Journal of Logic & Analysis 17:FDS2 (2025) 1–22
ISSN 1759-9008

A Degree Structure on Representations of Irrational Numbers

AMIR M. BEN-AMRAM
LARS KRISTIANSEN

Abstract: We study a degree structure on representations of irrational numbers. (Typical examples of representations are Cauchy sequences, Dedekind cuts and base–10 expansions.) We prove that the structure is a distributive lattice with a least and a greatest element. The maximum degree is the degree of the representation by continued fractions. The minimum degree is the degree of the representation by Weihrauch intersections.

2020 Mathematics Subject Classification 03D78 (primary); 03D30 (secondary)

Keywords: Representation of real numbers, resource-bounded computation

1 Introduction

The goal of this article is to prove some properties of a degree structure introduced in Ben-Amram et al [1]. We will to a certain extent provide motivations, examples and intuitive explanations, but for additional background, the readers might have to turn to the first section of [1] and maybe also introductory sections of earlier papers, eg, Kristiansen [8, 9] and Georgiev et al [3].

Different ways of representing real numbers are discussed in very early work on computable analysis. Both Mazur [12] and Specker [17] consider representations by Cauchy sequences, numerical expansions (eg, in base 2 or 10) and Dedekind cuts.[1] They conclude that these three representations yield the same class of computable real numbers, but they do also realise that the representations do not yield the same class of *primitive recursive* real numbers. Specker proves the strict inclusions[2]

(1) $$\mathcal{P}_D \subset \mathcal{P}_{10} \subset \mathcal{P}_C$$

[1]Specker's paper [17] is published in 1949. Mazur's book [12] was not published until 1963, but the book gives a systematic exposition of results obtained by Banach and Mazur in the period 1936–39 and, moreover, results obtain by Mazur in the first few years that follow the Second World War (before 1950). See the foreword of the book for more details.

[2]In Mazur [12], (1) is proved in detailed for non-strict inclusions. When the proof is completed, it is commented that Specker [17] has proven that the three classes are different.

Published: February 2025 https://doi.org/10.4115/jla.2025.17.FDS2

where $\mathcal{P}_D$, $\mathcal{P}_{10}$ and $\mathcal{P}_C$, respectively, denotes the class of real numbers that have a primitive recursive Dedekind cut, a primitive recursive base–10 expansion and a primitive recursive Cauchy sequence. Other early work on computable analysis, by Mostowski [13, 14], Lehman [11] and others, complements the insights won by Mazur and Specker. Eg, the representation by continued fractions yields the same class of computable real numbers as the representations above, but the class of real numbers that have a primitive recursive continued fraction is strictly included in $\mathcal{P}_D$; see [11]. For more on primitive recursive representation of real numbers, see Skordev [16] and Chen et al [2].

The early founders of computable analysis seemed to have realised that the class of computable real numbers is a natural and robust class: any reasonable representation that works in a computable setting yields the same class of computable real numbers. But they did also realise that it might not always be all that easy to convert one representation into another: sometimes it cannot be done primitive recursively (otherwise every representation would yield the same class of primitive recursive reals). The degree structure we define in the next section is motivated and inspired by these profound insights.

We will define an ordering relation $\preceq_S$ over the representations. This ordering relation will induce a degree structure on the representations. We prove that this structure is a distributive lattice. Moreover, we prove that the structure has a least and a greatest element. The maximum degree is the degree of the representation by continued fractions. The minimum degree is the degree of the representation by Weihrauch intersections.

2 The Degree Structure

We identify an irrational number α with its Dedekind cut. The Dedekind cut of an irrational α is the function $\alpha\colon \mathbb{Q} \longrightarrow \{0,1\}$ where

$$\alpha(q) = \begin{cases} 0 & \text{if } q < \alpha \\ 1 & \text{if } q > \alpha. \end{cases}$$

Our subject in this paper is *representations of irrational numbers* and we take a computability-theoretic viewpoint. A Dedekind cut has been defined as a function with this in mind. We shall discuss other representations, which are also functions with countable domain and codomain. We require of a representation to be not just a mapping of functions to real numbers (there are too many such mappings to be of interest), but one that is computationally equivalent, in a sense defined below, to the Dedekind cut. Next, we give the necessary definitions, followed by a few examples.

We will work with oracle Turing machines, and Φ_M^f denotes the function computed by the Turing machine M using the function f as an oracle; in particular, when α is an irrational number, then Φ_M^α denotes the function computed by M using the Dedekind cut of α as an oracle.

Definition 2.1 A class of functions R is a *representation (of the irrational numbers)* if:

(1) There exists a Turing machine M with the following property: For every $f \in R$ there exists irrational α in the interval $(0,1)$ such that $\alpha = \Phi_M^f$. When $\alpha = \Phi_M^f$, we say that f *represents* α and that f is an *R–representation* of α.

(2) There exists a Turing machine N with the following property: For every irrational α in the interval $(0,1)$ there exists an R–representation f of α such that $f = \Phi_N^\alpha$.

We say that an oracle Turing machine M *converts an R_1–representation into an R_2–representation* if for any $f \in R_1$ representing α there exists $g \in R_2$ representing α such that $g = \Phi_M^f$.

We will use R, Q, P (possibly decorated) to denote representations.

Note that a representation will not contain representations of irrationals outside the interval $(0,1)$. That is convenient when working with certain representations. Observe that it follows from our definitions that any representation can be converted (by an algorithm) to and from the representation by Dedekind cuts. This is the space which we intend to explore by dividing it into "degrees".

Example We define a Cauchy sequence for α as a function $C\colon \mathbb{N}^+ \longrightarrow \mathbb{Q}$ with the property $|C(n) - \alpha| < n^{-1}$. Let $\mathcal{C}$ be the class of all Cauchy sequences for all irrational numbers in the interval $(0,1)$. Then $\mathcal{C}$ is a representation according to the definition above.

First we observe that we can compute the Dedekind cut of an irrational α in any Cauchy sequence for α. In order to compute $\alpha(q)$, we search for the least n such that $|C(n) - q| > n^{-1}$. This search terminates as q is rational and α is irrational (the algorithm might not terminate if α is rational). If $q < C(n)$, it will be the case that $\alpha(q) = 0$ (we have $q < \alpha$), otherwise, we have $q > C(n)$, and then it will be case that $\alpha(q) = 1$ (we have $q > \alpha$). Thus there will be an oracle Turing machine M such that $\alpha = \Phi_M^f$ whenever f is a Cauchy sequence for α. Now, M has the following property: For every $f \in \mathcal{C}$ there exists irrational α such that $\alpha = \Phi_M^f$. This shows that clause (1) of Definition 2.1 is satisfied.

