An overview of the ordinal calculator

Paul Budnik

March 24, 2011

Abstract

An ordinal calculator has been developed as an aid for understanding the countable ordinal hierarchy and as a research tool that may eventually help to expand it. A GPL licensed version is available in C++ code and as an interactive command line calculator. It includes ordinals uniquely expressible in Cantor normal form, the Veblen hierarchies and a form of ordinal projection from countable admissible ordinal hierarchies to themselves. This projection is based on a weakened version of notations for recursive ordinals that can be applied to countable admissible ordinals. Iterative projections using this notation may be one key to expanding the ordinal hierarchy. These ideas have their roots in a philosophy of mathematical truth that sees *objectively true* mathematics as connected to recursive processes. It suggests that computers are an essential adjunct to human intuition for extending the combinatorially complex parts of objective mathematics.

Contents

List of Figures

List of Tables

Introduction

An ordinal calculator has been developed as an aid for understanding the countable ordinal hierarchy and as a research tool that may eventually help to expand it. A GPL licensed version is available in C++ code and as an interactive command line calculator. It includes ordinals uniquely expressible in Cantor normal form (that are $< \varepsilon_0$), the Veblen hierarchies and a form of ordinal projection from countable admissible ordinal hierarchies to themselves. This projection is based on a weakened version of notations for recursive ordinals that can be applied to countable admissible ordinals. Iterative projections using this notation may be one key to expanding the ordinal hierarchy. The well orderings that can be generated in this weakened form of recursive ordinal notations will always be recursive and incomplete. They can be projected onto themselves in complex ways that is somewhat reminiscent of the Mandelbrot set[9].

Loosely speaking there are two dimensions to the power of axiomatic mathematical systems: definability and provability. The former measures what structures can be defined and the latter what statements about these structures are provable. Provability is usually expandable by expanding definability, but there are other ways to expand provability. In arguing for the necessity of large cardinal axioms a number of arithmetic statements have been shown to require such axioms to decide[6]. This claim is relative to the linear ranking of generally accepted axiom systems. However any arithmetic (or even hyperarithmetic) statement can be decided by adding to second order arithmetic a finite set of axioms that say certain integers do or do not define notations for recursive ordinals in the sense of Kleene's $\mathcal{O}$[8]. This follows because Kleene's $\mathcal{O}$ is a Π^1_1 complete set[13] and a TM with an oracle that makes a decision must do so after a finite number of queries.

Large cardinal axioms are needed to decide these statements because it has not been possible to construct a sufficiently powerful axiom system about notations for recursive ordinals. This can change. Any claim that large cardinal axioms are needed to decide arithmetic statements is relative to the current state of mathematics.

Large cardinal axioms seem to implicitly define large recursive ordinals that may be beyond the ability of the unaided human mind to define explicitly. Thus the central motivation of this work is to use the computer as a research tool to augment human intuition with the enormous combinatorial power of today's computers.

There is the outline of a theory of objective mathematical truth that underlies this approach in Section 4. This theory sees objective mathematics as logically determined by a recursive enumerable sequence of events. (The relationship between thee events may be complex[1], but these events, by themselves, must decide the statement.) Objective mathematics includes arithmetic and hyperarithmetic statements and some statements requiring quantification over the reals.

1 Ordinal notations and the Cantor normal form

An ordinal notation system assigns strings to an initial fragment of the recursive ordinals. It contains a recursive process for deciding the relative size ($<, =$, or $>$) of the ordinals represented by each string and a recursive process for deciding if a given string represents an ordinal. Section 3 describes a weakened version of this ordinal notation system for the Church-Kleene ordinal (the ordinal of the recursive ordinals) and larger countable ordinals.

Greek letters represent both notations, the finite strings that represent ordinals, and the ordinals themselves. The relative size of notations is the relative size of the ordinals they represent.

The ordinal calculator uses `C++` `class`[2] `Ordinal` and `virtual` member functions[3]. to implement ordinal notations in an expandable way. Thus the routine that determines the relative size of the ordinals represented by two strings, `compare`, is a `virtual` function that can continue to work correctly on an expanded hierarchy built on top of existing `class`es. The ordinal calculator uses a parser, lexical analyzer and a semantic checker to define valid ordinal notations. It uses the `static`[4] function `fixedPoint` to convert notations to a unique representation.

From the above and for any notation α in the system, one can enumerate all notations $< \alpha$. However this is too inefficient to be practical. Thus there is a `virtual Ordinal` member function `limitElement` on the integers that outputs an increasing sequence of ordinal

[1] The valid relationships cannot be precisely defined without limiting the definition beyond what is intended. An objective formal system can always be extended in both provability and definability.

[2] Constructs from the programming language `C++` in which the ordinal calculator is written, will be given in `tty` font.

[3] In `C++` a member function is a routine with access to the internal structure of the specific `class` instance (such as that defining an ordinal notation) from which it is called. A `virtual` member function can be overridden by a function with the same name and parameters in a sub`class` of the base `class`. This means calling the function from an object of this `class` will always invoke the highest level `virtual` function that is defined for this object even if the object in context is only declared to be in the base `class`.

[4] A `static` function is a `class` member function that can be called without a specific class instance. `fixedPoint` must be defined for each new sub`class`.

notations. The union of the ordinals represented by the outputs of α.`limitElement`[5] is the ordinal represented by α.

1.1 Cantor normal form

Every ordinal, α, can be represented as shown in Equation 1.

$$\alpha_1 > \alpha_2 > \alpha_3 > ... > \alpha_k$$

The α_k are ordinal notations and the n_k are integers > 0.

$$\alpha = \omega^{\alpha_1} n_1 + \omega^{\alpha_2} n_2 + \omega^{\alpha_3} n_3 + ... + \omega^{\alpha_k} n_k \tag{1}$$

Because $\varepsilon_0 = \omega^{\varepsilon_0}$, the Cantor normal form gives unique representation only for ordinals $< \varepsilon_0$. Each term in Equation 1 is represented by a member of `class CantorNormalElement`. The terms are linked in decreasing order in `class Ordinal`. This base `class` can represent any ordinal $< \varepsilon_0$. The integers used to define finite ordinals[6] are scanned and processed with a library that supports arbitrarily large integers[7]. The `Ordinal` instance representing ω is predefined as the variable `omega`. Larger ordinals in the base class are constructed using the integers, ω and three ordinal operators, $+, \times$ and exponentiation. $\varepsilon_0 = \bigcup \omega, \omega^\omega, \omega^{\omega^\omega} \omega^{\omega^{\omega^\omega}} ...$ and thus is the smallest ordinal not constructable with these operators and predefined values.

1.2 Interactive mode

The ordinal calculator has a command line interactive mode that supports most functions without requiring C++ coding. In this mode one can write ordinal expressions directly using the symbols '`*`' for multiply and '`^`' for exponentiation. These expressions can be assigned to variable names. `omega` is defined by the single character '`w`' as well as `omega`. To list α.`limitElement(1)` through α.`limitElement(n)` use the interactive command α.`listElts(n)`.

2 The Veblen hierarchy

The Veblen hierarchy[16, 10, 7] extends the Cantor normal form by defining functions that grow much faster than ordinal exponentiation and using these to define notations for ordinals much larger than ε_0. It is developed in two stages. The first involves expressions of a fixed finite number of variables. The second involves functions definable as limits of sequences of

[5]Sometimes mathematical notation is combined with C++ code. In this example the C++ definition of a member function is combined with a Greek letter to represent the C++ object (an `Ordinal` notation) that the subroutine is called from.

