

Countability of Infinite Paths in the Infinity Tree: Proof of the Continuum Hypothesis in a Non-Cantorian Infinity Theory

Philip C. Jackson

TalaMind LLC, Troy, MI, USA

Email: dr.phil.jackson@talamind.com

How to cite this paper: Jackson, P.C. (2025) Countability of Infinite Paths in the Infinity Tree: Proof of the Continuum Hypothesis in a Non-Cantorian Infinity Theory. *Advances in Pure Mathematics*, 15, 73-90.
<https://doi.org/10.4236/apm.2025.151003>

Received: November 22, 2024

Accepted: January 24, 2025

Published: January 27, 2025

Copyright © 2025 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

A previous paper showed that the real numbers between 0 and 1 could be represented by an infinite tree structure, called the 'infinity tree', which contains only a countably infinite number of nodes and arcs. This paper discusses how a finite-state Turing machine could, in a countably infinite number of state transitions, write all the infinite paths in the infinity tree to a countably infinite tape. Hence it is argued that the real numbers in the interval $[0, 1]$ are countably infinite in a non-Cantorian theory of infinity based on Turing machines using countably infinite space and time. In this theory, Cantor's Continuum Hypothesis can also be proved. And in this theory, it follows that the power set of the natural numbers $P(\mathbb{N})$ is countably infinite, which contradicts the claim of Cantor's Theorem for the natural numbers. However, this paper does not claim there is an error in Cantor's arguments that $[0, 1]$ is uncountably infinite. Rather, this paper considers the situation as a paradox, resulting from different choices about how to represent and count the continuum of real numbers.

Keywords

Infinity, Countable, Uncountable, Diagonalization, Real Numbers, Infinity Tree, Continuum Hypothesis, Turing Machine, Infinite Time Turing Machine, Non-Cantorian

1. Introduction

A previous paper discussed how the infinite set of real numbers between 0 and 1 could be represented by a tree structure with a countably infinite number of nodes, called the 'infinity tree'. [1] There is a one-to-one correspondence between infinite

paths from the root node of the tree and infinite decimal expressions for numbers in the interval $[0, 1)$. That paper did not claim to prove that the number of infinite paths is countably infinite, and noted that such a proof would contradict Cantor's diagonalization argument [2] that the real numbers in the interval $[0,1]$ are uncountably infinite.

This present paper shows that a Turing machine could write all the infinite paths of the infinity tree to a constant, finite set of countably infinite tapes, if it performs a countably infinite number of processing steps, requiring only a constant finite set of processing states. Hence this paper finds that the real numbers in the interval $[0,1]$ are countably infinite, and proposes a non-Cantorian theory of infinity based on Turing machines using countably infinite space and time. In this theory, Cantor's Continuum Hypothesis is true, since there is no set with cardinality between that of the integers and the real numbers. And in this theory, the power set of the natural numbers $P(\mathbb{N})$ is countably infinite, which contradicts the claim of Cantor's Theorem for the natural numbers.

However, this paper does not claim there is an error in Cantor's arguments that $[0, 1]$ is uncountably infinite. Rather, this paper considers the situation as a paradox, resulting from different choices about how to represent and count the continuum of real numbers.

2. Types of Infinity Discussed in This Paper

It may help to briefly list the different kinds of infinity discussed in this paper, and briefly describe how this paper discusses them:

- The “countable infinity” of the natural numbers: 0, 1, 2, 3, 4, ...
- The “countable infinity” of spaces on a Turing machine's tape. I shall consider a Turing machine which has twelve tapes, with each tape having a countable infinity of spaces. I will discuss an “infinite time Turing machine” which performs a countable infinity of processing steps. (A tape may be regarded as countably infinite in both directions. However, my discussion will only use a countably infinite number of cells in a single direction from a starting point on each tape).
- The “infinity tree” [1] which has a countable infinity of nodes, a countable infinity of arcs, and an infinity of infinite paths descending from the root node, each path corresponding to an infinite decimal expression in the range $[0, 1)$, *i.e.*, $[0.0000\dots, 0.9999\dots]$.

It should be noted that this paper includes 0 in the natural numbers.¹ However, we shall later see that this doesn't matter when considering Cantor's Theorem.

The central question studied in this paper will be whether the number of infinite paths in the infinity tree is countably infinite or is an uncountable infinity. This paper will show that all the infinite paths in the infinity tree could be written onto a countably infinite tape of a Turing machine, in countably infinite Turing ma-

¹Per Wikipedia: “In mathematics, the natural numbers are the numbers 0, 1, 2, 3, and so on, possibly excluding 0.” [Enderton, Herbert B. (1977) *Elements of Set Theory*, Academic Press, p. 66].

chine state transitions. Hence it will be argued that the set of infinite paths in the infinity tree is countably infinite, and that the set of infinite decimal expressions in the interval $[0, 1)$ is countably infinite.

It should be noted that this paper does not use ‘limit equivalences’, such as “ $0.3999\dots = 0.4000\dots$ ”. This paper is not concerned with such equivalences, which would only reduce the number of decimals to be considered in the range $[0, 1)$. Hence such equivalences would work in favor of this paper’s argument that the set of decimals in the range $[0, 1)$ is countably infinite.

