

Polynomial-Time Hierarchy of Computable Reals

Qingliang CHEN^{1,2,3,†}, Kaile SU^{3,4}, Lijun WU⁵

¹Department of Computer Science, Jinan University, Guangzhou 510632, China

²Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai, China

³ Key laboratory of High Confidence Software Technologies, Ministry of Education, Peking University, Beijing, China

⁴Institute for Integrated and Intelligent Systems, Griffith University, Brisbane, Australia

⁵School of Computer Science and Engineering, University of Electronic Science and Technology, Chengdu, 610054, China

Abstract

In mathematics, various representations of real numbers have been investigated and all these representations are proved to be mathematically equivalent. Furthermore, it is known that all effective versions of these representations lead to the same class of “computable real numbers”. However, when subrecursive (such as primitive recursive) is taken into account, these representations can lead to different notions of “computable real numbers”. This paper will look into the polynomial-time version of the problem for computable real numbers under different representations. We will summarize the known results to exhibit the comprehensive hierarchy they form. Our goal is to clarify systematically how the polynomial-time computability depends on the representations of the real numbers..

Keywords: Computable Real Numbers; Polynomial-Time Computability

1. Introduction

The computability of real numbers is introduced by Alan Turing in his seminal paper [15]. According to Turing, “the ‘computable’ numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means”. In order to define the “finite means” precisely, he introduces the nowadays well-known Turing machines. Since Turing machines compute exactly the computable functions on natural numbers, Turing defines actually the real numbers with computable decimal expansions as computable real numbers. Namely, x is computable if there is a computable function $f : \mathbb{N} \rightarrow \{0, 1, \dots, 9\}$

such that $x = \sum_{s=0}^{\infty} f(s) \cdot 10^{-(s+1)}$. Here we consider only the real numbers in the interval $[0, 1]$. As it was

pointed out by Robinson [13], Myhill [11], Rice [12] and others, the computability of real numbers can be equivalently defined by means of Cauchy sequences, Dedekind cuts and other representations of real

[†] Corresponding author.

Email addresses: tsingliangchen@gmail.com (Qingliang CHEN)

numbers. That is, the computability of reals is independent of their representations. The class of computable reals will be denoted by **EC** (for Effectively Computable).

Besides the computability, the subrecursive real numbers like primitive recursive (p.r., for short) and polynomial time computable real numbers have also been discussed. The different notions of subrecursive real numbers could be defined if different representations are used. Specker [14] is the first who investigates this problem and he shows that decimal expansions, Dedekind cuts and Cauchy sequences lead to three different versions of p.r. real numbers. Later on, Peter [4], Mostowski [10], and Lehman [8] investigated other versions of p.r. reals and showed some relations between the notions of p.r. real numbers based on different representations. For a comprehensive discussion of this problem, please refer to our paper in [3].

As for the polynomial-time computable real numbers, Ko [5] has shown that the polynomial version of Cauchy representation is more expressive than the standard left cut representation and binary expansion, and later he showed that the continued fraction representation is less expressive than the standard left cut representation but a slightly modified representation based on the principal convergents of real numbers can be equivalent to it, by polynomial Turing reduction [6]. However, not every important representation of real numbers have been discussed as far as polynomial-time computability is concerned and there is no a systematical overview about the polynomial-time hierarchy of real numbers under all different representations so far.

This paper aims to address the deficit. We summarize the known results about polynomial-time computable reals which we can find in literature. We will analyze systematically the dependence of polynomial-time computability of reals on the representations.

This paper is organized as follows. Firstly we recall the representations in the computability theory for real numbers in the next section, then we will survey and explore the hierarchy in these representations in section 3 and 4 by Cauchy sequences, binary expansions, Dekedind cut and continued fraction. And we will conclude the paper in the last section.

2. Representations of Real Numbers

In this section, we recall the representations of real numbers [16] which will be discussed in this paper. First we explain the classical form of the representations. Since we are interested in the effectivizations of the representations to different levels, all representations will be defined again in a uniform way such that they depend on some given class F of functions. According to the choice of the class F , various computability of different levels about real numbers can be defined. These notions depend also on the selected representations.