[6]The syntax for defining the ordinal 12 named '`a`' is '`Ordinal a(12);`' in C++ and '`a=12`' in the interactive ordinal calculator.

[7]The package used is MPIR, (Multiple Precision Integers and Rationals) based on the package GMP (GNU Multiple Precision Arithmetic Library). Either package can be used, but only MPIR is supported on Microsoft operating systems.

$\varphi(0, \alpha_2)$	$= \omega^{\alpha_2}$
$\alpha 1$ and $\alpha 2$ are successors	
$\varphi^0(\alpha_1 + 1, \alpha_2 + 1)$	$= \varphi(\alpha_1 + 1, \alpha_2)$
$\varphi^{i+1}(\alpha_1 + 1, \alpha_2 + 1)$	$= \varphi(\alpha_1, \varphi^i(\alpha_1 + 1, \alpha_2 + 1)) + 1$
thus using $\varphi^i(\alpha_1, \alpha_2)$ thus	
$\varphi(\alpha_1 + 1, \alpha_2 + 1)$	$= \bigcup_{i \in N} \varphi^i(\alpha_1 + 1, \alpha_2 + 1)$
alternatively	
$\varphi(\alpha_1 + 1, \alpha_2 + 1)$	$= \bigcup \varphi(\alpha_1 + 1, \alpha_2) + 1,\ \varphi(\alpha_1, \varphi(\alpha_1 + 1, \alpha_2) + 1) + 1,$
	$\varphi(\alpha_1, \varphi(\alpha_1, \varphi(\alpha_1 + 1, \alpha_2) + 1) + 1) + 1, ...$
α_2 is a limit	
$\varphi(\alpha_1, \alpha_2)$	$= \bigcup_{\beta \in \alpha_2} \varphi(\alpha_1, \beta)$
α_1 is a limit, α_2 is a successor	
$\varphi(\alpha_1, \alpha_2 + 1)$	$= \bigcup_{\beta \in \alpha_1} \varphi(\beta, \varphi(\alpha_1, \alpha_2))$
See Table 2 for examples.	

Table 1: Two parameter Veblen function definition

functions of an increasing number of variables. The idea is to give an inductive definition of the largest possible ordinal at each stage, using what has previously been defined including the limit of infinite sequences of previously defined expressions.

2.1 Finite parameter Veblen functions

The Veblen hierarchy starts with a function of two variables, $\varphi(\alpha_1, \alpha_2)$, based on ω^α. $\varphi(0, \alpha_2)$ is defined as ω^{α_2}. $\varphi(1, \alpha_2)$ is defined as the α_2 fixed point of ω^α and is written as ε_{α_2}. Loosely speaking the Veblen hierarchy generalizes the idea of fixed points function to functions that yield limits of what is obtainable by finite iteration of previously defined functions. Thus, for example $\varphi(2, \alpha_2 + 1) = \bigcup \varphi(2, \alpha 2), \varphi(1, \varphi(2, \alpha 2) + 1), \varphi(1, \varphi(1, \varphi(2, \alpha 2) + 1) + 1), ...$ Each element of the sequence, past the first, takes the previous element as the second parameter.

The Veblen hierarchy first develops this approach for fixed finite functions and then for sequences of fixed finite functions of increasing length. These functions are fully described in [16, 10, 12]. The algorithms used in the ordinal calculator are documented in [3]. The goal here is to give an understanding of these definition in part by using tables of examples generated by the ordinal calculator.

Table 1 defines the two parameter Veblen function. Examples of this function are in in Table 2. Three parameter Veblen function examples are shown in Table 3 and larger examples in Table 4.

2.2 Infinite parameter Veblen functions

The limit of what is definable with the finite parameter Veblen function is the union of the sequence $\varphi(1),\ \varphi(1, 0),\ \varphi(1, 0, 0), ...,$. This is defined to be φ_1. The infinite parameter Veblen function is written as $\varphi_\gamma(\alpha_1, \alpha_2, .., \alpha_k)$. If α_{n-1} or a more significant α parameter is nonzero, then the γ parameter is unchanged in defining `limitElement` i. e. the ordinal represented by

α	α.limitElement(n)		
	n=1	n=2	n=3
ε_1	ε_0	ω^{ε_0+1}	$\omega^{\omega^{\varepsilon_0+1}}$
ε_2	ε_1	ω^{ε_1+1}	$\omega^{\omega^{\varepsilon_1+1}}$
ε_ω	ε_1	ε_2	ε_3
$\varphi(2,1)$	$\varphi(2,0)$	$\varepsilon_{\varphi(2,0)+1}$	$\varepsilon_{\varepsilon_{\varphi(2,0)+1}+1}$
$\varphi(2,2)$	$\varphi(2,1)$	$\varepsilon_{\varphi(2,1)+1}$	$\varepsilon_{\varepsilon_{\varphi(2,1)+1}+1}$
$\varphi(2,5)$	$\varphi(2,4)$	$\varepsilon_{\varphi(2,4)+1}$	$\varepsilon_{\varepsilon_{\varphi(2,4)+1}+1}$
$\varphi(2,\omega)$	$\varphi(2,1)$	$\varphi(2,2)$	$\varphi(2,3)$
$\varphi(3,1)$	$\varphi(3,0)$	$\varphi(2,\varphi(3,0)+1)$	$\varphi(2,\varphi(2,\varphi(3,0)+1)+1)$
$\varphi(3,2)$	$\varphi(3,1)$	$\varphi(2,\varphi(3,1)+1)$	$\varphi(2,\varphi(2,\varphi(3,1)+1)+1)$
$\varphi(\omega,1)$	$\varepsilon_{\varphi(\omega,0)+1}$	$\varphi(2,\varphi(\omega,0)+1)$	$\varphi(3,\varphi(\omega,0)+1)$
$\varphi(\omega,9)$	$\varepsilon_{\varphi(\omega,8)+1}$	$\varphi(2,\varphi(\omega,8)+1)$	$\varphi(3,\varphi(\omega,8)+1)$
$\varphi(\omega,\omega)$	$\varphi(\omega,1)$	$\varphi(\omega,2)$	$\varphi(\omega,3)$
$\varphi(\omega,\omega+2)$	$\varepsilon_{\varphi(\omega,\omega+1)+1}$	$\varphi(2,\varphi(\omega,\omega+1)+1)$	$\varphi(3,\varphi(\omega,\omega+1)+1)$

