

11-2020

Possibility to Algorithmically Check: Yet Another Reason Why Current Definitions Have Been Selected in Elementary Mathematics

Christian Servin

Olga Kosheleva

Vladik Kreinovich

Follow this and additional works at: https://scholarworks.utep.edu/cs_techrep



Part of the [Mathematics Commons](#)

Comments:

Technical Report: UTEP-CS-20-112

Published in *Applied Mathematical Sciences*, 2020, Vol. 14, No. 18, pp. 881-886.

Possibility to Algorithmically Check: Yet Another Reason Why Current Definitions Have Been Selected in Elementary Mathematics

Christian Servin¹, Olga Kosheleva², and Vladik Kreinovich³

¹Computer Science and

Information Technology Systems Department

El Paso Community College (EPCC)

919 Hunter Dr., El Paso, TX 79915-1908, USA

cservin1@epcc.edu

^{2,3}Departments of ¹Teacher Education and ³Computer Science

University of Texas at El Paso

500 W. University

El Paso, TX 79968, USA

olgak@utep.edu, vladik@utep.edu

Abstract

At first glance, many definitions in mathematics – especially in elementary mathematics – seem arbitrary. Why is 1 not considered a prime number? Why is a square considered to be a particular case of a parallelogram – in some old textbooks, a parallelogram was defined in such a way as to exclude the square. In his 2018 article, Art Duval explained many such definitions by a natural requirement to make the corresponding results (theorems) as simple as possible. However, elementary mathematics is not just about theorems and proofs, it is also about computations. In this paper, we show that from the computational viewpoint, it is also preferable, e.g., to view a square as a particular case of a parallelogram.

1 Formulation of the Problem

Why current definitions have been selected? In mathematics, some definitions seem to be, at first glance, rather arbitrary. For example:

- Is 1 a prime number? At first glance, it should be, since it has no factors other than 1 and itself – but current definitions do not consider 1 to be prime.
- Is a square a parallelogram? In the past, some textbooks specifically

defined a parallelogram in such a way so as not to include a square, but all modern textbook consider a square a particular case of a parallelogram.

Is there simply a historical reason why the current definitions were selected – or is there some logic in selecting such definitions?

Duval’s approach. In his blog [1], Art Duval explains that many current definitions have been motivated by the desire to make formulations of the corresponding results (theorems) simpler. For example, one of the main facts about prime numbers is that any positive integer number can be uniquely represented as a product of prime numbers. This works if we use the current definition of a prime number. However, if we consider 1 as a prime number 2, then we would have different product representations: e.g., 6 could be then represented in at least two different ways:

- as $6 = 2 \cdot 3$ or
- as $6 = 1 \cdot 2 \cdot 3$.

As shown in [1], several other examples of seemingly arbitrary current definitions can be similarly explained.

How about computations? Mathematics – especially elementary mathematics – is not only about theorems and proofs, it is also about computations. From the very beginning, mathematical results were used to compute the values of desired quantities: e.g., mathematical formulas allows us to estimate distances – be it distances to the mountain peaks or to stars – based on the difference in observation angles.

A natural question is: can some definitions be explained by this need for computations – and thus for algorithms?

What we do in this paper. In this paper, we show that indeed, some of the current definitions can be explained by this need for algorithms.

2 What Is Algorithmic and What Is Not: A Brief Reminder

Need for this section. To understand why some definitions lead to somewhat more algorithmic results than others, we need to recall the main ideas and results related to algorithmic processing of real numbers; for details, see, e.g., [2].

How real numbers are represented in computable mathematics. Real numbers are ubiquitous in elementary mathematics, they are a mathematical construction that describes possible values of physical quantities, such as length, angle, area, etc.

In real life, the only way to know the values of such quantities is to measure them. Measurements can be made very accurate, but they are never absolute accurate: there is always some accuracy $\varepsilon > 0$. In other word, whatever value

r we measure, it is, in general, different from the actual (unknown) value x of the corresponding physical quantity: $|r - x| \leq \varepsilon$.

The more accurate the measurement, the more information we get about the actual value x . To get complete information about the quantity, we need to have an infinite sequence of measurements with higher and higher accuracy. In other words, for every natural number n , we need to have a measurement result r_n obtained with at least n binary digits accuracy, i.e., for which $\varepsilon \leq 2^{-n}$ and thus, $|r_n - x| \leq 2^{-n}$.

All we know about the actual value x is this sequence of rational numbers – rational because at present, measurement results are presented in binary form and thus, represent a rational number of the type $\frac{p}{2^q}$ for some integers p and q . At any given moment of time, we only have finitely many such values r_1, \dots, r_n .

When we talk about algorithms applied to real numbers, we mean algorithms that use these values r_1, \dots, r_n as input – because these values are the only information that we have about the real number. An algorithm has to stop and produce the result. At this point, the algorithm only uses finitely many values r_i .