For simplicity, we consider only the real numbers in the unity interval $[0,1]$. If a real number x is notion this interval, then there is a $y \in [0;1]$ and a natural number n such that $x = y+n$ or $x = y-n$. In this case, the real numbers x and y should have the same computability level in any reasonable sense.

In this paper, the basic objects are integers and strings in $\{0,1\}^*$. The length of a string w is denoted by $l(w)$. In order to define precisely the complexity of “computing” a real number, we need to use specific models such as Turing machines or random access machines. What’s more, we need a uniform representation of rational numbers that can provide a natural measure of complexity since the computation

objects we are dealing with here are rational numbers. In this paper, Turing machines model is chosen for its simplicity. Also, instead of the set of rational numbers, we use the set of dyadic rational numbers because the length of a dyadic number is a natural measure of its complexity [7].

A *dyadic rational number* d is a rational number that has finite binary expansions, that is, $d = m/2^n$ for some integers m, n and $n \geq 0$. The set of dyadic rationals is $D = \{m/2^n : m \in \mathbb{Z}; n \in \mathbb{N}\}$. Each dyadic rational d has infinitely many binary representations with arbitrarily many trailing zeros. For each such representation t , we use $l(t)$ to denote its length. And $l(d)$ denotes its shortest binary representation. We use D_n to be the class of dyadic rationals with at most n bits in the fractional part of its binary representations.

Now we give the precise definition of the relativization of all important representations to a class F of functions.

Definiton 1. Let F be a class of functions $f: \mathbb{N} \rightarrow D, f: \mathbb{N} \rightarrow \mathbb{N}$ or $f: \mathbb{N} \rightarrow \{0,1\}$ and let $x \in [0,1]$ be a real number.

1. x has an F -Cauchy representation ($x \in F_{CS}$) if there is a function $f: \mathbb{N} \rightarrow D$ in F such that $f(n) \in D_n$ and $|f(n) - x| \leq 2^{-n}$.

2. x has a F -Dedekind cut representation ($x \in F_{DC}$) if there is a function $f: \mathbb{N} \rightarrow \{0,1\}$ and $g: \mathbb{N} \rightarrow D$ in F such that $f(n) = 1$ if and only if $g(n) < x$. Or intuitively, the Dedekind cut set $\{d \mid d \in D \text{ and } d < x\}$ can be decided by some $f \in F$.

3. x has a binary representation ($x \in F_{BIN}$) if there is a function $f: \mathbb{N} \rightarrow \{0,1\}$ in F such that

$$x = \sum_{s=0}^{\infty} f(s) \cdot 2^{-(s+1)}$$

4. x has a continued fraction representation ($x \in F_{CON}$) if there is a function $f: \mathbb{N} \rightarrow \mathbb{N}$ in F such that

$$x = f(0) + \frac{1}{f(1) + \frac{1}{f(2) + \dots}}$$

denoted by $x = [f(0), f(1), \dots]$.

When we limit the function classes F to be polynomial-time computable functions, it will lead to the definitions of various versions of “polynomial-time computable real numbers”. Denote by $\mathbf{P}_{CON}, \mathbf{P}_{DC}, \mathbf{P}_{BIN}$, and \mathbf{P}_{CS} the classes of real numbers which have polynomial-time continued fraction, polynomial-time Dedekind cut, polynomial-time binary expansion and polynomial-time Cauchy representations respectively. We will see that the hierarchy among these classes is as follows.

$$\mathbf{P}_{CON} \subsetneq \mathbf{P}_{DC} = \mathbf{P}_{BIN} \subsetneq \mathbf{P}_{CS}$$

3. Dedekind Cut, Binary Expansion and Cauchy Representations

The Dedekind Cut, binary and Cauchy representations are the most frequently discussed ones in literatures. And we can see in the following that the separation of the classes is also due to the application of diagonalization.

Theorem 2 (Ko [5]). The class of real numbers with polynomial-time computable Dedekind cut and

polynomial-time computable binary representations defines exactly the same class of computable real numbers, i.e., $\mathbf{P}_{DC} = \mathbf{P}_{BIN}$.

