Proceedings of the 4th International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/2025.22647

Proof of the Nonexistence of an Algorithm That Tells If a Point Is on a Closed Set's Boundary or Exterior

Jiale Fan[1*], Jincan Li[2]

[1]*Valley Christian High School, San Jose, USA*
[2]*University of Leeds, Leeds, UK*
**Corresponding Author. Email: jerryfan5433@gmail.com*

Abstract: This study investigates the connection between the undecidability of certain sets and the problem of determining the position of a point relative to the boundary or exterior of a closed set. We introduce a new algorithm, referred to as Algorithm A, which generates a closed set at regular intervals. By employing a reductio ad absurdum argument, we demonstrate that no algorithm exists that can consistently determine whether a point lies on the boundary or in the exterior of a closed set. This finding highlights the limitations of algorithmic approaches in the context of constructive mathematics.

Keywords: Algorithm, undecidable, closed set

1. Introduction

Constructive mathematics deals with strict use of constructive proofs and axioms which are used to form more intricate proofs. Contrary to classical mathematics, which seeks to prove theorems and conjectures without generally considering its complement, constructive mathematics requires mathematicians to not only use a certain way of thinking to formulate a path to the proof but also create certain assumptions that make sense among them (thus the existence of axioms). Instead of dealing with abstract ideas, this area of mathematics utilizes intuitive, mathematical understanding to make a better sense of our world.

Proofs in constructive mathematics include systematic processes that can be used to support the validity of its arguments in a realistic way. This algorithmic way of thinking enables mathematicians to not utterly reject past proofs but to reconsider them in a different point of view. Constructive mathematics is popular within the realms of computer science where algorithms and logical thinking is crucial to its development.

The US school of mathematics was founded by Bishop[1] and the Russian school of mathematics was founded by Markov and Shanin. While these schools studied constructive mathematics, each school have different rules and guidelines for forming proofs and ideas. One example of such relates to Markov's principle[2], which states that if the inapplicability of an algorithm U to some input x has been refuted, then the algorithm is applicable to that input x. The Russian school permitted the use of Markov's principle, also known as constructive choice, which sometimes allows mathematicians to argue by contradiction. On the other hand, the US school generally does not accept Markov's principle, with one reason being that it does not tell how long it would take for an algorithm to terminate successfully given some inputs.

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2753-8818/2025.22647

A Constructive Sequence of Rational Numbers (CSRN) is an algorithm that transforms every natural number into its respective rational number. Constructive Real Numbers can be defined as a set of two programs: A computer-generated Cauchy sequence of numbers and a convergence regulator, which ensures the convergence of the sequence of numbers. The reason that we have the convergence regulator is because we can never truly predict what the next values of the Cauchy sequence will be, which is like how we can never know whether a flipped coin will be heads or tails despite the preconceived knowledge that the theoretical chance is 50 percent for each outcome. Therefore, the convergence regulator is used to ensure the predictability of the values of the generated Cauchy sequence. A constructive separable metric space is defined as a set of points where the distances between them can be described as a constructive real number within a topological space.

This research explores the ideas of boundaries and open sets while applying the knowledge of undecidability, topological spaces, and mathematical programs. Our general approach to proving whether a given point is outside or on its boundary is to find the undecidability of such program, meaning that it realistically cannot halt in a finite amount of time. Building on the ideas that Alan Turing had developed in 1936, we construct such theoretical program and test whether it can uphold the presented conditions.

2. Definition

Computable function: A function f with natural arguments and values is called computable if there exists an algorithm that computes f, that is, an algorithm A such that[3]:

If $f(n)$ is defined for a certain natural n, then the algorithm A halts on the input n and prints $f(n)$

If $f(n)$ is undefined, then the algorithm A does not halt on the input n.

Decidability: A set X of natural numbers is called decidable if there exists an algorithm that determines whether an arbitrarily given natural number n belongs to the set X. Such an algorithm must terminate for any n and give one of the two answers “yes” or “no”.

