On a Representation of an ω-Language with Application to Cantor's Diagonal Argument

J. Mark Inman
(mark.inman@egs.edu)

Key Words:
foundations, computability, ω-languages, computational grammars, formal languages, Cantor's diagonal argument, representation theory, Abelian group
Category: F.4.1, F.4.2, F.4.3, I.1.3

Abstract: We provide a review of Cantor's Diagonal Argument by offering a representation of a recursive ω-language by a construction of a context sensitive grammar whose language of finite length strings through the defined operation of addition is an Abelian Group. We then generalize Cantor's Diagonal Argument as an argument function whose domain is found as a row in the matrix upon which it is constructed, thus providing a non-trivial counterexample to Cantor's Diagonal Argument under this generalization.

1 Introduction

1.1 The Need for Review of Cantor's Diagonal Argument

Cantor's Diagonal Argument (CDA) [2, 6] is a method for providing proof of the nature of infinitely sized sets and the limits of computation. It is foundational for set theory, the logic of computation and computability. From time to time, it is our duty as logicians to review our foundations and propose purposeful constructions for understanding these foundations.

1.2 Implications

Our results imply that the conclusions which derive from CDA are a consequence of the construction of representation peculiar to a case when presented in the current representational practice. Further implications include that the consequences of CDA are not a foundational or a universal argument for representation in computation, nor the entire domain of ordinary mathematics, the superstructure, SN. Since CDA loses generality through representation, we believe that all general statements which derive from CDA must be revised as qualified by being a consequence of the specific circumstance of representation. More general proofs on the theorems derived from CDA should be revisited based on the findings of this paper.

1.3 Related Research

The properties of infinite computations can trace it's roots to the work of Büchi, who first proposed ω-regular languages and automaton that accepts infinite length words [1]. A recent argument in favor of *Total Representations*, in a text by that name, was proposed in May of this year by Victor Selivanov [5]. In this text, he claims that representations in computable analysis are mostly partial, and unifies terminology "by elements of the Baire space." Ludwig Staiger has provided a tremendous body of work on ω-languages including a chapter on ω-languages published in the third volume of the *Handbook of Formal Languages* [7]. While much of our text was drafted prior to our reading of these texts, these texts were instrumental in unifying this work into its final form.

1.4 Method and Foundations

Much of the literature on ω-languages is written in terms of topology or argues in favor of a topological approach to studying the properties of ω-languages [3, 4, 5, 7]. This is because of the power of topology to describe infinitly sized strings as metric spaces. However, we have decided to not take a topological approach to ω-languages, but rather a more set-theoretical approach. This choice in method is because of the near ubiquitous reliance topology has on CDA and Cantor's First Proof of Uncountability (CFPU) over infinite spaces. We have chosen to work within the axioms of ZFC set theory and employing a portion of the techniques of the Von Neumann Hierarchy without explicitly limiting ourselves to that universe [8].

Conventional thought on CDA is that we would need an entirely new axiom schema to find a counterexample to the method [2]. However, no new axioms are needed to find the counterexample to CDA found in this paper. We present in this paper a new grammar for *computable representation* of an ω-language through current set theoretical foundations and the principles of formal languages. The grammar is novel, because its string set will be one of the few sets formalized, as a computable essentially non-contracting, context-sensitive grammar whose fixed recognition word problem can be solved with a sub-polynomial time deterministic algorithm (available upon request). To what extent the grammar is equivilent to or relates with tree-adjoining grammars, combinatory categorial grammars, or other context sensitive grammars is left as an open problem for further research.

We will not assume CDA is a method which produces theorems, nor rely on any theorems derived from either CDA or CFPU. We also exclude consequences of CFPU because of it's similarity to the method of CDA. This requires that we assume the cardinality of the Real numbers as it relates to the Natural numbers is unknown, since these proofs are the very foundation that prove the Reals are strictly larger than the Natural numbers. We do not make any claims about the cardinality of the Real numbers as they relate to the Natural numbers in this paper, and leave this question open for further research after a thorough review of our findings. We will instead re-explore CDA by creating a construction of an essentially non-contracting, context-sensitive grammar, and its generated set of well formed strings, explore its properties and expressive power as a language, construct an ω-language with mapped equivalencies between the strings in the sets, and finally use a subset of the union of these sets in a specific construction of CDA. It will be constructed such that when this subset of the union is utilized in CDA, the resulting string from the diagonal is equal to an element accounted for in this subset, and is thus, a non-trivial exception to CDA.