This paper refers to the interval $[0, 1]$ in several places, where it is making the claim that the set of numbers in the range $[0, 1]$ is countably infinite. If we accept this paper’s argument that the set of numbers in the range $[0, 1)$ is countably infinite, then it follows that the range $[0, 1]$ is also countably infinite, since we can add the number 1 to the set $[0, 1)$ and the resulting set is still countably infinite. Indeed, the Turing machine described here always adds the number “1” to the list of decimals that it writes to a countably infinite tape.

3. Definition of the Infinity Tree

The ‘infinity tree’ [1] is defined as follows:

1. The root node has the label \bullet , representing a decimal point, implicitly preceded by 0. That is, the root node should be visualized as 0.
2. Each node in the tree has pointers to 10 different nodes below it, labeled with the digits 0 through 9. The 10 nodes are unique to the node above them. No other node points to any of them.
3. The tree is not reentrant. No two nodes point to the same other node. There is no circular path between nodes. No node points to itself.

This is a simple infinite tree, with a branching factor of 10 arcs from each node. Just a few of the nodes and branches of the first few levels of the ‘infinity tree’ are shown in **Figure 1**. For the infinite tree, the following assertions hold:

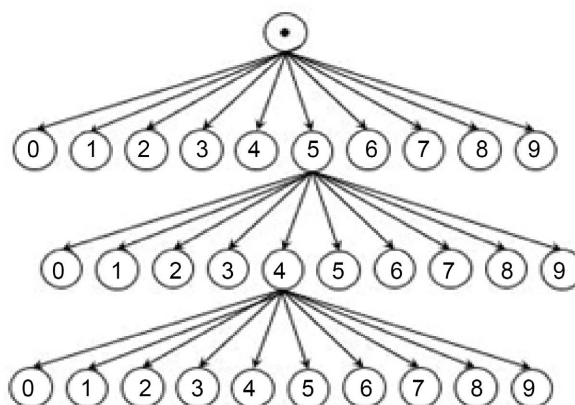


Figure 1. A partial illustration of the first 4 levels of the infinity tree.

1. Every finite path from the root node to a descendant corresponds to a finite

decimal number between 0 and 1. Therefore, every node below the root node also corresponds to a finite decimal number between 0.0 and 1.0, indicated by the path reaching it from the root node.

2. The nodes in the tree are countably infinite, *i.e.*, each node can be counted by a distinct, sequential natural number (1, 2, 3, ...). This can be done by navigating the tree breadth-first: The root node is counted 1. Its descendants are counted 2 through 11. The nodes at the next level down are counted 12 through 111, and so forth counting nodes breadth-first throughout the tree. Therefore, the set of all nodes in the infinite tree is countably infinite.

3. The arcs in the tree are also countably infinite, in the same breadth-first manner since each node has exactly ten arcs going from it.

4. For each non-root node N , there is a unique, finite path from the root node to node N . So, the number of finite paths to nodes in the tree is also countably infinite.

5. Every infinite path descending from the root node of the tree corresponds to an infinite decimal fraction, from zero up to .99999999... Mathematically, "0." followed by an infinite number of 9's is considered equal to 1. However, the tree does not literally include the path "1.0", nor does it need to, for the purpose of discussion in this paper.

6. Every infinite decimal from 0 up to but not including 1 corresponds to an infinite path in the tree descending from the root node.

7. So, a countably infinite set of nodes and arcs comprises the infinite set of infinite paths from 0 downward through the tree.

8. So, a countably infinite set of nodes and arcs in a tree represents the set of infinite decimals from 0 up to but not including 1.

9. Therefore, the set of infinite decimals between 0 and 1 can be mapped to and represented by a countably infinite set of nodes and arcs. Any number between 0 and 1 can be reached to any desired degree of precision, with countably many nodes and arcs.

The infinity tree shows something specific and non-ambiguous: that a countably infinite set of nodes and arcs can exactly represent the set of all infinite decimals in the range $[0, 1)$.

The following sections will discuss in more detail whether the set of infinite paths in the tree is countably infinite, or uncountably infinite. This question was left open in [1].

4. The Real Numbers in $[0, 1]$ Are Countably Infinite

4.1. The Basic Argument

To discuss this topic, we begin by considering just the first three levels of the infinity tree below its root node. A list of the paths from the root node to each node at the third level below the root would look like:

. 0 0 0
. 0 0 1

. 0 0 2
 ...
 . 9 9 8
 . 9 9 9

Now suppose we try to diagonalize this list. We can choose any number we like to replace the first 0 in “. 0 0 0”. Whatever number we choose for the first digit will select some other number already in the list. Then whatever number we choose for the second digit will select some other number already in the list. And whatever number we choose for the third digit will select some other number already in the list. This is what I mean by saying the infinity tree cannot be diagonalized to specify a number that it does not contain. This applies from the root node of the tree to any finite number of levels below the root node.

Now imagine that a Turing machine [3] is designed to simulate a breadth-first descent through the infinity tree and to write decimal expressions that correspond to paths from the root node to nodes at each level of the infinity tree below the root node.

After it has descended three levels through the tree, it will have written the above expressions on its tape. After it has simulated descent through four levels of the tree, it will have written expressions from .0000 to .9999 to its tape. And so on: As it simulates breadth-first descent through the tree, it always writes a complete set of finite expressions from the interval between 0 and 1 to its tape, and the set of finite expressions can never be diagonalized to find an expression it does not contain.