Topology: A family ξ of sets where the intersection of any two members of ξ is a member of ξ and the union of any collection of members of ξ is a member of ξ.

Union of the collection of the member sets is a member of ξ.

Intersection of finitely many elements of ξ is a member of ξ.

Empty set and the whole space are members of ξ.

Topological space: Defined as a pair (X, ξ) where ξ is a topology for set X.

Closed set: A subset A of a topological space (X, ξ) is closed if and only if its relative complement $X\backslash A$ is open, implying that a set is open if and only if its complement is closed[4].

Open set: A subset A of X of a topological space (X, ξ) is an open set of X if and only if A belongs to X.

Neighborhood: The neighborhood of a point is an open set containing it.

Interior point: A point x of a subset A of a topological space is an interior point of A if and only if there is an open subset of A containing x. The set of all interior points of A is the interior of A, denoted A°

Boundary: The boundary of subset A of a topological space X is the set of all points x which are interior to neither A nor $X \sim A$.

Exterior: The exterior of a set B, denoted by $Ext\ B$, is the interior of the complement of the set B.

3. Main theorem

For a General Constructive Topological space and a closed set S in it and a point p, it is not possible to algorithmically tell if the point p is in the exterior of S or on the boundary.

DOI: 10.54254/2753-8818/2025.22647

3.1. Construction of algorithm Y and X

We let Y be an unextendible algorithm where it cannot necessarily cover all integer inputs and we first assume that this algorithm certainly exists. The fact that this algorithm is unextendible implies that the domain of this program is undecidable, meaning that there exists no algorithm such that it can make decisions through this domain.

We also let X be an algorithm such that it has domain Z. For every second, this algorithm will produce a closed set F_s, where s is the respective second that the closed set was produced from. Throughout the run of this algorithm, we will define F_s depending on whether Y(n) had stopped. We also define the number of seconds passed as t and choose the number 1 as an arbitrary number for this proof (though any number could be chosen like 0).

Construction of our program:

Let us run program Y:

1. If algorithm Y(n) did not stop at t seconds, then let $F_s = [1, s + 1]$, $(s < t)$
2. If algorithm Y(n) does stop at t seconds, then let F_s be:

a. $[1.5, s+1]$ for $(s \geq t)$

b. $[1, s + 1]$ for $(s < t)$

Let us concurrently run X:

1. If X detects program Y as never terminating, then X will return the intersection of the sets that are outputted so far: $([1, 2] \cap [1, 3] \cap [1, 4] \cap \ldots \cap [1, t])$

2 If X detects program Y as terminated, then X will return the intersection of the sets that are outputted by Y: $[1, 2] \cap [1, 3] \cap [1, 4] \cap \ldots \cap [1, t] \cap [1.5, 2] \cap [1.5, 3] \ldots$

Given the executions of these two programs we can deduce that:

If Y never stops, then $\cap^{t}_{s=1} F_s = [1, 2]$

If Y stops at some point, then $\cap^{t}_{s=1} F_s = [1.5, 2]$

Note: these intersections done above are closed sets since intersection of closed sets are closed.

Therefore, by definition, our arbitrary number 1 is on the boundary of [1, 2] and exterior of [1.5, 2]. We can say that if the program never stops, then our arbitrary number 1 is on the boundary of the intersection generated by X, and if the program stops, then the arbitrary number is on the exterior (namely, not part of the set) of the set generated by X.

3.2. Proof of our final result

In this section, we employ a reductio ad absurdum argument to establish the nonexistence of the algorithm A(S, p), which purportedly determines whether a point p lies on the boundary or in the exterior of a closed set S .

To illustrate the contradiction inherent in this assumption, we consider the previously defined algorithms Y and X. The algorithm Y operates on integer inputs and is characterized as unextendible, while the algorithm X concurrently generates closed sets based on the behavior of Y. Specifically, we define the closed sets produced by X as follows:

1. If Y(n) does not terminate after t seconds, then X will return the intersection of the sets generated up to that point: $\cap^{t}_{s=1}[1, s + 1] = [1, 2]$.