1.5 Notation

Let the Natural numbers, $\{1, 2, 3...\}$, be denoted by $\mathbb{N}$. Let ω be the rank of $\mathbb{N}$, from the Von Neumann Hierarchy. Let $\aleph_0$ be the cardinality of $\mathbb{N}$ and the cardinality of any set bijective to $\mathbb{N}$. Let a *symbol* be a monoid for the purpose of mathematical or computational representation. Let a *string* be some series of symbols and let the *asymptotic density* of a string be the frequency of the occurence of a symbol in a string. An *ω-expansion* is an expansion of free monoids on a string $Lim \rightarrow \omega$, and is denoted, for some free monoid, z, as z^{ω}. Define *ω-completion* as a mathematically induced output on the iterations of a sequence after unbounded iterations of this sequence such that this output is itself iterable and representable as such for some i, $0 < i < \omega$. Let *ω-density* be the asymptotic density of a string after ω-completion. Let *exhaustion* be a deductive and necessary change in output after ω-completion, where i at exhaustion is the $\omega^{th}+1$ iteration. Also, at the ω^{th} iteration, let i be equal to the symbol at the rank of the set of iterations. We denote CR as the set of computable real numbers which are those real numbers that can be calculated to arbitrary precision by some finite state Turing Machine. Let a *special order* of some set S be some ordering of S such that we may determine the value of any distinct element of S by some operation on S which guarantees the result is in S. Let an *asymptotic ω-expansion* of a series of symbols in a string be a recursive expansion of the symbols such that the $Lim \rightarrow \omega$ has the same ω-density as the original string.

Let $\sum$ be a set of terminal symbols which is its alphabet. Let V be a set of non-terminal starting symbols and R be a set of rules, each in the form $x\mathrm{A}x \longrightarrow xw$ for some strings $w \in \sum$, $x \in \mathrm{V} \bigcup \sum$ when $\mathrm{A} \in \mathrm{V}$.

Let $\sum$ be an alphabet with at least 6 elements and the empty string ϵ, let $\sum_1^{\infty}$ be the set of all strings over $\sum$ which includes all ω length strings and let $\sum_2^{\infty}$ be the set of strings which also include strings of length ω over a subset $\sum_2$ of at least two elements in $\sum$. Let $\sum_3^{\omega}$ be the subset of $\sum_2^{\infty}$ which are only the recursive, countably infinite length strings over $\sum_2$ so that the set of all symbols in each string has a cardinality $\aleph_0$ and iterable. A language of the class of computable transfinite number **T** has a subset of unique finite length strings $\sum_4^*$ of $\sum_1^\infty$ that is either bijective or surjective (but not injective) to the set $\sum_3^\omega$. The alphabet $\sum_4$ must also have a subset of at least three elements not a member of $\sum_2$ with at least one of these symbols to distinguish context between other symbols of value.

Remark. Note that the total cardinality of the set of strings in $\sum_3^\omega$ is $\aleph_0^2$ and fully representable by an $\sum_4^*$ mapping via Cantor pairing between the sets.

Let $\sum_2^*$ be the set of finite length strings over $\sum_2$. For any language L in the class **T**, the set of strings $\sum^*$ over L, $\sum^* = \sum_4^* \bigcup \sum_2^*$.

2 Representing an ω-language by a Computable Surjection

2.1 A grammar for a language in T

Definition 1. Let the grammar L_T(G) be a tuple, {V, $\sum$, R, S}. S $\in$ V, $\sum_2 \subset \sum$, $\sum_3 \subset \sum$, $\sum_4 \subset \sum$, such that:

V $\bigcap \sum = \emptyset$,
V={S, A, B, C, Q, U},
$\sum$={ ε, 0, 1, ***0***, ***1***, [,] },
$\sum_2$={0,1},
$\sum_3$={0,1},
$\sum_4$={ ε, 0, 1, ***0***, ***1***, [,] },