Table 2: Two parameter Veblen function exanples

α	α.limitElement(n)		
	n=1	n=2	n=3
Γ_1	$\varphi(1,0,1)$	$\varphi(1,0,\varphi(1,0,1)+1)$	$\varphi(1,0,\varphi(1,0,\varphi(1,0,1)+1)+1)$
$\varphi(1,1,1)$	Γ_1	$\varphi(1,0,\Gamma_1+1)$	$\varphi(1,0,\varphi(1,0,\Gamma_1+1)+1)$
$\varphi(1,1,2)$	$\varphi(1,1,1)$	$\varphi(1,0,\varphi(1,1,1)+1)$	$\varphi(1,0,\varphi(1,0,\varphi(1,1,1)+1)+1)$
$\varphi(1,1,\omega)$	$\varphi(1,1,1)$	$\varphi(1,1,2)$	$\varphi(1,1,3)$
Γ_ω	Γ_1	Γ_2	Γ_3
$\varphi(1,\omega,1)$	$\varphi(1,1,\Gamma_\omega+1)$	$\varphi(1,2,\Gamma_\omega+1)$	$\varphi(1,3,\Gamma_\omega+1)$
$\varphi(1,2,1)$	Γ_2	$\varphi(1,1,\Gamma_2+1)$	$\varphi(1,1,\varphi(1,1,\Gamma_2+1)+1)$
$\varphi(1,2,2)$	$\varphi(1,2,1)$	$\varphi(1,1,\varphi(1,2,1)+1)$	$\varphi(1,1,\varphi(1,1,\varphi(1,2,1)+1)+1)$
$\varphi(1,2,5)$	$\varphi(1,2,4)$	$\varphi(1,1,\varphi(1,2,4)+1)$	$\varphi(1,1,\varphi(1,1,\varphi(1,2,4)+1)+1)$
$\varphi(3,1,1)$	$\varphi(3,1,0)$	$\varphi(3,0,\varphi(3,1,0)+1)$	$\varphi(3,0,\varphi(3,0,\varphi(3,1,0)+1)+1)$
$\varphi(2,2,\omega)$	$\varphi(2,2,1)$	$\varphi(2,2,2)$	$\varphi(2,2,3)$
$\varphi(4,3,2)$	$\varphi(4,3,1)$	$\varphi(4,2,\varphi(4,3,1)+1)$	$\varphi(4,2,\varphi(4,2,\varphi(4,3,1)+1)+1)$
$\varphi(\omega,3,2)$	$\varphi(\omega,3,1)$	$\varphi(\omega,2,\varphi(\omega,3,1)+1)$	$\varphi(\omega,2,\varphi(\omega,2,\varphi(\omega,3,1)+1)+1)$

Table 3: Three parameter Veblen function examples.

α	α.limitElement(n)		
	n=1	n=2	n=3
$\varphi(1,0,0,0)$	Γ_0	$\varphi(\Gamma_0+1,0,0)$	$\varphi(\varphi(\Gamma_0+1,0,0)+1,0,0)$
$\varphi(1,0,0,1)$	$\varphi(1,0,0,0)$	$\varphi(\varphi(1,0,0,0)+1,0,1)$	$\varphi(\varphi(\varphi(1,0,0,0)+1,0,1)+1,0,1)$
$\varphi(\omega,0,0,1)$	$\varphi(1,\varphi(\omega,0,0,0)+1,0,1)$	$\varphi(2,\varphi(\omega,0,0,0)+1,0,1)$	$\varphi(3,\varphi(\omega,0,0,0)+1,0,1)$
$\varphi(\omega,0,0,4)$	$\varphi(1,\varphi(\omega,0,0,3)+1,0,4)$	$\varphi(2,\varphi(\omega,0,0,3)+1,0,4)$	$\varphi(3,\varphi(\omega,0,0,3)+1,0,4)$
$\varphi(\omega,0,0,0)$	$\varphi(1,0,0,0)$	$\varphi(2,0,0,0)$	$\varphi(3,0,0,0)$
$\varphi(\omega,0,0,\omega)$	$\varphi(\omega,0,0,1)$	$\varphi(\omega,0,0,2)$	$\varphi(\omega,0,0,3)$
$\varphi(1,0,0,0,0)$	$\varphi(1,0,0,0)$	$\varphi(\varphi(1,0,0,0)+1,0,0,0)$	$\varphi(\varphi(\varphi(1,0,0,0)+1,0,0,0)+1,0,0,0)$

Table 4: More than three parameter Veblen function examples

φ_1	$= \varphi(1),\ \varphi(1,0),\ \varphi(1,0,0), ...,$
γ and the least significant α are successors, the other α_i are 0	
$\varphi^0_{\gamma+1}(\alpha+1)$	$= \varphi_{\gamma+1}(\alpha)$
$\varphi^{i+1}_{\gamma+1}(\alpha+1)$	$= \varphi_\gamma(\varphi^i_{\gamma+1}(\alpha+1)+1)$
thus using $\varphi^i_\gamma(\alpha)$ thus	
$\varphi_{\gamma+1}(\alpha+1)$	$= \bigcup_{i\in N} \varphi^i_{\gamma+1}(\alpha+1)$
α a limit	
$\varphi_\gamma(\alpha)$	$= \bigcup_{\beta\in\alpha} \varphi_\gamma(\beta)$
γ a limit	
$\varphi_\gamma(\alpha+1)$	$= \bigcup_{\beta\in\gamma} \varphi_\beta(\varphi_\gamma(\alpha)+1)$
See Table 6 for examples.	

Table 5: Definition of $\varphi_\gamma(\alpha)$

α	α.limitElement(n)		
	n=1	n=2	n=3
φ_1	ω	ε_0	Γ_0
$\varphi_1(1,0,0)$	$\varphi_1(1,0)$	$\varphi_1(\varphi_1(1,0)+1,0)$	$\varphi_1(\varphi_1(\varphi_1(1,0)+1,0)+1,0)$
$\varphi_1(1,0,1)$	$\varphi_1(1,0,0)$	$\varphi_1(\varphi_1(1,0,0)+1,1)$	$\varphi_1(\varphi_1(\varphi_1(1,0,0)+1,1)+1,1)$
$\varphi_3(1)$	φ_3+1	$\varphi_2(\varphi_3+1,0)$	$\varphi_2(\varphi_3+1,0,0)$
$\varphi_3(2)$	$\varphi_3(1)+1$	$\varphi_2(\varphi_3(1)+1,0)$	$\varphi_2(\varphi_3(1)+1,0,0)$
$\varphi_5(\omega)$	$\varphi_5(1)$	$\varphi_5(2)$	$\varphi_5(3)$
φ_ω	φ_1	φ_2	φ_3
$\varphi_\omega(1)$	$\varphi_1(\varphi_\omega+1)$	$\varphi_2(\varphi_\omega+1)$	$\varphi_3(\varphi_\omega+1)$
$\varphi_{\omega^\omega}(8)$	$\varphi_\omega(\varphi_{\omega^\omega}(7)+1)$	$\varphi_{\omega^2}(\varphi_{\omega^\omega}(7)+1)$	$\varphi_{\omega^3}(\varphi_{\omega^\omega}(7)+1)$

Table 6: Veblen function with variable number of parameters examples

this notation can be built up from the union of smaller ordinals with notations with the same γ value. In this case the α_i are treated as they are in the finite parameter Veblen function. The Veblen function, $\varphi_\gamma(\alpha)$ defined in Table 6, illustrates the one major difference between the finite and infinite length Veblen functions. Examples of the infinite length function are shown in Table 6.

3 Countable admissible ordinals and projection

The ordinal calculator goes beyond the Veblen hierarchy using a form of ordinal projection based on countable admissible ordinals $\geq \omega_1^{CK}$. This uses a weakened version of recursive ordinal notation that applies to these larger ordinals. The idea is to replace the defining function `limitElement` on the integers with `limitOrd` that accepts ordinal notations of a specified type. For types greater than the integers, the domain of `limitOrd` is incomplete in any finitely specifiable notation system. For these larger ordinals the following no longer holds.

$$\bigcup_{i \in N} \alpha.\texttt{limitElement(i)} = \alpha \tag{2}$$

Instead the following holds[8].

$$\alpha = \bigcup_{\beta:\alpha.\texttt{isValid}(\beta)} \alpha.\texttt{limitOrd}(\beta) \tag{3}$$

If α is a recursive ordinal then `isValid` is defined to be 'is an integer'.

ω_1.`limitOrd`[9] can be defined as the identity function and equation 3 will be valid as the system is expanded. ω_1.`limitOrd` is not defined this way, but as a string manipulation that can work with an incomplete expandable hierarchy. ω_1.`limitOrd` is defined to grow faster than lower level functions.