The machine only requires a finite amount of time and processing to simulate descent to any finite level of the tree, although the amount of time and processing goes up by a multiple of 10 for each level. To write the expressions for paths to the n th level of the infinity tree to a Turing machine tape, the machine requires space for representing paths from the root to 10^n nodes of the tree. Each path is of the form “0.” followed by n digits for choices in the tree along the path. Including a blank space after each path, the path requires $n + 3$ cells on a tape, *i.e.*, $O(n)$ space on a tape. Representing all 10^n paths to level n requires $(n+3)10^n = n10^n + (3*10^n)$ cells on a tape, *i.e.*, $O(n10^n)$ space on a tape. Using one more cell for writing “1” after all the paths to level n , the space required on a tape is still $O(n10^n)$.

The Turing machine never runs out of space on the infinite tape to write the paths to each finite level of the infinity tree. The Turing machine never runs out of discrete time steps to continue processing. For a Turing machine, countably infinite time never ends.

Since the infinity tree has a countably infinite number of levels, and the machine only uses finite time to process each level, the machine requires only countably infinite time and processing to simulate descent through all the levels of the tree. Also, the machine only requires countably infinite space on its tape to write the expressions for all the paths that it generates, representing numbers in the interval $[0, 1]$, if it always writes 1 after writing 0.999... 9 for each level of the tree.

So, in countably infinite time, the Turing machine would write to a countably infinite tape all the decimal expressions in the interval $[0, 1)$, plus the number 1. The set of starting locations for all these numbers would always be a subset of the countably infinite spaces on the Turing machine tape. So, the set containing the starting points of all these numbers would be countably infinite.

Therefore, we can argue that the set of real numbers in $[0, 1]$ is countably infinite. This of course is a surprising conclusion, since it contradicts Cantor's arguments that the real numbers in $[0, 1)$ are uncountably infinite. Cantor's arguments are discussed in Section 7.

4.2. Use of Twelve Tapes

For simplicity and efficiency, the Turing machine could actually use twelve countably infinite tapes. One of these tapes would be designated as the "input level tape", and the other as the "output level tape".

The machine would simulate performing a series of passes through successive levels of the infinity tree. In each pass, the machine would use decimal expressions on the input level tape representing all paths from the root node to level n of the tree, to support generating decimal expressions on the output level tape representing all paths from the root node to level $n+1$ of the tree.

The machine would use the other ten tapes (the "branch tapes") to write decimal expressions representing extensions from the path to an individual node of the tree to paths for its ten child nodes at the next level of the tree. Only one node of the tree is processed as a parent at a time using these branch tapes. The ten branch tapes are individually named the "branch0tape", "branch1tape", "branch2tape", ..., "branch9tape". (Appendix A gives pseudocode defining the operation of the machine.)

After computing and writing all information for level $n + 1$ of the infinity tree to the output level tape, the machine copies the information from the output level tape back to the input level tape. Both tapes then have the same information written to them, and the machine is ready to compute the expressions representing paths from the root node to the next level $n + 2$ of the tree.

And so on, for the countably infinite set of levels of the infinity tree.

For a Turing machine ([4], p. 298), initially all the cells on the twelve infinite tapes are blank, except for at most a finite number of cells on each tape that are initially non-blank. Without loss of generality, each tape can have a first, left-most cell, and a countably infinite number of cells to the right, *i.e.*, each tape can be "singly infinite" ([5], p. 129).

For the Turing machine described here, initially each tape would have the symbol '*' on its first cell, to support the state-transition function recognizing that the machine is at the left-most cell of the tape.² All tapes will have only blank spaces on every other cell, except for the machine's 'input-level' tape, which initially has "0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1" on its left-most cells. That is, the input-

²The symbol '*' at the starting position of each tape is not shown in the Appendix A pseudocode.

level tape initially has 11 finite numerical expressions, separated by single blank spaces between them.

The Turing machine would effectively generate and write the continuum of real numbers in $[0, 1]$ within a countably infinite set of cells on a tape, using a countably infinite number of processing steps.

So again, the Turing machine supports the conclusion that the continuum of real numbers in the interval $[0, 1]$ is countably infinite. Likewise, it supports the conclusion that the infinity tree contains only a countable infinity of countably infinite paths descending from its root node. One cannot diagonalize the infinity tree to specify any real number in the interval $[0, 1]$ that the Turing machine discussed here does not generate, in writing all real numbers in that interval to a countably infinite tape.

4.3. Infinite Time Turing Machines

The Turing machine discussed here may be considered as an “infinite time Turing machine” (ITTM). However, it should be noted that there are different treatments and discussions of the ITTM concept. Stannett [6] surveyed the topic. It appears that most (and perhaps all) previously published papers on ITTM’s do not challenge the thinking that Cantor was correct in arguing that the real numbers are uncountably infinite. It does not appear that previous papers on ITTM’s have discussed the Turing machine design presented here.

Regarding ITTM’s, Hamkins [7] wrote: “The idea is to allow a Turing machine to compute for infinitely many steps, preserving the information produced while doing so, and allowing the machine to continue with more computation afterwards.”

This description implicitly represents time by the computation steps of a Turing machine. A unit of time (a discrete time step) corresponds to a single computation step, within which: a) a Turing machine reads a symbol on a tape; b) based on the machine’s current state, the machine may write a symbol on a tape, and then optionally move one space left or right on a tape; and c) optionally the machine may change its state. Precisely what happens within each computation step is specified by the finite state transition table for the Turing machine. For example, the table could specify that the machine will in parallel write a different symbol on different tapes, and then move one space left or right on each tape, all within a single computation step. [4] [5]

For a Turing machine, countably infinite time corresponds to a countably infinite sequence of computation steps. Likewise, countably infinite space corresponds to a Turing machine’s constant, finite set of countably infinite tapes. This paper will not discuss or consider what happens “after” countably infinite time and will not rely on “limit states” for each cell on a tape. This paper contends only that a certain, specific Turing machine would be able to generate and write all the real numbers in $[0, 1]$ within countably infinite computation steps, using a constant, finite number of countably infinite tapes.