R={S$\longrightarrow$ ε, S$\longrightarrow$A, S$\longrightarrow$B, S$\longrightarrow$C, S$\longrightarrow$Q, A$\longrightarrow$1, A$\longrightarrow$0, A$\longrightarrow$0A, A$\longrightarrow$1A, A$\longrightarrow$0C, A$\longrightarrow$1B, B$\longrightarrow$0, B$\longrightarrow$0A, B$\longrightarrow$***0***, B$\longrightarrow$***0***C, C$\longrightarrow$1, C$\longrightarrow$1A, C$\longrightarrow$***1***B, C$\longrightarrow$***1***, Q$\longrightarrow$[Q], Q$\longrightarrow$[U], Q$\longrightarrow$S[U], Q$\longrightarrow$[U]S, Q$\longrightarrow$S[U]S[U]S, U$\longrightarrow$01B, U$\longrightarrow$10C, U$\longrightarrow$0A, U$\longrightarrow$1A, U$\longrightarrow$***0***C, U$\longrightarrow$***1***B}

Definition 2. Let L_T be a language of the elements in $\sum_4^*$ which generate only from grammars generated by the non-terminals B, C, Q and U that cannot be generated exclusively by the non-terminal A. Let the elements of $\sum_4^*$ have an onto mapping to the recursive strings of infinite length in $\sum_3^\omega$ (i.e. those strings from $\sum_3^\omega$ which are readable by Büchi automaton).

2.2 ρ^*

Definition 3. We define ρ^* as the set of all well formed strings over L_T.

2.3 The Cardinality of ρ^*

Let CR be the set of computable Real numbers.
The Grammar $L(CR)$ is a tuple, {V, Σ, R_r, S}. S $\in$ V.
V $\bigcap$ $\Sigma = \emptyset$
V={S, A}
$\sum$={0,1}
R_r={S$\longrightarrow$A, A$\longrightarrow$0A, A$\longrightarrow$1A, A$\longrightarrow$1, A$\longrightarrow$0}.

Remark. Any computable binary number inclusive the open interval (0,1) can be generated using the grammar $L(CR)$ by construction of $L(CR)$.

Proposition 4. *$\rho^* \geq CR$. Let $\sum^*_{L(CR)}$ be the set $\sum^*$ in the grammar $L(CR)$. Let $\sum^*_{L_T(G)}$ be the set $\sum^*$ in the grammar $L_T(G)$. Given a bijection between $\mathbb{R}$ and the open interval (0,1), (0,1) has the same cardinality as $\mathbb{R}$, all computable elements of (0,1) are in $\sum^*_{L(CR)}$ by construction of $L(CR)$, and $\sum^*_{L(CR)} \subset \rho^*$ by the grammar construction of ρ^* which contains the rule subset r of R, which has at least the expressive power of R_r. $\sum^*_{L(CR)} \subset \sum^*_{L_T(G)}$ therefore $\rho^* \geq CR$.* $\square$

3 ρ^* is an Abelian Group

3.1 Addition in ρ^*

Remark. To correctly define addition, +, in ρ^*, we will first define several functions which will generate the rules of operation. These rules, when applied to some string, will allow us to properly define a sum between two strings, $w_1 + w_2$.

Definition 5. LENGTH(x) is a function which outputs the length of a string within brackets.

Definition 6. PLUS1(w) $= w + 1 = 1 + w$ and follows these rules:

PLUS1(w0) := w1
PLUS1(w1) := PLUS1(w)0
PLUS1($w\boldsymbol{1}$) := PLUS1(w)$\boldsymbol{0}$
PLUS1($w\boldsymbol{0}$) := $w\boldsymbol{0}$1
PLUS1($w[x0]$) := $w[x0]1$
PLUS1($w[x\boldsymbol{1}]$) := $w[x\boldsymbol{1}]$PLUS1(x)0
PLUS1($w[x1]$) := $w[x1]$PLUS1(x)$\boldsymbol{0}$
PLUS1(ε)w := Halt

Definition 7. PLUS$\boldsymbol{1}$( ): PLUS$\boldsymbol{1}(w) = w + \boldsymbol{1} = \boldsymbol{1} + w$ and follows these rules:

PLUS$\boldsymbol{1}(w0) :=$ PLUS$\boldsymbol{1}(w)1$
PLUS$\boldsymbol{1}(w1) :=$ PLUS$\boldsymbol{1}$(PLUS1$(w))0$
PLUS$\boldsymbol{1}(0w) := 1$ PLUS$\boldsymbol{1}(w)$
PLUS$\boldsymbol{1}(1w) := 0$ PLUS$\boldsymbol{1}(w)$
PLUS$\boldsymbol{1}(w\boldsymbol{1}) :=$ PLUS$\boldsymbol{1}(w)\boldsymbol{1}0$
PLUS$\boldsymbol{1}(w\boldsymbol{0}) :=$ PLUS$\boldsymbol{1}(w)\boldsymbol{1}$
PLUS$\boldsymbol{1}(w[x0]) :=$ PLUS$\boldsymbol{1}(w)$[PLUS$\boldsymbol{1}(x)1$]
PLUS$\boldsymbol{1}(w[x1]) :=$ PLUS$\boldsymbol{1}(w)$[PLUS$\boldsymbol{1}$(PLUS1$(x))0$]
PLUS$\boldsymbol{1}(w[1x]) :=$ PLUS$\boldsymbol{1}$(PLUS1(w))[PLUS$\boldsymbol{1}(1x)$]
PLUS$\boldsymbol{1}(w[0x]) :=$ PLUS$\boldsymbol{1}(w)$[1 PLUS$\boldsymbol{1}(x)$]
PLUS$\boldsymbol{1}(w[x\boldsymbol{1}]) :=$ PLUS$\boldsymbol{1}(w)$[PLUS$\boldsymbol{1}(x)\boldsymbol{1}0$]
PLUS$\boldsymbol{1}(\varepsilon)w :=$ Halt

Definition 8. For all strings w_1, x_1, w_2, x_2, w_3, w_4 when LENGTH(x_1) = LENGTH(x_2) and given a concatenation function between two strings, AND such that 'aa' AND 'bb'='aabb'; w_1 AND $[x_1]$=w_3; w_2 AND $[x_2]$=w_4, PLUS(w_3, w_4) $= w_3 + w_4 = w_4 + w_3$:

PLUS($w_1 0$, $w_2 0$) := PLUS(w_1, w_2)0
PLUS($w_1 1$, $w_2 0$) := PLUS(w_1, w_2)1
PLUS($w_1 0$, $w_2 1$) := PLUS(w_1, w_2)1
PLUS($w_1 1$, $w_2 1$) := PLUS1(PLUS(w_1, w_2))0
PLUS($w_1\boldsymbol{1}$, $w_2\boldsymbol{0}$) := PLUS(w_1, w_2)$\boldsymbol{1}$
PLUS($w_1\boldsymbol{0}$, $w_2\boldsymbol{1}$) := PLUS(w_1, w_2)$\boldsymbol{1}$
PLUS($w_1\boldsymbol{1}$, $w_2\boldsymbol{1}$) := PLUS1(w_1, w_2))$\boldsymbol{0}$1
PLUS($w_1\boldsymbol{0}$, $w_2\boldsymbol{0}$) := PLUS(w_1, w_2)$\boldsymbol{0}$
PLUS($w_1\boldsymbol{1}$, $w_2 1$) := PLUS(w_1, PLUS$\boldsymbol{1}$(w_2))0
PLUS($w_1\boldsymbol{1}$, $w_2 0$) := PLUS(w_1, PLUS$\boldsymbol{1}$(w_2))1
PLUS($w_1\boldsymbol{0}$, $w_2 0$) := PLUS($w_1\boldsymbol{0}$, w_2)0
PLUS($w_1\boldsymbol{0}$, $w_2 1$) := PLUS($w_1\boldsymbol{0}$,w_2)1

