arXiv:cs/0508069v2 [cs.LO] 22 Feb 2006

Real Hypercomputation and Continuity*

Martin Ziegler**

University of Paderborn, 33095 GERMANY

Abstract. By the sometimes so-called *Main Theorem* of Recursive Analysis, every computable real function is necessarily continuous. We wonder whether and which kinds of *hyper*computation allow for the effective evaluation of also discontinuous $f : \mathbb{R} \to \mathbb{R}$. More precisely the present work considers the following three super-Turing notions of real function computability:

– relativized computation; specifically given oracle access to the Halting Problem $\emptyset'$ or its jump $\emptyset''$;
– encoding input $x \in \mathbb{R}$ and/or output $y = f(x)$ in weaker ways also related to the Arithmetic Hierarchy;
– non-deterministic computation.

It turns out that any $f : \mathbb{R} \to \mathbb{R}$ computable in the first or second sense is still necessarily continuous whereas the third type of hypercomputation does provide the required power to evaluate for instance the discontinuous Heaviside function.

1 Motivation

What does it mean for a Turing Machine, capable of operating only on discrete objects, to compute a real number x:

$\rho_{\mathrm{b},2}$: To determine its binary expansion, i.e., to decide $A \subseteq \mathbb{N}$ with $x = \sum_{n\in A} 2^{-n}$?

ρ_{Cn}: To compute a sequence (q_n) of rational numbers eventually converging to x?

ρ: To compute a *fast* convergent sequence $(q_n) \subseteq \mathbb{Q}$ for x, i.e. with $|x - q_n| \leq 2^{-n}$ (in other words: to approximate x with effective error bounds)?

$\rho_<$: To approximate x from below, i.e., to compute (q_n) such that $x = \sup_n q_n$?

All these notions make sense in being closed under arithmetic operations like addition and multiplication. In fact they are well (known equivalent to variants) studied in literature***; e.g. [Tur36], [BH02], [Tur37], [Wei01] in order.

Now what does it mean for a Turing Machine $\mathcal{M}$ to compute a real function $f : \mathbb{R} \to \mathbb{R}$? Most naturally it means that $\mathcal{M}$ realizes effective evaluation $x \mapsto f(x)$ in that, upon input of $x \in \mathbb{R}$ given in one of the above ways, it outputs $y = f(x)$ also in one (not necessarily the same) of the above ways.

* An extended abstract of this work, mostly lacking proofs, has appeared as [Zie05].

** supported by the German Research Fundation DFG with project Zi1009/1-1

*** Their above names by indexed Greek letters are taken from [Wei01, Section 4.1].

And, again, many possible combinations have already been investigated. For instance the standard notion of real function computation in Recursive Analysis [Grz57,PER89,Ko91,Wei01] refers (or is equivalent) to input and output given according to ρ. Here, the Main Theorem of Computable Analysis implies that any computable f will necessarily be continuous [Wei01, THEOREM 4.3.1].

We are interested in ways of lifting this restriction, that is, in the following

Question 1. Does hypercomputation in some sense permit the computational evaluation of (at least certain) discontinuous real functions?

That is related to the Church-Turing Hypothesis: A Turing Machine's ability to simulate *every* physical process would imply all such processes to behave continuously—a property G. LEIBNIZ was convinced of ("*Natura non facit saltus*") but which we nowadays know to be violated for instance by the Quantum Hall Effect awarded a Nobel Prize in 1985. Since this (nor any other) discontinuous physical process cannot be simulated on a classical Turing Machine, it constitutes a putative candidate for a system capable of realizing hypercomputation.

1.1 Summary

The standard (and indeed the most general) way of turning a Turing Machine into a hypercomputer is to grant it access to an oracle like, say, the Halting Problem $\emptyset'$ or its iterated jumps like $\emptyset''$ and $\emptyset^{(d)}$ in KLEENE's Arithmetic Hierarchy. However regarding computational evaluation of real functions, closer inspection in Section 3.1 reveals that this Main Theorem relies solely on information rather than recursion theoretic arguments and therefore requires continuity also for oracle-computable real functions with respect to input and output of form ρ. (For the special case of an $\emptyset'$–oracle, this had been observed in [Ho99, THEOREM 16].)

A second idea, applicable to real but not to discrete computability, changes the input and output representation for x and $y = f(x)$ from ρ to a weaker form like, say, ρ_{Cn}. This relates to the Arithmetic Hierarchy, too, however in a different way: Computing x in the sense of ρ_{Cn} is equivalent to computing x in the sense of ρ [Ho99, THEOREM 9] *relative* (i.e., given access) to the Halting Problem $\emptyset'$ and thus suggests to write $\rho' := \rho_{\text{Cn}}$. Most promisingly, the Main Theorem [Wei01, COROLLARY 3.2.12] which requires continuity of $(\rho \to \rho)$–computable real functions applies to ρ but not to ρ' because the latter lacks the technical property of *admissibility*.

It therefore came to quite a surprise when BRATTKA and HERTLING established that any $(\rho' \to \rho')$–computable f (that is, with respect to input x and output $f(x)$ encoded according to ρ_{Cn}) still satisfies continuity; see [Wei01, EXERCISE 4.1.13d] or [BH02, SECTION 6].

Section 3.2 contains an extension of this and a series of related results. Specifically we manage to prove that continuity is necessary for $(\rho'' \to \rho'')$–computability of f; here, $\rho \precneqq \rho' \precneqq \rho'' \precneqq \dots$ denote the first levels of an entire hierarchy of real number representations explained in Lemma 5 which emerge naturally from the Real Arithmetic Hierarchy of Weihrauch and Zheng [ZW01].

In Section 4, we closer investigate the two approaches to real function hypercomputation. Specifically it is established (Section 4.1) that the hierarchy of real number representation actually does yield a hierarchy of weakly computable real functions. Furthermore a comparison of both oracle-supported and weakly computable (and each hence necessarily continuous) real functions in Section 4.2 reveals a relativized version of the Effective Weierstraß Theorem to fail.

Our third approach to real hypercomputation (Section 5) finally allows the Turing Machines under consideration to behave *non*deterministically. Remarkably and in contrast to the classical (Type-1) theory, this does significantly increase their principal capabilities. For example, all quasi-strongly δ–$\mathbb{Q}$–analytic functions in the sense of Chadzelek and Hotz [CH99]—and in particular many discontinuous real functions—now become computable as well as conversion among the aforementioned representations ρ_{Cn} and $\rho_{\mathrm{b},2}$.

2 Arithmetic Hierarchy and Reals

In [Ho99], Ho observed an interesting relation between computability of a real number x in the respective senses of ρ and ρ_{Cn} in terms of oracles: $x = \lim_n q_n$ for an (eventually convergent) computable rational sequence (q_n) iff x admits a *fast* convergent rational sequence computable with oracle $\emptyset'$, that is, a sequence $(p_m) \subseteq \mathbb{Q}$ recursive in $\emptyset'$ with $|x - p_m| \leq 2^{-m}$. This suggests to use ρ' synonymously for ρ_{Cn}; and denoting by $\Delta_1\mathbb{R} = \mathbb{R}_c$ the set of reals computable in the sense of Recursive Analysis (that is with respect to ρ), it is therefore natural to write, in analogy to Kleene's classical Arithmetic Hierarchy, $\Delta_2\mathbb{R}$ for the set of all $x \in \mathbb{R}$ computable with respect to ρ'. Weihrauch and Zheng extended these considerations and obtained for instance [ZW01, Corollary 7.3] the following characterization of $\Delta_3\mathbb{R}$: A real $x \in \mathbb{R}$ admits a fast convergent rational sequence recursive in $\emptyset''$ iff x is computable in the sense of ρ'' defined as follows:

$\boldsymbol{\rho''}$**:** $x = \lim_i \lim_j q_{\langle i,j \rangle}$ for some computable rational sequence (q_n)

where $\langle \cdots \rangle : \mathbb{N}^* \to \mathbb{N}$ denotes some fixed computable pairing or, more generally, tupling function. Similarly, $\Sigma_1\mathbb{R}$ contains of all $x \in \mathbb{R}$ computable with respect to $\rho_<$ whereas $\Sigma_2\mathbb{R}$ includes all x computable in the sense of $\rho'_<$ defined as follows:

$\boldsymbol{\rho'_<}$**:** $x = \sup_i \inf_j q_{\langle i,j \rangle}$ for some computable rational sequence (q_n).

To $\Sigma_2\mathbb{R}$ belongs for instance the radius of convergence $r = 1/\limsup_{n\to\infty} \sqrt[n]{a_n}$ of a computable power series $\sum_{n=0}^{\infty} a_n x^n$ [ZW01, Theorem 6.2]. More generally we take from [ZW01, Definition 7.1 and Corollary 7.3] the following

Definition 2 (Real Arithmetic Hierarchy). *Let* $d = 0, 1, 2, \ldots$

$\rho_<^{(d)}$: $\Sigma_{d+1}\mathbb{R}$ *consists of all* $x \in \mathbb{R}$ *of the form* $x = \sup_{n_1} \inf_{n_2} \ldots \Theta_{n_{d+1}} q_{\langle n_1,\ldots,n_{d+1}\rangle}$ *for a computable rational sequence* (q_n), *where* $\Theta{=}\sup$ *or* $\Theta{=}\inf$ *depending on d's parity;*

$\rho_>^{(d)}$: $\Pi_{d+1}\mathbb{R}$ *similarly for* $x = \inf_{n_1} \sup_{n_2} \ldots$

$\rho^{(d)}$: $\Delta_{d+1}\mathbb{R}$ *contains all* $x \in \mathbb{R}$ *of the form* $x = \lim_{n_1} \lim_{n_2} \ldots \lim_{n_d} q_{\langle n_1,\ldots,n_d\rangle}$ *for a computable rational sequence* (q_n).

(For an extension to levels beyond ω see [Bar03]. . .)

The close analogy between the discrete and this real variant of the Arithmetic Hierarchy is expressed in [ZW01] by a variety of elegant results like, e.g.,

Fact 3. *a)* $x \in \Delta_d\mathbb{R}$ *iff deciding its binary expansion is in* Δ_d.

b) x *is computable with respect to* $\rho^{(d)}$ *iff there is a* $\emptyset^{(d)}$*–computable fast convergent rational sequence for* x.

c) x *is computable with respect to* $\rho_<^{(d)}$ *iff* x *is the supremum of a* $\emptyset^{(d)}$*–computable rational sequence.*

d) $\Delta_d\mathbb{R} = \Sigma_d\mathbb{R} \cap \Pi_d\mathbb{R}$.

e) $\Sigma_d\mathbb{R} \cup \Pi_d\mathbb{R} \subsetneq \Delta_{d+1}\mathbb{R}$.

Proof. a) Theorem 7.8, b+c) Lemma 7.2, d) Definition 7.1, and e) Theorem 7.8 in [ZW01], respectively. □

2.1 Type-2 Theory of Effectivity

Specifying an encoding formalizes how to feed some general form of input like graphs or integers into a Turing Machine with fixed alphabet Σ. Already in the discrete case, the complexity of a problem usually depends heavily on the chosen encoding; e.g., numbers in unary versus binary. This dependence becomes even more important when dealing with objects from a continuum like the set of reals and their computability. While Recursive Analysis usually considers one particular encoding for $\mathbb{R}$, the Type-2 Theory of Effectivity (TTE) due to Weihrauch provides (a convenient formal framework for studying and comparing) a variety of encodings for different universes. Formally speaking, a *representation* α for $\mathbb{R}$ is a partial† surjective mapping $\alpha :\subseteq \Sigma^\omega \to \mathbb{R}$; and an infinite string $\bar\sigma \in \mathrm{dom}(\alpha)$ is regarded as an α–name for the real number $x = \alpha(\bar\sigma)$.

In this way, $(\alpha \to \beta)$–computing‡ a real function $f : \mathbb{R} \to \mathbb{R}$ means to compute a transformation on infinite strings $F :\subseteq \Sigma^\omega \to \Sigma^\omega$ such that any α–name $\bar\sigma$ for $x = \alpha(\bar\sigma)$ gets transformed to a β–name $\bar\tau = F(\bar\sigma)$ for $f(x) = y$, that is, satisfying $\beta(\bar\tau) = y$; cf. [Wei01, Section 3]. Converting α–names to β–names thus amounts to $(\alpha \to \beta)$–computability of $\mathrm{id} : \mathbb{R} \to \mathbb{R}$, $x \mapsto x$, and is called reducibility "$\alpha \preceq \beta$" [Wei01, Definition 2.3.2]. Computational equivalence, that

† indicated by the symbol "$\subseteq$", whose absence here generally refers to total functions

‡ We use this notation instead of [Wei01]'s (α, β)–computability to stress its connection (but not to be confused) with $[\alpha \to \beta]$–computability appearing in Section 4.2.

is mutual reducibility $\alpha \preceq \beta$ and $\beta \preceq \alpha$, is denoted by "$\alpha \equiv \beta$" whereas "$\alpha \precneqq \beta$" means $\alpha \preceq \beta$ but $\beta \not\preceq \alpha$.

We borrow from TTE also two ways of constructing new representations from giving ones: The conjunction $\alpha \wedge \tilde{\alpha}$ of α and $\tilde{\alpha}$ is the least upper bound with respect to "$\preceq$" [Wei01, Lemma 3.3.8]; and for (finitely or countably many) representations $\alpha_i :\subseteq \Sigma^\omega \to A_i$, their product $\prod_i \alpha_i$ denotes a natural representation for the set $\prod_i A_i$ [Wei01, Definition 3.3.3.2]. In particular, in order to encode $x \in \mathbb{R}$ as a rational sequence $(q_n) \in \mathbb{Q}^\omega$, we (often implicitly) refer to the representation $[\nu_\mathbb{Q}]^\omega :\subseteq \Sigma^\omega \to \mathbb{Q}^\omega$ due to [Wei01, Definition 3.1.2.4 and Lemma 3.3.16].

2.2 Arithmetic Hierarchy of Real Representations

Observe that (the characterizations due to Fact 3 of) each level of the Real Arithmetic Hierarchy gives rise not only to a notion of computability for real numbers but also canonically to a representation for $\mathbb{R}$; for instance let

$\boldsymbol{\rho}$: encode (arbitrary!) $x \in \mathbb{R}$ as a fast convergent rational sequence (q_n);
$\boldsymbol{\rho_<}$: encode $x \in \mathbb{R}$ as a rational sequence (q_n) with supremum $x = \sup_n q_n$;
$\boldsymbol{\rho'}$: encode $x \in \mathbb{R}$ as a rational sequence (q_n) with limit $x = \lim_n q_n$;
$\boldsymbol{\rho'_<}$: encode $x \in \mathbb{R}$ as $(q_n) \subseteq \mathbb{Q}$ with $x = \sup_i \inf_j q_{\langle i,j\rangle}$;
$\boldsymbol{\rho''}$: encode $x \in \mathbb{R}$ as $(q_n) \subseteq \mathbb{Q}$ with $x = \lim_i \lim_j q_{\langle i,j\rangle}$.