For notations for ordinals $\geq \omega_1^{CK}$, the `compare` member function works as defined in Section 1. It is a `virtual` function that can be overridden as new sub`class`es of `Ordinal` are added. As a consequence the ordinal hierarchy defined at any point in this process has a recursive well ordering even though ordinals much larger than ω_1^{CK} are represented. Thus the hierarchy can be embedded within itself in complex ways a bit like the Mandelbrot set. That is used in to expand ordinals represented in the ordinal calculator.

3.1 Semantics for countable admissible ordinals

The syntax used in version 0.3 of the ordinal calculator is summarized in Figure 1. The syntax and semantics for Cantor normal form expressions (Section 1.1), as well as finite length (Section 2.1) and infinite length (Section 2.2) Veblen functions have been discussed. The syntax and semantics for the remainder of Figure 1 are described in this section (equations 7 to 11) and Section 3.3 (equations 12 to 14).

κ in ω_κ refers to the admissible level (ω_κ^{CK}). The other parameters in Equation 7, γ and the α_i, work mostly as they do in the Veblen hierarchy. New rules are needed only when one of two conditions are met. This is reflected in the software where `virtual` functions are used to create `limitElement` and `limitOrd` outputs. In these `virtual` functions, unspecified parameters are filled in with the current values in the ordinal notation from which the `virtual` function is called. This allows lower `class` functions to do much of the work for objects of a higher sub`class`.

New algorithms are required when κ and the least significant α (α_m in Equation 7) are the only nonzero parameters. κ can be a successor or a limit. α must be a successor. (If the least significant α is a limit than the standard definition can be used with `virtual` functions.) These new algorithms are illustrated in Table7.

Equation 8 is used to define `limitOrd`. $\omega_{\alpha+1}.\texttt{limitOrd}(\eta) = \omega_{\alpha+1}[\eta]$[10]. The definition of $\omega_{\alpha+1}[\eta]$ diagonalizes previously defined functions as illustrated in Table 8 lines 2 to 5.

---

[8] In the following `isValid` is an abbreviation for `isValidLimitOrdParam` used in the ordinal calculator C++ code.

[9] Because the ordinal calculator only references countable ordinals ω_1 is used to represent ω_1^{CK}. In general ω_α^{CK} is written as ω_α.

[10] If α is a limit then $\omega_\alpha.\texttt{limitOrd}(\eta) = \omega_{\alpha.\texttt{limitOrd}(\eta)}$.

Cantor normal form

$$\alpha_1 > \alpha_2 > \alpha_3 > ... > \alpha_k$$

α and the α_i are ordinal notations
n_i are nonzero integers

$$\alpha = \omega^{\alpha_1} n_1 + \omega^{\alpha_2} n_2 + \omega^{\alpha_3} n_3 + ... + \omega^{\alpha_k} n_k \tag{4}$$

Veblen functions

$$\alpha = \varphi(\alpha_1, \alpha_2, ..., \alpha_k) \tag{5}$$

$$\alpha = \varphi_\gamma(\alpha_1, \alpha_2, ..., \alpha_m) \tag{6}$$

Notations for countable admissible ordinals with projection

$$\alpha = \omega_{\kappa,\gamma}(\alpha_1, \alpha_2, ..., \alpha_m) \tag{7}$$

$$\alpha = \omega_\kappa[\eta] \tag{8}$$

$$\alpha = [[\delta]]\omega_\kappa[\eta] \tag{9}$$

$$\alpha = [[\delta]]\omega_{\kappa,\gamma}(\alpha_1, \alpha_2, ..., \alpha_m) \tag{10}$$

$$\alpha = [[\delta]]\omega_\kappa[[\eta]] \tag{11}$$

Notations for a stronger form of ordinal projection

$$\alpha = [[\delta_1 \leftharpoonup \sigma_1, \delta_2 \leftharpoonup \sigma_2, ..., \delta_m \leftharpoonup \sigma_m]]\omega_\kappa[\eta] \tag{12}$$

$$\alpha = [[\delta_1 \leftharpoonup \sigma_1, \delta_2 \leftharpoonup \sigma_2, ..., \delta_m \leftharpoonup \sigma_m]]\omega_\kappa[[\eta]] \tag{13}$$

$$\alpha = [[\delta_1 \leftharpoonup \sigma_1, \delta_2 \leftharpoonup \sigma_2, ..., \delta_m \leftharpoonup \sigma_m]]\omega_{\kappa,\gamma}(\alpha_1, \alpha_2, ..., \alpha_k) \tag{14}$$

There are several restrictions on equations 7 to 14.

1. If κ is a limit then no η parameter is allowed.
2. The most significant δ cannot be a limit.
3. If any other δ is a limit, then the associated σ must be 0.
4. If the σ associated with the least significant δ is a limit, then no η parameter is allowed.

Figure 1: Ordinal calculator notation syntax

κ and the least significant α are successors
other α_i are 0
$\omega^0_{\kappa+1}(\alpha+1) \quad = \omega_{\kappa+1}(\alpha)$
$\omega^{i+1}_{\kappa+1}(\alpha+1) \quad = \omega_{\kappa,\omega^i_{\kappa+1}(\alpha+1)+1}(\alpha)$
thus using $\omega^i_\kappa(\alpha)$ thus
$\omega_{\kappa+1}(\alpha+1) \quad = \bigcup_{i\in N} \omega^i_{\kappa+1}(\alpha+1)$
See line 6 in Table 8 for an example.
κ is a limit
$\omega_\kappa(\alpha+1) \quad = \bigcup_{\zeta\in\kappa} \omega_{\zeta,\omega_\kappa(\alpha)+1}$
See line 7 in Table 8 for an example.