PLUS($w_1[x0]$, $w_2 0$) := PLUS($w_1[0x]$, w_2)0
PLUS($w_1[x0]$, $w_2 1$) := PLUS($w_1[0x]$, w_2)1
PLUS($w_1[x1]$, $w_2 0$) := PLUS($w_1[1x]$, w_2)1
PLUS($w_1[x1]$, $w_2 1$) := PLUS(PLUS1($w_1[0x]$), w_2)0
PLUS($w_1[x0]$, $w_2\boldsymbol{1}$) := PLUS(PLUS$\boldsymbol{1}$(w_1), w_2)[PLUS$\boldsymbol{1}(x)1$]
PLUS($w_1[x1]$, $w_2\boldsymbol{1}$) := PLUS(PLUS$\boldsymbol{1}$(w_1), w_2)[PLUS$\boldsymbol{1}(x)0$]
PLUS($w_1[x\boldsymbol{0}]$, $w_2\boldsymbol{1}$) := PLUS(PLUS$\boldsymbol{1}$(w_1), w_2)[PLUS$\boldsymbol{1}$(x)$\boldsymbol{1}$]
PLUS($w_1[x\boldsymbol{1}]$, $w_2\boldsymbol{1}$) := PLUS(PLUS$\boldsymbol{1}$(w_1), w_2)[PLUS1(x)$\boldsymbol{1}$0]

PLUS($w_1[x\boldsymbol{1}]$, $w_2\boldsymbol{0}$) := PLUS($w_1[\boldsymbol{1}x$, $w_2)\boldsymbol{1}]$
PLUS($w_1[x\boldsymbol{0}]$, $w_2\boldsymbol{0}$) := PLUS(w_1, w_2)$[x\boldsymbol{0}]$

Definition 9. To define P*, let us suppose Kleene star expansions of strings over ρ^*: '(w_1)*([(w_2)*])*(w_3)*' and an expansion rule such that when there exists the terminal ***0*** in a string, there can be a Kleene star expansion of '0' either to the left or the right or both of ***0*** such that '0****0***0*' is a valid string and similarly for ***1*** such that '1****1***1*' is a valid string. P* is the union of ρ^* and all Kleene star expansions in ρ^*.

Definition 10. Let P$^{\backprime}$ be all strings in P* not in ρ^*.

Definition 11. Let P$^\omega$ be all the strings in P union all ω-expansions and Kleene star expansions of symbols in P*.

Definition 12. LENDIFF( ): For all x in strings w_3, w_4:
LENDIFF(LENGTH(x_1)*LENGTH(x_2)) := LENGTH(x_y) where x_y is a concatenation of x_1 LENGTH(x_y) times.
LENDIFF(LENGTH(x_1)*LENGTH(x_2)) := LENGTH(x_z) where x_z is a concatenation of x_2 LENGTH(x_z) times.

When LENGTH(x_1) != LENGTH(x_2):

Definition 13. PLUS($w_1[x_y]$, $w_2[x_z]$) := PLUS(w_3, w_4)

Remark. Addition is Commutative: By the definition of the PLUS function, addition is commutative in ρ^*; the equation $a+b=b+a$ holds for all a, b in ρ^*.

Remark. Let addition be associative in ρ^* such that the equation $(a+b)+c = a+(b+c)$ holds for all a, b and c in ρ^*.

Remark. The identity element exists as 0, through the following iteration:

PLUS(w, 0)=w

Definition 14. Let *String Equality* be the condition where expanded strings from a language, either by Kleene star expansions or ω-expansions, have equality with strings from which they expanded. Let w_1 be a string from ρ^* and let w_2 be a string expanded from w_1; $w_2 \in$ P$^\omega$. Such that:

PLUS(w_1, 0) := PLUS(w_2, 0)

Proposition 15. *Addition in ρ^* is closed. PLUS( ) will only yield answers in ρ^* or P' by the iterations of PLUS( ). For any strings $w_1 \in \rho^*$, $w_2 \in \rho^*$, , PLUS(w_1, w_2):=w_a, if $w_a \in$ P', there exists a String Equality in ρ^*, such that $w_a = w$ where w is in ρ^* and PLUS(w_1, w_2):=w through reflexivity. If $w_a \notin$ P', it must be in ρ^*. Therefore, addition is closed in ρ^*.*

Proposition 16. *There is an additive inverse for each $w \in \rho^*$. For every w_a, there exists some w_b through iteration in PLUS(w_a, w_b) such that the output string will be some combination of terminals 0* and **0***, which through String Equality=0.*

Lemma 17. *ρ^* is an Abelian Group under addition. From the previous two propositions, and because or definition of PLUS( ) includes commutitivity, by the definition of an Abelian Group as having closure, associativity, an identity element, an inverse element and commutativity, ρ^* is an Abelian Group under addition.* $\square$