As already pointed out, the first three of them are already known and used in TTE as ρ, $\rho_<$, and ρ_{Cn}, respectively [Wei01, Section 4.1]. In general one obtains, similar to Definition 2, a hierarchy of real representations as follows:

Definition 4. *Let $\rho^{(0)} := \rho$, $\rho_<^{(0)} := \rho_<$, $\rho_>^{(0)} := \rho_>$. Now fix $1 \leq d \in \mathbb{N}$: A $\rho^{(d)}$–name for $x \in \mathbb{R}$ is (a $[\nu_\mathbb{Q}]^\omega$–name for) a rational sequence (q_n) such that*

$$x \;=\; \lim_{n_1} \lim_{n_2} \ldots \lim_{n_d} q_{\langle n_1,\ldots,n_d\rangle} \;.$$

A $\rho_<^{(d)}$–name for $x \in \mathbb{R}$ is a (name for a) sequence $(q_n) \subseteq \mathbb{Q}$ such that

$$x \;=\; \sup_{n_1} \inf_{n_2} \ldots \Theta_{n_{d+1}} q_{\langle n_1,\ldots,n_{d+1}\rangle} \;.$$

A $\rho_>^{(d)}$–name for $x \in \mathbb{R}$ is a sequence $(q_n) \subseteq \mathbb{Q}$ such that $x \;=\; \inf_{n_1} \sup_{n_2} \ldots$

Regarding Fact 3, one may see ρ' and ρ'' as the first and second *Jump* of ρ, respectively; same for $\rho'_<$ and $\rho_<$.

Results from [ZW01] about the Real Arithmetic Hierarchy are easily rephrased in terms of these representations. Fact 3d) for example translates as follows:

x is $\rho^{(d)}$–computable iff it is both $\rho_<^{(d)}$–computable and $\rho_>^{(d)}$–computable.

Observe that this is a non-uniform claim whereas closer inspection of the proofs in particular of Lemma 3.2 and Lemma 3.3 in [ZW01] reveals them to hold fully uniformly so that we have

Lemma 5. $\rho \;\equiv\; \rho_< \wedge \rho_> \;\precneqq\; \rho_< \;\precneqq\; \rho' \;\equiv\; \rho'_< \wedge \rho'_> \;\precneqq\; \rho'_< \;\precneqq\; \rho'' \;\equiv\; \ldots\,.$

Moreover, the uniformity of [ZW01, Lemma3.2] yields the following interesting

Scholium[§] **6** *Let $\tilde\rho'_<$ denote the representation encoding $x \in \mathbb{R}$ as $(q_n) \subseteq \mathbb{Q}$ with $x = \liminf_n q_n$; and $\hat\rho'_<$ similarly with the additional requirement that $q_n < x$ for infinitely many n.*
Then it holds $\hat\rho'_< \equiv \tilde\rho'_< \equiv \rho'_<$ ($\hat\rho'_< \preceq \tilde\rho'_< \preceq \rho'_<$ being the trivial direction).

3 Computability and Continuity

Recursive Analysis has established as folklore that any computable real function is continuous. More precisely, computability of a partial function from/to infinite strings $f :\subseteq \Sigma^\omega \to \Sigma^\omega$ requires continuity with respect to the Cantor Topology τ_C [Wei01, Theorem 2.2.3]; and this requirement carries over to functions $f :\subseteq A \to B$ on other topological spaces (A, τ_A) and (B, τ_B) where input $a \in A$ and output $b = f(a)$ are encoded by respective *admissible* representations α and β. Roughly speaking, this property expresses that the mappings $\alpha :\subseteq \Sigma^\omega \to A$ and $\beta :\subseteq \Sigma^\omega \to B$ satisfy a certain compatibility condition with respect to the topologies τ_A/τ_B and τ_C involved. For $A = B = \mathbb{R}$, the (standard) representation ρ for example is admissible [Wei01, Lemma 4.1.4.1], thus recovering the folklore claim.

Now in order to treat and non-trivially investigate computability also of discontinuous real functions $f : \mathbb{R} \to \mathbb{R}$, there are basically two ways out: Either enhance the underlying Type-2 Machine model or resort to non-admissible representations. It turns out that for either choice, at least the straight-forward approaches fail:
- extending Turing Machines with oracles as well as
- considering weakened representations for $\mathbb{R}$.

3.1 Type-2 Oracle Computation

Specifically concerning the first approach, most results in Computable Analysis relativize. Specifically we make

Observation 7. *Let $\mathcal{O} \subseteq \Sigma^*$ be arbitrary. Replace in* [Wei01, Definition 2.1.1] *the Turing Machine $\mathcal{M}$ by $\mathcal{M}^{\mathcal{O}}$, that is, one with oracle access to $\mathcal{O}$. This Type-2 Computability* in $\mathcal{O}$ *still satisfies*

a) closure under composition [Wei01, Theorem 2.1.12]*;*
b) computability of string functions requires continuity [Wei01, Theorem 2.2.3]*;*
c) same for computable functions on represented spaces with respect to admissible *representations* [Wei01, Corollary 3.2.12].

[§] A scholium is "*a note amplifying a proof or course of reasoning, as in mathematics*" [Mor69]

In particular, the Main Theorem of Recursive Analysis [Wei01, Theorem 4.3.1] *relativizes.*

A strengthening and counterpart to Observation 7b), we have

Lemma 8. *For a partial function on infinite strings* $f :\subseteq \Sigma^\omega \to \Sigma^\omega$*, the following are equivalent:*

- *There exists an oracle* $\mathcal{O}$ *such that* f *is computable relative to* $\mathcal{O}$*;*
- f *is Cantor-continuous and* $\mathrm{dom}(f)$ *is a* G_δ*–set.*

Compare this with Type-1 Theory (that is, computability on finite strings) where every function $f :\subseteq \Sigma^* \to \Sigma^*$ is recursive in some appropriate $\mathcal{O} \subseteq \Sigma^*$.

Proof (Lemma 8). If f is recursive in $\mathcal{O}$, then it is also continuous by Observation 7b), that is, the relativized version of [Wei01, Theorem 2.2.3]. Furthermore the relativization of [Wei01, Theorem 2.2.4] reveals $\mathrm{dom}(f)$ to be a G_δ–set.

Conversely suppose that continuous f has G_δ domain. Then $f = h_\omega$ for some monotone total function $h : \Sigma^* \to \Sigma^*$ according to [Wei01, Theorem 2.3.7.2] where, by [Wei01, Definition 2.1.10.2], $h_\omega :\subseteq \Sigma^\omega \ni \bar\sigma \mapsto \sup_n h(\sigma_1 \dots \sigma_n)$ denotes the (existing and unique) extension of h from Σ^* to $\subseteq \Sigma^\omega$. A classical Type-1 function on finite strings, this h is recursive in a certain oracle $\mathcal{O} \subseteq \Sigma^*$. The relativization of [Wei01, Lemma 2.1.11.2] then asserts also $h_\omega = f$ to be computable in $\mathcal{O}$. □

The conclusion of this subsection is that oracles do not increase the computational power of a Type-2 Machine sufficiently in order to handle also discontinuous functions. So let us proceed to the second approach to real hypercomputation:

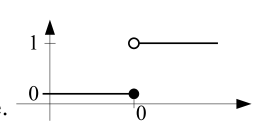
3.2 Weaker Representations for Reals

In the present section we are interested in relaxations of the standard representation ρ for single reals and their effect on the computability of function evaluation $x \mapsto f(x)$. Since, with exception of ρ, none of the ones introduced in Definition 4 is admissible with respect to the usual Euclidean[¶] topology on $\mathbb{R}$ [Wei01, Lemma 4.1.4, Example 4.1.14.1], the relativized Main Theorem (Observation 7c) is not applicable. Hence, chances are good for evaluation $x \mapsto f(x)$ to become computable even for discontinuous $f : \mathbb{R} \to \mathbb{R}$; and indeed we have the following

Example 9. Heaviside's function

$$h : \mathbb{R} \to \mathbb{R}, \qquad x \mapsto 0 \ \text{ for } \ x \le 0, \quad x \mapsto 1 \ \text{ for } \ x > 0$$

is both $(\rho_< \to \rho_<)$–computable and $(\rho'_< \to \rho'_<)$–computable.



Proof. Given $(q_n) \subseteq \mathbb{Q}$ with $x = \sup_n q_n$, exploit $(\nu_\mathbb{Q} \to \nu_\mathbb{Q})$–computability of the restriction $h|_\mathbb{Q} : \mathbb{Q} \to \{0,1\}$ to obtain $p_n := h(q_n)$. Then indeed, $(p_n) \subseteq \mathbb{Q}$

[¶] it might be admissible w.r.t. some other, typically artificial topology, though has $\sup_n p_n = h(x)$: In case $x \leq 0$, $q_n \leq 0$ and hence $p_n = 0$ for all n; whereas in case $x > 0$, $q_n > 0$ and hence $p_n = 1$ for some n.

Let $x \in \mathbb{R}$ be given by a rational double sequence $(q_{i,j})$ with $x = \sup_i \inf_j q_{i,j}$. Proceeding from $q_{i,j}$ to $\tilde{q}_{i,j} := \max\{q_{0,j}, \ldots, q_{i,j}\}$, we assert $\inf_j \tilde{q}_{i+1,j} \geq \inf_j \tilde{q}_{i,j}$. Now compute $p_{i,j} := h(\tilde{q}_{i,j} - 2^{-i})$. Then in case $x \leq 0$, it holds $\forall i \exists j : \tilde{q}_{i,j} \leq 2^{-i}$, i.e., $p_{i,j} = 0$ and thus $\sup_i \inf_j p_{i,j} = 0 = h(x)$. Similarly in case $x > 0$, there is some i_0 such that $\inf_j \tilde{q}_{i_0,j} > x/2$ and thus $\inf_j \tilde{q}_{i,j} > x/2$ for all $i \geq i_0$. For $i \geq i_0$ with $2^{-i} \leq x/2$, it follows $p_{i,j} = 1$ $\forall j$ and therefore $\sup_i \inf_j p_{i,j} = 1 = h(x)$. □

So real function hypercomputation based on weaker representations indeed does allow for effective evaluation of some discontinuous functions. On the other hand, they still impose well-known topological restrictions:

Fact 10. *Consider* $f : \mathbb{R} \to \mathbb{R}$.

a) If f is $(\rho \to \rho)$–computable, then it is continuous.
b) If f is $(\rho \to \rho_<)$–computable, then it is lower semi-continuous.
c) If f is $(\rho_< \to \rho_<)$–computable, then it is monotonically increasing.
d) If f is $(\rho' \to \rho')$–computable, then it is continuous.

The claims remain valid under oracle-supported computation.

Claim a) is the Main Theorem. For b) see [WZ00] and recall, e.g. from [Ran68, CHAPTER 6.7], that $f : \mathbb{R} \to \mathbb{R}$ is lower semi-continuous iff $f(\lim_n x_n) \leq \liminf_n f(x_n)$ for all convergent sequences (x_n); equivalently: $f^{-1}\big[(y, \infty)\big]$ is open for any $y \in \mathbb{R}$. The establishing of d) in [BH02, SECTION 6] caused some surprise. We briefly sketch the according proofs as a preparation for those of Theorem 11 below.

Proof. a) Suppose for a start that Heaviside's function, in spite of its discontinuity at $x = 0$, be $(\rho \to \rho)$–computable by some Type-2 Machine $\mathcal{M}$. Feed the rational sequence $q_n := 2^{-n}$, a valid ρ–name for x, to this $\mathcal{M}$. By presumption it will then spit out a sequence $(p_m)_m \subseteq \mathbb{Q}$ with $|p_m - y| \leq 2^{-m}$ for $y = h(x) = 0$; in particular, $|p_2 - \tilde{y}| > 2^{-2}$ for $\tilde{y} := 1$. Up to output of p_2, $\mathcal{M}$ has executed a finite number $N \in \mathbb{N}$ of operations and in particular read at most the initial part $p_0, p_1, \ldots, p_N$ of the input.
Now re-use $\mathcal{M}$ in order to evaluate h at $\tilde{x} := p_N > 0$ ρ–encoded as the rational sequence $(\tilde{q}_n) := (q_0, q_1, \ldots, q_N, q_N, \ldots)$ coinciding with (p_n) for $n \leq N$. Being a deterministic machine, $\mathcal{M}$ will then proceed exactly as before for its first N steps; in particular the output $(\tilde{p}_m)$ agrees with (p_m) up to $m = 2$. Hence $|\tilde{p}_2 - \tilde{y}| > 2^{-2}$ contradicting that $\mathcal{M}$ is supposed to output a ρ–name for $\tilde{y} = h(\tilde{x})$.