4.4. Experimental Validation

The author has implemented a detailed simulation of a Turing machine for which pseudocode is given in Appendix A, in a computer program showing that with only a constant set of 24 states, a constant set of 227 state transition rules, and a constant set of twelve tapes, the Turing machine could iteratively generate all the paths in the infinity tree to each level of the tree. The simulation uses the symbol ‘*’ at the starting position on each tape. (Including headings, the state transition table has 41 rows and 229 columns of information. Including the state transition table and the Visual Basic code for this computer program as additional Appendices would more than double the length of this paper).

Admittedly, it is counterintuitive to find that a finite-state Turing machine can write a countable infinity of countably infinite decimal expressions to a single countably infinite tape, and also write copies of all these expressions to a second countably infinite tape, using only countably infinite discrete time steps and state transitions. Such is the power of combining countably infinite space with countably infinite time for a Turing machine.

5. Cantor’s Continuum Hypothesis

Cantor advanced the Continuum Hypothesis in 1878, which states that “there is no set whose cardinality is strictly between that of the integers and the real numbers.” [8] Establishing the truth or falsity of the hypothesis was the first of Hilbert’s 23 problems. [9] Cantor was unable to prove or disprove the hypothesis during his lifetime, and it has remained unproven until now. In 1940, Gödel showed that the hypothesis is consistent with the axioms of set theory, including the Axiom of Choice. [10] In 1963, Cohen showed that the hypothesis “cannot be derived from the other axioms of set theory, including the Axiom of Choice”. [11]

By showing in Section 4 above that the cardinality of the real numbers is the same as that of the natural numbers, this paper has also shown that there is no set with cardinality between that of the natural numbers and that of the real numbers. Hence Cantor’s Continuum Hypothesis is shown to be true, if one considers the possible computations of a Turing machine within countably infinite space and time, in a non-Cantorian theory of infinity.

6. Cantor’s Theorem

Cantor’s Theorem provides a proof that the power set of the natural numbers \mathbb{N} has cardinality greater than that of the natural numbers. In this paper’s non-Cantorian infinity theory, it follows that the power set of the natural numbers $P(\mathbb{N})$ is countably infinite, which contradicts the claim of Cantor’s Theorem for the natural numbers.

This paper considers zero to be a natural number, as noted in Section 2, although later in this section, we will see that it doesn’t matter whether zero is considered to be a natural number or not, for this paper’s conclusion about Cantor’s Theorem.

To represent sets of natural numbers which may include zero, we can map any subset S of the natural numbers $\mathbb{N} = 0, 1, 2, 3, \dots$ to a real number $0.b_1b_2b_3\dots$ where $b_i = 1$ if $i - 1$ is in S and $b_i = 0$ if $i - 1$ is not in S . That is, the first digit indicates whether 0 is a member of the subset or not, the second digit indicates whether 1 is a member or not, the third digit indicates if 2 is a member or not, etc. For example, the subset $\{0, 3, 5\}$ would be represented by 0.100101 followed by infinite 0's. Any set is considered a subset of itself, and \mathbb{N} would be represented by 0.111111...

By definition, the empty set is also an element of a power set. In this paper's mapping, the empty set will be mapped to the infinite decimal 0.000... A non-empty subset of the natural numbers will not map to the infinite decimal 0.000....

This mapping gives us a way to represent the power set of natural numbers $P(\mathbb{N})$ by a subset of the real numbers in $[0, 1)$. If the set of all real numbers in $[0, 1)$ is countably infinite, as shown in Section 4 of this paper, then $P(\mathbb{N})$ would also be countably infinite.

However, this would contradict Cantor's Theorem that the power set of the natural numbers \mathbb{N} has cardinality greater than that of the natural numbers. To address this contradiction, we need to discuss how Cantor's Theorem is proved.

A standard proof of Cantor's Theorem goes as follows:³

"Suppose that \mathbb{N} is equinumerous with its power set $P(\mathbb{N})$. Let us see a sample of what $P(\mathbb{N})$ looks like:

$$P(\mathbb{N}) = \{\emptyset, \{1, 2\}, \{1, 2, 3\}, \{4\}, \{1, 5\}, \{3, 4, 6\}, \{2, 4, 6, \dots\}, \dots\}.$$

$P(\mathbb{N})$ contains infinite subsets of \mathbb{N} , e.g. the set of all positive even numbers $\{2, 4, 6, \dots\} = \{2k: k \text{ in } \mathbb{N}\}$, along with the empty set \emptyset .