2. Conversely, if Y(n) terminates at t seconds, then X will return the intersection of the sets generated by Y up to that moment: $\cap^{t}_{s=1}[1, s + 1] \cap \cap^{\infty}_{s=t+1}[1.5, s + 1] = [1.5, 2]$.

Thus, we arrive at two distinct scenarios based on the behavior of Y:

- In the case where Y does not halt, the intersection yields [1, 2].
- In the event that Y halts, the intersection yields [1.5, 2].

It is noteworthy that these intersections are closed sets, as the intersection of closed sets remains closed. Consequently, we can deduce that if Y does not terminate, the point 1 is situated on the

DOI: 10.54254/2753-8818/2025.22647

boundary of the set [1, 2]. Conversely, if Y does terminate, the point 1 resides in the exterior of the set [1.5, 2].

This leads us to a contradiction: if the algorithm A(S, p) were to exist, it would imply a consistent method to ascertain the position of p relative to the closed set S. However, our construction demonstrates that the behavior of Y engenders an undecidable scenario, thereby proving that no such algorithm A(S, p) can exist.

4. Conclusion

This paper has rigorously established the nonexistence of a universal algorithm capable of determining whether a point lies on the boundary or in the exterior of a closed set within the framework of constructive separable metric spaces. A constructive separable metric space is defined as an algorithmically given set of points with the distances between them being computable real numbers (CRNs) and with a countable, algorithmically given topological base. Importantly, the real line of constructive real numbers serves as an example of such a space.

Building on the undecidability of the Halting Problem[5,6] , we have shown that for a vast class of constructive separable topological metric spaces[7], including the real line of constructive real numbers, it is not possible to algorithmically determine whether a given point is on the boundary of a set or in its exterior. This finding underscores the profound connections between computability theory and topology, particularly within the realm of constructive mathematics.

Moreover, while there may be specific subclasses of closed constructive sets where algorithmic methods are effective[8], this study concludes that such methods cannot be universally applied. The inherent undecidability of these problems invites further theoretical exploration into the boundaries of computability in this context, potentially leading to the development of more advanced frameworks or identifying conditions that allow for decidable cases.

Acknowledgment

We sincerely express our thanks to Viktor Chernov and Vladimir Chernov who formulated this problem and to Neoscholar company for organizing the summer program where the problem was studied.

References

[1] *E. Bishop, D. Bridges: Constructive analysis Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], 279 Springer-Verlag, Berlin (1985)[1,5]*

[2] *B. A. Kushner: Lectures on constructive mathematical analysis (in Russian) Monographs in Mathematical Logic and Foundations of Mathematics. Iz- dat. "Nauka", Moscow, 1973. 447 pp., English translation in Translations of Mathematical Monographs, 60 American Mathematical Society, Providence. [2]*

[3] *Shen, A., and Nikolai Konstantinovich Vereshchagin. Computable Functions. American Mathematical Society, 2003. [3]*

[4] *Kelly, John L. General Topology John L. Kelly. Springer, 1975. [4]*

[5] *A. M. Turing: Corrections, Proc. Lond. Math. Soc., ser. 2, 43 (1937), 544-546 [5]*

[6] *E. Bishop, D. Bridges: Constructive analysis Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], 279 Springer-Verlag, Berlin (1985)*

[7] *N. A. Sanin (Shanin): Constructive real numbers and constructive functional spaces (in Russian) Trudy Mat. Inst. Steklov 67 (1962) 15–294. English translation in Amer. Math. Soc., Providence R.I. (1968) [6]*

[8] *N. Vereshchagin & A. Shen: Algorithms and randomness. In Proceedings of the 14th Annual IEEE Conference on Computational Complexity (1999), pp. 12–21. IEEE. [7]*