For the case of a general function $f : \mathbb{R} \to \mathbb{R}$ with discontinuity at some $x \in \mathbb{R}$, let $y = f(x) \neq \lim_k f(x_k) = \tilde{y}$ with a real sequence x_k converging to x. There exists $M \in \mathbb{N}$ with $|y - \tilde{y}| > 2^{-M+2}$; by possibly proceeding to an appropriate subsequence of (x_k), we may suppose w.l.o.g. that $|x - x_k| \leq 2^{-k-2}$ and $|f(x_k) - \tilde{y}| \leq 2^{-M}$. Then there is a rational double sequence $(q_{k,n})$ such that $|x_k - q_{k,n}| \leq 2^{-n-1}$; thus $|x - q_{n,n}| \leq 2^{-n}$. We may therefore feed

$(q_{n,n})$ as a ρ–name in order to evaluate f at x and obtain in turn a ρ–name $(p_m) \subseteq \mathbb{Q}$ for y. As before, p_M is output after having only read some finite initial part $(q_{n,n})_{n \leq N}$ of the input. Then

$$|q_{n,n} - x_N| \leq |q_{n,n} - x_n| + |x_n - x| + |x - x_N| \leq 2^{-n-1} + 2^{-n-2} + 2^{-N-2} \leq 2^{-n}$$

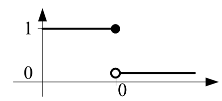
for $n \leq N$ reveals this very initial part to also be the start of a valid ρ–name for $\tilde{x} := x_N$ whereas

$$2^{-M+2} < |y - \tilde{y}| \leq |y - p_M| + |p_M - f(\tilde{x})| + |f(\tilde{x}) - \tilde{y}| \leq 2^{-M} + |p_M - f(\tilde{x})| + 2^{-M}$$

shows that $(p_m)_{m \leq M}$ is not a valid initial part of a ρ–name for $f(\tilde{x})$: contradiction.

b) We prove $(\rho \to \rho_<)$–uncomputability of the flipped Heaviside Function

$$\overline{h} : 0 \geq x \mapsto 1, \qquad 0 < x \mapsto 0$$



as a prototype lacking lower semi-continuity.
Consider again the ρ–name $q_n := 2^{-n}$ for $x = 0$ which the hypothetical Type-2 Machine transforms into a $\rho_<$–name for $y = \overline{h}(x) = 1$, that is, a sequence $(p_m) \subseteq \mathbb{Q}$ with $\sup_m p_m = y$; In particular $p_M \geq \frac{2}{3}$ for some $M \in \mathbb{N}$ gets output having read only $(q_n)_{n \leq N}$ for some $N \in \mathbb{N}$. The latter finite segment is also the initial part of a valid ρ–name for $\tilde{x} = q_N > 0$ whereas $(p_m)_{m \leq M}$ has sup $\geq \frac{2}{3}$ and thus is not the initial part of a valid $\rho_<$–name for $\tilde{y} = \overline{h}(\tilde{x}) = 0$: contradiction.
This proof for the case $\overline{h}$ carries over to an arbitrary $f : \mathbb{R} \to \mathbb{R}$ just like in a), that is, by replacing $q_n = 2^{-n}$ with rational approximations to a general sequence $x_n \in \mathbb{R}$ witnessing violated lower semi-continuity of f in that $f(\lim_n x_n) > \liminf_n f(x_n)$.

c) As in a) and b), we treat for notational simplicity the case of $f : \mathbb{R} \to \mathbb{R}$ violating monotonicity in that $f(0) = 1$ and $f(1) = 0$; the general case can again be handled similarly. Feed the $\rho_<$–name $(q_n) = (0, 0, \ldots)$ for $x = 0$ into a machine which be presumption produces a sequence $(p_m) \subseteq \mathbb{Q}$ with $\sup p_m = 1$ and in particular $p_M \geq \frac{2}{3}$ for some $M \in \mathbb{N}$. Up to output of p_M, only $(q_n)_{n \leq N}$ has been read for some $N \in \mathbb{N}$. Now consider the rational sequence $(\tilde{q}_n)$ consisting of N zeros followed by an infinity of 1s, that is, a valid $\rho_<$–name for $\tilde{x} = 1$. This new input will cause the machine to output a sequence $(\tilde{p}_m) \subseteq \mathbb{Q}$ coinciding with (p_m) for $m \leq M$; in particular $\tilde{p}_M \geq \frac{2}{3}$ contradicting that $(\tilde{p}_m)$ is supposed to satisfy $\sup_m \tilde{p}_m = f(\tilde{x}) = 0$.

d) Suppose that, in spite of its discontinuity at $x = 0$, $\overline{h}$ be $(\rho' \to \rho')$–computable by some Type-2 Machine $\mathcal{M}$.
Consider the sequence $q^{(1)} := \big(q^{(1)}_n\big) \subseteq \mathbb{Q}$, $q^{(1)}_n :\equiv 1$, which is by definition a valid ρ'–name for $1 =: x^{(1)} = \lim_n q^{(1)}_n$. So upon input of $q^{(1)}$, $\mathcal{M}$ will generate a corresponding sequence $p^{(1)} \subseteq \mathbb{Q}$ as a ρ'–name for $y^{(1)} = \overline{h}\big(x^{(1)}\big) = 0$, that is, satisfying $\lim_m p^{(1)}_m = 0$; in particular, $p^{(1)}_{m_1} \leq \frac{1}{3}$ for some $m_1 \in \mathbb{N}$. Up to this output, $\mathcal{M}$ has read only a finite initial part of the input $q^{(1)}$, say, up to $n \leq n_1$.

Next consider the sequence $q^{(2)} \subseteq \mathbb{Q}$ defined by $q^{(2)}_n := 1$ for $n \le n_1$ and $q^{(2)}_n := \frac{1}{2}$ for $n > n_1$: a valid ρ'–name for $x^{(2)} = \frac{1}{2}$ which $\mathcal{M}$ by presumption transforms into a sequence $p^{(2)} \subseteq \mathbb{Q}$ with $\lim_m p^{(2)}_m = y^{(2)} = \overline{h}\big(x^{(2)}\big) = 0$; in particular, $q^{(2)}_{m_2} \le \frac{1}{3}$ for some $m_2 > m_1$. However, due to $\mathcal{M}$'s deterministic behavior and since $q^{(1)}$ and $q^{(2)}$ initially coincide, it still holds $p^{(2)}_{m_1} \le \frac{1}{3}$.
Now by repeating the above argument we obtain a sequence of sequences $q^{(k)} \subseteq \mathbb{Q}$, each constant for $n \ge n_k$ of value (and thus a valid ρ'–name for) $x^{(k)} = 2^{-k+1}$ and transformed by $\mathcal{M}$ into a sequence $p^{(k)} \subseteq \mathbb{Q}$ satisfying $p^{(k)}_{m_i} \le \frac{1}{3}$ for $i = 1, \ldots k$ with strictly increasing $(n_k), (m_k) \subseteq \mathbb{N}$. The ultimate sequence $q^{(\omega)} \subseteq \mathbb{Q}$, well-defined by $q^{(\omega)}_n := q^{(k)}_n$ for $n \le n_k$ (and in fact the limit of the sequence of sequences $\big(q^{(k)}\big)_k$ with respect to Baire's Topology), therefore converges to (and is hence a valid ρ'–name for) $x^{(\omega)} = 0$; and it gets mapped by $\mathcal{M}$ to a sequence $q^{(\omega)} \subseteq \mathbb{Q}$ satisfying $q^{(\omega)}_m \le \frac{1}{3}$ for infinitely many m contradicting that a valid ρ'–name for $y^{(\omega)} = \overline{h}\big(x^{(\omega)}\big) = 1$ should have $\lim_m = 1$.

Being only information-theoretic, the above arguments obviously relativize. □

The main result of the present section is an extension of Fact 10 to one level up on the hierarchy of real representations from Definition 4. This suggests similar claims to hold for the entire hierarchy and might not be as surprising any more as Fact 10d) in [BH02]; nevertheless, already this additional step makes proofs significantly more involved.

Theorem 11 (First Main Theorem of Real Hypercomputation). *Consider $f : \mathbb{R} \to \mathbb{R}$.*

a) If f is $(\rho' \to \rho'_<)$–computable, then it is lower semi-continuous.
b) If f is $(\rho'_< \to \rho'_<)$–computable, then it is monotonically increasing.
c) If f is $(\rho'' \to \rho'')$–computable, then it is continuous.

The claims remain valid under oracle-supported computation.

We point out that the proofs of Fact 10 proceed by constructing an input for which a presumed machine $\mathcal{M}$ fails to produce the correct output. They differ however in the 'length' of these constructions: for Claims a) to c), the counter-example inputs are obtained by running $\mathcal{M}$ for a finite number of steps on a single, fixed argument; whereas in the proof of Claim d), $\mathcal{M}$ is repeatedly started on an adaptively extended sequence of arguments. The latter argument may thus be considered as of length ω, the first infinite ordinal. Our proof of Theorem 11c) will be even longer and is therefore put into the following subsection.

3.3 Proof of Theorem 11

As in the proof of Fact 10, we treat the special case of the flipped Heaviside Function $\overline{h}$ for reasons of notational convenience and clarity of presentation; the according arguments can be immediately extended to the general case.

Claim 12. $\overline{h} : \mathbb{R} \to \mathbb{R}$ *is not* $(\rho' \to \rho'_<)$*–computable.*

Proof. Suppose a Type-2 Machine $\mathcal{M}$ $(\rho' \to \rho'_<)$–computes $\overline{h}$. In particular, upon input of $x^{(1)} = 1$ in form of the sequence $q^{(1)} = (q_n^{(1)})$ with $q_n^{(1)} :\equiv 1$, $\mathcal{M}$ will output a rational double sequence $p^{(1)} = (p_{k,\ell}^{(1)})$ with $0 = y^{(1)} := \overline{h}(x^{(1)}) = \sup_k \inf_\ell p_{k,\ell}^{(1)}$. Observe that $p_{1,\ell_1}^{(1)} \leq \frac{1}{3}$ for some ℓ_1. When writing $p_{1,\ell_1}^{(1)}$, $\mathcal{M}$ has only read a finite part of $(q_n^{(1)})$, say, up to n_1.

Now consider $x^{(2)} := \frac{1}{2}$, given by way of the sequence $q^{(2)}$ with $q_n^{(2)} := 1$ for $n < n_1$ and $q_n^{(2)} := \frac{1}{2}$ for $n \geq n_1$. Then, too, $\mathcal{M}$ will output a double sequence $p^{(2)}$ with $0 = y^{(2)} = \sup_k \inf_\ell p_{k,\ell}^{(2)}$. Observe that, similarly, some $p_{2,\ell_2}^{(2)} \leq \frac{1}{3}$ is output having read only a finite part of $(q_n^{(2)})$, say, up to n_2. Moreover, as $q^{(1)}$ and $q^{(2)}$ coincide up to n_1 and since $\mathcal{M}$ operates deterministically, $p_{1,\ell_1}^{(2)} = p_{1,\ell_1}^{(1)} \leq \frac{1}{3}$.

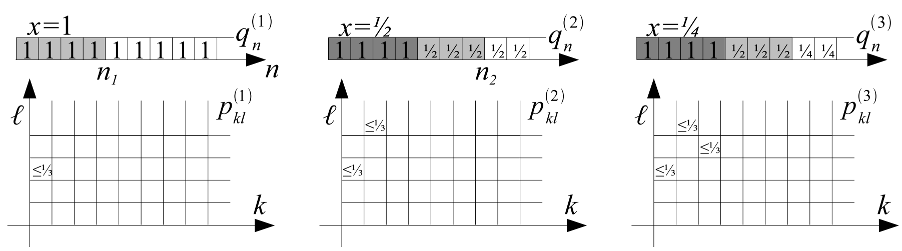


Fig. 1. Illustration to the iterative construction employed in the proof of Claim 12

Continuing this process with $x^{(k)} := 2^{-k+1}$ for $k = 3, 4, \ldots$ as indicated in Figure 1 eventually yields a rational sequence $q^{(\omega)}$ with $\lim_n q_n^{(\omega)} =: x^{(\omega)} = 0$, upon input of which $\mathcal{M}$ outputs a double sequence $p^{(\omega)}$ such that $p_{k,\ell_k}^{(\omega)} \leq \frac{1}{3}$ for all $k = 1, 2, \ldots$. In particular, $y^{(\omega)} := \sup_k \inf_\ell p_{k,\ell}^{(\omega)} \leq \frac{1}{3}$ whereas $\overline{h}(x^{(\omega)}) = 1$: contradiction. □

Notice that the above proof involves one-dimensionally indexed sequences (q_n) for input and two-dimensionally indexed ones $(p_{k,\ell})$ for output. We now proceed a step further in proof difficulty, namely involving two-dimensional indices for both input and output in order to establish Item b).

Claim 13. *Let* $f : \mathbb{R} \to \mathbb{R}$ *violate monotonicity in that* $f(0) = 1$ *and* $f(1) = 0$*. Then,* f *is not* $(\rho'_< \to \rho'_<)$*–computable.*

Proof. We construct a $\rho'_<$–name for $x = 0$ from an iteratively defined sequence of initial segments of $\rho'_<$–names for $x = 1$:

Start with $q^{(1)}_{i,j} := 1$ for all i,j. Then, $q^{(1)} = (q^{(1)}_{i,j})$ is obviously a $\rho'_<$–name for $x = 1$ and thus yields by presumption, upon input to $\mathcal{M}$, a $\rho'_<$–name $p^{(1)}_{k,\ell}$ for $f(1) = 0$, that is, with $0 = \sup_k \inf_\ell p^{(1)}_{k,\ell}$. In particular, $p^{(1)}_{1,\ell_1} \leq \frac{1}{3}$ for some ℓ_1.