Table 7: Definition of $\omega_\kappa(\alpha)$ with κ a successor and limit

	α	α.limitElement(n)			
		n=1	n=2	n=3	n=4
1	ω_1	$\omega_1[1]$	$\omega_1[2]$	$\omega_1[3]$	$\omega_1[4]$
2	$\omega_1[1]$	ω	φ_ω	$\varphi_{\varphi_\omega+1}$	$\varphi_{\varphi_{\varphi_\omega+1}+1}$
3	$\omega_1[3]$	$\omega_1[2]$	$\varphi_{\omega_1[2]+1}$	$\varphi_{\varphi_{\omega_1[2]+1}+1}$	$\varphi_{\varphi_{\varphi_{\omega_1[2]+1}+1}+1}$
4	$\omega_1[\omega]$	$\omega_1[1]$	$\omega_1[2]$	$\omega_1[3]$	$\omega_1[4]$
5	$\omega_3[3]$	$\omega_3[2]$	$\omega_{2,\omega_3[2]+1}$	$\omega_{2,\omega_{2,\omega_3[2]+1}+1}$	$\omega_{2,\omega_{2,\omega_{2,\omega_3[2]+1}+1}+1}$
6	$\omega_3(5)$	$\omega_3(4)$	$\omega_{2,\omega_3(4)+1}(4)$	$\omega_{2,\omega_{2,\omega_3(4)+1}(4)+1}(4)$	$\omega_{2,\omega_{2,\omega_{2,\omega_3(4)+1}(4)+1}(4)+1}(4)$
7	$\omega_\omega(5)$	$\omega_{1,\omega_\omega(4)+1}$	$\omega_{2,\omega_\omega(4)+1}$	$\omega_{3,\omega_\omega(4)+1}$	$\omega_{4,\omega_\omega(4)+1}$
8	$[[1]]\omega_1$	$[[1]]\omega_1[[1]]$	$[[1]]\omega_1[[2]]$	$[[1]]\omega_1[[3]]$	$[[1]]\omega_1[[4]]$
9	$[[1]]\omega_1[[1]]$	ω	$\omega_1[\omega]$	$\omega_1[\omega_1[\omega]]$	$\omega_1[\omega_1[\omega_1[\omega]]]$
10	$[[3]]\omega_3[[1]]$	ω	$\omega_3[\omega]$	$\omega_3[\omega_3[\omega]]$	$\omega_3[\omega_3[\omega_3[\omega]]]$
11	$[[1]]\omega_1[[2]]$	$[[1]]\omega_1[[1]]$	$\omega_1[[[1]]\omega_1[[1]]]$	$\omega_1[\omega_1[[[1]]\omega_1[[1]]]]$	$\omega_1[\omega_1[\omega_1[[[1]]\omega_1[[1]]]]]$
12	$[[1]]\omega_1[[3]]$	$[[1]]\omega_1[[2]]$	$\omega_1[[[1]]\omega_1[[2]]]$	$\omega_1[\omega_1[[[1]]\omega_1[[2]]]]$	$\omega_1[\omega_1[\omega_1[[[1]]\omega_1[[2]]]]]$
13	$[[1]]\omega_1(1)$	$[[1]]\omega_1$	$\varphi_{[[1]]\omega_1+1}$	$\varphi_{\varphi_{[[1]]\omega_1+1}+1}$	$\varphi_{\varphi_{\varphi_{[[1]]\omega_1+1}+1}+1}$
14	$[[2]]\omega_2(1)$	$[[2]]\omega_2$	$\omega_{1,[[2]]\omega_2+1}$	$\omega_{1,\omega_{1,[[2]]\omega_2+1}+1}$	$\omega_{1,\omega_{1,\omega_{1,[[2]]\omega_2+1}+1}+1}$
15	$[[2]]\omega_3(1)$	$[[2]]\omega_3$	$[[2]]\omega_{2,[[2]]\omega_3+1}$	$[[2]]\omega_{2,[[2]]\omega_{2,[[2]]\omega_3+1}+1}$	$[[2]]\omega_{2,[[2]]\omega_{2,[[2]]\omega_{2,[[2]]\omega_3+1}+1}+1}$
16	$[[5]]\omega_6[1]$	$[[5]]\omega_5$	$[[5]]\omega_{5,[[5]]\omega_5+1}$	$[[5]]\omega_{5,[[5]]\omega_{5,[[5]]\omega_5+1}+1}$	$[[5]]\omega_{5,[[5]]\omega_{5,[[5]]\omega_{5,[[5]]\omega_5+1}+1}+1}$
17	$[[5]]\omega_7[4]$	$[[5]]\omega_7[3]$	$[[5]]\omega_{6,[[5]]\omega_7[3]+1}$	$[[5]]\omega_{6,[[5]]\omega_{6,[[5]]\omega_7[3]+1}+1}$	$[[5]]\omega_{6,[[5]]\omega_{6,[[5]]\omega_{6,[[5]]\omega_7[3]+1}+1}+1}$
18	$\omega_5(8)$	$\omega_5(7)$	$\omega_{4,\omega_5(7)+1}(7)$	$\omega_{4,\omega_{4,\omega_5(7)+1}(7)+1}(7)$	$\omega_{4,\omega_{4,\omega_{4,\omega_5(7)+1}(7)+1}(7)+1}(7)$
19	$\omega_{5,8}$	$\omega_{5,7}(\omega_{5,7}+1)$	$\omega_{5,7}(\omega_{5,7}+1,0)$	$\omega_{5,7}(\omega_{5,7}+1,0,0)$	$\omega_{5,7}(\omega_{5,7}+1,0,0,0)$
20	$\omega_\omega(8)$	$\omega_{1,\omega_\omega(7)+1}$	$\omega_{2,\omega_\omega(7)+1}$	$\omega_{3,\omega_\omega(7)+1}$	$\omega_{4,\omega_\omega(7)+1}$
21	$\omega_{\omega,8}$	$\omega_{\omega,7}(\omega_{\omega,7}+1)$	$\omega_{\omega,7}(\omega_{\omega,7}+1,0)$	$\omega_{\omega,7}(\omega_{\omega,7}+1,0,0)$	$\omega_{\omega,7}(\omega_{\omega,7}+1,0,0,0)$

Table 8: Countable admissible level ordinal notations

$\omega_1[1]$	$= \bigcup \omega, \varphi_\omega, \varphi_{\varphi_\omega}, \varphi_{\varphi_{\varphi_\omega}}, \varphi_{\varphi_{\varphi_{\varphi_\omega}}}, \ldots,$
$\omega_1[\eta+1]$	$= \bigcup \omega_1[\eta], \varphi_{\omega_1[\eta]+1}, \varphi_{\varphi_{\omega_1[\eta]+1}+1}, \ldots,$
$\omega^0_{\kappa+1}[\eta+1]$	$= \omega_1[\eta]$
$\omega^{i+1}_{\kappa+1}[\eta+1]$	$= \varphi_{\kappa, \omega^i_{\kappa+1}[\eta+1]+1}$
thus using $\omega^i_\kappa[\eta]$ thus	
$\omega_{\kappa+1}[\eta+1]$	$= \bigcup_{i \in N} \omega^i_{\kappa+1}[\eta+1]$
See line 5 in Table 8 for an example.	
η is a limit	
$\omega_\kappa[\eta]$	$= \bigcup_{\zeta<\eta} \omega_\kappa[\zeta]$
If κ is a limit then η must be 0.	

Table 9: Definition of $\omega_\kappa[\eta]$

The notation system at and above the limit of recursive ordinals definable in the system is incomplete. Equation 9 through Equation 11 take advantage of this in various forms of ordinal projection. The δ parameter in these equations ($\delta \leq \kappa$), is the admissible level the ordinal is projected onto. With the $[\eta]$ suffix the restriction is $\delta < \kappa$. When $[[\alpha+1]]\omega_{\alpha+1}[[1]]$ is expanded with `limitOrd` (or `limitElement`) the $[[\delta]]$ prefix is dropped as shown in lines 9 and 10 in Table 8.

Ordinal projection allows arbitrarily large notations in a fixed, and thus limited, system to be used in specifying an ordinal notation. However the definition of the ordinal is modified so that any invocation of `limitOrd` for evaluating this ordinal, will have a value for which `isValid` is true. This requires a notation $< \omega_\delta$. Since this notation is defining a new ordinal $< \omega_\delta$ not all parameters that meet this definition are legal. There is a definition of `isValid` in Section 3.2.

Equation 9 evaluates just as Equation 8 does (see Table 9). δ is copied without changing it. This is illustrated by lines 16 and 17 in Table 8. The exception (mentioned above) is when the δ prefix is dropped as shown in lines 9 and 10 in Table 8.

Equation 10 is evaluated as Equation 7 is in all cases with the value of δ being copied without change. (The $[[\delta]]$ prefix only changes when an $[[\eta]]$ or $[\eta]$ suffix is present which requires the α_i and γ parameters to be 0.) Examples are shown in lines 13, 14 and 15 in Table 8.

Equation 11 in combination with Equation 9 is used to define the projection of a successor δ down one level in κ. Equation 11 diagonalizes equation 9 as shown in lines 9 and 10 in Table 8. In evaluating this notation, any result of the form $[[\delta]]\omega_\delta[\eta]$ drops the $[[\delta]]$ prefix as no longer meaningful. See lines 9 through 12 in Table 8 for examples.

3.2 `limitType`, `maxLimitType` and `isValid`

α.`limitType` is an `Ordinal` `virtual` member function which designates an upper limit on the ordinals that can be used by α.`limitOrd`. α.`maxLimitType` is the maximum of `limitType` valid in evaluating this ordinal and any smaller one.