4 Cantor's Diagonal Argument Yields a Non-trivial Exception in $\mathbf{P_0^\omega}$

4.1 Cantor's Argument Function

For Cantor's Diagonal Argument, one may generalize an *argument function*, for proof by contradiction, through constructing an arbitrary sequence of infinite length strings in any order into an $\mathbb{N}$ by $\mathbb{N}$ matrix of symbols and assuming all rows of the matrix contains all string values. Then, one proceeds to the i^{th}column, j^{th} row of the matrix, where $(i,\ j)$ moves along the diagonal of the matrix- (1,1), (2,2), (3,3)... etc. The symbol at that position is then changed to a different symbol within the language of the system and concatenated to produce a new string, which when so constructed, the new string of symbols can't be found in any row or column of the matrix, thus providing a string of a value not listed in the matrix. The new string is considered transcendental in Cantor's Universe and the proof by contradiction tells us that not every string value can be calculated in Cantor's Universe through iterative process, and as such, $\mathbb{R}$ is "uncountable" and has a cardinality strictly larger than $\mathbb{N}$. This is necessarily true for all previous constructions of the diagonal argument because the i^{th} string must contain the symbol found at $(i,\ j)$ which can not be in the constructed string at $(i,\ j)$ from the diagonal and there did not yet exist a mapping relationship suitable for exception to CDA. [6]

The following construction maps the set ρ^* with equivalent strings in P^ω, showing that the diagonal argument applied to a special ordering of subsets in P^ω, yields a non-trivial exception to Cantor's argument function.

Remark. It should be noted that because the elements of P^ω are not the Real numbers, and thus, the trivial exception where $.\bar{9}$=1 is not a concern because this evaluation or any equivalent evaluation does not exist in P^ω by construction and this exception simply cannot occur. We will not concern ourselves with this exception, since it is trivial.

4.2 Additional Preliminaries and Terminology

Consider an argument function $f(S)$ over a set S yielding a domain $f(S) \to g(S)$. And some countable set of matrices reducible via Cantor Pairing to size $\mathbb{N} \times \mathbb{N}$, $\{f(S)\}$ to which we will apply $f(S)$.

A *counterexample* to an argument function, which is a proof by contradiction, is the condition when there exists $f(S) \rightarrow g(S)$ such that for any $g(S) \in \{f(S)\}$ iff $g(Q) \notin \{f(Q)\}$ OR if any $g(S) \notin \{f(S)\}$ iff $g(Q) \in \{f(Q)\}$ for sets S and Q.

Let *right-concatenation* be the concatination operator * such that when w_1 right concatenates over w_2, $w_1{*}w_2{=}w_2w_1$

Let $str(f(\,x_j\,))$ be the cumulative string of the output of the function $f(x_j)$ right-concatenated over $f(x_{j-1})$, right-concatenated over $f(x_{j-2})$... right-concatenated over $f(x_{j-j})$. Similarly for $str(g(\,x_j\,))$.

Let the argument function $f(\mathbb{R})$ be a construction of Cantor's diagonal method on the $\mathbb{N} \times \mathbb{N}$ matrix $\{f(\mathbb{R})\}$ of arbitrary unbounded binary strings. They are listed in any order, assuming that all possible string values of $\mathbb{R}$ are listed. Let $g(\mathbb{R}){=}str(g(x_j))$ where x_j is the value, 0 or 1 at the coordinate (i,j) for all j on the constructed matrix at row j such that $j{=}i$. It is well established that in Cantor's Universe, $g(\mathbb{R}) \notin \{f(\mathbb{R})\}$, and we can produce a string which is in $\mathbb{R}$ but not in $\{f(\mathbb{R})\}$.