In order to verify that clause (2) of the definition is satisfied, we observe that we can compute a Cauchy sequence C for α if we have access to the Dedekind cut of an irrational α in the interval $(0,1)$. We can, eg, use the equations

$$C(1) = \frac{1}{2} \quad \text{and} \quad C(i+1) = \begin{cases} C(i) - 2^{-i-1} & \text{if } C(i) > \alpha \\ C(i) + 2^{-i-1} & \text{if } C(i) < \alpha \end{cases}$$

to compute $C(n)$ for arbitrary n. Hence, there exists a Turing machine N with the following property: For every irrational $\alpha \in (0,1)$ there exists a $\mathcal{C}$–representation f of α such that $f = \Phi_N^\alpha$. This shows that also clause (2) is satisfied, and we conclude that $\mathcal{C}$ is a representation according to Definition 2.1.

Example Let α be an irrational number in the interval $(0,1)$, and let $E_2^\alpha\colon \mathbb{N}^+ \longrightarrow \{0,1\}$ be the function that yields the ith digit of the base–2 expansion of α; more precisely, let E_2^α be such that $\alpha = \sum_{i=1}^\infty E_2^\alpha(i)2^{-i}$. The representation by base–2 expansions is the set $\mathcal{E}_2$, where:

$$\mathcal{E}_2 \;=\; \{\, E_2^\alpha \mid \alpha \text{ is an irrational in the interval } (0,1) \,\}$$

The representation by base-b expansions $\mathcal{E}_b$ is defined similarly for any $b \geq 2$. We leave to the reader to verify that $\mathcal{E}_b$ indeed is a representation according to Definition 2.1.

Definition 2.2 A function $t\colon \mathbb{N} \longrightarrow \mathbb{N}$ is a *time bound* if (i) $n \leq t(n)$, (ii) t is increasing and (iii) t is time-constructible: there is a multi-tape Turing machine that, on input 1^n, computes $t(n)$ in $\Theta(t(n))$ steps.

Definition 2.3 Let t be a time-bound and let R be a representation. Then, $O(t)_R$ denotes the class of all irrational α in the interval $(0,1)$ such that at least one R–representation of α is computable by a Turing machine running in time $O(t(n))$ (where n is the length of the input).

Let R_1 and R_2 be representations. The relation $R_1 \preceq_S R_2$ holds if for any time-bound t there exists a time-bound s such that

$$O(t)_{R_2} \;\subseteq\; O(s)_{R_1}.$$

If the relation $R_1 \preceq_S R_2$ holds, we will say that the representation R_1 is *subrecursive* in the representation R_2.

Intuitively, if we can convert an R_2–representation f_2 of α into an R_1–representation f_1 of α, while satisfying a time-bound, then the relation $R_1 \preceq_S R_2$ will hold. If such a conversion exists, then there exists a time-bounded oracle Turing machine M such that $f_1 = \Phi_M^{f_2}$, and thus, if f_2 is computable in time $O(t)$, then f_1 is computable in time $O(s)$ for some time-bound s (and the inclusion $O(t)_{R_2} \subseteq O(s)_{R_1}$ holds). Note that the complexity bound s is not specified but only required to exist. Informally, this means that the conversion does not make use of *unbounded* search. The case of converting the representation $\mathcal{E}_2$, that is, the representation by base–2 expansions, into the representation by Dedekind cut may serve to illustrate this notion. Let $\mathcal{D}$ denote the representation by Dedekind cuts. Consider an irrational whose base–2 expansion starts by $0.0101010101\ldots$. Clearly the number is close to $^1/_3$. But in order to determine on which side of $^1/_3$ it falls we have to search for the first pair of bits which is not 01. Thus unbounded search is unavoidable, and we have $\mathcal{D} \not\preceq_S \mathcal{E}_2$, On the other hand, if we have access to the Dedekind cut of α, unbounded search is not needed to generate the ith bit of the base–2 expansion of α (to compute the value of $E^\alpha(i)$). Thus, we have $\mathcal{E}_2 \preceq_S \mathcal{D}$. Indeed, we have $\mathcal{E}_b \preceq_S \mathcal{D}$ for any $b \geq 2$. It is easy to see that we can use the Dedekind cut of α to generate the digits $\mathtt{D}_1, \mathtt{D}_2, \mathtt{D}_3, \ldots$ of the base-b expansion of α one by one. First we use the Dedekind cut to determine $\mathtt{D}_1$; then we use the Dedekind cut to determine $\mathtt{D}_2$; and thus we proceed up to the desired position.

To show that the relation $R_1 \preceq_S R_2$ holds, the natural way is to exhibit a *reduction* in the form of an time-bounded oracle Turing machine which computes the R_1 representation given the R_2 representation.

Definition 2.4 Let R and Q be representations. The relation $R \equiv_S Q$ holds when $R \preceq_S Q$ and $Q \preceq_S R$. If the relation $R \equiv_S Q$ holds, we will say that the representation R is *subrecursively equivalent* to the representation Q.

The relation $R \prec_S Q$ holds when $R \preceq_S Q$ and $Q \not\preceq_S R$.

It is obvious that $\equiv_S$ is an equivalence relation, and thus the next definition makes sense.

Definition 2.5 Let R be a representation. We define the *S–degree* of R, denoted $\deg_S(R)$, as the equivalence class given by:

$$\deg_S(R) \ = \ \{ \ Q \mid Q \equiv_S R \ \}$$

The set of all S–degrees, denoted $\mathcal{S}$, is given by:

$$\mathcal{S} \ = \ \{ \ \deg_S(R) \mid R \text{ is a representation } \}$$

We will use $\mathbf{a}, \mathbf{b}, \mathbf{c}$ (possible decorated) to denote S–degrees. We will use $\leq$ and $<$ to denote the ordering relations induced on the S–degrees by $\preceq_S$ and $\prec_S$, respectively.

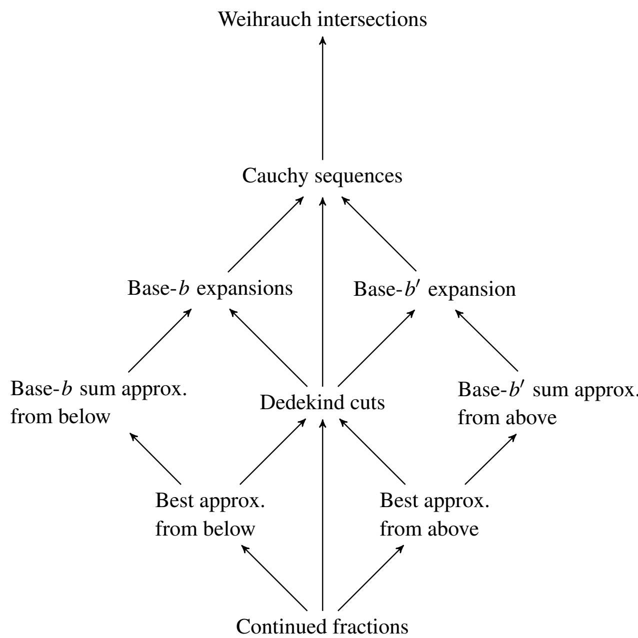

Figure 1: Overview.