Proof: ($\mathbf{P}_{DC} \subseteq \mathbf{P}_{BIN}$) Assume that $x \notin \mathbf{D}$ (the case for $x \in \mathbf{D}$ is trivial) and the function for the binary expansion of x is f . Then for each n , the relation

$$f(n+1) = 1 \iff \sum_{i=1}^n f(i) \cdot 2^{-i} + 2^{-(n+1)} < x.$$

So, $f(n+1)$ can be computed from $f(0), f(1), \dots, f(n)$ by making only one question to the Dedekind cut D_x of x . So if $x \in P_{DC} \Rightarrow x \in P_{BIN}$ ($\mathbf{P}_{BIN} \subseteq \mathbf{P}_{DC}$). Assume that $x \notin \mathbf{D}$ (the case for $x \in \mathbf{D}$ is trivial) and the function for the binary expansion of x is f , then the greatest dyadic rational number $d \in D_n$ such that $d < x$ is

$$g(n) = \sum_{i=1}^n f(i) \cdot 2^{-i}.$$

So in order to decide a dyadic rational $d \in D_n$ whether or not $d < x$, we can compute the binary expansion of x to the first n bits and then compare the number d with $g(n)$. Therefore, if $x \in P_{BIN} \Rightarrow x \in P_{DC}$

Theorem 3 (Ko [5]). *The class of real numbers with polynomial-time computable Dedekind cut is strictly contained in the class of real numbers with polynomial-time computable Cauchy representations, i.e., $\mathbf{P}_{DC} \subsetneq \mathbf{P}_{CS}$.*

Proof: We show that there exists a real number x such that $x \in \mathbf{P}_{CS}$ while $x \notin \mathbf{P}_{DC}$. This can be done by a complexity-theoretic variant of the diagonalization.

First of all, we define a linear time honest function $T(n)$ inductively (In computational complexity theory, a linear time honest function is a function $T(n)$ that can be computed by a Turing machine in time $O(T(n))$):

$$T(n) := \begin{cases} T(1) = 1 \\ T(n+1) = 2^{2^{T(n)}} \end{cases}$$

Since $T(n)$ is time-constructible, we know that, there exists a set $A \subseteq \{0\}^*$ such that its characteristic function $\chi_A(0^n)$ can be computed in time $T(n)$ by some Turing machine but not in time $\log T(n)$ by any Turing machine. This can be proved by the application of diagonalization to the complexity classes such as that in the Time Hierarchy Theorem in [1], since $T(n)$ grows fast than $\log T(n)$.

Without loss of generality, let $0 \in A$. Define

$$x = \sum_{i=1}^{\infty} (2\chi_A(0^i) - 1)2^{-T(i)}.$$

First, we claim that x is a polynomial-time computable Cauchy real. The computation of the Cauchy function $\phi(0^n)$ such that $|x - \phi(0^n)| \leq 2^{-n}$ is as follows:

(1) Find the integer k such that $T(k) \leq n < T(k+1)$. This can be done in $O(n)$ steps because $T(n)$ is linear time honest.

(2) Then compute and output

$$\phi(0^n) = \sum_{i=1}^k (2\chi_A(0^i) - 1)2^{-T(i)}.$$

which can be done in $O(\sum_{i=1}^k T(i)) = O(n)$ steps. It is obvious that $|x - \phi(0^n)| \leq 2^{-(T(k+1)-1)} \leq 2^{-n}$ and $\phi \in \mathbf{P}$.

Secondly, we claim that $x \notin \mathbf{P}_{DC}$ by the way of contradiction. We will show that, if Dedekind cut set of x Dx can be computed in time $p(n)$ for some polynomial p , then we will construct a Turing machine which computes $\chi_A(0^n)$ in time $\log T(n)$ for almost all $n \in \mathbf{N}$ and thus establish a contradiction.

Suppose M_A is the Turing machine that computes $\chi_A(0^n)$ in time $T(n)$. We construct another Turing machine which works as follows.