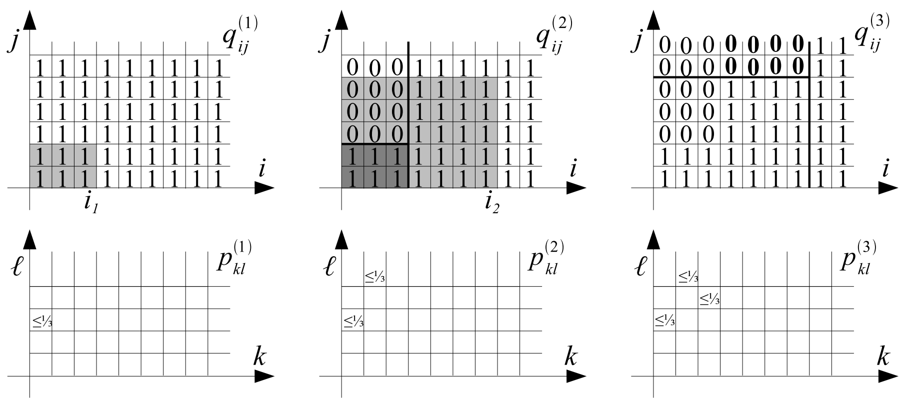


Fig. 2. Illustration to the iterative construction employed in the proof of Claim 13

Until output of $p^{(1)}_{1,\ell_1}$, $\mathcal{M}$ has read only finitely many entries of $q^{(1)}$; say, up to i_1 and j_1, that is, covered in Figure 2 by the light gray rectangle. Now consider $q^{(2)}$ defined as in this figure: Since $\inf_j q^{(2)}_{i,j} = 0$ for $i \leq i_1$ and $\inf_j q^{(2)}_{i,j} = 1$ for $i > i_1$, $\sup_i \inf_j q^{(2)}_{i,j} = 1$, that is, this is again valid $\rho'_<$–name for $x = 1$; and again, $\mathcal{M}$ will by presumption convert $q^{(2)}$ into a $\rho'_<$–name $p^{(2)}$ for $f(1) = 0$. In particular, $p^{(2)}_{2,\ell_2} \leq \frac{1}{3}$ for some ℓ_2; and, being a deterministic machine, $\mathcal{M}$'s operation on the initial part (dark gray) on which input $q^{(2)}$ coincides with input $q^{(1)}$ will first have generated the same initial output, namely $p^{(2)}_{1,\ell_1} = p^{(1)}_{1,\ell_1} \leq \frac{1}{2}$.
Again, until output of $p^{(2)}_{2,\ell_2}$, $\mathcal{M}$ has read only a finite part of $q^{(2)}$ of, say, up to $i_2 > i_1$ (light gray). By now considering input $q^{(3)}$ with $\inf_j q^{(3)}_{i,j} = 0$ for $i \leq i_2$ as in Figure 2, we arrive at $p^{(3)}$ and ℓ_3 with $p^{(3)}_{1,\ell_1}, p^{(3)}_{2,\ell_2}, p^{(3)}_{3,\ell_3} \leq \frac{1}{3}$; and so on with $i_3, q^{(4)}, p^{(4)}, \ell_4, i_4, \ldots$
Finally observe that continuing these arguments eventually leads to a rational double sequence $q^{(\omega)} = (q^{(\omega)}_{i,j})$ which has $\inf_j q^{(\infty)}_{i,j} = 0$ for $i \leq i_\infty = \infty$—and is therefore a valid $\rho'_<$–name for $x = 0$ (rather than $x = 1$)—but gets mapped by $\mathcal{M}$ to $p^{(\omega)} = (p^{(\omega)}_{k,\ell})$ with $\inf_\ell p^{(\infty)}_{k,\ell} \leq p^{(\infty)}_{k,\ell_k} \leq \frac{1}{3}$ for all k. Since $f(0) = 1$, this contradicts our presumption that $\mathcal{M}$ maps $\rho'_<$–names for x to $\rho'_<$–names for $f(x)$. $\square$

The above proofs involving ρ' and $\rho'_<$ proceeded by constructing an infinite sequence of inputs $q^{(1)}, q^{(2)}, \ldots, q^{(\omega)}$ (each possibly a multi-indexed sequence of its own). For finally asserting Claim c) involving ρ'', we will extend this method from length ω, the first infinite ordinal, to an even longer one.

Claim 14. $\overline{h} : \mathbb{R} \to \mathbb{R}$ *is not* $(\rho'' \to \rho'')$*–computable.*

Proof. Outwit a Type-2 Machine $\mathcal{M}$, presumed to realize this computation, as follows:

i) Take $q^{(1)}$ to be the constant double sequence 1, i.e., $q^{(1)}_{i,j} := 1$ for all i, j. Being a ρ''–name for 1, it is by presumption mapped to a ρ''–name $p^{(1)}$ for $\overline{h}(1) = 0$, that is, satisfying $\lim_k \lim_\ell p^{(1)}_{k,\ell} = 0$. In particular, almost every column $\#k$ contains an entry $\#\ell$ with $p^{(1)}_{k,\ell} \leq \frac{1}{3}$. Until output of the first such $p^{(1)}_{k_1,\ell_1}$, $\mathcal{M}$ has read only a finite part of $q^{(1)}$—say, up to i_1, j_1.

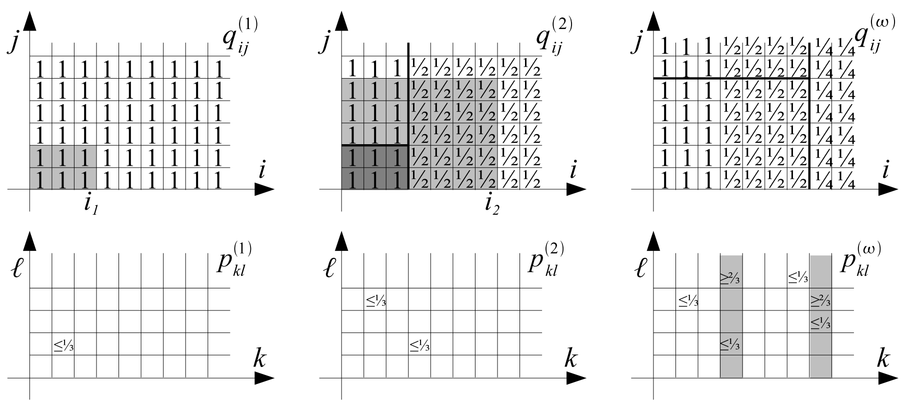


Fig. 3. The first infinitely long iterative construction employed in the proof of Claim 14

ii) Observe that this Argument i) equally applies to the scaled input sequence $2^{-m} \cdot q^{(1)}$ for any m. So define $q^{(2)}_{i,j} := q^{(1)}_{i,j}$ for $j \leq j_1$ (i.e., inherit the initial part of $q^{(1)}$) and $q^{(2)}_{i,j} :\equiv \frac{1}{2}$ for $j > j_1$. Now upon input of this $q^{(2)}$, $\mathcal{M}$ will output $p^{(2)}$ with, again, infinitely many $p^{(2)}_{k,\ell} \leq \frac{1}{3}$, the first one—$(k_2, \ell_2)$, say—after having read $q^{(2)}$ only up to some (i_2, j_2). Furthermore $\mathcal{M}$'s determinism implies $p^{(2)}_{k_1,\ell_1} = p^{(1)}_{k_1,\ell_1} \leq \frac{1}{3}$.
By repeating for $m = 2, 3, \ldots$, we eventually obtain—similarly to the proof of Claim 13—an input sequence $q^{(\omega)}$ with $q^{(\omega)}_{i,j}$ with $\lim_i \lim_j q^{(\omega)}_{i,j} = 0$, that is, a valid ρ''–name for $x = 0$ (rather than 1). This is mapped by $\mathcal{M}$ to $p^{(\omega)}$ with $p^{(\omega)}_{k_m,\ell_m} \leq \frac{1}{3}$ for all m. On the other hand, $p^{(\omega)}$ is by presumption a ρ''–name for $\overline{h}(0) = 1$. Therefore, there are infinitely many m with $p^{(\omega)}_{m,\ell} \geq \frac{2}{3}$ for some $\ell > \ell_m$ and $p^{(\omega)}_{m,\ell_m} \leq \frac{1}{3}$; see the grey columns in the right part of Figure 3.

iii) Since this gives no contradiction yet, we proceed by considering the first such column m containing an entry $\leq \frac{1}{3}$ as well as an entry $\geq \frac{2}{3}$. Take the initial part of the input $q^{(\omega)}$ — up to (i_ω, j_ω), say, depicted in grey in the left part of Figure 4 — that $\mathcal{M}$ has read until output of both of them; extend it with $\frac{1}{2}$s in top direction and with 1s to the right. Feed this ρ''–name for $x = 1$ into $\mathcal{M}$ until output of an entry $p_{k,\ell} \leq \frac{1}{3}$ in some column k beyond m. Then repeat extending to the right with 1s replaced by $\frac{1}{2}$s for a second entry $p_{k,\ell} \leq \frac{1}{3}$.

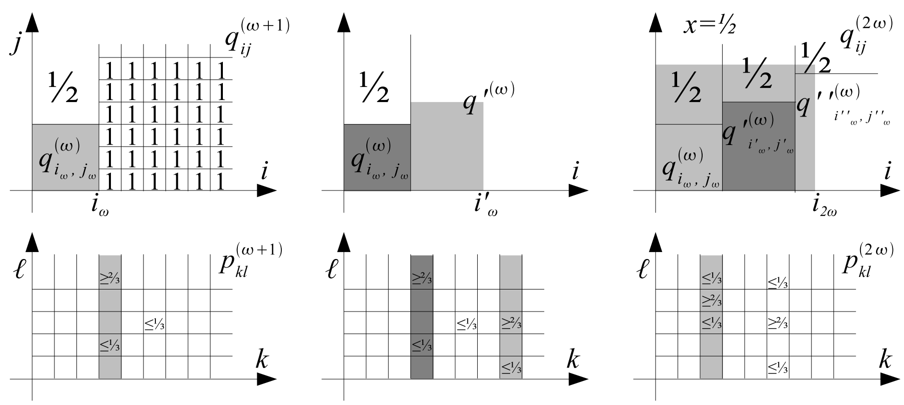


Fig. 4. Second infinitely long iterative construction employed in the proof of Claim 14

More generally, proceed similarly as in ii) and extend $q^{(\omega)}_{i_\omega, i_\omega}$ to the right in such a way with some ρ''–name $q'^{(\omega)}$ for $x = 0$ as to obtain another column m' with both entries $\leq \frac{1}{3}$ and $\geq \frac{2}{3}$; see the middle part of Figure 4. Again, $\mathcal{M}$ outputs the latter two entries having read only a finite part; say, up to (i'_ω, j'_ω).

Now extend this part, too, with $\frac{1}{2}$ in top direction and with another $q''^{(\omega)}$ obtained, again, as in ii) for a third column m'' with both entries $\leq \frac{1}{3}$ and $\geq \frac{2}{3}$; and so on.

This eventually leads to an input $q^{(2\omega)}$ which, due to the extensions to the top, represents a ρ''–name for $x = \frac{1}{2}$ and is thus mapped by presumption to a ρ''–name $p^{(2\omega)}$ for $\overline{h}(\frac{1}{2}) = 0$. In particular, almost every column of $p^{(2\omega)}$ has almost every entry $\leq \frac{1}{3}$ while maintaining infinitely many columns with preceding entries $\leq \frac{1}{3}$ and $\geq \frac{2}{3}$; see the right part of Figure 4. This asserts the existence of infinitely many columns in $p^{(2\omega)}$ containing $\leq \frac{1}{3}$, $\geq \frac{2}{3}$, and $\leq \frac{1}{3}$ in order. And again, already a finite initial part of $q^{(2\omega)}$ up to some $(i_{2\omega}, j_{2\omega})$ gives rise to the first such triple.

iv) Notice that the arguments in iii) similarly yield the existence of an appropriate, scaled counter-part $\frac{1}{2}q'^{(2\omega)}$ of $q^{(2\omega)}$, of some $\frac{1}{4}q''^{(2\omega)}$, and so on, all leading to output containing infinitely many columns with alternating triples

as above. We now construct input $q^{(3\omega)}$ leading to output $p^{(3\omega)}$ containing an infinity of columns, each with four entries $\leq \frac{1}{3}$, $\geq \frac{2}{3}$, $\leq \frac{1}{3}$, $\underline{\text{and}} \geq \frac{2}{3}$.
To this end, take the initial part of $q^{(2\omega)}$ leading to output of the first column with alternating triple in the above sense; then extend it with the initial part of the scaled version $\frac{1}{2}q'^{(2\omega)}$ leading to another column with such a triple; and so on. Observing that, due to the scaling, the thus obtained $q^{(3\omega)}$ represents a ρ''–name for $x = 0$, almost every column of the output $p^{(3\omega)}$ representing $\overline{h}(0) = 1$ contains entries $\geq \frac{2}{3}$ in addition to the infinitely many columns with triples as above; see the left part of Figure 5.

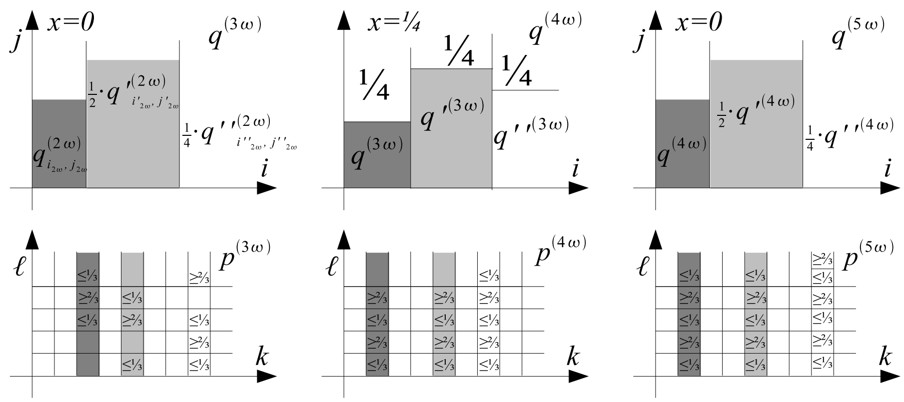


Fig. 5. Third, fourth, and fifth infinitely long iterative construction employed in the proof of Claim 14

v) Our next step is a ρ''–name $q^{(4\omega)}$ for $x = \frac{1}{4}$ giving rise to $p^{(4\omega)}$ with infinitely many columns containing alternating quintuples. This is obtained by repeating the arguments in iv) to obtain initial segments of (variants of) $q^{(3\omega)}$, stacking them horizontally—in order to obtain an infinity of columns with alternating quadruples—while extending in top direction with $\frac{1}{4}$; see the middle part of Figure 5. This forces $\mathcal{M}$ to output a ρ''–name $q^{(4\omega)}$ for $\overline{h}(\frac{1}{4}) = 0$ and thus with in almost every column almost every entry being $\leq \frac{1}{4}$, thus extending the alternating quadruples to quintuples.

vi) Noticing that the vertical extension in v) was similar to step iii), we now take a step similar to iv) based on horizontally stacked initial parts of scaled counterparts of $q^{(4\omega)}$ in order to obtain a ρ''–name $q^{(5\omega)}$ for $x = 0$ which $\mathcal{M}$ maps to some $p^{(5\omega)}$ containing infinitely many alternating six-tuples.
Then again construct a ρ''–name $q^{(6\omega)}$ for $x = \frac{1}{8}$ by horizontally stacking initial segments of (variants of) $q^{(5\omega)}$ while extending them vertically with $\frac{1}{8}$ and so on.

Now for the bottom line: By proceeding the above construction, one eventually obtains a rational double sequence $q^{(\omega^2)}$ with $\lim_j q^{(\omega^2)}_{i,j} = 0$ for all i — that is, a ρ''–name for $x = 0$ — mapped by $\mathcal{M}$ to some $p^{(\omega^2)}$ containing (infinitely many) columns $\#k$ with infinitely many alternating entries $\leq \frac{1}{3}$ and $\geq \frac{2}{3}$ — contradicting that, for ρ''-names $p = (p_{k,\ell})$, $\lim_\ell p_{k,\ell}$ is required to exist for every k. □

4 Hierarchies of Hypercomputable Real Functions

The present section investigates and compares the first levels of the two hierarchies of hypercomputable real functions induced by the two approaches to real function hypercomputation considered in Section 3: based on oracle support and based on weakened encodings.