The directed graph in Figure 1 gives an overview of the relationship between some natural degrees. The nodes depict degrees of representations, and each degree is labeled with one of the most well known representations in the degree. For two representations R_1 and R_2, there is a directed path from a node labeled R_1 to a node labeled R_2 if and only if $R_2 \preceq_S R_1$. (The paths in the graph also tell us when it is possible, and when it is not possible, to convert one representation into another without resorting to unbounded search. If there is a directed path from R_1 to R_2, unbounded search is not needed in order to convert an R_1–representation into an R_2–representation, and if there is no directed path from R_1 to R_2, unbounded search is needed.) This implies that Figure 1 shows an upside-down picture of the degree structure, that is, if a degree **a** lies below a degree **b**, then **a** is depicted above **b** in the figure. The least degree shown in the figure is the degree of the Weihrauch intersections. The greatest degree shown in the figure is the degree of the continued fractions. We will prove that there are no degrees below the degree of the Weihrauch intersections and above the degree of the continued fractions.

Some explanation may be called for regarding the degrees of base–b expansions. The figure shows two such degrees, with bases b and b'. In fact, the relation between two such degrees depends on the relation of b to b'. This is specified by the next theorem.

Theorem 2.6 (Kristiansen [9]) *$\mathcal{E}_b \preceq_S \mathcal{E}_{b'}$ if and only if every prime that divides b also divides b'.*

The same rule applies to the degrees of base-b sum approximations (also studied in [9]).

For more on the degrees and the representations appearing in Figure 1, see Ben-Amram et al [1], Kristiansen [8, 9], Georgiev et al [3], Kristiansen [10], Georgiev [4, 5] and Hiroshima & Kawamura [6].

3 The Structure is a Lattice

Definition 3.1 For strings x, y, we denote by $\langle x, y\rangle$ an encoding of the pair (x, y).

The precise encoding does not matter, but we require one where pairing and unpairing can be done efficiently (within quadratic time) on a Turing machine. For example, we could use $x\#y$ where # is a special symbol.

Definition 3.2 Let f and g be functions with the signatures $f\colon A_1 \longrightarrow B_1$ and $g\colon A_2 \longrightarrow B_2$. We define the function $f \times g\colon A_1 \times A_2 \longrightarrow B_1 \times B_2$ by:

$$f \times g(\langle x, y\rangle) \;=\; \langle f(x), g(y)\rangle$$

Let R and Q be representations. We define $\text{join}[R, Q]$ by:

$$\text{join}[R, Q] = \{\, f \times g \mid f \text{ is an } R\text{–representation of } \alpha \text{ and } g \text{ is a } Q\text{–representation of } \alpha \,\}$$

Lemma 3.3 *Let R_0 and R_1 be representations. Then $\text{join}[R_0, R_1]$ is a representation.*

Proof Since R_i (for $i = 0, 1$) is a representation, we have Turing machines M_i and N_i such that:

- For every $f \in R_i$ there exists irrational $\alpha \in (0, 1)$ such that $\alpha = \Phi^f_{M_i}$.
- For every irrational $\alpha \in (0, 1)$ there exists R_i–representation f of α such that $f = \Phi^\alpha_{N_i}$.

Let ε be the first oracle query performed by M_1 on input $1/2$ (this is just an arbitrary choice). Let M be the oracle Turing machine that simulates M_0, while replacing any oracle query with input w by code that

- writes $\langle w, \varepsilon\rangle$ on the query tape;

- queries the oracle, obtaining a result in the form $\langle x, y\rangle$; and
- extracts x and uses it as the result of the query.

It should be obvious that the following claim holds.

> **(Claim 1)** For every $f \in \mathrm{join}[R_0, R_1]$, there exists irrational $\alpha \in (0,1)$ such that $\alpha = \Phi_M^f$.

We have constructed M from M_0. It is easy to see that we might as well have constructed M from M_1.

Let N be the oracle Turing machine given by:

$N^f =$ on input $\langle x, y\rangle$ do:
  Run N_0^f on input x and store the output z_0.
  Run N_1^f on input y and store the output z_1.
  Give output $\langle z_0, z_1\rangle$.

It should be obvious that the following claim holds.

> **(Claim 2)** For every irrational $\alpha \in (0,1)$ there exists a $\mathrm{join}[R_0, R_1]$–representation of α such that $f = \Phi_N^\alpha$.

It follows straightforwardly from Definition 2.1 and the two claims that $\mathrm{join}[R_0, R_1]$ is a representation. □

Lemma 3.4 *We have*

$$R \preceq_S R' \quad \text{and} \quad Q \preceq_S Q' \quad \Rightarrow \quad \mathrm{join}[R, Q] \preceq_S \mathrm{join}[R', Q']$$

for any representations R, R', Q, Q'.

Proof Assume $R \preceq_S R'$ and $Q \preceq_S Q'$. Then, by the definition of $\preceq_S$, for any time bound t there exist time bounds s_1, s_2 such that:

(2) $$O(t)_{R'} \subseteq O(s_1)_R \quad \text{and} \quad O(t)_{Q'} \subseteq O(s_2)_Q$$

We prove $\mathrm{join}[R, Q] \preceq_S \mathrm{join}[R', Q']$. By the definition of $\preceq_S$, we have to prove that for any time bound t there exists time bound s such that:

(3) $$O(t)_{\mathrm{join}[R',Q']} \subseteq O(s)_{\mathrm{join}[R,Q]}$$

Fix t and assume $\alpha \in O(t)_{\mathrm{join}[R',Q']}$ (we will find a time bound s such that $\alpha \in O(s)_{\mathrm{join}[R,Q]}$). By this assumption, we have an $O(t)$–time Turing machine M such that

$\Phi_M = f \times g$ where $f \times g$ is some join$[R', Q']$–representation of α. From M we can easily construct $O(t)$–time Turing machines M_1, M_2 such that Φ_{M_1} is an R'–representation of α and Φ_{M_2} is a Q'–representation of α. Thus, we have $\alpha \in O(t)_{R'}$ and $\alpha \in O(t)_{Q'}$, and by (2), we have time bounds s_1, s_2 such that $\alpha \in O(s_1)_R$ and $\alpha \in O(s_2)_Q$. Thus, there exists an $O(s_1)$–time Turing machine N_1 such that Φ_{N_1} is an R–representation of α, and there exists an $O(s_2)$–time Turing machine N_2 such that Φ_{N_2} is an Q–representation of α. Let $s(n) = \max(s_1(n) + s_2(n), n^2)$. From N_1 and N_2 we can construct an $O(s)$–time Turing machine N such that Φ_N is a join$[R, Q]$–representation of α (see Figure 2), and thus, $\alpha \in O(s)_{\mathrm{join}[R,Q]}$. This proves that (3) holds. □

N = "On input $\langle x, y\rangle$ do:
  Run N_1 on input x, store the output z_1.
  Run N_2 on input y, store the output z_2.
  Give output $\langle z_1, z_2\rangle$."

Figure 2: A Sipser-style construction of N from N_1 and N_2.