Now that we have an idea of what the elements of $P(\mathbb{N})$ are, let us attempt to pair off each element of \mathbb{N} with each element of $P(\mathbb{N})$ to show that these infinite sets are equinumerous. In other words, we will attempt to pair off each element of \mathbb{N} with an element from the infinite set $P(\mathbb{N})$, so that no element from either infinite set remains unpaired. Such an attempt to pair elements would look like this:

$$\begin{array}{l} 1 \leftrightarrow \{4, 5\} \\ 2 \leftrightarrow \{1, 2, 3\} \\ 3 \leftrightarrow \{4, 5, 6\} \\ 4 \leftrightarrow \{1, 3, 5\} \\ \vdots \\ \vdots \\ \vdots \end{array}$$

Given such a pairing, some natural numbers are paired with subsets that contain the very same number. For instance, in our example the number 2 is paired with the subset $\{1, 2, 3\}$, which contains 2 as a member. Let us call

³This proof of Cantor's Theorem is quoted from https://en.wikipedia.org/wiki/Cantor%27s_theorem, which says that "Cantor gave essentially this proof in a paper published in 1891 'Über eine elementare Frage der Mannigfaltigkeitslehre' ... Ernst Zermelo has a theorem (which he calls 'Cantor's Theorem') that is identical to the form above in the paper that became the foundation of modern set theory ('Untersuchungen über die Grundlagen der Mengenlehre I'), published in 1908."

such numbers *selfish*. Other natural numbers are paired with subsets that do not contain them. For instance, in our example the number 1 is paired with the subset {4, 5}, which does not contain the number 1. Call these numbers *non-selfish*. Likewise, 3 and 4 are non-selfish.

Using this idea, let us build a special set of natural numbers. This set will provide the contradiction we seek. Let B be the set of all non-selfish natural numbers. By definition, the power set $P(\mathbb{N})$ contains all sets of natural numbers, and so it contains this set B as an element. If the mapping is bijective, B must be paired off with some natural number, say b . However, this causes a problem. If b is in B , then b is selfish because it is in the corresponding set, which contradicts the definition of B . If b is not in B , then it is non-selfish and it should instead be a member of B . Therefore, no such element b which maps to B can exist.

Since there is no natural number which can be paired with B , we have contradicted our original supposition, that there is a bijection between \mathbb{N} and $P(\mathbb{N})$.

Note that the set B may be empty. This would mean that every natural number x maps to a subset of natural numbers that contains x . Then, every number maps to a nonempty set and no number maps to the empty set. But the empty set is a member of $P(\mathbb{N})$, so the mapping still does not cover $P(\mathbb{N})$.

Through this proof by contradiction, we have proven that the cardinality of \mathbb{N} and $P(\mathbb{N})$ cannot be equal. We also know that the cardinality of $P(\mathbb{N})$ cannot be less than the cardinality of \mathbb{N} because $P(\mathbb{N})$ contains all singletons, by definition, and these singletons form a “copy” of \mathbb{N} inside of $P(\mathbb{N})$. Therefore, only one possibility remains, and that is that the cardinality of $P(\mathbb{N})$ is strictly greater than the cardinality of \mathbb{N} , proving Cantor’s theorem.”

The lines that I have underscored in the preceding quotation would not apply with the mapping that I’ve defined above, and the proof by contradiction would fail: The empty set \emptyset would map to the infinite decimal $0.000\dots$. Each subset S of the natural numbers $\mathbb{N} = 0, 1, 2, 3, \dots$ would map to a real number $0.b_1b_2b_3\dots$ where $b_i = 1$ if $i - 1$ is in S and $b_i = 0$ if $i - 1$ is not in S .

So, if the set of all real numbers in $[0, 1)$ is countably infinite, then $P(\mathbb{N})$ would also be countably infinite. And Section 4 above argues that the set of all real numbers in $[0, 1]$ is countably infinite, by considering the computations of a Turing machine to generate these numbers in countably infinite space and time. Thus, Cantor’s Theorem is contradicted within the non-Cantorian infinity theory presented in this paper: One can argue reasonably that \mathbb{N} is equinumerous with its power set $P(\mathbb{N})$, if the real numbers in $[0, 1]$ are countably infinite.

“Hilbert’s Hotel” shows a 1-1 correspondence between the natural numbers $0, 1, 2, 3, \dots$ and $1, 2, 3, 4, \dots$.⁴ So, the argument presented in this paper also shows that the set $1, 2, 3, 4, \dots$ is equinumerous with its power set. Hence, it doesn’t matter

⁴I thank an anonymous mathematician for noting this point about Hilbert’s Hotel.

whether 0 is included as a natural number, in considering Cantor's Theorem.

Cantor's Theorem is also contradicted by considering the possible computations of a Turing machine to generate the set of infinite paths in the infinity tree within countably infinite space and time: Recall that at each level of the infinity tree, the machine writes all the paths from the root of the tree to that level, to a tape. This is a finite list of finite length expressions. The number of expressions increases by 10 for each level. The length of each expression increases by 1 for each level. For level n of the tree, the machine writes 10^n expressions, each of length n to the tape. The tree has a countably infinite number of levels. If the machine were to traverse \aleph_0 levels, it would write 10^{\aleph_0} expressions each of length \aleph_0 onto a tape with \aleph_0 squares. In effect, given countably infinite time, the machine would map the 10^{\aleph_0} expressions into \aleph_0 .

It may seem impossible or paradoxical that a tape with \aleph_0 squares can hold 10^{\aleph_0} expressions, with each expression of length \aleph_0 , yet that is precisely what would be accomplished in countably infinite time by the Turing machine. The Turing machine never runs out of space on a countably infinite tape, and infinite time never stops.

And it is well known that countable infinities can contain multiple countable infinities within them: For example, the countable infinity of natural numbers contains the countable infinities of even numbers, odd numbers, and prime numbers, and also countable infinities of randomly chosen natural numbers.