(1) For a given input 0^n and $n > 0$, first simulate M_A on inputs $0, 0^2, \dots, 0^{n-1}$ and compute

$$d = \sum_{i=1}^{n-1} (2\chi_A(0^i) - 1)2^{-T(i)}.$$

(2) Then, we can determine the value of $\chi_A(0^n)$ by computing the membership of d in Dx since $d \in Dx$ iff $\chi_A(0^n) = 1$. Intuitively, if $d \in Dx$ we have to add something to approximate x , so $\chi_A(0^n) = 1$, and vice versa.

Then the computation time above is bounded by

$$O(\sum_{i=1}^{n-1} T(i)) + p(T(n-1)) < 2^{T(n-1)} = \log T(n).$$

for almost all n . Then we have a contradiction.

4. Continued Fraction Representation

First of all, we list some basic facts about this representation [9]. Let $x = [a_0, a_1, \dots]$ and

$$p_k/q_k = [a_0, a_1, \dots, a_k] = a_0 + \frac{1}{a_1 + \frac{1}{\dots + \frac{1}{a_k}}}$$

Lemma 4. For any $k \geq 2$,

$$p_k = p_{k-2} + a_k p_{k-1}, \quad q_k = q_{k-2} + a_k q_{k-1}.$$

Lemma 5. For any $k \geq 2$, $q_k \geq 2^{(k-1)^2}$, the fractions of the form with $0 < r < a_{k+1}$,

$$\frac{p_{k-1} + r p_k}{q_{k-1} + r q_k}$$

are called *intermediate fractions*.

Lemma 6. For any $k \geq 0$, the following sequence

$$\frac{p_k}{q_k}, \frac{p_k + p_{k+1}}{q_k + q_{k+1}}, \frac{p_k + 2p_{k+1}}{q_k + 2q_{k+1}}, \dots, \frac{p_k + a_{k+2}p_{k+1}}{q_k + a_{k+2}q_{k+1}} = \frac{p_{k+2}}{q_{k+2}}, x,$$

$$\frac{p_k + (a_{k+2} + 1)p_{k+1}}{q_k + (a_{k+2} + 1)q_{k+1}}, \dots, \frac{p_k + (a_{k+2} + i)p_{k+1}}{q_k + (a_{k+2} + i)q_{k+1}}, \dots, \frac{p_{k+1}}{q_{k+1}}$$

is monotone increasing if k is even, and is decreasing if k is odd.

Lemma 7. For any $k \geq 0$, if $x \neq p_k/q_k$, then

$$\frac{1}{q_k(q_{k+1} + q_k)} < \left| x - \frac{p_k}{q_k} \right| < \frac{1}{q_k q_{k+1}}.$$

A fraction a/b is a best approximation to x if $|a/b - x| \leq |c/d - x|$ for all c/d with $0 < d \leq b$.

Lemma 8. Let $q_{k+1} > b$. Then, $a/b \leq p_k/q_k$ iff $a/b \leq x$ if k is even; and $a/b < p_k/q_k$ iff $a/b < x$ if k is odd.

For any real number x , let $CONx$ and DCx denote its continued fraction and Dedekind cut representations respectively.

Theorem 9 (Ko [6]).

1. $CONx \equiv_T DCx$.

2. Assume that x is a real number such that $l(CONx(n)) \leq p(n)$ for some polynomial p . Then,

$$CONx \equiv_T^P DCx.$$

3. There exists a real number x such that DCx is polynomial-time computable, but $CONx$ is not polynomially length-bounded, and hence, $CONx \not\equiv_T^P DCx$.

Proof:

1. Assume that x is rational, then it is obvious that both $CONx$ and DCx are recursive. Therefore, $CONx \equiv_T DCx$.

If x is irrational, let $x = [a_0, a_1, \dots]$ and $p_k/q_k = [a_0, a_1, \dots, a_k]$ for $k \geq 0$. To compute DCx from $CONx$, we have the following Algorithm 1.

By Lemma 6, for each even k , $p_k/q_k < x$ and for each odd k , $p_k/q_k > x$. So, the algorithm always outputs the right answer when it halts. Furthermore, $\{p_k/q_k\}$ converges to x and $a/b \neq x$.

Therefore, the algorithm always halts.