Lemma 3.5 *We have*

$$R \equiv_S R' \quad \text{and} \quad Q \equiv_S Q' \quad \Rightarrow \quad \mathrm{join}[R, Q] \ \equiv_S \ \mathrm{join}[R', Q']$$

for any representations R, R, Q, Q'.

Proof This follows straightforwardly from Lemma 3.4 and the definition of $\equiv_S$. □

Lemma 3.5 shows that the next definition makes sense.

Definition 3.6 We define the *join* of the S–degrees $\mathbf{a}$ and $\mathbf{b}$, written $\mathbf{a} \cup \mathbf{b}$, by

$$\mathbf{a} \cup \mathbf{b} = \deg_S(\mathrm{join}[R, Q])$$

where R and Q are any representations such that $\mathbf{a} = \deg_S(R)$ and $\mathbf{b} = \deg_S(Q)$.

We are now ready to prove that the set of S–degrees is an upper semi-lattice, that is, every pair of degrees has a least upper bound (lub).

Theorem 3.7 *Let* $\mathbf{a}$, $\mathbf{b}$ *be* S*–degrees. The degree* $\mathbf{a} \cup \mathbf{b}$ *is the lub of* $\mathbf{a}$ *and* $\mathbf{b}$.

Proof It is obvious that $\mathbf{a} \leq \mathbf{a} \cup \mathbf{b}$ and $\mathbf{b} \leq \mathbf{a} \cup \mathbf{b}$. Let $\mathbf{c}$ be any degree such that $\mathbf{a} \leq \mathbf{c}$ and $\mathbf{b} \leq \mathbf{c}$. We prove that $\mathbf{a} \cup \mathbf{b} \leq \mathbf{c}$ (and thus $\mathbf{a} \cup \mathbf{b}$ will be the least degree that lies above both $\mathbf{a}$ and $\mathbf{b}$).

Let $\mathbf{c} = \deg_S(R)$. Furthermore, let $\mathbf{a} = \deg_S(Q)$ and $\mathbf{b} = \deg_S(P)$. As $\mathbf{a} \leq \mathbf{c}$ and $\mathbf{b} \leq \mathbf{c}$, we have $Q \preceq_S R$ and $P \preceq_S R$, and then, Lemma 3.5 yields $\mathrm{join}[Q,P] \preceq_S \mathrm{join}[R,R]$. It is easy to see that $\mathrm{join}[R,R] \equiv_S R$. Thus, we have $\mathrm{join}[Q,P] \preceq_S R$. By Definition 2.5 and Definition 3.6, we have $\mathbf{a} \cup \mathbf{b} \leq \mathbf{c}$. □

In order to prove that every pair of degrees also has a greatest lower bound (glb), we will define a meet operator.

Definition 3.8 Let $\bot \notin A$ (just pick a value that is not in A). Fix an arbitrary value y in the set B (this y will act as a dummy, and it does not matter which y we pick). For any function $f\colon\ A \longrightarrow B$, let

$$\mathbf{inl}_0(f), \mathbf{inl}_1(f)\colon\ A \cup \{\bot\} \longrightarrow \{0,1\} \times B$$

be the functions given by

$$\mathbf{inl}_0(f)(x) = \begin{cases} \langle 0, f(x) \rangle & \text{if } x \in A \\ \langle 0, y \rangle & \text{if } x = \bot \end{cases}$$

and

$$\mathbf{inl}_1(f)(x) = \begin{cases} \langle 1, f(x) \rangle & \text{if } x \in A \\ \langle 1, y \rangle & \text{if } x = \bot\,. \end{cases}$$

For any representations R and Q, we define $\mathrm{meet}[R,Q]$ by:

$$\mathrm{meet}[R,Q] \quad = \quad \{\, \mathbf{inl}_0(f) \mid f \text{ is an } R\text{–representation} \,\} \;\cup \; \{\, \mathbf{inl}_1(f) \mid f \text{ is a } Q\text{–representation} \,\}$$

Lemma 3.9 *Let R_0 and R_1 be representations. Then $\mathrm{meet}[R_0,R_1]$ is a representation.*

Proof Since R_i (for $i = 0,1$) is a representation, we have Turing machines M_i and N_i such that:

- For every $f \in R_i$ there exists an irrational $\alpha \in (0,1)$ such that $\alpha = \Phi^f_{M_i}$.
- For every irrational $\alpha \in (0,1)$ there exists an R_i–representation f of α such that $f = \Phi^{\alpha}_{N_i}$.

For any Turing machine M with oracle $f\colon\ A \longrightarrow B$, let $\widehat{M}$ denote a Turing machine with oracle $f\colon\ A \cup \{\perp\} \longrightarrow \{0,1\} \times B$ such that:

$$\Phi_M^f \quad = \quad \Phi_{\widehat{M}}^{\mathbf{inl}_0 f} \quad = \quad \Phi_{\widehat{M}}^{\mathbf{inl}_1 f}$$

The oracle Turing machine $\widehat{M}$ works like M, but $\widehat{M}$'s oracle will give answers of the form $\langle i, y\rangle \in \{0,1\} \times B$ and $\widehat{M}$ simply ignores the left component i. Let M be the oracle Turing machine given by:

$M^f =$ on input w do:

Check if $f(\perp) = \langle 0, y\rangle$ for some y (otherwise, $f(\perp) = \langle 1, y\rangle$ for some y).

If YES, run $\widehat{M_0}^f$ on input w (and give the same output as $\widehat{M_0}^f$).

If NO, run $\widehat{M_1}^f$ on input w (and give the same output as $\widehat{M_1}^f$).

(Claim 1) For every $f \in \text{meet}[R_0, R_1]$, there exists irrational $\alpha \in (0,1)$ such that $\alpha = \Phi_M^f$.

In order to see that the claim holds, pick an arbitrary f in the set $\text{meet}[R_0, R_1]$. Then, we either have $f = \mathbf{inl}_0(f_0)$ for some $f_0 \in R_0$, or $f = \mathbf{inl}_1(f_1)$ for some $f_1 \in R_1$. Let us say that $f = \mathbf{inl}_1(f_1)$ where $f_1 \in R_1$ (do a symmetric argument if $f = \mathbf{inl}_0(f_0)$ where $f_0 \in R_0$). By the construction of M, we have $\Phi_M^f = \Phi_{\widehat{M_1}}^{f_1} = \alpha$ where $\alpha \in (0,1)$ is the irrational number represented by f_1. Hence, the claim holds.

Let N be the oracle Turing machine given by:

$N^f =$ on input w do:

If $w = \perp$, give output $\langle 0, y\rangle$ (where y is some fixed value).

If $w \neq \perp$, run N_0^f on input w and store the output z.

Give output $\langle 0, z\rangle$.

We have constructed N from N_0. We may also construct N from N_1. Let N be the oracle Turing machine given by

$N^f =$ on input w do:

If $w = \perp$, give output $\langle 1, y\rangle$ (where y is some fixed value).