7. Cantor's Arguments for Uncountability of Real Numbers

7.1. Cantor's First Uncountability Argument

Before using diagonalization, Cantor [12] gave a different uncountability argument for the real numbers. He wrote:

“Suppose we have an infinite sequence of real numbers, $\omega_1, \omega_2, \dots, \omega_n, \dots$ where the sequence is given according to any law and where the numbers are distinct from each other. Then in any given interval $(\alpha \dots \beta)$ a number η (and consequently infinitely many such numbers) can be determined such that it does not occur in the series; this shall now be proved. ...”

This uncountability argument does not apply to the countably infinite sequence of real numbers in the interval $[0, 1]$ that would be generated in a countably infinite sequence of steps by a Turing machine (and all of them written to a countably infinite tape) using the method discussed in this paper, because this paper's approach agrees that there are no first and second numbers in the open interval $(0, 1)$, and that for any pair of numbers $\alpha < \beta$ in the interval, there are infinitely many numbers between them. This can also be seen by considering the infinity tree for the interval $[0, 1)$.

Yet for the infinity theory proposed by this paper, it doesn't matter that there are no first and second numbers in the open interval $(0, 1)$, nor in any open interval (α, β) within $[0, 1)$. In countably infinite time, the Turing machine would write

to a countably infinite tape all the decimal expressions in the interval $[0, 1)$, plus the number 1. The set of starting points for all these numbers would always be a subset of the countably infinite spaces on the Turing machine tape. So, the set containing the starting points of all these numbers would be countably infinite. Therefore, the set of real numbers in $[0, 1]$ is countably infinite, in the infinity theory advocated by this paper.

7.2. Cantor's Diagonalization Argument

To begin, *I do not claim to find an error in Cantor's diagonalization argument.* However, some discussion needs to be given in this paper.

Cantor's diagonalization argument [2] considered infinite strings of two mutually exclusive characters, m and w . Essentially, he considered infinite binary strings: we could use $m = 0$ and $w = 1$. For any infinite sequence of such infinite strings, he defined a diagonal infinite string that differed on every diagonal element, by changing m to w and w to m , *i.e.*, flipping the binary element. He then argued that the diagonal infinite string could not exist anywhere in the countably infinite sequence of infinite binary strings. He did not require or specify any particular order for the list of infinite binary strings.

However, if we consider a binary infinity tree, with only two branches, labeled 0 and 1 from each node, then the 2^n paths from the root node to each level $n \geq 1$ below the root node correspond to all the finite binary sequences of length n . A new number specified by flipping the first n characters of the first n rows in a list of binary sequences would always exist somewhere in a list that included all 2^n initial sequences of the two characters 0 and 1. The new sequence would always exist in the binary infinity tree, and it would always exist somewhere on a countably infinite Turing machine tape, in the expressions written by a Turing machine running for infinite time, simulating a breadth-first navigation of the binary infinity tree. If the Turing machine writes "0." at the start of each binary sequence that it writes to its tape, then if it runs for countably infinite time steps, simulating breadth-first descent through all the countably infinite levels of the binary infinity tree, it will never write more than a countably infinite number of "." characters on its tape. So, the Turing machine's execution would demonstrate that the set of all infinite binary sequences is a countably infinite set, contradicting Cantor's diagonalization argument.

Similarly, the definition and design of the decimal infinity tree are such that it cannot be diagonalized: For any number p that Cantor specifies in diagonalizing a hypothetical countably infinite list of decimal numbers, p also specifies a path that exists in the decimal infinity tree. And each number in Cantor's hypothetical countably infinite list of decimal numbers also specifies a path that exists in the decimal infinity tree.

So, if a Turing machine could run for countably infinite time, then it could simulate a breadth-first descent through the countably infinite levels of the decimal infinity tree and write a list of decimal expressions corresponding to all the infinite

paths descending from the root node of the infinity tree. This would correspond to all the decimal numbers in the interval $[0, 1]$, if the Turing machine always writes “1” after it writes the decimal expressions for each level of the infinity tree. The set of starting points for these decimal expressions would be countably infinite, since they are all written on a single countably infinite tape. Hence if a Turing machine could run for countably infinite time, then it would demonstrate that the number of decimal expressions in $[0, 1]$ is \aleph_0 , countably infinite.

However, since I do not claim to find an error in Cantor’s diagonalization argument yet reach a different conclusion by considering a Turing machine argument, we appear to have a paradox. This is further discussed in Section 9 below.

8. Computability of Real Numbers

This paper has focused on countability of the real numbers, yet it is appropriate to also discuss computability. A real number r is said to be *computable* if there is a Turing machine which, given n on its initial tape, prints the decimal expression of r to the n th digit on an output tape.⁵ That is, r is computable if it can be written to a tape to any desired finite degree of precision, by a finite-state Turing machine performing a finite number of computation steps.

There is a countable infinity of different Turing machines, so if we consider Turing machines which each only print out the decimal expression of a different real number r , then the set of real numbers that are computable is countably infinite. If the set of real numbers in the interval $[0, 1)$ were uncountably infinite, then there would be uncountably many real numbers which are not individually computable by different Turing machines.

Yet the entire infinity tree is computable by a single Turing machine, with a constant finite set of states, a constant finite set of state transition rules, and a constant set of twelve countably infinite tapes, as discussed in Section 4 of this paper. All the numerical expressions to any finite level of the tree can be computed by the Turing machine and written on an output tape with a total number of symbols that is finite. The number of output symbols and computation steps required goes up by a factor of 10 for each level, yet these numbers are always finite at each level.