4.1 Weakly Computable Real Functions

For every $(\alpha \to \beta)$–computable function $f : A \to B$, one may obviously replace representation α for A by a stronger one and β for B by a weaker one while maintaining computability of f:

$f \quad (\alpha \to \beta)$–computable $\wedge \;\; \alpha, \preceq \alpha \;\; \wedge \; \beta \preceq \beta' \quad \Rightarrow \quad f \quad (\alpha, \to \beta')$–computable.

However if both α and β are made, say, weaker then $(\alpha' \to \beta')$–computability of f may in general be violated. For $\alpha = \beta = \rho_<$, though, we have seen in Example 9 that the implication "$(\rho_< \to \rho_<) \Rightarrow (\rho'_< \to \rho'_<)$" does hold at least for the case of f being Heaviside's function. By the following result, it holds in fact for every f:

Theorem 15 (Second Main Theorem of Real Hypercomputation). *Consider $f : \mathbb{R} \to \mathbb{R}$.*

a) If f is $(\rho \to \rho)$–computable, then it is also $(\rho' \to \rho')$–computable.
b) If f is $(\rho \to \rho_<)$–computable, then it is also $(\rho' \to \rho'_<)$–computable.
c) If f is $(\rho_< \to \rho_<)$–computable, then it is also $(\rho'_< \to \rho'_<)$–computable.
d) If f is $(\rho' \to \rho')$–computable, then it is also $(\rho'' \to \rho'')$–computable.
e) If f is $(\rho'' \to \rho'')$–computable, then it is also $(\rho''' \to \rho''')$–computable.

The claims remain valid under oracle-supported computation.

As a consequence, we obtain the following partial strengthening of Lemma 5:

Corollary 16. *It holds $\rho \equiv \rho_< \wedge \rho_> \precneq_t \rho_< \precneq_t \rho' \equiv \rho'_< \wedge \rho'_> \precneq_t \rho'_< \precneq_t \rho''$ where "$\preceq_t$" denotes* continuous *reducibility of representations* [Wei01, Def. 2.3.2].

Proof. The positive claims follow from Lemmas 5 and 8. For a negative claim like "$\rho'_< \not\preceq_t \rho'$" suppose the contrary. Then by Lemma 8, with the help of some appropriate oracle $\mathcal{O}$, one can convert $\rho'_<$–names to ρ'–names. As Heaviside's function h is $(\rho' \to \rho'_<)$–computable by Example 9 and Theorem 15, composition with the presumed conversion implies $(\rho' \to \rho')$–computability of h relative to $\mathcal{O}$—contradicting Theorem 11c). □

Proof (Theorem 15d). Let f be $(\rho' \to \rho')$–computable and x given by a ρ''–name, that is, a rational sequence $q = (q_n)$ with $x = \lim_i \lim_j q_{\langle i,j\rangle}$. For each i, compute by assumption from the ρ'–name $q_{\langle i,\cdot\rangle} = (q_{\langle i,j\rangle})_j$ of $x_i := \lim_j q_{\langle i,j\rangle}$ a ρ'–name of $f(x_i)$, that is, a sequence $p = p_{\langle i,\cdot\rangle} = (p_{\langle i,j\rangle})_j$ with $f(x_i) = \lim_j p_{\langle i,j\rangle}$. Continuity of f due to Fact 10c) asserts

$$\lim_i \lim_j p_{\langle i,j\rangle} \;=\; \lim_i f(x_i) \;\stackrel{!}{=}\; f\big(\lim_i x_i\big) \;=\; f\big(\lim_i \lim_j q_{\langle i,j\rangle}\big) \;=\; f(x)$$

this sequence p to be a ρ''–name for $y = f(x)$. □

Where the last proof exploited Fact 10c), the next one relies on Theorem 11c):

Proof (Theorem 15e). A ρ'''–name for $x \in \mathbb{R}$ is a rational sequence $a = (q_n)$ with $x = \lim_i \lim_j \lim_k q_{\langle i,j,k\rangle}$. For each i, exploit $(\rho'' \to \rho'')$–computability of f to obtain, from the ρ''–name $q_{\langle i,\cdot,\cdot\rangle}$ of $x_i := \lim_j \lim_k q_{\langle i,j,k\rangle} \in \mathbb{R}$, a sequence $p_{\langle i,\cdot,\cdot\rangle}$ with $\lim_j \lim_k p_{\langle i,j,k\rangle}$ as ρ''–name of $f(x_i)$. Similarly to case d), this sequence p constitutes a ρ'''–name for $y = f(x)$ by continuity of f due to Theorem 11c). □

Proof (Theorem 15a). Let f be $(\rho \to \rho)$–computable. Its $(\rho' \to \rho')$–computability is established as follows: Given $(q_n) \subseteq \mathbb{Q}$ with $x = \lim_n q_n$, apply the assumption to evaluate $f(q_n)$ for each n up to error 2^{-n}; that is, obtain $p_n \in \mathbb{Q}$ with $|p_n - f(q_n)| \leq 2^{-n}$. Since f is continuous by Fact 10a), it follows $f(x) = \lim_n f(q_n) = \lim_n p_n$ so that (p_n) is a ρ'–name for $y = f(x)$. □

It is interesting that the latter proof works in fact uniformly in f, i.e., we have

Scholium 17. *The* apply *operator* $C(\mathbb{R}) \times \mathbb{R} \ni (f, x) \mapsto f(x)$ *is* $([\rho \to \rho] \times \rho' \to \rho')$*–computable.*

Similarly, Theorem 15b) follows from Lemma 18 below together with the observation that every $(\rho \to \rho_<)$–computable f has a computable $[\rho \to \rho_<]$–name [WZ00, Corollary 5.1(2) and Theorem 3.7]; here, $[\rho \to \rho_<]$ denotes a natural representation for the space $\mathrm{LSC}(\mathbb{R})$ of lower semi-continuous functions $f : \mathbb{R} \to \mathbb{R}$ considered in [WZ00]. Specifically, a $[\rho \to \rho_<]$–name for such a f is an enumeration of all rational triples (a, b, c) such that $c < \min f\big[[a, b]\big]$—the latter making sense as a lower semi-continuous function attains its minimum (though not necessarily its maximum) on any compact set. $[\rho \to \rho_<]$ indeed *is* a representation for $\mathrm{LSC}(\mathbb{R})$ because different lower semi-continuous functions give rise to different such collections $\{(a, b, c) \in \mathbb{Q}^3 : \ldots\}$; cf. [WZ00, Lemma 3.3].

Lemma 18. $\mathrm{LSC}(\mathbb{R}) \times \mathbb{R} \ni (f, x) \mapsto f(x)$ *is* $([\rho \to \rho_<] \times \rho' \to \rho'_<)$*–computable.*

Proof. Let $(a_k, b_k, c_k)_k$ denote the given $[\rho \to \rho_<]$–name of $f \in \mathrm{LSC}(\mathbb{R})$ and $(q_n)_n$ the given ρ'–name for $x \in \mathbb{R}$. Our goal is to $\rho'_<$–compute $y := f(x)$. Define the sequence $p = (p_m)_m \subseteq \mathbb{Q} \cup \{+\infty\}$ by

$$p_{\langle k,\ell,n\rangle} \;:=\; \begin{cases} \max\big\{c_m \;:\; m \leq k \wedge [a_m, b_m] \supseteq [a_k, b_k]\big\} & \\ \qquad\qquad\qquad\qquad \text{if } q_n \in (a_k, b_k) \;\wedge\; |b_k - a_k| = 2^{-\ell} & \\ +\infty \qquad\qquad\qquad\quad \text{otherwise} & \end{cases} \tag{1}$$

From the given information, one can obviously compute p. Moreover this sequence satisfies

- $\liminf p \geq y$:
  Let $\epsilon > 0$ be arbitrary. Since f is lower semi-continuous, its preimage $f^{-1}\big[(y-\epsilon,\infty)\big] \ni x$ is an open set and therefore contains an entire ball around x. In fact, the center of this ball may be chosen as rational and its diameter of the form 2^{-L} for some $L \in \mathbb{N}$; formally (see Figure 6):

$$\exists K, L, K' \in \mathbb{N}: \quad x \in (a_{K'}, b_{K'}) \subseteq [a_K, b_K] \subseteq f^{-1}\big[(y-\epsilon,\infty)\big] \;\wedge$$
$$|b_K - a_K| = 2^{-L} \;\wedge\; a_{K'} = a_K + \tfrac{3}{2}\cdot 2^{-L-2} \;\wedge\; b_{K'} = b_K - \tfrac{3}{2}\cdot 2^{-L-2} \quad (2)$$

  where we have exploited that *every* rational pair (a,b) occurs in the list representing the $[\rho\!\to\!\rho_<]$–name. Moreover, as it consists of all rational triples (a,b,c) with $c < \min f\big[[a,b]\big]$,

$$\exists M \geq K: \;\; [a_K, b_K] = [a_M, b_M] \wedge c_M \geq \min f\big[[a_M, b_M]\big] - \epsilon \overset{(*)}{\geq} y - 2\epsilon \quad (3)$$

  with (*) a consequence of $[a_K, b_K] \subseteq f^{-1}\big[(y-\epsilon,\infty)\big]$ in Equation (2). Finally,

$$\lim_n q_n = x \in (a_{K'}, b_{K'}) \qquad \Rightarrow \qquad \exists N: \; \forall n \geq N: \;\; q_n \in (a_{K'}, b_{K'}) \;. \quad (4)$$

  So putting things together, for each $n \geq N$, $\ell \geq L'$, and $k \geq M$, we either have $p_{\langle k,\ell,n\rangle} = +\infty \geq y - 2\epsilon$; or we are in the first case of Equation (1), thus
  - $q_n \in (a_k, b_k)$ with $|b_k - a_k| \leq 2^{-\ell}$
  - $q_n \in (a_{K'}, b_{K'})$ by Equation (4)
  - hence $[a_k, b_k] \subseteq [a_K, b_K]$ by Equation (2) due to $\ell \geq L'$; cf. Figure 6.
  - So $[a_k, b_k] \subseteq [a_M, b_M]$ by Equation (3)
  - implying $p_{\langle k,\ell,n\rangle} \geq c_M \geq y - 2\epsilon$ by Equations (1) and (3) since $k \geq M$.

  Summarizing, it holds $p_{\langle k,\ell,n\rangle} \geq y - 2\epsilon$ for all $(k,\ell,n) \in \mathbb{N}^3$ not belonging to the finite set $\{0,1,\ldots,N-1\} \times \{0,1,\ldots,L'-1\} \times \{0,1,\ldots,M-1\}$ of exceptions. Consequently $\liminf p \geq y - 2\epsilon$; even $\liminf p \geq y$ because $\epsilon > 0$ was arbitrary.

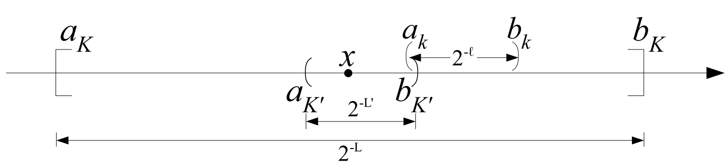


Fig. 6. Nesting of some rational intervals of dyadic length contained in $f^{-1}\big[(y-\epsilon,\infty)\big]$.
The parameters are chosen in such a way that, whenever $(a_{K'}, b_{K'})$ meets some other (a_k, b_k) of length $|b_k - a_k| = 2^{-\ell}$ for $\ell \geq L' := L+2$, then $[a_k, b_k]$ is entirely contained within the larger $[a_K, b_K]$.

- $\liminf p \leq y$
  Indeed: Since the $[\rho \to \rho_<]$–name contains in particular all rational pairs (a_k, b_k) and these intervals are dense in $\mathbb{R}$, there exists to every $\ell \in \mathbb{N}$ some k such that $|b_k - a_k| = 2^{-\ell}$ and $x \in (a_k, b_k)$. Furthermore it holds $q_n \in (a_k, b_k)$ for some sufficiently large n because $\lim_n q_n = x$. We have thus infinitely many triples (k, ℓ, n) for which $p_{\langle k,\ell,n\rangle}$ is defined by the first case in Equation (1) and thus agrees with some $c_m < \min f\big[[a_m, b_m]\big] \leq f(x) = y$ as $x \in (a_k, b_k) \subseteq [a_m, b_m]$.