If $w \neq \perp$, run N_1^f on input w and store the output z.

Give output $\langle 1, z\rangle$.

and our proofs will still go through.

> **(Claim 2)** For every irrational $\alpha \in (0,1)$ there exists a $\mathrm{meet}[R_0,R_1]$–representation f of α such that $f = \Phi_N^\alpha$.

In order to verify the claim, pick an arbitrary irrational α in the interval $(0,1)$. Then there exists an R_0–representation f_0 of α. Let $f = \mathbf{inl}_0(f_0)$. Then $f \in \mathrm{meet}[R_0,R_1]$ and, moreover, f is a $\mathrm{meet}[R_0,R_1]$–representation of α since $\alpha = \Phi_M^f$. It is easy to see that we have $f = \Phi_N^\alpha$. Thus, we conclude that the claim holds.

It follows straightforwardly from Definition 2.1 and the two claims that $\mathrm{meet}[R_0,R_1]$ is a representation. □

Lemma 3.10 *Let $f\colon A \longrightarrow B$ be any function, and let t be a time-bound such that $t(n) \geq n^2$. The following three assertions are equivalent: (1) There exists an $O(t)$–time Turing machine M such that $\Phi_M = f$. (2) There exists an $O(t)$–time Turing machine M_0 such that $\Phi_{M_0} = \mathbf{inl}_0(f)$. (3) There exists an $O(t)$–time Turing machine M_1 such that $\Phi_{M_1} = \mathbf{inl}_1(f)$.*

Proof Each of these machines can be converted to each of the others with little effort. The assumption $t(n) \geq n^2$ ensures that the process of stripping the first component from $\langle 0,x\rangle$ or $\langle 1,x\rangle$, or adding such a component, remains within $O(t)$ time. □

Lemma 3.11 *We have*

$$R \equiv_S R' \quad \text{and} \quad Q \equiv_S Q' \quad \Rightarrow \quad \mathrm{meet}[R,Q] \equiv_S \mathrm{meet}[R',Q']$$

for any representations R, R', Q, Q'.

Proof It is sufficient to prove that:

$$R \preceq_S R' \text{ and } Q \preceq_S Q' \quad \Rightarrow \quad \mathrm{meet}[R,Q] \preceq_S \mathrm{meet}[R',Q'] \tag{4}$$

The proof of (4) is rather straightforward, and we leave the details to the reader. □

Lemma 3.11 shows that the next definition makes sense.

Definition 3.12 We define the *meet* of the S–degrees $\mathbf{a}$ and $\mathbf{b}$, written $\mathbf{a} \cap \mathbf{b}$, by

$$\mathbf{a} \cap \mathbf{b} = \deg_S(\mathrm{meet}[R,Q])$$

where R and Q are any representations such that $\mathbf{a} = \deg_S(R)$ and $\mathbf{b} = \deg_S(Q)$.

By the next theorem, every pair of degrees has a greatest lower bound, and thus our degree structure is a lattice. Moreover, it is a distributive lattice as we have:

$$\mathbf{a} \cup (\mathbf{b} \cap \mathbf{c}) \;=\; (\mathbf{a} \cup \mathbf{b}) \cap (\mathbf{a} \cup \mathbf{c}) \tag{5}$$

It is straightforward, but rather tedious, to prove that (5) holds, and we leave the details to the reader. (When the join operator distributes over the meet operator in a lattice, then the meet operator will also distribute over the join operator.)

Theorem 3.13 *Let* $\mathbf{a}$, $\mathbf{b}$ *be* S*–degrees. The degree* $\mathbf{a} \cap \mathbf{b}$ *is the glb of* $\mathbf{a}$ *and* $\mathbf{b}$.

Proof We have $\mathbf{a} \cap \mathbf{b} \leq \mathbf{a}$ and $\mathbf{a} \cap \mathbf{b} \leq \mathbf{b}$ by Lemma 3.10. Let $\mathbf{c}$ be any S–degree that lies below both $\mathbf{a}$ and $\mathbf{b}$, that is, $\mathbf{c} \leq \mathbf{a}$ and $\mathbf{c} \leq \mathbf{b}$. We have to prove $\mathbf{c} \leq \mathbf{a} \cap \mathbf{b}$ (thus, $\mathbf{a} \cap \mathbf{b}$ will be the greatest degree that lies below both $\mathbf{a}$ and $\mathbf{b}$).

Let $\mathbf{c} = \deg_S(Q)$, let $\mathbf{a} = \deg_S(R_0)$ and let $\mathbf{b} = \deg_S(R_1)$. Fix an arbitrary time-bound t. Since $\mathbf{c} \leq \mathbf{a}$, there exists time-bound s_0 such that

$$O(t)_{R_0} \subseteq O(s_0)_Q \tag{6}$$

and, since $\mathbf{c} \leq \mathbf{b}$, there exists time-bound s_1 such that:

$$O(t)_{R_1} \subseteq O(s_1)_Q \tag{7}$$

Let $s(n) = \max(s_0(x), s_1(x))$. Then s is a time-bound. We will prove that:

$$O(t)_{\mathrm{meet}[R_1,R_2]} \subseteq O(s)_Q \tag{8}$$

It follows straightforwardly from (8) and our definitions that $\mathbf{c} \leq \mathbf{a} \cap \mathbf{b}$.

In order to prove (8), assume $\alpha \in O(t)_{\mathrm{meet}[R_0,R_1]}$. Then there exists an $O(t)$–time Turing machine M such that Φ_M is a meet$[R_1, R_2]$–representation of α. Either we have (i) $\Phi_M = \mathbf{inl}_0(f_0)$ where f_0 is an R_0–representation of α, or we have (ii) $\Phi_M = \mathbf{inl}_1(f_1)$ where f_1 is an R_1–representation of α. In case (i), we apply Lemma 3.10 and get an $O(t)$–time Turing machine N such that $\Phi_N = f_0$. Thus, we can conclude that $\alpha \in O(t)_{R_0}$, and then by (6), we have $\alpha \in O(s_0)_Q$. In case (ii), we apply Lemma 3.10 and get an $O(t)$–time Turing machine N' such that $\Phi_{N'} = f_1$. Now we can conclude that $\alpha \in O(t)_{R_1}$, and by (7), we have $\alpha \in O(s_1)_Q$.

This proves that we for any $\alpha \in O(t)_{\mathrm{meet}[R_0,R_1]}$, have $\alpha \in O(s_0)_Q$ or $\alpha \in O(s_1)_Q$. Hence, (8) holds when s is given by $s(n) = \max(s_0(n), s_1(n))$. □

4 Minimum and Maximum Degrees

It turn outs that our lattice has a top element and a bottom element.

Definition 4.1 A function $I\colon\ \mathbb{N} \longrightarrow \mathbb{Q} \times \mathbb{Q}$ is a *Weihrauch intersection* for the real number α if the left component of the pair $I(i)$ is strictly less than the right component of the pair $I(i)$ (for all $i \in \mathbb{N}$) and

$$\{\ \alpha\ \} \ = \ \bigcap_{i=0}^{\infty} I_i^O$$

where I_i^O denotes the open interval given by the pair $I(i)$.