To compute $CONx$ from DCx , we note that for each $k \geq 2$, the value $a_k = CON_x(k)$ can be computed from p_{k-2}/q_{k-2} and p_{k-1}/q_{k-1} along with DCx . Then by Lemma 6, we have the following Algorithm 2:

Algorithm 1 Computation from CON_x to DC_x

On input $a/b, b > 0$,
 for $k = 0, 1, \dots$, compute p_k/q_k from $CON_x(0), \dots, CON_x(k)$
 until
 if there is an even k such that $a/b \leq p_k/q_k$, output “yes, $a/b \in DC_x$ ”
 or there is an odd k such that $a/b \geq p_k/q_k$, output “no, $a/b \notin DC_x$ ”

Algorithm 2 Computation from DC_x to CON_x

On input $k \geq 2$,
 for $m = 1, 2, \dots$, compute the intermediate convergents
 $r_{k,m} = (p_{k-2} + mp_{k-1}) / (q_{k-2} + mq_{k-1})$
 until both $r_{k,m}$ and p_{k-1}/q_{k-1} are greater than x or less than x (by asking DC_x).
 output “ $CON_x(k) = m - 1$ ”

2. Assume that $l(an) \leq p(n)$ for some polynomial p . Then we modify Algorithm 1 as following Algorithm

3. By Lemma 7, $q_k > b$ iff $k \geq \lceil 2 \log b + 1 \rceil$. Therefore, according to Lemma 8, $a/b \leq p_k/q_k$ iff $a/b \leq x$,

we can easily verify the Algorithm 3 is correct. Since the computation of p_k/q_k from $CONx(0), CONx(1), \dots, CONx(k)$ can be done in polynomial time with $O(k)$ arithmetic operations, the above algorithm halts in polynomial time.

Next, we need to modify Algorithm 2 to become Algorithm 4. We define, for each k , $b_k = 2^{p(k)}$. Then, trivially $b_k \geq a_k$. We will perform a binary search. Again, Lemma 8 will imply that the algorithm is correct.

Furthermore, the binary search halts in $\lceil \log b_k \rceil = p(k)$ steps. So, $CONx \equiv_T^P DCx$.

3. Let $x = [a_0, a_1, a_2, \dots]$ where $a_0 = 1$, $a_n = 2^{2^n}$ for all n . Then, $l(CON_x(n))$ grows as an exponential function. We prove that DC_x is polynomial-time computable by giving the following Algorithm 5. We should note that by Lemma 8, Algorithm 5 is correct if $q_{k+1} > b$. Indeed, by Lemma 4, we can prove by induction that $2^{2^k} \leq q_k \leq 2^{2^{k+2}}$. So the correctness follows. Furthermore, since $0 \leq x \leq 1$, $q_k \leq 2^{2^{k+2}}$ implies $p_k \leq 2^{2^{k+2}}$ and so the computation of p_k/q_k costs $O(\log \log b)$ many arithmetic operations on numbers of length $O(l(b))$. This justifies that DC_x is polynomial-time computable.

Theorem 9 shows that the polynomial-time nonequivalence of DC_x and CON_x is basically due to the fact that the growth rate of $l(CON_x(n))$ may be faster than any polynomial for some x . This suggests the following final conclusion.

Algorithm 3 Modified Computation from CON_x to DC_x

On input a/b , $b > 0$,
 for $k = 0, 2, 4, \dots$,
 if $k \geq \lceil 2 \log b + 1 \rceil$
 compute p_k/q_k from $CON_x(0), CON_x(1), \dots, CON_x(k)$
 if $a/b \leq p_k/q_k$
 output "yes, $a/b \in DC_x$ "
 else
 output "no, $a/b \notin DC_x$ "

Algorithm 4 Modified Computation from DC_x to CON_x

On input $k \geq 2$,
 search for a c such that $1 \leq c \leq b_k$ and
 $r_{k,c} = (p_{k-2} + cp_{k-1})/(q_{k-2} + cq_{k-1})$ and $r_{k,c+1}$ lie on the opposite sides of x
 output " $a_k = c$ ".