Concluding, we have $\liminf_m p_m = y$. Although p may attain the value $+\infty$, this can easily be overcome by proceeding to $\tilde{p}_m := p_m$ for $p_m \neq \infty$ and $\tilde{p}_m := \max\{0, \tilde{p}_0, \ldots, \tilde{p}_{m-1}\}$ for $p_m = \infty$ because this transformation $p \mapsto \tilde{p}$ on sequences obviously does not affect their $\liminf < \infty$. This yields a $\tilde{\rho}'_<$–name for y which can finally be converted to the desired $\rho'_<$–name due to the easy part of Scholium 6. ☐

In order to obtain a similar uniform claim yielding Theorem 15c), recall that every $(\rho_< \to \rho_<)$–computable function $f : \mathbb{R} \to \mathbb{R}$ is necessarily both monotonically increasing and lower semi-continuous (Fact 10b+c). This suggests

Definition 19. *Let* MLSC($\mathbb{R}$) *denote the class of all monotonically increasing, lower semi-continuous functions $f : \mathbb{R} \to \mathbb{R}$. A $[\rho_< \to \rho_<]$–name for $f \in$* MLSC($\mathbb{R}$) *is an enumeration of the set $\{(a, c) \in \mathbb{Q}^2 : c < f(a)\}$.*

Lemma 20. *a) Distinct $f, g \in$* MLSC($\mathbb{R}$) *have different sets $\{(a, c) : \ldots\}$ according to Definition 19; that is, $[\rho_< \to \rho_<]$ constitutes a well-defined representation.*

b) A function $f \in$ MLSC($\mathbb{R}$) *is $(\rho_< \to \rho_<)$–computable iff it has a computable $[\rho_< \to \rho_<]$–name.*

c) Let $f \in$ MLSC($\mathbb{R}$)*, $(a_k, c_k)_k$ with $\{(a, c) \in \mathbb{Q}^2 : c < f(a)\} = \{(a_k, c_k) : k \in \mathbb{N}\}$, $x \in \mathbb{R}$, and $q = (q_n) \subseteq \mathbb{Q}$ with $x = \liminf_n q_n$. Then, the rational sequence p defined by*

$$p_{\langle k,n,\ell\rangle} := \begin{cases} \max\big\{c_m \ : \ m \leq k \ \wedge \ a_m \geq a_k\big\} & \text{if } \ a_k < q_n < a_k + 2^{-\ell} \\ +\infty & \text{otherwise} \end{cases}$$

satisfies $\liminf p = f(x) =: y$.

d) Therefore, the apply *operator* MLSC($\mathbb{R}$) $\times\, \mathbb{R} \ni (f, x) \mapsto f(x)$ *is $([\rho_< \to \rho_<] \times \rho'_< \to \rho'_<)$–computable.*

Proof. a) Let $f, g \in$ MLSC($\mathbb{R}$) with $f \neq g$, that is, w.l.o.g. $f(x_0) < g(x_0)$ for some $x_0 \in \mathbb{R}$. There exists some $c_0 \in \mathbb{Q}$ with $f(x_0) < c_0 < g(x_0)$. Being monotonically increasing and lower semi-continuous, their pre-images $f^{-1}\big[(c_0, \infty)\big] \not\ni x_0$ and $g^{-1}\big[(c_0, \infty)\big] \ni x_0$ on open half-interval (c_0, ∞) are again open half-intervals (x_f, ∞) and (x_g, ∞), respectively. As x_0 belongs to the second but not to the first, we have $x_g < x_0 < x_f$ and therefore $x_g < a_0 < x_f$ for some $a_0 \in \mathbb{Q}$. Then $a_0 \in (x_g, \infty) = g^{-1}\big[(c_0, \infty)\big]$ yields $c_0 < g(a_0)$ whereas $a_0 \notin (x_f, \infty) = f^{-1}\big[(c_0, \infty)\big]$ asserts $c_0 \not< f(a_0)$.

b) Let $\mathcal{M}$ denote a Type-2 Machine $(\rho_< \to \rho_<)$–computing $f \in \mathrm{MLSC}(\mathbb{R})$. Evaluating f at $a \in \mathbb{Q}$ by simulating $\mathcal{M}$ on the $\rho_<$–name $(a, a, a, \ldots)$ for a thus yields a $\rho_<$–name for $f(a)$ which is (equivalent to) a list of all $c \in \mathbb{Q}$ with $c < f(a)$ [Wei01, LEMMA 4.1.8]. So dove-tailing this simulation for all $a \in \mathbb{Q}$ yields the desired $[\rho_< \to \rho_<]$–name for f.
Conversely, knowing a $[\rho_< \to \rho_<]$–name $(a_k, c_k)_k$ for $f \in \mathrm{MLSC}(\mathbb{R})$ and given an increasing sequence $(q_n) \subseteq \mathbb{Q}$ with $x = \sup_n q_n$, let

$$p_n := c_n \text{ if } a_n \le q_n, \qquad p_n := -\infty \text{ otherwise} .$$

Then, in the first case, $p_n = c_n < f(a_n) \le f(q_n) \le f(x) =: y$ by monotonicity, and $p_n = -\infty \le y$ in the second; hence $\sup_n p_n \le y$. To see $\sup_n p_n \ge y$, fix arbitrary $\epsilon > 0$ and consider the open half-interval $f^{-1}\big[(y - \epsilon, \infty)\big] = (x_\epsilon, \infty)$ containing x and thus also some rational $a = a_K \in (x_\epsilon, x)$, $K \in \mathbb{N}$. Furthermore $q_n \nearrow x$ yields some $N \in \mathbb{N}$ such that $q_n \in (a_K, x)$ for all $n \ge N$. And finally there exists $M \ge N$ with $a_M = a_K$ and $c_M \ge f(a_M) - \epsilon$. Together this asserts $q_M > a_K = a_M$ because $M \ge N$ and thus $p_M = c_M \ge f(a_K) - \epsilon > y - 2\epsilon$ due to $a_k \in f^{-1}\big[(y - \epsilon, \infty)\big]$.

c) Take arbitrary $\epsilon > 0$. As f is increasing and lower semi-continuous, the preimage $f^{-1}\big[(y-\epsilon, \infty)\big]$ is an open half-interval (x_ϵ, ∞) containing x. Therefore there exist $K, L \in \mathbb{N}$ such that $x_\epsilon < a_K$ and $a_K + 2^{-L} < x$; furthermore, the sequence $(a_k, c_k)_k$ containing *all* rational pairs (a, c) with $c < f(a)$, there is $M \ge K$ such that $a_M = a_K$ and $c_M \ge f(a_M) - \epsilon$; and finally, since $\liminf_n q_n = x > a_M + 2^{-L}$, it holds $q_n > a_M + 2^{-L}$ for all $n \ge N$ with an appropriate $N \in \mathbb{N}$. Observe that $q > a_M + 2^{-L}$ and $a < q < a + 2^{-L}$ implies $a \ge a_M$; so together we have for all $n \ge N$, $\ell \ge L$, and $k \ge M$ that $p_{\langle k,n,\ell\rangle}$ is either $+\infty$ or $\ge c_M \ge f(a_K) - \epsilon \ge f(x_\epsilon) - \epsilon \ge y - 2\epsilon$ due to monotonicity of f and by definition of $x_\epsilon < a_K$. This proves $\liminf p \ge y$ because ϵ was arbitrary.
To see the reverse inequality "$\liminf p \le y$", take arbitrary $\ell \in \mathbb{N}$. There exists $k \in \mathbb{N}$ with $a_k < x < a_k + 2^{-\ell}$ and, because of $\liminf_n q_n = x$, also $n \in \mathbb{N}$ with $a_k < q_n < a_k + 2^{-\ell}$. We therefore have infinitely many triples (n, k, ℓ) for which $p_{\langle n,k,\ell\rangle}$ agrees with a certain $c_m < f(a_m) \le f(a_k) \le f(x) = y$.

d) Given a $\rho'_<$–name for x, one can obtain a sequence $(q_n) \subseteq \mathbb{Q}$ with $x = \liminf_n q_n$ by virtue of Scholium 6. From this, the sequence $p \subseteq \mathbb{Q}$ with $\liminf p = f(x)$ according to c) is obviously computable and yields, again by Scholium 6, a $\rho'_<$–name for $y = f(x)$. $\square$

Concluding this subsection, the classes of $(\rho^{(d)} \to \rho^{(d)})$–computable real functions $f : \mathbb{R} \to \mathbb{R}$ form, for $d = 0, 1, \ldots$ respectively, a hierarchy. By Fact 3, this hierarchy is strict as can be seen from the constant functions $f(x) \equiv c$ with $c \in \Delta_{d+1}\mathbb{R}$.

4.2 Arithmetic Weierstraß Hierarchy

Section 4.1 established the sequence $\rho, \rho', \rho'', \ldots$ of increasingly weaker representations for $\mathbb{R}$ to yield the strict hierarchy of $(\rho \to \rho)$–computable, $(\rho' \to \rho')$–

computable, and $(\rho'' \to \rho'')$–computable functions $f : [0,1] \to \mathbb{R}$. We now compare these classes with those induced by the other kind of real hypercomputation suggested in Section 3: relative to the Halting Problem $H = \emptyset'$ and its iterated jumps $\emptyset''$, ...

Such a comparison makes sense because both weakly and oracle-computable real functions are necessarily continuous according to Fact 10d)/Theorem 11c) and Lemma 8, respectively.

The classical WEIERSTRASS Approximation Theorem establishes any continuous real function $f : [0,1] \to \mathbb{R}$ to be the uniform limit $f = \mathrm{ulim}_n P_n$ of a sequence of rational polynomials $(P_n) \subseteq \mathbb{Q}[X]$. Here, 'ulim' suggestively denotes uniform convergence of continuous functions on $[0,1]$, that is the requirement

$$\sup_{0\leq x\leq 1} |f(x) - P_n(x)| \;\; =: \;\; \|f - P_n\| \;\to\; 0 \qquad \text{as} \;\; n \to \infty \;\; .$$

The famous Effective Weierstraß Theorem due to POUR-EL, CALDWELL, and HAUCK relates effectively evaluable to effectively approximable real functions:

Fact 21. *A function $f : [0,1] \to \mathbb{R}$ is $(\rho \to \rho)$–computable if and only if it holds*

$[\rho\to\rho]$: *There exists a computable sequence of (degrees and coefficients of) rational polynomials $(P_n) \subseteq \mathbb{Q}[X]$ such that* $\quad \|f - P_n\| \;\leq\; 2^{-n} \quad (5)$

Proof. See [PER89, SECTION 0.7], [PEC75], or [Hau76].
The notion "$[\rho \to \rho]$" is justified as the list $(P_n)_n$ constitutes (or is equivalent to) a $[\rho\to\rho]$–name for $f = \mathrm{ulim}_n P_n$; cf. [Wei01, bottom of p.160]. □

The aforementioned other approach to continuous real hypercomputation arises from allowing the fast convergent sequence $(P_n)_n \subseteq \mathbb{Q}[X]$ to be computable in $\emptyset'$ or $\emptyset''$. The $\emptyset'$–computable $f : [0,1] \to \mathbb{R}$ have in fact already been characterized by HO as Claim a) of the following

Lemma 22. *a) To a real function $f : [0,1] \to \mathbb{R}$, there exists a $\emptyset'$–computable sequence of polynomials (P_n) satisfying Equation (5) if and only if it holds*

$[\rho\to\rho]'$: *There is a computable sequence $(Q_m) \subseteq \mathbb{Q}[X]$ converging uniformly (although not necessarily 'fast') to f, that is, with $f = \mathrm{ulim}_{m\to\infty} Q_m$.*

b) For an arbitrary oracle A, the sequence (of discrete degrees and numerators/denominators of the coefficients of) $(P_n)_n \subseteq \mathbb{Q}[X]$ is A'–computable iff there exists an A–computable sequence $(Q_{n,m})_{n,m} \subseteq \mathbb{Q}[X]$ such that

$$\forall n \; \exists M \; \forall m \geq M : \quad P_n = Q_{n,m} \;\; .$$

c) To a real function $f : [0,1] \to \mathbb{R}$, there exists a $\emptyset''$–computable sequence of polynomials (P_n) satisfying Equation (5) if and only if it holds

$[\rho\to\rho]''$: *There is a computable sequence $(Q_m) \subseteq \mathbb{Q}[X]$ s.t. $f = \mathrm{ulim}_i \, \mathrm{ulim}_j \, Q_{\langle i,j\rangle}$.*

Notice the similarity of Claims a+c) to Fact 3b).

Proof. a) See [Ho99, THEOREM 16].

b) is a straight-forward extension of SHOENFIELD's Limit Lemma [Soa87, LEMMA III.3.3] and its generalization to sequences of rational numbers [ZW01, LEMMA 4.1].

c) If $\emptyset''$–computable $(P_n)_n \subseteq \mathbb{Q}[X]$ satisfies Equation (5), then by virtue of the relativization of [Ho99, THEOREM 16] there exists some $\emptyset'$–computable $(\tilde{P}_n)_n \subseteq \mathbb{Q}[X]$ converging to the same f uniformly on $[0,1]$. By Claim a) in turn, $\tilde{P}_n = \operatorname{ulim}_m Q_{n,m}$ for some computable sequence $(Q_{n,m}) \subseteq \mathbb{Q}[X]$. Conversely if $f = \operatorname{ulim}_n \tilde{P}_n$ with $\tilde{P}_n := \operatorname{ulim}_m Q_{n,m}$ for a computable $(Q_{n,m})$, then let $P_n := Q_{n,m_n}$ where

$$m_n \quad := \quad \min\left\{m \in \mathbb{N} \,\middle|\, \forall k, \ell \geq m \,:\, \|Q_{n,k} - Q_{n,\ell}\| \leq 2^{-n}\right\} \ . \qquad (6)$$

This sequence $(m_n)_n$ is well-defined and yields $\|P_n - \tilde{P}_n\| \leq 2^{-n}$, so $f = \operatorname{ulim}_n \tilde{P}_n = \operatorname{ulim}_n P_n$. Moreover, the minimum in Equation (6) is taken over a co-r.e. set — $r := \|Q_{n,k} - Q_{n,\ell}\| \cdot 2^n$ being ρ–computable by virtue of [Wei01, COROLLARY 6.2.5] and the complementary condition "$r > 1$" ρ–r.e. open and hence recursive in $\emptyset'$. Similar to Equation (6), this $\emptyset'$–computable sequence $(P_n)_n \subseteq \mathbb{Q}[X]$ converging uniformly though just ultimately to f can be turned into a $\emptyset''$–computable, fast convergent one. □

We thus have two hierarchies of hypercomputable continuous real functions:
- $[\rho \to \rho]$, $[\rho \to \rho]'$, $[\rho \to \rho]''$, ...
- $(\rho \to \rho)$, $(\rho' \to \rho')$, $(\rho'' \to \rho'')$, ...

By the Fact 21, their respective ground-levels coincide. Our next result compares their respective higher levels. They turn out to lie skewly to each other (Claim c).

Theorem 23. *a) Let $f : [0,1] \to \mathbb{R}$ be $[\rho \to \rho]'$–computable (in the sense of Lemma 22a). Then, f is $(\rho' \to \rho')$–computable.*
b) Let $f : [0,1] \to \mathbb{R}$ be $(\rho' \to \rho')$–computable. Then, f is $[\rho \to \rho]''$–computable.
c) There is a $(\rho' \to \rho')$–computable but not $[\rho \to \rho]'$–computable $f : [0,1] \to \mathbb{R}$.

The idea to c) is that every $[\rho \to \rho]'$–computable $f : [0,1] \to \mathbb{R}$ has a modulus of uniform continuity recursive in $\emptyset'$; whereas a $(\rho' \to \rho')$–computable f, although uniformly continuous as well, in general does not.

Before proceeding to the proof, we first provide a tool which turns out to be useful in the sequel. It is well-known in Recursive Analysis that, although equality of real numbers is ρ–undecidable due to the Main Theorem, inequality is at least semi-decidable. The following lemma generalizes this to ρ' and to $(\rho' \to \rho'_<)$–computable functions:

Lemma 24. *a) Let $f : \mathbb{R} \to \mathbb{R}$ be $(\rho \to \rho_<)$–computable. Then the property*

$$\left\{(a,b,c) \in \mathbb{Q}^3 \,\middle|\, \exists x \in [a,b] : f(x) > c\right\}$$

whether f on $[a,b]$ exceeds c is semi-decidable.
b) Let $f : \mathbb{R} \to \mathbb{R}$ be $(\rho' \to \rho'_<)$–computable. Then the property

$$\left\{(a,b,c) \in \mathbb{Q}^3 \,\middle|\, \exists x \in [a,b] : f(x) > c\right\}$$

whether f on $[a,b]$ exceeds c is semi-decidable relative to $\emptyset'$.

c) *Let $f : \mathbb{R} \to \mathbb{R}$ be $(\rho' \to \rho')$–computable. Then the property*

$$\big\{(a, b, c, m) \in \mathbb{Q}^3 \times \mathbb{N} \,\big|\, \forall x \in [a, b] : c - 2^{-m} \leq f(x) \leq c + 2^{-m}\big\}$$

is decidable relative to $\emptyset''$.

