

A NEW PROOF THAT THE SET OF COMPLEX NUMBERS IS DENUMERABLE

J. Ulisses Ferreira

Trv. Pirapora 36 Costa Azul 41.770-220
Salvador, Brazil

ABSTRACT

This short paper suggests that there might be numerals that do not represent numbers. It introduces an alternative proof that the set of complex numbers is denumerable, and also an algorithm for denumerating them. Both the proof and the contained denumeration are easy to be understood.

KEYWORDS

enumeration, Georg Cantor, diagonal process

1. INTRODUCTION

In 2004 [1] and 2005 [2], I demonstrated that \mathbb{C} , i.e. the set of the complex numbers, is denumerable. Roughly speaking, those papers observes that Cantor had no right for the last word by showing a number that had not been in the denumeration, for the real set (hence, any denumeration of it) was infinite. Before that, it had already been known that \mathbb{Q} , i.e. the rational numbers, is denumerable, and that repetitions of the same numbers, such as $1/2$, $2/4$ and $3/6$, in any denumeration such as

$$0, 1/1, -1/1, 1/2, -1/2, 2/1, -2/1, 1/3, -1/3, 2/2, -2/2, 3/1, -3/1...$$

caused no problem. The order of magnitude of the numbers does not matter either. Furthermore, I observed that the Cantor's diagonal process was based on numerals, instead of numbers. A numeral is a sequence of symbols that represents a number in accordance with some given numeral system, i.e. it is a mathematical notation of a number.

One cannot arbitrarily invent any set of new numbers. Historically, the zero and negative integers were discovered because there was some need: the subtraction operation. Then, the rational numbers were discovered since there was the need for dividing numbers, and so on. Thus, except for the natural numbers, the notion of a number expresses quantity but it is also the result from a sequence of arithmetical operations. Pi appeared because of the geometry, and so on. Thus, there may exist irrational numerals between 0 and 1 that do not represent numbers, possibly most of them. This means that using randomizing functions such as the well known **randomize** and **rand** in a programming language, even if the computer memory was infinite, will produce numerals based on decimal digits, but not necessarily numbers. Since the memory is finite, they do not even produce irrational numbers in such a representation. Not even $10/3$ can be represented on a

computer using decimal digits only. Cantor himself chose a numeral system, but it was not even adequate.

Cantor's diagonal process just seems to work because it was a deduction over symbols. In [1,2], I also proved that not only \mathbb{R} but also \mathbb{C} are denumerable. To regard a set as denumerable means that it is possible to build some denumeration of that set, but I showed a denumeration that was in accordance with a kind of data structure. In this short paper, I introduce a proof and a denumeration that are easier to be understood.

Note that MMXII (Roman) and 2012 (Arabic) are two different numerals that represent the same number, and that this is much older than Cantor and our culture in this century. Whether there are or not numerals in decimal Arabic notation that are supposed to represent some irrational number but do not represent any number, the following proof works.

2. THE PROOF

Here, I use the coding and decoding system previously used by Kurt Gödel in the proof of his historic theorem of incompleteness, published in 1931 [3], which, in its turn, was inspired by Whitehead and Russell's work entitled *Principia Mathematica* [4]. The used numbers are better known as "Gödel Numbers".

As a professional originally from the area of programming languages design and implementation, whenever I do not have suitable means for representing any arithmetic expression, I simply write the expression as if I was using a subset of some full programming language, such as BASIC, Pascal, Java, Haskell or any other, taking into consideration that I use the symbol "^" to represent power like in BASIC, instead of "*" like in FORTRAN, and that some languages do not provide the power function. In this way, those are the *numerals* that I chose for representing the set \mathbb{C} of the complex numbers: for instance, I represent the constant e by simply "e"; For the irrational number π , I simply represent it by "pi"; The irrational number which is the square root of -1, I represent by "sqrt(-1)", but as I wish to also represent all complex numbers, I simply write "i" for representing the same number, and so on. A few examples of numerals of mine are "10/3", "sqrt(2)", "1+3/(sin(30)*2)", "log(2,10)-pi", "e^32", "2+6*i", each of them representing its corresponding complex number. Furthermore, since they are arithmetic expressions only, not logical propositions nor predicates nor function definitions, there is no self-references or paradoxes involved.