Algorithm 5 Computation of DC_x

On input a/b , $b > 0$,
 search for a k such that $2^{2^k} \leq b \leq 2^{2^{k+1}}$ and we have $2^k \leq l(b)$
 compute p_k/q_k
 if " $a/b < p_k/q_k$ " or $(a/b = p_k/q_k$ and k is even)
 output " $a/b < x$ ".

Corollary 10 (Ko [6]). *The class of real numbers with polynomial-time computable continued fraction is strictly contained in the class of real numbers with polynomial-time computable Dedekind cut representation, i.e., $\mathbf{P}_{CON} \subsetneq \mathbf{P}_{DC}$.*

5. Conclusion

In this paper we summarize several known results about polynomial-time computable real numbers under different representations which are scattered in literatures and analyze systematically the dependence of

polynomial-time computability of reals on the representations. We have seen that, the polynomial-time computable reals under different representations form a comprehensive hierarchy:

$$\mathbf{P}_{\text{CON}} \subsetneq \mathbf{P}_{\text{DC}} = \mathbf{P}_{\text{BIN}} \subsetneq \mathbf{P}_{\text{CS}}$$

It is also very natural to discuss these representations in other complexity classes such as the **NC** and **Log-Space** computable reals.[2].

Acknowledgement

This work is supported by National Basic Research 973 Program of China grant 2010CB328103; National Natural Science Foundations grant 60725207, 61003056 and 61073033; ARC Future Fellowship FT0991785, Key research project of Ministry of Education in China grant No. 210257; Open Funds of Key Laboratory of Embedded System and Service Computing, the Fundamental Research Funds for the Central Universities of China and Guangdong Distinguished Young Scholar Nurturing Program grant No.LYM09028.

References

- [1] Sanjeev Arora, Boaz Barak, “Computational Complexity: A Modern Approach”, *Cambridge University Press*, 2009.
- [2] Fuxiang Yu, On the Representations of NC and Log-Space Real Numbers, Computing and Combinatorics, *13th Annual International Conference, COCOON 2007, Banff, Canada, July 16-19, 2007, Proceedings. Lecture Notes in Computer Science 4598*, Springer 2007.
- [3] Qingliang Chen, Kaile Su, Xizhong Zheng, Primitive Recursiveness of Real Numbers under Different Representations. *Electr. Notes Theor. Comput. Sci. 167: 303-324 (2007)*.
- [4] R. P’eter, Rekursive Funktionen, (*Akademischer Verlag*, 1951).
- [5] Ko, Ker-I. On the definitions of some complexity classes of real numbers, *Math. Systems Theory*,16 (1983):95–109.
- [6] Ko, Ker-I. On the continued fraction representation of computable real numbers, *Theoretical Computer Science*, 47(1986):299–313.
- [7] Ko,Ker-I. “Complexity Theory of Real Functions”, *Progress in Theoretical Computer Science*. Birkhauser, Boston, MA,1991.
- [8] Lehman,R.S. On primitive recursive real numbers. *Fundamenta Mathematicae*, 49(1960/61):105–118.
- [9] A.Ya.Khintchine, Continued fractions (transl.by P.Wynn) ,*Noordhoff, Groningen, The Netherlands, 1963*
- [10] Mostowski, Andrzej. On computable sequences. *Fundamenta Mathematicae*, 44(1957):37–51.
- [11] Myhill, John. Criteria of constructibility for real numbers. *Journal of Symbolic Logic*, 18(1953):7–10.
- [12] Rice, H. Gordon. Recursive real numbers. *Proc. Amer. Math. Soc.* 5(1954):784–791.
- [13] Robinson, Raphael M. Review of “R’ozsa P’eter, Rekursive Funktionen”. *Journal of Symbolic Logic*, 16(1951):280–282.
- [14] Specker,Ernst. Nicht konstruktiv beweisbare S’atze der Analysis. *Journal of Symbolic Logic*, 14(1949):145–158.
- [15] Turing,Alan M. On computable numbers, with an application to the “Entscheidungsproblem”. *Proceedings of the London Mathematical Society*, 42(1936):230–265.
- [16] Weihrauch,Klaus. “Computable Analysis: An Introduction”. *Springer*, Berlin Heidelberg, 2000.