Proof. a) is standard; c) follows from b) which is established as follows:
By lower semi-continuity of f due to Theorem 11a), if f exceeds c on the compact interval $[a, b]$, then it does so on some *rational* x. Feeding, for any such $x \in [a, b] \cap \mathbb{Q}$, the ρ'–name $(x, x, x, x, \ldots)$ for x into the Type-2 Machine computing f reveals the mapping $\mathbb{Q} \ni x \mapsto f(x)$ to be $(\nu_{\mathbb{Q}} \to \rho'_<)$–computable. With $\emptyset'$–oracle, it thus becomes $(\nu_{\mathbb{Q}} \to \rho_<)$–computable by virtue of [ZW01, Lemma 4.2]. Since $\{(y, c) : y > c\}$ is $(\rho_< \times \nu_{\mathbb{Q}})$–semi-decidable, the claim follows. □

Proof (Theorem 23).

a) Let $(P_n) \subseteq \mathbb{Q}[X]$ denote a computable sequence converging uniformly (yet not necessarily fast) to f. Let $x \in [0, 1]$ be given as the limit of a sequence $(q_n) \subseteq \mathbb{Q}$. Then, $p_n := P_n(q_n)$ eventually converges to $f(x)$.
b) Let $x \in [0, 1]$ be given by (an equivalent to) its ρ–name in form of two rational sequences (a_n) and (b_n) with $\{x\} = \bigcap_n [a_n, b_n]$. There exists a rational sequence (c_m) forming a ρ–name for $f(x)$, that is, satisfying $c - 2^{-m} \leq f(x) \leq c + 2^{-m}$ for all m; and by virtue of Lemma 24c), such a sequence can be found with the help of a $\emptyset''$–oracle. This reveals that f is $\emptyset''$–recursive in the sense of [Ho99, Section 4] and thus, similarly to [Ho99, Corollary 17], $[\rho \to \rho]''$–computable.
c) Let $h : \mathbb{N} \to \mathbb{N}$ denote a $\emptyset'$–computable injective total enumeration of some subset $H = h[\mathbb{N}] \in \Sigma_2 \setminus \Delta_2$. Observe that $a_m := 2^{-h(m)}$ is a ρ'–computable real sequence converging to 0 with modulus of convergence [Wei01, Definition 4.2.2] lacking $\emptyset'$–recursivity; compare [Wei01, Exercise 4.2.4c)]. Let $\varphi : \mathbb{R} \to \mathbb{R}$ denote some $(\rho \to \rho)$–computable unit pulse, that is, vanishing outside $[0, 1]$ and having height $\max_x \varphi(x) = \varphi(\frac{1}{2}) = 1$; a piecewise linear '*hat*' function for instance will do fine but we can even choose φ as in [PER89, Theorem 1.1.1] to obtain the counter-example

$$f(x) \ := \ \sum_{m \in \mathbb{N}} a_m \cdot \varphi(2^m x - 1) \ , \tag{7}$$

(that is, a non-overlapping superposition of scaled translates of such pulses) to be C^∞. By Theorem 15a), $x \mapsto a_m \cdot \varphi(2^m x - 1)$ is $(\rho' \to \rho')$–computable; in fact even uniformly in m: Given $(q_n)_n \subseteq \mathbb{Q}$ with $x = \lim_n q_n$, one can for each $M \in \mathbb{N}$ obtain a sequence $(p_{k,M})_k \subseteq \mathbb{Q}$ with $\lim_k p_{k,M} = \sum_{m \leq \boldsymbol{M}} a_m \cdot \varphi(2^m x - 1) =: f_M$. The functions f_M converge uniformly (though not effectively) to f because of the disjoint supports of the terms $\varphi(2^m x - 1)$ in Equation (7). Therefore $\lim\limits_M p_{M,M} = f(x)$, thus establishing $(\rho' \to \rho')$–computability of f.
Suppose f was $[\rho \to \rho]'$–computable. Then, by virtue of [Ho99, Lemma 15],

it has a $\emptyset'$–recursive modulus of uniform continuity; cf. [Wei01, DEFINITION 6.2.6.2]. In particular given $n \in \mathbb{N}$, one can $\emptyset'$–compute $m \in \mathbb{N}$ such that $x := 2^{-m}$ and $y := \frac{3}{2}x$ satisfy $2^{-n} \geq |f(x) - f(y)| = |0 - a_m|$ contradicting that (a_m) has no $\emptyset'$–recursive modulus of continuity. □

5 Type-2 Nondeterminism

Concerning the two kinds of real hypercomputation considered so far—based on oracle-support and weak real number encodings that is—recall that the according proofs of Fact 10 and Theorem 11 crucially rely on the underlying Turing Machines to behave deterministically. This raises the question whether nondeterminism might yield the additional power necessary for evaluating discontinuous real functions like Heaviside's.

In the discrete (i.e., Type-1) setting where any computation is required to terminate, the finitely many possible choices of a nondeterministic machine can of course be simulated by a deterministic one—however already here subject to the important condition that all paths of the nondeterministic computation indeed terminate, cf. [STvE89]. In contrast, a Type-2 computation realizes a transformation from/to infinite strings and is therefore a generally non-terminating process. Therefore, nondeterminism here involves an infinite number of guesses which turns out cannot be simulated by a deterministic Type-2 machine.

We also point out that nondeterminism has already before been revealed not only a useful but indeed the most natural concept of computation on Σ^ω. More precisely, BÜCHI extended Finite Automata from finite to infinite strings and proved that here, as opposed deterministic, *non*deterministic ones are closed under complement [Tho90] and thus the appropriate model of computation. Since

Chomsky-Level	Σ^*	Σ^ω
3: regular	Finite Automata	Büchi Automata (nondeterministic)
2: context-free		
1: context-sensitive		
0: unrestricted	(Type-1) Turing Machines	*non*deterministic Type-2 Machines

Fig. 7. Models of Computation in Chomsky's Hierarchies over finite/infinite strings

automata and Turing Machines constitute the bottom and top levels, respectively, of CHOMSKY's Hierarchy of classical languages $L \subseteq \Sigma^*$ (Type-1 setting), we suggest that over infinite strings Σ^ω (Type-2 setting) both their respective counterparts, that is Büchi Automata *and Type-2 Machines* be considered nondeterministically; compare Figure 7.

The concept of nondeterministic computation of a function $f :\subseteq \Sigma^* \to \Sigma^*$ (as opposed to a decision problem) is taken from the famous IMMERMAN-SZELEPSCÉNYI Theorem in computational complexity; cf. for instance [Pap94, the paragraph preceding THEOREM 7.6]: For $\bar{x} \in \mathrm{dom}(f)$, some computing paths of the according machine $\mathcal{M}$ may fail by leading to rejecting states, as long as

1) there *is* an accepting computation of $\mathcal{M}$ on $\bar{x}$ and
2) *every* accepting computation of $\mathcal{M}$ on $\bar{x}$ yields the correct output $f(\bar{x})$.

This notion extends straight-forwardly from Type-1 to the Type-2 setting:

Definition 25. *Let A and B be sets with respective representations $\alpha :\subseteq \Sigma^\omega \to A$ and $\beta :\subseteq \Sigma^\omega \to B$. A function $f :\subseteq A \to B$ is called* nondeterministically *$(\alpha \to \beta)$–computable if some nondeterministic one-way Turing Machine $\mathcal{M}$,*

- *upon input of any α–name $\bar{\sigma} \in \Sigma^\omega$ for some $a \in \mathrm{dom}(f)$,*
- *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 $b = f(a)$.*

This definition is sensible insofar as it leads to closure under composition:

Observation 26. *Let $f :\subseteq A \to B$ be nondeterministically $(\alpha \to \beta)$–computable and $g :\subseteq B \to C$ be nondeterministically $(\beta \to \gamma)$–computable. Then, $g \circ f :\subseteq A \to C$ is nondeterministically $(\alpha \to \gamma)$–computable.*

A subtle point in Definition 25, the nondeterministic machine may 'withdraw' a guess as long as it does so within finite time.

Example 27 ('Deciding' the Arithmetic Hierarchy). Let $P \subseteq \mathbb{N}$ be recursive,

$$A \;=\; \big\{x \in \mathbb{N} \,|\, \forall y_1 \in \mathbb{N} \exists z_1 \in \mathbb{N} \;\; \forall y_2 \exists z_2 \ldots \forall y_k \exists z_k : \langle x; y_1, z_1, \ldots, y_k, z_k\rangle \in P\big\}$$

on (or below) level Π_{2k} of Kleene's Arithmetic Hierarchy. Then the function $\tilde{\chi}_A : \mathbb{N} \to \{0,1\} \times \{\text{␣}\}^\omega$ is nondeterministically computable:
Observe that $x \in A$ iff

$$\exists f_1, f_2, \ldots, f_k : \mathbb{N} \to \mathbb{N} \; \forall y_1, y_2, \ldots, y_k \in \mathbb{N} \;:\; \langle x; y_1, f(y_1), \ldots, y_k, f(y_k)\rangle \in P$$

So given $x \in \mathbb{N}$, let $\mathcal{M}_+$ output "1" and then verify, while continuously spitting out blanks "␣", that $\chi_A(x) = 1$ indeed holds. To this end, the machine starts 'guessing' the values of $\bar{f} = (f_1, \ldots, f_k)$ restricted to $\{0, 1, \ldots, n\}$ for $n = 1, 2, \ldots$ Simultaneously by means of dove-tailing, $\mathcal{M}_+$ tries all $\bar{y} \in \{0, 1, \ldots, n\}^k$ and aborts in case that the assertion "$\langle x; y_1, f(y_1), \ldots, y_k, f(y_k)\rangle \in P$" fails.
Now if $x \in A$, then an appropriate $\bar{f}$ exists, is ultimately 'found' by $\mathcal{M}_+$, and leads to indefinite execution; whereas if $x \notin A$, then $\mathcal{M}_+$ will eventually terminate for any guessed $\bar{f}$.
Since $\mathbb{N} \setminus A \in \Pi_{2k+2}$, a machine $\mathcal{M}_-$ can output "0" and then similarly verify $\chi_A(x) = 0$. The final machine $\mathcal{M}$, upon input of $x \in \mathbb{N}$, nondeterministically chooses to proceed either like $\mathcal{M}_+$ or like $\mathcal{M}_-$. Its computation satisfies the requirements of Definition 25. □

‖ This condition is slightly stronger than the one required in [Zie05, DEFINITION 14].

The power of nondeterministic computation permits conversion forth and back among representations on the Real Arithmetic Hierarchy from Definition 2:

Theorem 28 (Third Main Theorem of Real Hypercomputation). *For each $d = 0, 1, 2, \ldots$, the identity $\mathbb{R} \ni x \mapsto x$ is nondeterministically $(\rho^{(d+1)} \to \rho^{(d)})$–computable. It is furthermore nondeterministically $(\rho \to \rho_{b,2})$–computable.*

Proof. Consider first the case $d = 0$. Let $x \in \mathbb{R}$ be given by a sequence $(q_n) \subseteq \mathbb{Q}$ eventually converging to x. Then, there exists a fast convergent Cauchy subsequence $(q_{n_k})_k$, that is, satisfying

$$\forall k \geq \ell : \quad |q_{n_k} - q_{n_\ell}| \;\leq\; 2^{-\ell-1} \tag{8}$$

and thus forming a ρ–name for x. To find this subsequence, guess iteratively for each $k \in \mathbb{N}$ some $n_k > n_{k-1}$ and check whether it complies with Inequality (8) for the (finitely many) $\ell \leq k$; if it does not, we may abort this computation in accordance with Definition 25.

For $d = 1$, let $x = \lim_n x_n$ with $x_n = \lim_m q_{n,m}$. Then apply the case $d = 0$ to convert for each n the ρ'–name $(q_{n,m})_m$ of $x_n \in \mathbb{R}$ into an according ρ–name, that is, a sequence $p_{n,m}$ satisfying $|x_n - p_{n,m}| \leq 2^{-m}$. Its diagonal $(p_{n,n})_n$ then has $|x - p_{n,n}| \leq |x - x_n| + 2^{-n} \to 0$ and is thus a ρ'–name for x. Higher levels d can be treated similarly by induction.

For $(\rho \to \rho_{\mathrm{b},2})$–computability, let $x \in (0, 2)$ be given by a fast convergent sequence $(q_n) \subseteq \mathbb{Q}$. We guess the leading digit $b \in \{0, 1\}$ for x's binary expansion $b.*$; in case $b = 0$, check whether $x > 1$—a ρ–semi decidable property—and if so, abort; similarly in case $b = 1$, abort if it turns out that $x < 1$. Otherwise (that is, proceeding while simultaneously continuing the above semi-decision process via dove-tailing) replace x by $2(x - b)$ and repeat guessing the next bit. ☐

It is also instructive to observe how, in the case of non-unique binary expansion (i.e., for dyadic x), nondeterminism in the above $(\rho \to \rho_{\mathrm{b},2})$–computation generates, in accordance with the third requirement of Definition 25, both possible expansions.

Theorem 28 implies that *non*deterministic computability of real functions is largely independent of the representation under consideration — in striking contrast to the classical case (Corollary 16) where the effectivity subtleties arising from different encodings had confused already Turing himself [Tur37].

Corollary 29. *a) With respect to nondeterministic reduction "$\preceq_n$", it holds* $\rho_{b,2} \;\equiv_n\; \rho \;\equiv_n\; \rho_< \;\equiv_n\; \rho' \;\equiv_n\; \rho'_< \;\equiv_n\; \rho'' \;\equiv_n\; \ldots$.

b) The entire Real Arithmetic Hierarchy of Weihrauch *and* Zheng *is nondeterministically computable.*

Proof. a) follows from Lemma 5 and Theorem 28.

b) Let $x \in \Delta_{d+1}\mathbb{R}$ for some $d \in \mathbb{N}$. Then, $x \in \mathbb{R}$ is $\rho^{(d)}$–computable by Definition 2; hence nondeterministically also $\rho_{\mathrm{b},2}$–computable by a). Alternatively combine Example 27 with Fact 3a). ☐

In particular, this kind of hypercomputation allows for nondeterministic $(\rho \to \rho)$–evaluation of Heaviside's function by appending to the $(\rho \to \rho_<)$–computation in Example 9 a conversion from $\rho_< \preceq \rho'$ back to ρ. Section 5.1 establishes many more real functions, both continuous and discontinuous ones, to be nondeterministically computable, too.

