhu Sudan), now publishers (2005); preliminary version: <http://www.cs.rutgers.edu/~muthu/stream-1-1.ps>
- [13] P. Odifreddi: “*Classical Recursion Theory*”, North-Holland (1989).
- [14] H. Rogers, Jr.: “*Theory of Recursive Functions and Effective Computability*”, MIT Press (1987).
- [15] C.H. Papadimitriou: “*Computational Complexity*”, Addison-Wesley (1994).
- [16] H. Putnam: “Trial and Error Predicates”, pp.49–57 in *Journal of Symbolic Logic* vol.**30:1** (1965).
- [17] J. Schmidhuber: “Hierarchies of Generalized Kolmogorov Complexities and Nonenumerable Universal Measures Computable in the Limit”, pp.587–612 in *International Journal of Foundations of Computer Science* vol.**13:4** (2002).
- [18] M. Schröder: “Admissible Representations in Computable Analysis”, pp.471–480 in *Proc. 2nd Conference on Computability in Europe* (CiE’06), Springer LNCS vol.**3988**.
- [19] E. Spaan, L. Torenvliet, P. van Emde Boas: “Nondeterminism, Fairness and a Fundamental Analogy”, pp.186–193 in *The Bulletin of the European Association for Theoretical Computer Science* (EATCS Bulletin) vol.**37** (1989).
- [20] Thomas, W.: “Automata on Infinite Objects”, pp.133–191 in *Handbook of Theoretical Computer Science*, vol.**B** (Formal Models and Semantics), Elsevier (1990).
- [21] K. Weihrauch: “*Computable Analysis*”, Springer (2000).
- [22] K. Weihrauch, X. Zheng: “Computability on continuous, lower semi-continuous and upper semi-continuous real functions”, pp.109–133 in *Theoretical Computer Science* vol.234 (2000).
- [23] Wikipedia, the free encyclopedia: “Computer terminal”, http://en.wikipedia.org/wiki/Computer_terminal
- [24] M. Ziegler: “Computability on Regular Subsets of Euclidean Space”, pp.157–181 in *Mathematical Logic Quarterly* **48+1** (2002).
- [25] M. Ziegler: “Computability and Continuity on the Real Arithmetic Hierarchy and the Power of Type-2 Nondeterminism”, pp.562–571 in *Proc. 1st Conference on Computability in Europe* (CiE’2005), Springer LNCS vol.**3526**.
- [26] M. Ziegler: “Real Hypercomputation and Continuity”, to appear in *Theory of Computing Systems*; pre-print <http://arxiv.org/cs.LG/0508069>
- [27] X. Zheng, K. Weihrauch: “The Arithmetical Hierarchy of Real Numbers”, pp.51–65 in *Mathematical Logic Quarterly* vol.**47** (2001).
- [28] N. Zhong, K. Weihrauch: “Computability Theory of Generalized Functions”, pp.469–505 in *J. ACM* vol.**50** (2003).