One of historically first negative results: it is not algorithmically possible to check whether two real numbers are equal. By using the above understanding of algorithms operating on real numbers, we can check which properties of real numbers are algorithmically checkable and which are not. One of the historically first results of this type is the result that *it is not algorithmically possible to check whether two real numbers are equal or not*.

This impossibility can be proved by reduction to a contradiction. Indeed, suppose that such an algorithm exists. This means that for every two numbers x and y – represented by the corresponding sequences r_n and s_n – for some N , we will be able to make a conclusion – about whether the two numbers are equal or not – based on the first N values of each sequence, i.e., based on the values r_1, \dots, r_N , and s_1, \dots, s_N .

In particular, for the case when we consider the values $x = y = 0$ represented by sequences $r_1 = r_2 = \dots = 0$ and $s_1 = s_2 = \dots = 0$, this algorithm should correctly conclude that the numbers x and y are equal to each other. In other words, based on the sequences $r_1 = \dots = r_N = 0$ and $s_1 = \dots = s_N = 0$, this algorithm should return the answer “yes, the numbers x and y are equal to each other”.

However, the same approximations $s_1 = \dots = s_N = 0$ can also represent a slightly different value $z = 2^{-N}$. Indeed, for all $n \leq N$, we have

$$|s_n - z| = 2^{-N} \leq 2^{-n}.$$

To be more precise, the number $z = 2^{-N}$ can be represented by a sequence s'_1, s'_2, \dots for which:

- $s'_1 = \dots = s'_N = 0$ and
- $s'_{N+1} = s'_{N+2} = \dots = 2^{-N}$.

Thus, if we apply our hypothetic equality-checking algorithm to the following two real numbers:

- the number $x = 0$ represented by the sequence $r_n = 0$ and
- the number $z = 2^{-N}$ represented by the sequence s'_n ,

then our assumed-to-exist algorithm:

- will use exactly the same binary inputs $r_1 = \dots = r_N = 0$ and $s_1 = \dots = s_N = 0$ as for the previous inputs $x = 0$ and $y = 0$, and
- thus, this algorithm will make the exact same conclusion: that the input real numbers are equal to each other.

This leads to a problem:

- For $x = y = 0$, equality was a valid conclusion.
- However, for $x = 0$ and $z = 2^{-N}$, this conclusion is wrong, since these two numbers are different: $x = 0 \neq z = 2^{-N}$.

We thus get a contradiction with the assumption that the algorithm always (correctly) checks whether two real numbers are equal or not.

This contradiction shows that it is not algorithmically possible to check whether two real numbers are equal or not.

3 How This Explains the Current Choice of Definitions

Square and parallelogram: reminder. In the current definition, a square can be defined as a parallelogram in which:

- all sides are equal, and
- all angles are equal to 90° .

Two possible definitions. As we have mentioned earlier, there are two possible definitions of a parallelogram. In both definitions, a generic parallelogram is a parallelogram, the difference is whether a square is defined as a parallelogram or not:

- in an old definition, a square is *not* considered a particular case of a parallelogram, while
- in the current definition, a square *is* considered a particular case of a parallelogram.

From the algorithmic viewpoint, why is the current definition better?

In both definitions, if all we have are sides and angles, we cannot algorithmically decide whether the corresponding object is a square or not:

- this would require checking whether sides are equal and whether angles are equal to 90 degrees, and
- we have shown that this checking is not algorithmically possible.

However, for checking whether a given rectangle with mutually parallel sides is a parallelogram, the situation is different:

- In the old definition, we would also not be able to check whether a given object is a parallelogram – because for that, we would need to check that it is not a square, and this would require checking that either the sides are different or the angles are different from 90 degrees.
- In contrast, in the current definition, no such checking is needed – every quadrangle with parallel sides is automatically a parallelogram.

In this sense, the current definition is more algorithmic and thus preferable.

Comments.

- Similar ideas can be applied to other current definitions – e.g., the definitions mentioned in [1].
- It is interesting that two different arguments – from proofs and from computations – lead to the same selection of the definitions. Maybe there is a general reason for this coincidence?

Acknowledgments

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), and HRD-1834620 and HRD-2034030 (CAHSI Includes).

The authors are thankful to Art Duval for valuable discussions.

References

- [1] A. Duval, *What Is 0^0 , and Who Decides and What Does it Matter? Definition in Mathematics*, American Mathematical Society Blog on Teaching and Learning Mathematics, November 1, 2019, <https://blogs.ams.org/matheducation/2018/11/01/what-is-00-and-who-decides-and-why-does-it-matter-definitions-in-mathematics/>
- [2] K. Weihrauch, *Computable Analysis*, Springer Verlag, Berlin, 2000.