5.1 Nondeterministic and Analytic Computation

We now show that Type-2 nondeterminism includes the algebraic so called BCSS-model of real number computation due to Blum, Cucker, Shub, and Smale [BSS89,BCSS98] employed for instance in Computational Geometry [PS85, Section 1.4]. As a matter of fact, nondeterministic real hypercomputation even covers all *quasi-strongly δ–$\mathbb{Q}$–analytic* functions $f :\subseteq \mathbb{R}^d \to \mathbb{R}$ in the sense of Chadzelek and Hotz [CH99, Definition 5]. The latter can be considered a synthesis of the Type-2 (i.e., infinite approximate) and the BCSS (i.e., finite exact) model of real number computation. Its computational power admits an elegant characterization (see Lemma 31b+c) in terms of the following

Definition 30. *A ρ_H–name for $x \in \mathbb{R}$ is some $(q_n)_n \subseteq \mathbb{Q}$ such that*

$$\exists N\ \forall n \geq N : \quad |q_n - x| \leq 2^{-n} \quad . \tag{9}$$

The encoding sequence of rational approximations must thus converge fast with the exception of some initial segment of finite yet unknown length. It corresponds to ρ–computation by an *Inductive* Turing Machine in the sense of [Bur04] which is roughly speaking a Type-2 Machine but whose output tape(s) need not be one-way [Wei01, Section 2.1] provided that the contents of every cell ultimately stabilizes.

Lemma 31. *a) It holds $\rho \precnsim \rho_H \precnsim \rho'$.*
b) A function $f :\subseteq \mathbb{R}^N \to \mathbb{R}$ is $(\rho^N \to \rho_H)$–computable iff it is computable by a quasi-strongly δ–$\mathbb{Q}$–analytic machine.
c) $(\rho \to \rho_H)$–computability is equivalent to $(\rho_H \to \rho_H)$–computability.
d) The class of $(\rho_H \to \rho_H)$–computable functions is closed under composition.
The above claims relativize.

Proof. a) is immediate.
b) Observe that the *robustness* of the program π required in [CH99, top of p.157] amounts to the argument $x \in \mathbb{R}$ of f being accessible by rational approximations $q_n \in \mathbb{Q}$ of error $|q_n - x| \leq 2^{-n}$, that is, in terms of a ρ–name. The output $y = f(x)$ on the other hand proceeds by way of two sequences $(p_m)_m, (\epsilon_m)_m \subseteq \mathbb{Q}$ such that $\epsilon_m \to 0$ and $|p_m - y| \leq \epsilon_m$ holds for all sufficiently large m. By effectively proceeding to an appropriate subsequence, we can w.l.o.g. suppose $\epsilon_m = 2^{-m}$, hence (p_m) is ρ_H–name of y.
c) By a), every $(\rho_\mathrm{H} \to \rho_\mathrm{H})$–computable function is $(\rho \to \rho_\mathrm{H})$–computable, too. For the converse implication, take the Type-2 Machine $\mathcal{M}$ converting ρ–names for $x \in \mathbb{R}$ to ρ_H–names for $y = f(x)$. Let (q_n) satisfy Equation (9) for some unknown $N \in \mathbb{N}$.

Now simulate $\mathcal{M}$ on $(q_n)_{n\geq 0}$, implicitly supposing that it is a valid ρ–name, i.e., that $N = 0$. Simultaneously check consistency of Condition (9), that is, verify $|q_n - q_k| \leq 2^{-n+1} \forall k \geq n \geq N$. If (or, rather, when) the latter fails for some (k_0, n_0), $\mathcal{M}$ has output only finitely (say $M_0 \in \mathbb{N}$) many $p_m \in \mathbb{Q}$. In that case, restart $\mathcal{M}$ on $(q_n)_{n\geq 1}$ presuming $N = 1$ while, again, checking this presumption consistent with (9); but this time throw away the first M_0 elements of the sequence printed by $\mathcal{M}$. Continue analogously for $N = 2, 3, \ldots$.

We claim that this yields output of a ρ_{H}–name for y. Since (q_n) is a valid ρ_{H}–name, a feasible N will eventually be found. Before that happens, the several partial runs of $\mathcal{M}$ have produced only finitely (say $M \in \mathbb{N}$) many rational numbers p_m; and after that, the final simulation generates by presumption a valid ρ_{H}–name for y. Out of this sequence $(p_m)_m$, the first M entries may have been exchanged by outputs of previous simulation trials; however according to Definition 30, the representation ρ_{H} is immune against such finite modifications.

d) Quasi-strongly δ–$\mathbb{Q}$–analytic functions are closed under composition according to [CH99, LEMMA 2]; now apply b+c). □

A BCSS (or, equivalently, an $\mathbb{R}$–) machine $\mathcal{M}$ is permitted to store a finite number of *arbitrary* real constants $r_1, \ldots, r_k$ [CH99, Instruction 1(b) in TABLE 1 on p.154] and use it for instance to solve the Halting or any other fixed discrete problem [BSS89, EXAMPLE 6]. Slightly correcting [CH99, THEOREM 3], $\mathcal{M}$'s simulation by a rational machine thus *requires* knowledge of $\bar{r} := (r_1, \ldots, r_k) \in \mathbb{R}^k$; e.g. by virtue of oracle access to ($\mathcal{O} := \{\mathrm{bin}(n) : \sigma_n = 1\} \subseteq \{0,1\}^*$ as natural encoding of a $\rho^k_{\mathrm{b},2}$–name $\bar{\sigma} \in \{0,1\}^\omega$ of) $\bar{r}$—compare [BV99] for the case of simulating ρ–semi decidability.

Proposition 32. *a) A function $f :\subseteq \mathbb{R} \to \mathbb{R}$ computable by a BCSS–machine with constants $\bar{r} \in \mathbb{R}^k$ is also $(\rho_H \to \rho_H)$–computable* relative to $\bar{r}$.

b) Every $(\rho \to \rho)$–computable function $f :\subseteq \mathbb{R} \to \mathbb{R}$ is also $(\rho_H \to \rho_H)$–computable.

c) Let $f :\subseteq \mathbb{R} \to \mathbb{R}$ be $(\rho_H \to \rho_H)$–computable relative to some oracle $\mathcal{O} \subseteq \Sigma^*$ in (Kleene's) Arithmetic Hierarchy. Then f is nondeterministically Type-2 computable.

Proof. a) See (the proof of) [CH99, THEOREM 3].

b) Combine Lemma 31a+c).

c) The nondeterministic simulation can answer queries to $\mathcal{O}$ due to Example 27. As $\rho \equiv_{\mathrm{n}} \rho_{\mathrm{H}} \equiv_{\mathrm{n}} \rho'$ by Corollary 29a) and Lemma 31a), the claim follows. □

Let us illustrate Proposition 32a) with the following

Example 33. Heaviside's Function $h : \mathbb{R} \to \{0,1\}$ is trivially BCSS–computable. It is also $(\rho_{\mathrm{H}} \to \rho_{\mathrm{H}})$–computable by means of *conservative branching*: Given $x \in \mathbb{R}$ by virtue of $(q_n) \subseteq \mathbb{Q}$ with (9) and unknown $N \in \mathbb{N}$, let $p_n := 0$ if $q_n \leq 2^{-n}$ and $p_n := 1$ otherwise.

Indeed if $x \leq 0$ then, for all $n \geq N$, $q_n \leq 2^{-n}$ and thus $p_n = 0 = f(x)$. If on the other hand $x > 0$, $x > 2^{-M}$ for some $M \in \mathbb{N}$; then, for all $n \geq \max\{M+1, N\}$, $q_n > 2^{-n}$ so $p_n = 1 = f(x)$. □

Of course the class of nondeterministic Type-2 Machines (and thus also that of the nondeterministically computable real functions) is still only countably infinite: most (even constant) functions $f : \mathbb{R} \to \mathbb{R}$ actually remain infeasible to this kind of real hypercomputation.

6 Conclusion

Recursive Analysis is often criticized for being unable, due to its Main Theorem, to non-trivially treat discontinuous functions. Although one can in Type-2 Theory devise sensible computability notions for, say, generalized (and in particular discontinuous) functions as for instance in [ZW03], evaluation $x \mapsto f(x)$ of an L^2 function or a distribution f at a point $x \in \mathbb{R}$ does not make sense here already mathematically. Regarding the Main Theorem's connection to the Church-Turing Hypothesis indicated in the introduction, the present work has investigated whether and which models of hypercomputation allows for lifting that restriction.

A first idea, relativized computation on oracle Turing Machines, was ruled out right away. A second idea, computation based on weakened encodings of real numbers, renders evaluation $x \mapsto h(x)$ of Heaviside's function—although discontinuous—for instance $(\rho \to \rho_<)$–computable. The drawback of this notion of real hypercomputation: it lacks closure under composition.

Example 34. Let $f : \mathbb{R} \to \mathbb{R}$, $f(0) := 0$ and $f(x) := 1$ for $x \neq 0$. Let $g(x) := -x$. Then both f and g are $(\rho \to \rho_<)$–computable but their composition $g \circ f : 0 \mapsto 0,\ 0 \neq x \mapsto -1$ lacks lower semi–continuity.

Requiring both argument x and value $y = f(x)$ to be encoded in the same way—say, ρ, ρ', or ρ''—asserts closure under both composition and negation $f \mapsto -f$; and the prerequisites of the Main Theorem applies only to the case $(\rho \to \rho)$. Surprisingly, $(\rho' \to \rho')$–computability and $(\rho'' \to \rho'')$–computability still require continuity! These results extend to $(\rho^{(d)} \to \rho^{(d)})$–computability for arbitrary d, although already the step from $d = 1$ to 2 made proofs significantly more involved.

These claims immediately relativize, that is, even a mixture of oracle support and weak real number encodings does not allow for hypercomputational evaluation of discontinuous functions. This is due to the purely information-theoretic nature of the arguments employed, specifically: the deterministic behavior of the Turing Machines under consideration.

So we have finally looked at nondeterminism as a further way of enhancing the underlying machine model beyond Turing's barrier. Over the Type-2 setting of infinite strings Σ^ω, this parallels Büchi's well-established generalization of finite automata to so-called *ω–regular* languages. While the practical realizability of Type-2 nondeterminism is admittedly even more questionable than that of classical $\mathcal{NP}$-machines, it does yield an elegant notion of hypercomputation with nice closure properties and invariant under various encodings.

A precise characterization of the class of nondeterministically computable real functions will be subject of future work.

References

Bar03. G. Barmpalias: "A Transfinite Hierarchy of Reals", pp.163–172 in *Mathematical Logic Quarterly* vol.**49(2)** (2003).

BSS89. L. Blum, M. Shub, S. Smale: "On a theory of computation and complexity over the real numbers", pp.1–46 in *Bull. Amer. Math. Soc.* vol.**21** (1989).

BCSS98. L. Blum, F. Cucker, M. Shub, S. Smale: "*Complexity and Real Computation*", Springer (1998).

BV99. P. Boldi, S. Vigna: "Equality is a jump", pp.49–64 in *Theoretical Computer Science* vol.**219** (1999).

BH02. V. Brattka, P. Hertling: "Topological Properties of Real Number Representations", pp.241–257 in *Theoretical Computer Science* vol.**284** (2002).

Bur04. M. Burgin: "Algorithmic Complexity of Recursive and Inductive Algorithms", pp.31–60 in *Theoretical Computer Science* vol.**317** (2004).

CH99. T. Chadzelek, G. Hotz: "Analytic Machines", pp.151–167 in *Theoretical Computer Science* vol.**219** (1999).

Grz57. A. Grzegorczyk: "On the Definitions of Computable Real Continuous Functions", pp.61–77 in *Fundamenta Mathematicae* **44** (1957).

Hau76. J. Hauck: "Berechenbare reelle Funktionenfolgen", pp.265–282 in *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* vol.**22** (1976).

Ho99. C.-K. Ho: "Relatively Recursive Real Numbers and Real Functions", pp.99–120 in *Theoretical Computer Science* vol.**210** (1999).

Ko91. Ker-I Ko: "*Complexity Theory of Real Functions*", Birkhäuser (1991).

Mor69. W. Morris (Editor): "*American Heritage Dictionary of the English Language*", American Heritage Publishing (1969).

Pap94. C.H. Papadimitriou: "*Computational Complexity*", Addison-Wesley (1994).

PEC75. M.B. Pour-El, J. Caldwell: "On a simple definition of computable functions of a real variable", pp.1–19 in *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* vol.**21** (1975).

PER89. M.B. Pour-El, J.I. Richards: "*Computability in Analysis and Physics*", Springer (1989).

PS85. F.P. Preparata, M.I. Shamos: "*Computational Geometry: An Introduction*", Springer (1985).

Ran68. J.F. Randolph: "*Basic Real and Abstract Analysis*", Academic Press (1968).

Soa87. R.I. Soare: "*Recursively Enumerable Sets and Degrees*", Springer (1987).

STvE89. 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).

Tho90. Thomas, W.: "Automata on Infinite Objects", pp.133–191 in *Handbook of Theoretical Computer Science*, vol.**B** (Formal Models and Semantics), Elsevier (1990).

Tur36. Turing, A.M.: "On Computable Numbers, with an Application to the Entscheidungsproblem", pp.230–265 in *Proc. London Math. Soc.* vol.**42(2)** (1936).

Tur37. Turing, A.M.: "On Computable Numbers, with an Application to the Entscheidungsproblem. A correction", pp.544–546 in *Proc. London Math. Soc.* vol.**43(2)** (1937).

Wei01. K. Weihrauch: "*Computable Analysis*", Springer (2001).

WZ00. 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).

Zie05. 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**.

ZW01. X. Zheng, K. Weihrauch: "The Arithmetical Hierarchy of Real Numbers", pp.51–65 in *Mathematical Logic Quarterly* vol.**47** (2001).

ZW03. N. Zhong, K. Weihrauch: "Computability Theory of Generalized Functions", pp.469–505 in *J. ACM* vol.**50** (2003).