For counterexample, since we are given that $g(\mathbb{R}) \notin \{f(\mathbb{R})\}$, we must find some set S where $g(\mathrm{S}) \in \{f(\mathrm{S})\}$, let the argument function $f(\mathrm{P}_0^\omega)$ be an equivalent construction of Cantor's diagonal method where a series of $\mathbb{N} \times \mathbb{N}$ matrices, $\{f(\mathrm{P}_0^\omega)\}$, is partitioned into two specially ordered matrices, $\{f(\mathrm{P}_1^\omega)\}$ and $\{f(\mathrm{P}_2^\omega)\}$. $\{f(\mathrm{P}_1^\omega)\} \subset \{f(\mathrm{P}_0^\omega)\}$ and $\{f(\mathrm{P}_2^\omega)\} \subset \{f(\mathrm{P}_0^\omega)\}$. These two partitions will be proper disjoint subsets of $\{f(\mathrm{P}_0^\omega)\}$ such that $\{f(\mathrm{P}_1^\omega)\} \bigcup \{f(\mathrm{P}_2^\omega)\}{=}\{f(\mathrm{P}_0^\omega)\}$.

Construct for CDA, $\{f(\mathrm{P}_0^\omega)\}$, in the following manner: Let ρ^* map to two sets, ρ_1^* and ρ_2^*, through choice, choosing the strings in ρ_1^* as all the strings where each string is denoted by the initial with Kleene star expansions, from left to right as follows: $0w$, $\boldsymbol{0}w$, $[^*0w$ or $[^*\boldsymbol{0}w$. And likewise, choose the strings in ρ_2^* as all the strings where each string is denoted by the initial, from left to right, $1w$,

$\boldsymbol{1}w$, $[^*1w$ or $[^*\boldsymbol{1}w$, thus creating a strictly bijective disjunction between the sets whose union is ρ^*. Choose some string set in ρ_1^* such that it maps to an infinite length string by String Equality in P^ω to form the set P_1^ω. Choose some string set in ρ_2^* such that it maps to some infinite length string by String Equality in P^ω to form the set P_2^ω. Let the union of P_1^ω and P_2^ω be P_0^ω. We will begin the proof of the existence of the counterexample to CDA by proceding with a construction of $f(\mathrm{P}_1^\omega)$ and $f(\mathrm{P}_2^\omega)$ individually, and then providing a retrograde on $str(g(\mathrm{P}_2^\omega))$ to construct the diagonal of $\{f(\mathrm{P}_0^\omega)\}$.

4.3

Theorem 18. *There is a Non-trivial Counterexample to Cantor's Diagonal Argument.*

Proof. //

1. $\rho*$ is an Abelian group by Lemma.
2. By String Equality we can create an Abelian Relationship in all extended sets related to ρ^* by providing an operation on a string in ρ^* to yield another result in ρ^*, which can then be made equal to some string in P^ω and its subsets, thus allowing for special order, utilizing exhaustion, if necessary.
3. We will choose the elements of P_1^ω and P_2^ω in such a way, that when proceding with the argument function down the diagonal of the respected matrices, $\{f(\mathrm{P}_1^\omega)\}$ and $\{f(\mathrm{P}_2^\omega)\}$, we will encounter only the symbols 0 and 1, respectively. This can be ensured by expanding other symbols when necessary. We will reserve the axiom of replacement if necessary when forming the sets to aid in our choice for the rows in the respective matrix sets.
4. Proceeding to $f(\mathrm{P}_1^\omega)$, we choose to write the row, $j = 1$, as some $0\boldsymbol{1}$ as it's defined mapped string, $0\boldsymbol{1}1^\omega$ and by the argument function, utilize the symbol from the 1^{st} column of this row such that $f(x_1)$:=0 and $str(g(x_1))$:=1.
5. Choose in the set such that the second row in $\{f(\mathrm{P}_1^\omega)\}$ is $\boldsymbol{01}$=$\boldsymbol{0}0^\omega 1^\omega \boldsymbol{1}$ and $f(x_2)$:=0, and $str(g(x_2))$:=11 in accordance with the argument function.
6. Thus, through ω-completion of the iterable iterations resulting in an unbounded series, for all j in $\{f(\mathrm{P}_1^\omega)\}$, $f(x_j)$:=0; $g(x_j)$:=1.
7. It follows through String Equality that $g(\mathrm{P}_1^\omega)$=$\boldsymbol{1}$.
8. Likewise, proceeding to $f(\mathrm{P}_2^\omega)$, we choose to write the row, $j = 1$, $1\boldsymbol{0}$ as it's defined mapped string as some $1\boldsymbol{0}0^\omega$ and by the argument function, utilize the symbol from the 1^{st} column of this row such that $f(x_1)$:=1 and $str(g(x_1))$:=0.