If α.`limitType` $>$ β.`maxLimitType`, then α.`isValid`(β) is true and α.`limitOrd`(β) yields a valid ordinal notation. If α.`limitType` $=$ β.`maxLimitType`, then additional tests are needed. This only applies to equation 14 with γ and α_i all 0. Thus α is of the form $[[\delta_1 \nearrow \sigma_1, \delta_2 \nearrow \sigma_2, ..., \delta_m \nearrow \sigma_m]]\omega_\kappa$. Additional restrictions for this special case are $\kappa = \delta_m$, $\sigma_m = 0$ and $\alpha > \beta$. This special case allows expressions of the form: $[[4]]\omega_6[[[[4]]\omega_6[[\omega]]]]$. This is needed because both a double bracketed prefix and a double bracketed suffix individually reduce the `limitType` by 1, but combined they only reduce the `limitType` by 1.

3.3 Semantics for extended ordinal projection

Equations 12 to 14 provide an extended form of ordinal collapsing. The idea is to index the δ level with an ordinal notation, σ, and further expand the indexing with a list of these pairs. The first value of δ continues to limit the size of η. Values of δ that follow it must either be increasing or equal with increasing values of σ. κ, in turn, must be $\geq$ the last δ. The first δ indicates the level at which the η parameter for the ordinal as a whole and any of its parameters is restricted. Subsequent δs do not affect this. The additional index or σ can be any ordinal notation with this restriction. It is not otherwise limited.

Recall that the ordinal hierarchy beyond the Church-Kleene ordinal is somewhat like the Mandelbrot set. Any recursive formalization of the hierarchy has a well ordering less than the Church Kleene ordinal and thus it can be embedded within itself at many places and to any finite depth. This nesting must be managed to avoid an infinite descending chain or inconsistency.

The semantics for equations 12 to 14 differs from that for previous equation when that semantics would generate a value of $\kappa < \delta_m$. This can only occur when $\kappa = \delta_m$ and other conditions are met. These occur when κ is decremented or expanded from a limit ordinal. In these cases κ is changed as the previous semantics requires and the $[[\delta_1 \nearrow \sigma_1, \delta_2 \nearrow \sigma_2, ..., \delta_m \nearrow \sigma_m]]$ prefix is adjusted to keep $\delta_m \leq \kappa$ without violating the constraints on this prefix. In some cases this means the prefix is shortened.

Decrementing $[[\delta_1 \nearrow \sigma_1, \delta_2 \nearrow \sigma_2, ..., \delta_m \nearrow \sigma_m]]$ when its least significant value is a a limit is illustrated by lines 1 and 2 in Table 11. Table 10 gives the rules for decrementing the $[[\delta_1 \nearrow \sigma_1, \delta_2 \nearrow \sigma_2, ..., \delta_m \nearrow \sigma_m]]$ prefix when its least significant value is a successor. The last column of this table references the line or lines of examples in Table 11.

When the least significant δ is decremented a range of values for the corresponding σ can be appended and the result will be a lower valued prefix. Thus the valid values for `limitOrd` are chosen to be a sequence of σs that diagonalize lower level definitions. This is illustrated by lines 7, 8 and some of those that follow in Table 11 and the lines in Table 10 that reference these lines.

4 Mathematical truth

Some mathematics is absolute (2+2=4) and others is contingent (parallel lines never meet). Mathematics focuses on absolute truth. It deals with parallel lines by asking which assumptions lead to which conclusions. These are absolute truths although different assumptions will hold in different situations.

The borderline between contingent and absolute truth in the domain of the infinite has remained open for as long as mathematics has been a recognized discipline. It goes back

All entries assume least significant δ or σ is a successor and $\delta_m = \kappa$.			
The comparisons in columns 1 and 2 are between the two least significant δs or σs.			
The **11** column refers to an example line or lines in Table 11.			
δ_m	σ_m	**next least (or smaller) prefix**	**11**
$\delta_m = \delta_{m-1}$	$\sigma_m - 1 = \sigma_{m-1}$	$[[\delta_1\nearrow\sigma_1, , ..., \delta_{m-1}\nearrow\sigma_{m-1}]]$	**5**
$\delta_m = \delta_{m-1}$	$\sigma_m - 1 > \sigma_{m-1}$	$[[\delta_1\nearrow\sigma_1, , ..., \delta_{m-1}\nearrow\sigma_{m-1}, \delta_m, \sigma_m - 1]]$	**7**
$(\delta_m > \delta_{m-1})$ $\vee(m = 1)$	successor	$[[\delta_1\nearrow\sigma_1, , ..., \delta_{m-1}\nearrow\sigma_{m-1}, \delta_m, \sigma_m - 1]]$	**9 10 11**
$(\delta_m + 1 = \delta_{m-1})$ $\wedge\delta_{m-1}$ a limit	0	$[[\delta_1\nearrow\sigma_1, , ..., \delta_{m-1}\nearrow\sigma_{m-1}]]$	**12**
$(\delta_m + 1 = \delta_{m-1})$ $\wedge\delta_{m-1}$ successor	0	$[[\delta_1\nearrow\sigma_1, , ..., \delta_{m-1}\nearrow\sigma_{m-1}, \delta_m - 1\nearrow\sigma_{m-1} + 1]]$	**13 14 15**
$\delta_m + 1 > \delta_{m-1}$	0	$[[\delta_1\nearrow\sigma_1, , ..., \delta_{m-1}\nearrow\sigma_{m-1}, \delta_m - 1\nearrow\sigma_{m-1} + 1]]$	**16 17 18**

Table 10: Computing next least $[[\delta_1\nearrow\sigma_1, \delta_2\nearrow\sigma_2, ..., \delta_m\nearrow\sigma_m]]$ prefix