We define the *representation by Weihrauch intersections*, denoted $\mathcal{W}$, by:

$$\begin{aligned} \mathcal{W} \ \ = \ \ \{\ I \mid \alpha \text{ is an irrational in the interval } (0,1) \\ \text{and } I \text{ is a Weihrauch intersection for } \alpha\} \end{aligned}$$

Let us verify that $\mathcal{W}$ indeed is a representation according to Definition 2.1. If we have access to the Dedekind cut of α, then we can obviously compute a Weihrauch intersection for α (unbouded search will not be needed). If we have access to a Weihrauch intersection I for an irrational α, then we can compute the Dedekind cut of α if we use unbounded search. In order to decide if a rational number q lies above or below α, we search for the least i such that q lies outside the interval $I(i)$. The search will terminate as q is rational and α is irrational. If q is less than or equal to the left component of $I(i)$, we know that q lies below α; otherwise, q lies above α. This shows that $\mathcal{W}$ is a representation.

The representation of reals by Weihrauch intersections is more or less the representation by nested intervals which is known from Weihrauch's seminal book on computable analysis [18]. For the sake of simplicity, we do not want the intervals to be nested, but any Weihrauch intersection can be easily converted to a nested one. For some related representations, see Skordev [16].

Theorem 4.2 (Minimum Degree) *Let* $\mathbf{0} = \deg_S(\mathcal{W})$. *For any S–degree $\mathbf{a}$, we have* $\mathbf{0} \leq \mathbf{a}$.

Proof Let R be a representation of degree $\mathbf{a}$. Let f denote any R–representation of α. There is a Turing machine M_0 such that

$$W_0 = \Phi_{M_0}^f$$

where $W_0 \in \mathcal{W}$ represents α. Our definitions ensure that such an M_0 exists: M_0 computes the Dedekind cut of α from f, and M_0 uses the Dedekind cut to compute W_0 (note that M_0 might carry out unbounded search). Now let

$$W(x) = \begin{cases} \Phi^f_{M_0}(y) & \text{where } y \text{ is the greatest } y \text{ such that} \\ & y < x \text{ and } M_0^f \text{ on input } y \text{ halts within } x \text{ steps;} \\ (0,1) & \text{if no such } y \text{ exists.} \end{cases}$$

Now, W is a Weihrauch intersection for α. Moreover, W can be computed subrecursively in f. Specifically, if f can be computed by a Turing machine running in time $O(t)$, then W can be computed by a Turing machine running in time $O(s)$ (for some s depending on t). Thus, for any time bound t there exists time bound s such that $O(t)_R \subseteq O(s)_{\mathcal{W}}$. Thus, by our definitions, we have $\mathcal{W} \preceq_S R$. It follows that $\mathbf{0} \leq \mathbf{a}$. □

Let $a_0, a_1, a_2, \ldots$ be an infinite sequence of integers where $a_1, a_2, a_3 \ldots$ are positive. The *continued fraction* $[a_0; a_1, a_2, \ldots]$ is defined by:

$$[\, a_0; a_1, a_2, a_3, \ldots \,] = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

We assume that the readers are familiar with continued fractions (those who are not may consult Khintchine [7] or Richards [15]). The continued fraction of the real number α is the unique sequence $a_0, a_1, a_2, \ldots$ such that $\alpha = [a_0; a_1, a_2, \ldots]$.

It is well known that we can compute the Dedekind cut of α if we have access to the continued fraction of α, and vice versa, we can compute a continued fraction of α if we have access to the Dedekind cut of α (this will require unbounded search). It is also well known that every irrational number α in the interval $(0, 1)$ can be written uniquely in the form $[0; a_1, a_2, \ldots]$ where $a_1, a_2, a_3, \ldots$ are positive integers. Moreover, if $a_1, a_2, a_3, \ldots$ are positive integers and $\alpha = [0; a_1, a_2, \ldots]$, then α is an irrational number in the interval $(0, 1)$ (all rational numbers have finite continued fractions). Hence, if we map each irrational in the interval α to the unique $f : \mathbb{N}^+ \longrightarrow \mathbb{N}^+$ such that $\alpha = [0; f(1), f(2), \ldots]$, then we have a bijection between the irrational numbers in the interval $(0, 1)$ and the total functions from $\mathbb{N}^+$ into $\mathbb{N}^+$. This implies that the set $\mathcal{C}_{[\,]}$ given by

$$(9) \qquad \mathcal{C}_{[\,]} = \{ f \mid f \text{ is a total function from } \mathbb{N}^+ \text{ into } \mathbb{N}^+ \}$$

is a representation according to Definition 2.1.

Definition 4.3 The *representation by continued fractions* is the set $\mathcal{C}_{[\,]}$ given by (9).

Theorem 4.4 (Maximum Degree) *Let* $\mathbf{1} = \deg_S(\mathcal{C}_{[\,]})$. *For any* S*–degree* $\mathbf{a}$, *we have* $\mathbf{a} \leq \mathbf{1}$.

The proof of Theorem 4.4 requires some preliminary work, and the next section is dedicated to proving this result, including the auxiliary definitions and facts.

5 The Proof of the Maximum Degree Theorem

Let us first give a precise description of the function-oracle Turing machines that we use.

Definition 5.1 A (parameterized) *function-oracle Turing machine* is a (multi-tape) Turing machine $M = (Q, q_0, F, \Sigma, \Gamma, \delta)$ with initial state $q_0 \in Q$, final states $F \subseteq Q$, input and tape alphabets Σ and Γ (with $\Sigma \subseteq \Gamma$ and $\{\llcorner\!\lrcorner\} \subseteq \Gamma \setminus \Sigma$), and transition function δ such that M has a special *query tape* and two distinct states $q_q, q_a \in Q$ (the *query* and *answer* states).

To be executed, M is provided with a total function $f\colon (\Gamma \setminus \{\llcorner\!\lrcorner\})^* \longrightarrow (\Gamma \setminus \{\llcorner\!\lrcorner\})^*$ (the oracle) prior to execution on any input. We write M^f for M when f has been fixed. We use Φ^f_M to denote the function computed by M^f.

The transition relation of M^f is defined as usual for Turing machines, except for the query state q_q: If M enters state q_q with the word x on its query tape, then (i) the contents of the query tape are instantaneously changed to $f(x)$, (ii) the query-tape head is reset to the origin, while other heads do not move, and (iii) M moves to state q_a. The *time- and space complexity* of a function-oracle machine is counted as for usual Turing machines, with the transition between q_q and q_a taking $\|f(x)\|$ time steps, the length of the string $f(x)$.

In general, $\|w\|$ denotes the number of symbols in the string w. Numbers are represented in binary. If $i \in \mathbb{N}$, then $\|i\|$ denotes the length of the binary representation of i.