Since the infinity tree contains all the real numbers in $[0, 1)$, each real number in $[0, 1)$ can effectively be computed by a Turing machine which computes the entire infinity tree. Since the Turing machine always writes 1 after computing expressions at each level of the tree, it effectively computes all the real numbers in $[0, 1]$.

When the infinity tree is computed in this manner, two reals are distinguished from each other even when they are arbitrarily close together, *i.e.* even when they differ by a branch at any finite depth in the infinity tree.

9. Accepting Paradoxes

In writing this paper, my strong preference has been to leave open the possibility that the proposed non-Cantorian theory of infinity and Cantor’s theory of infinity

⁵See Minsky (5), p. 159, and https://en.wikipedia.org/wiki/Computable_number.

may be equally valid alternative theories.

An analogy is the acceptance and co-existence of alternative Euclidean and non-Euclidean theories of geometry, as another situation which also involves reasoning about infinity: In Euclidean geometries, parallel lines remain the same distance from each other when extended to infinity. In non-Euclidean geometries parallel lines can curve away or curve toward each other, when extended indefinitely. Yet these different geometries do not have anything specific to do with the countability or uncountability of real numbers.

There is a similar conflict between this paper's proposed non-Cantorian theory of infinity and Cantor's theory of infinity: Cantor claims to prove the proposition P that the real numbers in $[0, 1]$ are uncountably infinite. I claim to prove the proposition $\neg P$, that they are countably infinite.

However, I do not claim there is an error in Cantor's diagonalization argument. Yet I also claim that my argument based on a Turing machine computing paths in the infinity tree is correct. And my approach is consistent with Cantor's first uncountability argument.

So, in this paper, I simply accept this situation as a paradox, resulting from different choices about how to represent and count the continuum of real numbers.

This paper also identifies and accepts another paradox, in claiming that a Turing machine tape with \aleph_0 squares can hold 10^{\aleph_0} decimal expressions, with each expression of length \aleph_0 symbols. There is no finite point in time where this paradox happens: At each finite step in the machine's operation, it always has countably infinite empty space on the Turing machine's output tape to write more expressions. This result is achieved if the machine performs countably infinite processing steps, *i.e.*, if it runs for countably infinite time. And it is well known that countable infinities can contain multiple countable infinities within them: For example, the countable infinity of natural numbers contains the countable infinities of even numbers, odd numbers, and prime numbers, and also countable infinities of randomly chosen natural numbers. So, this paradox is not an impossibility, and not a self-contradiction.

This paper also has the hypothetical premise that a Turing machine can run for countably infinite time steps, a topic that has been previously studied by other researchers, as noted in Section 4.3.

10. Other Related Work

It does not appear that there has been any previously published work presenting the approach of this paper.

There has been a long history of attempts by others to refute Cantor's arguments. Hodges [13] gave an analysis of errors in counter arguments to Cantor that were contained in papers he received either as a referee or as editor of the *Journal of Symbolic Logic*. It does not appear that Hodges' paper covers the arguments presented here: His paper does not include any references to Turing, nor to Turing machines. Hodges' paper includes only one use of the word "tree" which is meta-

phorical, not mathematical.

And again, to be clear, this paper does not claim there is an error in Cantor's diagonalization argument that $[0, 1]$ is uncountably infinite. Rather, this paper considers the situation as a paradox, resulting from different choices about how to represent and count the continuum of real numbers.

11. Conclusions

Considering the infinity tree [1], this paper has shown that all the real numbers in the continuum $[0, 1]$ can be computed by a Turing machine and written to a countably infinite tape in a countably infinite number of state transitions. A computer program has been written which simulates the operation of such a Turing machine. Hence it is argued that the real numbers in the interval $[0, 1]$ are countably infinite in a non-Cantorian theory of infinity based on a Turing machine with a constant finite set of states, a constant finite set of state transition rules, and a constant set of twelve countably infinite tapes, using countably infinite space and time to compute and write the decimal expressions of the real numbers in $[0, 1]$.

In this theory, Cantor's Continuum Hypothesis is true, since there is no set with cardinality between that of the integers and the real numbers. And in this non-Cantorian infinity theory, it follows that the power set of the natural numbers $P(\mathbb{N})$ is countably infinite, which contradicts the claim of Cantor's Theorem for the natural numbers. In this theory, all the real numbers in $[0, 1]$ are effectively computable by a Turing machine which computes the entire infinity tree.

However, this paper does not claim there is an error in Cantor's arguments that $[0, 1]$ is uncountably infinite. Rather, this paper considers the situation as a paradox, resulting from different choices about how to represent and count the continuum of real numbers.

Acknowledgements

The writing of this paper has benefitted from correspondences with two anonymous mathematicians, who both gave criticisms and disagreed with the conclusions of this paper. I believe their criticisms have been addressed throughout this paper, although they continue to disagree.