	α	α`.limitElement(n)`	
		`n=1`	`n=2`
1	$[[3\nearrow\omega]]\omega_3$	$[[3\nearrow1]]\omega_3$	$[[3\nearrow2]]\omega_3$
2	$[[3, \omega]]\omega_\omega$	$[[3, 4]]\omega_4$	$[[3, 5]]\omega_5$
3	$[[3, \omega]]\omega_\omega(1)$	$[[3, 4]]\omega_{4,[[3,\omega]]\omega_\omega+1}$	$[[3, 5]]\omega_{5,[[3,\omega]]\omega_\omega+1}$
4	$[[3, \omega]]\omega_{\omega^\omega}$	$[[3, \omega]]\omega_{\omega 2}$	$[[3, \omega]]\omega_{\omega^2+\omega}$
5	$[[2, 3\nearrow4, 3\nearrow5]]\omega_3[1]$	$[[2, 3\nearrow4]]\omega_3$	$[[2, 3\nearrow4]]\omega_{[[2,3\nearrow4]]\omega_3}$
6	$[[2, 3\nearrow4, 3\nearrow8]]\omega_3[[1]]$	ω	$[[2, 3\nearrow4, 3\nearrow8]]\omega_3[\omega]$
7	$[[2, 3\nearrow4, 3\nearrow8]]\omega_3[1]$	$[[2, 3\nearrow4, 3\nearrow7]]\omega_3$	$[[2, 3\nearrow4, 3\nearrow7, [[2, 3\nearrow4, 3\nearrow7]]\omega_3]]\omega_{[[2,3\nearrow4,3\nearrow7]]\omega_3}$
8	$[[2, 3\nearrow4, 3\nearrow8]]\omega_3[5]$	$[[2, 3\nearrow4, 3\nearrow8]]\omega_3[4]$	$[[2, 3\nearrow4, 3\nearrow7, [[2, 3\nearrow4, 3\nearrow8]]\omega_3[4]]]\omega_{[[2,3\nearrow4,3\nearrow8]]\omega_3[4]}$
9	$[[3\nearrow\omega + 1]]\omega_3[1]$	$[[3\nearrow\omega]]\omega_3$	$[[3\nearrow\omega, [[3\nearrow\omega]]\omega_3]]\omega_{[[3\nearrow\omega]]\omega_3}$
10	$[[1, 3\nearrow1]]\omega_3[1]$	$[[1, 3]]\omega_3$	$[[1, 3, [[1, 3]]\omega_3]]\omega_{[[1,3]]\omega_3}$
11	$[[1, 3\nearrow1]]\omega_3[4]$	$[[1, 3\nearrow1]]\omega_3[3]$	$[[1, 3, [[1, 3\nearrow1]]\omega_3[3]]]\omega_{[[1,3\nearrow1]]\omega_3[3]}$
12	$[[1, \omega, \omega + 1]]\omega_{\omega+1}[1]$	$[[1, \omega]]\omega_{\omega+1}$	$[[1, \omega]]\omega_{[[1,\omega]]\omega_{\omega+1}}$
13	$[[3, 4]]\omega_4[1]$	$[[3, 3\nearrow1]]\omega_4$	$[[3, 3\nearrow[[3, 3\nearrow1]]\omega_4 + 1]]\omega_4$
14	$[[3\nearrow5, 4]]\omega_4[1]$	$[[3\nearrow5, 3\nearrow6]]\omega_4$	$[[3\nearrow5, 3\nearrow[[3\nearrow5, 3\nearrow6]]\omega_4 + 1]]\omega_4$
15	$[[3\nearrow5, 4]]\omega_4[3]$	$[[3\nearrow5, 4]]\omega_4[2]$	$[[3\nearrow5, 3\nearrow[[3\nearrow5, 4]]\omega_4[2] + 1]]\omega_4$
16	$[[2, 4]]\omega_4[1]$	$[[2, 3]]\omega_4$	$[[2, 3\nearrow[[2, 3]]\omega_4 + 1]]\omega_4$
17	$[[2\nearrow5, 4]]\omega_4[1]$	$[[2\nearrow5, 3]]\omega_4$	$[[2\nearrow5, 3\nearrow[[2\nearrow5, 3]]\omega_4 + 1]]\omega_4$
18	$[[2\nearrow5, 4]]\omega_4[3]$	$[[2\nearrow5, 4]]\omega_4[2]$	$[[2\nearrow5, 3\nearrow[[2\nearrow5, 4]]\omega_4[2] + 1]]\omega_4$

Table 11: Countable nested embed admissible level ordinal notations

at least as far as Zeno's paradoxes. We have much deeper knowledge of the implications of various assumptions today and these can provide a guide for drawing this line.

I take liberty with the famous phrase of Leopold Kronecker to say "God is making the integers; all else is the work of man". 'made' was changed to 'making' to emphasize the rejection of completed infinite totalities while recognizing that the universe may have no finite bound. Of course no argument can decide the matter. But one can argue that, in a finite, but perhaps potentially infinite universe, some questions definable in ZF (Zermelo Frankel set theory)[4] are not objectively determined. It is reasonable to restrict absolute mathematics to objectively determined questions in the universe we inhabit. Other questions are meaningful, if at all, in a universe in which actual infinities exist, at least in a Platonic sense. In the physical universe it appears that real numbers exist, not as completed infinite totalities, but only as mathematical expressions, some of which can be interpreted as properties of recursive processes. There is is a finite algebra of that defines properties of processes that never end. The ordinal calculator uses elements of this algebra.

The position advocated here is not formalism. Most of mathematics can be justified as determined by a recursively enumerable sequence of events. There may be no objective fact that determines the result, but some relationship between a recursively enumerable sequence of events determine it. Each of these events could occur in a finite universe that is unbounded over time. Each individual event is logically determined by the assumptions that define the sequence of events. As a consequence some statements about the entire collection are objective. What these statements are cannot be precisely defined. As the properties become more complex and convoluted they become more questionable. Mathematics will always be incomplete with regard to both provability and definability.

Consider the sequence of questions: 'Does a TM, have an infinite number of outputs with the first output being a code for the ordinal 0?' This is property P_0. 'Does a TM have an infinite number of outputs with the first output being a code for the ordinal 1 and with an infinite subset of the outputs being the Gödel numbers of TMs that satisfy P_0?' This is the property P_1. In general $P_{\alpha+1}$ asks 'Does a TM have an infinite number of outputs the first of which is the code for a recursive ordinal notation for $\alpha+1$ and an infinite subset of which are of type P_α?' If α is a limit ordinal then there question is: 'Are there an infinite number of outputs, the first of which is a recursive ordinal notation for α and an infinite subset of which are the Gödel numbers of a TMs whose first output is a notation of a recursive ordinal $\beta < \alpha$ and that TM satisfies P_β. Iterating this up to any integer yields the arithmetic hierarchy. Iterating it up to any recursive ordinal yields the hyperarithmetic hierarchy. These two hierarchies are objectively determined. Some mathematics requiring quantification over the reals is also objectively determined as described in the next section.

4.1 Real numbers

Real numbers have been questioned for another reason. They are impredicative. Reals defined in ZF include those that require quantification over *all* reals to define. Some of the famous paradoxes that have led to inconsistent proposals for mathematics are impredicative. Removing all impredicative definitions is one way to avoid them.

The reals provably definable in ZF are both incomplete and impredicative. How can one quantify over them? Some such questions appear to be objective. Consider the question:

does a TM (Turing Machine) that accepts an arbitrarily long sequence of inputs halt for every possible sequence? This question seems objective because it is logically determined by a recursively enumerable sequence. These are the finite sequences of inputs for which the TM eventually halts before it accepts a new input. If these include an initial segment of every real number than the answer is yes and no otherwise. Some initial segments obviously cover every possible real number. For example, in considering binary reals less than 1, the sequences that start with .0 and .1 cover every real number. A TM that halts for every possible input sequence is called WF (well founded).

In contrast is the continuum hypothesis. This asks does there exist a set, α, whose 'size' is $>$ than the size of integers and $<$ than the size of the reals. The size of α is $>$ than the size of β if there exists an onto function from α to β, but no onto function that goes the other way. If none of these functions or sets have an objective existence as infinite totalities, but only have meaning as human created mathematical expressions and the relationships provable between them, then one can question the objectivity of the continuum hypothesis as others have done[5]. No question that *requires* an uncountable set of events to determine can be determined by a recursively enumerable of sequence of events.

Mathematics is inevitably incomplete with regard to definability as well as provability. Gödel proved the latter. Cantor may or may not have proved that there are *more* reals than integers (for that to be true reals must exist as completed infinite totalities at least in some Platonic sense), but his diagonalization method combined with the Lowenheim-Skolem theorem prove that any finite formal system must be incomplete with regard to the definability of reals. Just as one can always make a system, that includes first order arithmetic, stronger by adding the axiom that the original system was consistent to a new system, one can always expand a system that defines real numbers by adding an axiom that defines the diagonalization of all real numbers provably definable in the original system. As Emil Post observed over 60 years ago: "The conclusion is unescapable that even for such a fixed, well defined body of mathematical propositions [a formulation of the recursively enumerable sets], *mathematical thinking is, and must remain, essentially creative*[11]."

This can be reflected in the definition of reals. Reals need not exist as sets. We can treat them as mathematical expressions satisfying a property. Proving that something is true of all reals means it must be true of any entity defined now or in the future that satisfies the property.