The *input size* of a query is the number of non-blank symbols on the query tape when M enters state q_q.

5.1 Canonical Standard Versions of Oracle Machines

Let M^f be a function-oracle Turing machine that for any given total oracle $f\colon \mathbb{N}^+ \longrightarrow \mathbb{N}^+$ terminates on every input w. Thus, $\Phi_M^f(w)$ is a total computable function whenever f is a computable oracle. For any total and computable $f\colon \mathbb{N}^+ \longrightarrow \mathbb{N}^+$, we construct a deterministic non-oracle Turing machine $\widehat{M^f}$ which computes Φ_M^f. We will say that $\widehat{M^f}$ is the *canonical standard version* of the oracle Turing machine M^f.

We need some notation. Let $M^f = (Q, q_0, F, \Sigma, \Gamma, \delta)$, and let q_q and q_a, respectively, denote the query and the answer state of M^f. We assume some standard representation of the configurations of M^f, eg

$$\texttt{\#}uqv\texttt{\#}u'\texttt{-}v'\texttt{\#} \tag{10}$$

where $q \in Q$, $u, v, u', v' \in \Gamma^*$ may represent the configuration where M is in state q; the content of the work tape is $uv\llcorner^*$ and the head scans the first symbol of v; the content of the query tape is $u'v'\llcorner^*$ and the head scans the first symbol of v'.

For any configuration C of M^f

- $\mathit{state}(C)$ denotes the state of C

eg, if C is the configuration (10), then $\mathit{state}(C) = q$.

If $\mathit{state}(C) = q_q$, that is, if M^f is in a query state, then:

- $\mathit{query}(C)$ denotes the element of $\mathbb{N}^+$ represented on the query tape (natural number are written in binary notation) in the configuration C.
- C^y denotes the configuration M^f will be in if the oracle returns the natural number y.

Example: Let C be the configuration $\texttt{\#}uq_qv\texttt{\#100-\#}$. Then $\mathit{query}(C) = 4$ as 100 represent number 4; moreover, C^{17} is

$$\texttt{\#}uq_av\texttt{\#-10001\#}$$

as 10001 is 17 written in binary and the head of the query tape scans the first symbol of 10001.

If $\mathit{state}(C) \notin F \cup \{q_q\}$, that is, if M^f is not in a final state or in the query state, then

- $\mathit{next}(C)$ denotes (the unique) configuration that follows C when M^f carries out one transition.

Now we have all the notation we need to describe the canonical standard version $\widehat{M^f}$ of the oracle machine M^f.

$\widehat{M^f}$ = on input w do:
  Construct the start configuration C_1 of M^f on input w.
  Execute the recursive procedure EXE(C : configuration) given by pseudo code in Figure 3 with input C_1, that is, execute EXE(C_1).
  The execution will generate a finite sequence of configurations $C_1, \dots, C_m$ (one configuration C_i each time the procedure makes a recursive call EXE(C_i)); use C_m to compute $\Phi^f_M(w)$.
  Give the output $\Phi^f_M(w)$.

It is obvious that the canonical standard version $\widehat{M^f}$ of the oracle machine M^f computes the function Φ^f_M for each computable f. Next we will construct a time-bound s with the following property: if f is computable in time $O(t)$, then $\widehat{M^f}$ will run in time $O(s)$ on all but finitely many inputs.

Recall that we assumed M^f to terminate on every input, provided f is total. For the next step we need a stronger assumption, namely that M^f terminates even for a "cheating" oracle that can answer differently when posed the same query twice. This assumption implies no loss of generality, since we could instrument procedure EXE to record oracle queries and answers, and pull the answer from the record in case a query is repeated. For simplicity we have left this out of the code in Figure 3.

```
proc EXE(C : configuration)
if state(C) = q_q then
   begin
   y:=f(query(C)); EXE(C^y)
   end
else if state(C) ∉ F then EXE(next(C))
else return C
end proc
```

Figure 3: A recursive procedure given in pseudo code. The parameter is called by value.

5.2 Time-Bounds for Canonical Standard Versions

Let $\widehat{M^f}$ be a canonical standard version where $f: \mathbb{N}^+ \longrightarrow \mathbb{N}^+$ is computable in time $O(t)$. The recursive procedure in Figure 4 is constructed from M and t along the lines the recursive procedure in Figure 3 is constructed from M and f. The reader should note the similarities and the differences between the two procedures.

```
proc TB(C : configuration; step : integer)
if state(C) = q_q then
   begin
   for i:=1 to 2^{t^2(step)} do TB(C^i, step + ||i||)
   end
else if state(C) ∉ F then TB(next(C), step + 1)
end proc
```

Figure 4: A recursive procedure given in pseudo code. The two parameters are called by value.

Next we describe a standard Turing machine $\widetilde{M^t}$ that computes a time-bound.

$\widetilde{M^t}$ = on input 1^n do:

Set a binary counter *count* to n; the machine later increases the counter as further explained below.

Let $w_1, w_2, \ldots, w_k$ be all potential inputs to M^f such that $\|w_i\| \leq n$ (for $i = 1, \ldots k$), moreover, let C_1^i be the start configuration of M^f on input w_i.

For $i = 1, \ldots, k$, execute TB(C_1^i, n).

When all the calls to TB$(\ldots)$ have terminated, let the output be $count^2$ (the square of the final value of the counter).

We should elaborate on how *count* is maintained. We maintain it so that $count^2$, at the end of computation, will be an upper bound on the number of transitions actually performed (including those that maintain *count*). During the computation, *count* is represented in binary on its own tape. Moreover, we increase it at every step (ie, whenever *step* is incremented). The time it takes to increase the counter up to a value of i is $O(i)$.

(Claim) Let $f : \mathbb{N}^+ \longrightarrow \mathbb{N}^+$ be any function computable in time $O(t)$, and let s be the function computed by $\widetilde{M^t}$. Then (i) s is a time-bound, and (ii) there exists a natural number K such that $s(\|w\|)$ bounds the running-time of $\widehat{M^f}$ on input w whenever $\|w\| \geq K$.

We prove the claim. First we argue that function s is a time-bound. To this end we have to verify that:

- $n \leq s(n)$: this is immediate since we have a main loop that already causes *count* to be incremented at least n times.

- s is increasing: this is the case because the computation of $\widetilde{M^t}$ on input n includes everything that it does on input $n-1$, and more; note that *count* is continually incremented while simulating calls to TB on different inputs.
- s is time-constructible: it is so because the machine $\widetilde{M^t}$ itself computes $s(n)$ in $O(s(n))$ time.

Thus, we conclude that the first clause of the claim holds.

Next, we argue that the second holds. Since f is computable in time $O(t)$, there exist constants k_0, k_1 such that $k_0t(n)+k_1$ bounds the number of steps in a computation of f on any input of length n. Thus, for any query q, we have $\|f(q)\| \leq k_0t(\|q\|)+k_1$, and thus also $f(q) \leq 2^{k_0t(\|q\|)+k_1}$ since natural numbers are represented in binary. Pick K such that $t^2(n) > k_0t(n)+k_1$ for every $n \geq K$.