9. Choose in the set such that the second row in $\{f(\mathrm{P}_2^\omega)\}$ is $\boldsymbol{10}{=}\boldsymbol{1}1^\omega 0^\omega \boldsymbol{0}$ and $f(x_2){:=}1$, and $str(g(x_2)){:=}00$ in accordance with the argument function.

10. Thus, through ω-completion of the iterable iterations resulting in an unbounded series, for all j in $\{f(\mathrm{P}_2^\omega)\}$, $f(x_j){:=}1$; $g(x_j){:=}0$.

11. It follows through String Equality that $g(\mathrm{P}_2^\omega){=}\boldsymbol{0}$.

12. The retrograde of $g(\mathrm{P}_2^\omega){=}\boldsymbol{0}$ is reflexive, $\boldsymbol{0}$.

13. The special order of $\{f(\mathrm{P}_0^\omega)\}$ is the special order of $\{f(\mathrm{P}_1^\omega)\}$ union the retrograde of the special order of $\{f(\mathrm{P}_2^\omega)\}$, allowing for an inversion on the strings in the diagonal, because of retrograde, for consistency in the argument function.

14. Through String Equality, $g(\mathrm{P}_0^\omega){=}\boldsymbol{10}$

15. By choice of $\{f(\mathrm{P}_2^\omega)\}$ which is a subset of $\{f(\mathrm{P}_0^\omega)\}$, as illustrated by step 9 in this proof, the string $\boldsymbol{10}{=}\boldsymbol{1}1^\omega 0^\omega \boldsymbol{0}$ is in $\{f(\mathrm{P}_0^\omega)\}$.

16. Therefore, there exists a counterexample; when $g(\mathbb{R}) \notin \mathbb{R}$, $g(\mathrm{P}_0^\omega) \in \mathrm{P}_0^\omega$. ■

5 Discussion

Finding a model for a non-trivial exception to CDA provides a new understanding about the limits of computability. We would like to call on computer scientists, complexity theorists and physicists to contact the authors without hesitation if they would like to collaborate on projects related to the grammar set introduced in this paper and it's consequences on complexity, computability and the mathematics of physics. The authors believe that this finding has widespread foundational implications across many disciplines.

In the universe of this paper, we had to invoke the Axiom of Choice regularly. And even though we utilized rank, a procedure fom the Von Neumann Universe, it was economical to construct a set which has decreasing infinities and thus does not belong to NBG or the Von Neumann universe. The model presented here will hopefully lead to technological advances in computer science, complexity theory and physics as a new foundation for computation.

Upon publication of this article, we will procede to complete a subsequent installment of this work which will expand ρ^* from an Abelian group into a Ring through multiplication rules we have developed over ρ^* (but not included in this paper for the sake of simplicity), from which, our hypothesis is, we may find a computable field. It is our hope and ambition to connect directly with physicists who would like access to this computable field to build operators which describe quantum relationships which otherwise, before the creation of such a field, would difficult or impossible to compute.

References

1. Büchi, J.R., On a decision method in restricted second order arithmetic. *Proc. International Congress on Logic, Method, and Philosophy of Science. 1960*, pages 112, Stanford, CA, Stanford University Press. 1962
2. Lewis, Harry *Introduction to Formal Systems and Computation CS121* Harvard College December 2007 http://www.seas.harvard.edu/courses/cs121
3. Perrin, D. and Pin, J-E. *Infinite Words: Automata, Semigroups, Logic and Games* Pure and Applied Mathematics Vol 141, Elsevier, 2004.
4. Thomas, Wolfgang *Automata on Infinite Objects.* Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, pages 133-192. Elsevier Science Publishers, Amsterdam, 1990.
5. Selivanov, Victor *Total Representations* arXiv:1304.1239v2 [cs.LO]
6. Sipser, M. *Introduction to the Theory of Computation, 3rd ed.*, pages 202-209, Cengage Learning, Boston 2013
7. Staiger, L., *ω-Languages* G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, Volume 3, pages 339-387. Springer-Verlag, Berlin, 1997.
8. Suppes, Patrick *Axiomatic Set Theory* Mineola, NY: Dover Publications, Inc. 1972