4.2 Expanding mathematics

There is an alternative to large cardinal axioms for extending mathematics that is related to these axioms, but limits itself to questions logically determined by a recursively enumerable sequence of events. This involves generalizing the idea of a WF TM to define large countable admissible ordinals.

Sacks proved that the countable admissible ordinals are those constructed like ω_1^{CK} using TMs with oracles for previous levels in the countable admissible hierarchy[15]. An alternative is to generalize the concept of a WF TM by asking if a TM that accepts an arbitrary number of integer inputs, halts for every infinite sequence of Gödel numbers of WF TMs of a lower level. Given any countable ordinal, α. defined in this way, one can iterate the property of being WF for lower levels up to α. This does not exhaust the countable admissible ordinals.

They can be further extended with new axioms somewhat analogous to large cardinal axioms. In this hierarchy one has different types of limit types (as described in Section 3.2) just as one does in the cardinal hierarchy. One may be able to define something analogous to large cardinal axioms while retaining the objectivity of properties of TMs determined by recursively enumerable sequences of events.

4.3 The cardinal hierarchy

This approach sees cardinals beyond the integers as dealing with contingent rather than absolute mathematics. The set of all reals and the entire cardinal hierarchy provably definable in ZF may be objectively definable as properties of recursive processes albeit not within ZF. Cardinal hierarchies may have an objective meaning under certain assumptions just as Euclidean geometry does.

Large cardinal axioms appear to implicitly define iterative processes of a sort that is very difficult to develop directly. Perhaps, by focusing on the countable admissible hierarchy and developing analogous axioms, one can define the same and even more powerful iterative structures. These structures will be more complex, but they can be implemented as computer programs on which experiments can be performed. It is possible that this capacity will eventually lead to approaches that are more productive than is possible with the apparent elegance of large cardinal axioms.

5 Incompleteness and diversity

With the aid of computers mathematics may be expanded beyond what is practical without them. This can continue indefinitely, but as long as the development leads to a single system of accepted mathematics it will remain inside what I call a Gödel limit. This assumes we are in a finite, but potentially infinite, universe where all physical processes can be recursively defined. This may or may not be true, but it seems increasingly likely as we gain deeper understanding of both physics and the human brain.[11]

A Gödel limit is a sequence of ever more powerful mathematical models all of which may eventually be discovered in a single path of mathematical exploration and development. However all of the results are subsumed by a single more powerful result that will never be discovered inside the Gödel limit. The only way, under these assumptions, to avoid a Gödel limit is through an unbounded expansion of diversity. Gödel proved that no formal system can decide all mathematical questions, but, in a potentially infinite universe, every possible true model can be explored with unbounded diversity.

It seems that the mathematically capable human mind could have only evolved over billions of years on a planet with the enormous diversity of life on earth. That mind, an evolutionary legacy, allows us to be relatively certain about a great deal of mathematics by using the human mind in a cultural process of consensus. Using computers as tools for mathematical research may significantly extend the mathematics that is widely accepted. It is also likely to lead to alternative powerful extensions that seem at least plausible as large

[11] I have argued that the apparent irreducible randomness of quantum mechanics may be a chaotic like but deterministic effect of a totally discrete universe[2].

cardinal axioms are seen today by some. At that point some avenues of further progress will require a diversity of schools with no way to ultimately resolve all competing consistent approaches although some will be seen to fail omega consistency or some other generally accepted criteria.

Why would it be worth the effort to explore this mathematics? Existing mathematics goes far beyond what is commonly used in science and engineering. The answer to this question takes us outside of mathematics to questions of ultimate meaning and value. Bertrand Russell may have been the first, in 1927, at the end of the *Analysis of Matter* to comment that intrinsic nature and by implication intrinsic value only exists in conscious experience.

> As regards the world in general, both physical and mental, everything that we know of its intrinsic character is derived from the mental side, and almost everything that we know of its causal laws is derived from the physical side. But from the standpoint of philosophy the distinction between physical and mental is superficial and unreal[14, p. 402].

Science first abandoned the fundamental substances of earth, air, fire and water and later Newtonian billiard balls for pure mathematical models lacking any fundamental substance. This is made explicit in set theory where the fundamental entity is the empty set or nothing at all. Intrinsic nature and thus meaning and value exists only in conscious experience.

Nonetheless the evolution of consciousness has been an evolution of structure. Reproducing molecules have evolved to create the depth and richness of human consciousness. They have also evolved to the point where we can take conscious control over future human evolution. Human genetic engineering has already begun as a way to cure or prevent horrible diseases. Over time the techniques will be perfected to the point where one may consider using them for human enhancement. We will need to have a sense of meaning and values that is up to the challenge this capability presents.

The depth and richness of human consciousness seems to require the level of abstraction and self reflection that has evolved. These seem necessary for both richness of human consciousness and the ability to create mathematics. The ordinal numbers are the backbone of mathematics determining what problems are decidable and what objects are definable in a mathematical system. Do they also impose limits on the depth and richness of human consciousness? If so than diversity is critical to the unbounded exploration of possible conscious experience. This possibility is explored more fully in a video *Mathematical Infinity and Human Destiny* and a book[1].

References

[1] Paul Budnik. *What is and what will be: Integrating spirituality and science.* Mountain Math Software, Los Gatos, CA, 2006. 17

[2] Paul Budnik. Emergent Properties of Discretized Wave Equations. *Complex Systens*, 19(2), 2011. 16

[3] Paul Budnik. *A Computational Approach to the Ordinal Numbers: Documents ordCalc 0.3.* Mountain Math Software, Los Gatos, CA, 2011. 5

[4] Paul J. Cohen. *Set Theory and the Continuum Hypothesis*. W. A. Benjamin Inc., New York, Amsterdam, 1966. 14

[5] Solomon Feferman. Does mathematics need new axioms? *American Mathematical Monthly*, 106:99–111, 1999. 15

[6] Harvey M. Friedman. Finite functions and the necessary use of large cardinals. *ANN.OF MATH.*, 2:148, 1998. 2

[7] Jean H. Gallier. What's so Special About Kruskal's Theorem And The Ordinal Γ_0? A Survey Of Some Results In Proof Theory. *Annals of Pure and Applied Logic*, 53(2):199–260, 1991. 4

[8] S. C. Kleene. On notation for ordinal numbers. *Journal of Symbolic Logic*, 3(4), 1938. 2

[9] Benoît Mandelbrot. Fractal aspects of the iteration of $z \mapsto \lambda z(1-z)$ for complex λ and z. *Annals of the New York Academy of Sciences*, 357:49–259, 1980. 2

[10] Larry W. Miller. Normal functions and constructive ordinal notations. *The Journal of Symbolic Logic*, 41(2):439–459, 1976. 4, 5

[11] Emil L. Post. Recursively enumerable sets of positive integers and their decision problems. *Bulletin American Mathematical Society*, 50(5):284–316, 1944. 15

[12] Michael Rathjen. The art of ordinal analysis. In *Proceedings of the International Congress of Mathematicians, Volume II*, pages 45–69. European Mathematical Society, 2006. 5

[13] Hartley Rogers Jr. *Theory of Recursive Functions and Effective Computability*. McGraw Hill, New York, 1967. 2

[14] Bertrand Russell. *The Analysis of Matter*. Paul Kegan, London, 1927. 17

[15] Gerald Sacks. *Higher recursion theory*. Springer Verlag, New York, 1990. 15

[16] Oswald Veblen. Continuous increasing functions of finite and transfinite ordinals. *Transactions of the American Mathematical Society*, 9(3):280–292, 1908. 4, 5