Given a coding table for characters such as ASCII, the next step consist in calculating the Gödel numbers that correspond to those strings, using a function that I would naturally define having G as its identifier. Let c be the function that receives a character and results in its ASCII code (or, alternatively, any more suitable code from a shorter table), and, skipping 1, let us use the same sequence of the first smaller prime numbers such as the one used by Gödel while proving that theorem of his. Thus, I can give the following examples:

$$G("e") = 2^c('e').$$

$$G("pi") = 2^c('p') * 3^c('i').$$

$$G("10/3") = 2^c('1') * 3^c('0') * 5^c('/') * 7^c('3').$$

$$G("2+6*i") = 2^c('2') * 3^c('+') * 5^c('6') * 7^c('*') * 11^c('i').$$

By regarding any sequence of symbols, one can observe that there are infinite Gödel numbers that do not represent any complex number, but this is not a problem since both sets are infinite and

any subset of the natural numbers is denumerable. Therefore, the set of Gödel numbers that represent arithmetic expressions is also denumerable.

Since the function G always receives a complete expression as a string and results in a natural number, and, since I use the numerals that I chose as strings that contain arithmetic expressions, I can represent any complex number, the set of the complex numbers is denumerable.

Remark: Note that, as in any coding function using Gödel numbers, there is an inverse function that, from any natural number obtained from the above explained manner, results in the string that contains the original arithmetic expression.

Theorem 2. *There does exist some algorithm that denumerates the set of the complex numbers.*

Proof. I will prove constructively, showing my algorithm: first, I define an alphabet, a formal language for arithmetic expressions and write a parser for expressions of this language, which, in its turn, contains only the alphabet defined for constant expressions, operators and symbols such as “(”, “,” and “)” used in expressions, identifiers of all arithmetic functions such as “log” and “sin”, and reserved words of constant values, such as “e”, “pi” and “i”, but that language does not contain any variable. One typically uses BNF for doing so.

A question is to know what the first number of my denumeration is. Thus, I have to set an order. Let `IsaGNumber(n)` be our function that receives a natural number `n` and results in **true** if `n` is a Gödel number, and **false** otherwise. Let `G'` be the inverse function that receives a Gödel number and results in the corresponding sequence of symbols of the chosen alphabet, possibly the original arithmetic expression. Starting from the natural number `n = 2`, one applies `G'` and verifies to know whether the resulting string denotes any well-formed arithmetic expression, in accordance with the formal language previously defined. If true, this is the first numeral of my denumeration. Otherwise, one can try using `n = 3`, then `n = 4`, then `n = 5`, and so on, until one finds the first natural number `n` whose `G'(n)` application results in some arithmetic expression well formed. The obtained string is the first numeral, which, in its turn, represents the first number of my denumeration.

Moreover, one obtains this and the other numbers in order using the following algorithm, in Algol 60 or Pascal style:

```
var n, ok: integer; s: string;
n := 1; m := 0; ok := 1000;
while ok = ok do
begin
  n := n + 1;
  if IsGNumber(n) then
  begin
    (* Gprime means G' as above *)
    s := Gprime(n);
    if parser(s) = ok then
    begin
      m := m + 1;
      writeln(s)
    end
  end
end;
```

In the end, what is written by the above algorithm is a sequence of complex numbers, i.e. a denumeration of them.

Note that m is a variable that indicates the m th number being written in the denumeration. It is not used in the algorithm, but it shows to the reader an easier understanding that there is a one-to-one correspondence between what is being denumerated and the natural numbers in order of magnitude.

3. CONCLUSIONS

There are no transfinite numbers, and there is no set with cardinality greater than infinite. Cantor, Post and others collapsed the older notions of numbers and numerals, and that was a mistake. There is no need for the notion of “computable numbers” either, since what was called “computable number” is just a number whose numeral in the Hindu-Arabic decimal notation can represent another number up to some given precision.

REFERENCES

- [1] Ferreira, Ulisses, (2004) “The real set can be seen as denumerable”, In Hamid R. Arabnia, Iyad A. Ajwa, and George A. Gravanis, editors, Post-Conference Proceedings of the 2004 International Conference on Algorithmic Mathematics & Computer Science, CSREA Press, Las Vegas, Nevada, USA, pp523-526.
- [2] Ferreira, Ulisses (2005) The sets of real and complex numbers are denumerable. ACM SIGACT News 36(2): pp126-130.
- [3] Gödel, Kurt (1931) Über formal unentscheidbare sätze der principia mathematica und verwandter system i. Monatshefte Mathematik und Physik. 38, pp173-98.
- [4] Whitehead, Alfred & Russell, Bertrand (1910) Principia Mathematica, Cambridge University Press, 3 volumes, 13.

Author

Ferreira was born in Salvador in 1961, studied and did some research work in computer science at UFBA, UFPB, Sussex University, Edinburgh University and Trinity College in Dublin.