The machine $\widetilde{M^t}$ uses the counter *step* to bound the possible size of the oracle queries, and it simulates M^f over all oracle answers of length bounded by $t^2(\mathit{step})$. Now, fix some oracle f and fix some input w with length $n \geq K$. Then, $\widetilde{M^t}$ on input 1^n will perform a set of simulations which *includes* one that precisely simulates $\widehat{M^f}$ on input w; the simulation of $\widehat{M^f}$ on input w corresponds to one of the branches in the recursion tree created by calling the procedure TB. This makes it easy to see that clause (ii) of the claim holds.

5.3 The Proof

We can now give the proof of Theorem 4.4. Let R be any representation. It is sufficient to prove that $R \preceq_S \mathcal{C}_{[\,]}$ where $\mathcal{C}_{[\,]}$ is the representation by continued fractions. Thus, by the definition of $\preceq_S$, we have to prove that for any time-bound t there exists a time-bound s such that $O(t)_{\mathcal{C}_{[\,]}} \subseteq O(s)_R$.

Assume $\alpha \in O(t)_{\mathcal{C}_{[\,]}}$ (we will prove that there exists a time-bound s such that $\alpha \in O(s)_R$) . Let $f \in \mathcal{C}_{[\,]}$ be the continued fraction of α. By Definition 2.3, f is computable in time $O(t)$. By Definition 2.1, we have an oracle Turing machine M^f such that Φ^f_M is an R–representation of $\alpha \in (0,1)$ (this is true for any representation R, convert via the Dedekind cut if necessary). Now, $\mathcal{C}_{[\,]}$ is simply the set of total functions from $\mathbb{N}^+$ into $\mathbb{N}^+$, and hence, we can construct the canonical standard version $\widehat{M^f}$ of the oracle machine M^f, moreover, we can construct the Turing machine $\widetilde{M^t}$. By the first clause of the claim above, $\widetilde{M^t}$ computes a time-bound, and by the second clause, there exists a fixed number K such that $s(\|w\|)$ bounds the running-time of $\widehat{M^f}$ on input w whenever $\|w\| \geq K$. Let M_0 be the Turing machine given by

M_0 =on input w do:
Check if $\|w\| < K$.
If $\|w\| < K$: give the output w' where w' is the output of the oracle Turing machine M^f on input w (use a hard-wired table).
If $\|w\| \geq K$: run the Turing machine $\widehat{M^f}$ on input w; give the same output as $\widehat{M^f}$.

Now, M_0 computes an R–representation of α, moreover, M_0 runs in time $O(s)$. By Definition 2.3, we have $\alpha \in O(s)_R$.

References

[1] **A M Ben-Amram**, **L Kristiansen**, **J G Simonsen**, *On Representations of Real Numbers and the Computational Complexity of Converting Between such Representations* (2023); arXiv:2304.07227

[2] **Q Chen**, **K Su**, **X Zheng**, *Primitive Recursiveness of Real Numbers under Different Representations*, Electronic Notes in Theoretical Computer Science 167 (2007) 303–324; http://doi.org/10.1016/j.entcs.2006.08.018

[3] **I George**, **L Kristiansen**, **F Stephan**, *Computable Irrational Numbers with Representations of Surprising Complexity*, Annals of Pure and Applied Logic 172 (2021); http://doi.org/10.1016/j.apal.2020.102893

[4] **I Georgiev**, *Interplay Between Insertion of Zeros and the Complexity of Dedekind Cuts*, Computability (1 Jan. 2024) 1–25; http://doi.org/10.3233/COM-230469

[5] **I Georgiev**, *Subrecursive Graphs of Representations of Irrational Numbers*, from: "Unity of Logic and Computation. CiE 2023", LNCS 13967, Springer-Verlag (2023) 154–165; http://doi.org/10.1007/978-3-031-36978-0_13

[6] **K Hiroshima**, **A Kawamura**, *Elementarily Traceable Irrational Numbers.*, from: "Unity of Logic and Computation. CiE 2023", LNCS 13967, Springer-Verlag (2023) 135–140; http://doi.org/10.1007/978-3-031-36978-0_11

[7] **A Y Khintchine**, *Continued Fractions*, P. Noordhoff, Ltd. (1963). Translated by P. Wynn.

[8] **L Kristiansen**, *On Subrecursive Representability of Irrational Numbers*, Computability 6 (2017) 249–276; http://doi.org/https://doi.org/10.3233/COM-160063

[9] **L Kristiansen**, *On Subrecursive Representability of Irrational Numbers, Part II*, Computability 8 (2019) 43–65; http://doi.org/10.3233/COM-170081

[10] **L Kristiansen**, *On Subrecursive Representation of Irrational Numbers: Contractors and Baire Sequences*, from: "CiE 2021: Connecting with Computability", LNCS 12813, Springer-Verlag (2021) 308–317; http://doi.org/10.1007/978-3-030-80049-9_28

[11] **R S Lehman**, *On Primitive Recursive Real Numbers*, Fundamenta Mathematica 49 (1961) 105–118; http://doi.org/10.4064/fm-49-2-105-118

[12] **S Mazur**, *Computable Analysis*, volume 33 of *Rozprawy Matematyczne (Instytut Matematyczny Polskiej Akademii Nauk)*, Panstwowe Wydawnictwo Naukowe (1963). Edited by A. Grzegorczyk and H. Rasiowa.

[13] **A Mostowski**, *A Lemma Concerning Recursive Functions and its Applications*, Bull. Acad. Polon. Sci. Cl. III 1 (1953) 277–280.

[14] **A Mostowski**, *On Computable Sequences*, Fundamenta Mathematica 44 (1957) 37–51; http://doi.org/DOI: 10.4064/fm-44-1-37-51

[15] **I Richards**, *Continued Fractions Without Tears*, Mathematics Magazine 54 (1981) 163–172; http://doi.org/10.1080/0025570X.1981.11976921

[16] **D Skordev**, *Characterization of the Computable Real Numbers by Means of Primitive Recursive Functions*, from: "Computability and Complexity in Analysis (4th International Workshop, CCA 2000, Swansea, UK, September 2000)", LNCS 2064, Springer (2001) 296–309; http://doi.org/10.1007/3-540-45335-0_17

[17] **E Specker**, *Nicht Konstruktiv Beweisbare Sätze der Analysis*, Journal of Symbolic Logic 14 (1949) 145–158; http://doi.org/10.2307/2267043

[18] **K Weihrauch**, *Computable Analysis*, Texts in Theoretical Computer Science. An EATCS Series, Springer Verlag (2000)

Qiryat Ono, Israel

Department of Mathematics, University of Oslo, Norway

Department of Informatics, University of Oslo, Norway

benamram.amir@gmail.com, larsk@math.uio.no

Received: 15 July 2023 Revised: 30 May 2024