The publication of [1] led to a brief correspondence with a philosopher who has written a separate unpublished paper [14] that describes an algorithm for iterative generation "in \aleph steps" of a list of all real numbers in $[0, 1]$, but does not discuss Turing machines *per se*, and does not discuss Cantor's Continuum Hypothesis or Cantor's Theorem.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Jackson, P.C. (2023) The Infinity Tree: Representing Infinities of Real Numbers with

- Countably Infinite Tree Structures. *Advances in Pure Mathematics*, **13**, 198-205. <https://doi.org/10.4236/apm.2023.134013>
- [2] Cantor, G. (1891) On an Elementary Question in the Theory of Manifolds. In: Ewald, W.B., Ed., *From Immanuel Kant to David Hilbert: A Source Book in the Foundations of Mathematics*, Vol. 2, Oxford University Press, 920-922.
- [3] Turing, A.M. (1937) On Computable Numbers, with an Application to the Entscheidungs Problem. *Proceedings of the London Mathematical Society*, **2**, 230-265. <https://doi.org/10.1112/plms/s2-42.1.230>
- [4] Hopcroft, J.E., Motwani, R. and Ullman, J.D. (2007) Introduction to Automata Theory, Languages, and Computation. 3rd Edition, Addison-Wesley.
- [5] Minsky, M.L. (1967) Computation: Finite and Infinite Machines. Prentice-Hall.
- [6] Stannett, M. (2006) The Case for Hypercomputation. *Applied Mathematics and Computation*, **178**, 8-24. <https://doi.org/10.1016/j.amc.2005.09.067>
- [7] Hamkins, J.D. (2002) Infinite Time Turing Machines. *Minds and Machines*, **12**, 521-539. <https://doi.org/10.1023/a:1021180801870>
- [8] Cantor, G. (1878) Ein Beitrag zur Mannigfaltigkeitslehre. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, **1878**, 242-258. <https://doi.org/10.1515/crll.1878.84.242>
- [9] Hilbert, D. (1902) Mathematical Problems. *Bulletin of the American Mathematical Society*, **8**, 437-479. <https://doi.org/10.1090/s0002-9904-1902-00923-3>
- [10] Gödel, K. (1940) The Consistency of the Continuum Hypothesis. Princeton University Press.
- [11] Cohen, P.J. (1963) The Independence of the Continuum Hypothesis. *Proceedings of the National Academy of Sciences*, **50**, 1143-1148. <https://doi.org/10.1073/pnas.50.6.1143>
- [12] Cantor, G. (1874) On a Property of the Set of All Real Algebraic Numbers. In: Ewald, W.B., Ed., *From Immanuel Kant to David Hilbert: A Source Book in the Foundations of Mathematics*, Vol. 2, Oxford University Press, 839-843.
- [13] Hodges, W. (1998) An Editor Recalls Some Hopeless Papers. *Bulletin of Symbolic Logic*, **4**, 1-16. <https://doi.org/10.2307/421003>
- [14] Styrman, A. (2023) A Constructive Proof That the Real Numbers Are Countable. Unpublished Paper.

Appendix A. Pseudocode for a Turing Machine

The following is pseudocode for a Turing machine which this paper argues would generate the real numbers in $[0,1]$ in countably infinite time and space.

create initial state

define tapes

'each tape has a countable infinity of cells

input level tape

output level tape

branch0tape, branch1tape, branch2tape, branch3tape,

branch4tape, branch5tape, branch6tape, branch7tape,

branch8tape, branch9tape

create / postulate initial starting state of tapes:

initially, all tapes have only blank spaces on every cell

position tape heads to start of all tapes

write 0.0, 0.1, 0.2, ..., 0.9, 1 to the input level tape

'*i.e.*, 11 finite expressions, each followed by a blank space on the tape.

'Commas and the ellipsis "..." would not be written to the tape.

'A cell on the tape would only have a digit, the decimal point, or a blank

'written to it.

position input level tape head to start of tape

loop

'an infinite loop to create expressions for paths from root to each level of infinity tree

for each expression on input level tape

'at each finite step there are a finite number of finite expressions,

'separated by spaces

if the first character of the expression is not "1" then

copy the expression to each branch tape

add 0 to end of expression on branch0tape

add 1 to end of expression on branch1tape

add 2 to end of expression on branch2tape

add 3 to end of expression on branch3tape

add 4 to end of expression on branch4tape

add 5 to end of expression on branch5tape

add 6 to end of expression on branch6tape
 add 7 to end of expression on branch7tape
 add 8 to end of expression on branch8tape
 add 9 to end of expression on branch9tape

position branch tape heads to start of branch tapes
 append copy of branch0tape expression to output tape
 append " " (empty space) to output tape
 append copy of branch1tape expression to output tape
 append " " (empty space) to output tape
 append copy of branch2tape expression to output tape
 append " " (empty space) to output tape
 append copy of branch3tape expression to output tape
 append " " (empty space) to output tape
 append copy of branch4tape expression to output tape
 append " " (empty space) to output tape
 append copy of branch5tape expression to output tape
 append " " (empty space) to output tape
 append copy of branch6tape expression to output tape
 append " " (empty space) to output tape
 append copy of branch7tape expression to output tape
 append " " (empty space) to output tape
 append copy of branch8tape expression to output tape
 append " " (empty space) to output tape
 append copy of branch9tape expression to output tape
 append " " (empty space) to output tape
 else 'expression is "1" on input level tape
 append "1" to output level tape 'signifying end of new expressions
 end if
 end for

position input level tape head to start of input level tape
 position output level tape head to start of output level tape

for each expression on output level tape 'a finite number of finite expressions
 copy expression to input level tape
 end for

'at this point, both tapes have the same content
 position input level tape head to start of input level tape
 position output level tape head to start of output level tape

end loop 'infinite loop creating expressions for paths from the root
 'to each level of the infinity tree