A Tutorial Overview of Ordinal Notations

Jacques Bailhache (jacques.bailhache@gmail.com)

January-April 2018

1 Interest of transfinite ordinal numbers

The domain of transfinite ordinal numbers, or ordinals, has the particularity of being the only mathematical domain that cannot be automated. In all other domains of mathematics, it is at least theoretically possible to deduce the theorems automatically from a formal system consisting of a finite set of axioms and rules. But Gödel proved that given any formal system of a theory sufficiently powerful to contain arithmetics, it is possible to build a proposition that expresses its own unprovability in this formal system. This proposition, which is very huge, has also a meaning as an ordinary arithmetic proposition, but is very useless in ordinary arithmetics. If the formal system is consistent, then this proposition is undecidable.

At first sight one could think that we just have to add this proposition to the system as an axiom, but this augmented system also have its own Gödelian proposition. By adding successively Gödelian propositions, we obtain an infinite sequence of systems, and the system defined as the union of all these systems also has its Gödelian proposition, and so on. But according to Solomon Feferman in "Penrose's Gödelian argument" http://math.stanford.edu/ feferman/papers/penrose.pdf p.9 :

"one obtains completeness for all arithmetical sentences in a progression based on the transfinite iteration of the so-called global or uniform reflection principle"

The uniform reflection principle, which is something similar to adding the Gödelian proposition as an axiom, is described for example in John Harrison's paper "Metatheory and Reflection in Theorem Proving: A Survey and Critique" http://www.cl.cam.ac.uk/ jrh13/papers/reflect.ps.gz p.18 :

$$\vdash \forall n. Pr(\lceil \phi[n] \rceil) \Rightarrow \phi[n]$$

Harrison also says p.19 :

"Feferman showed that a transfinite iteration based on it proves all true sentences of number theory".

So we can say that the construction of transfinite ordinals can store all the creative part of mathematics.

2 Mathematical reminders

2.1 Combinatory logic and lambda calculus

These theories are formalization of the notion of information processing.

Everything is represented by information processing or functions, even data. An elementary piece of data, like a boolean data (true or false), can be represented by a function with two variables, which gives the first one if the value is true, or the second one if the value is false. A structured information, for example a couple of value, is represented by a function that, when applied to the boolean true value, gives the first value of the couple, and when applied to the boolean false, gives the second value of the couple.

A function with two variables is represented by a function that, when applied to the first variable, gives another function which, when applied to the second variable, gives the final result, and so on. A function that gives several result can be represented by a function that gives a structured containing all the results. So we have to consider only functions that, when applied to one variable, also called "argument" or sometimes "parameter", give one result.

The application of a function f to a variable x is written "f x".

"(f x) y" may be written simply "f x y".

The following cases can be distinguished according to the relationship between the variable to which the function is applied and the result of the application of this function to this variable :

- The result is the variable itself : the function is called identity, written "I". For any x, we have I x = x.
- The result is y which does not depend on the variable. The function is a constant function which always gives y as result. It is written "K y". For any x and y, we have K y x = y.

- The result is the result of the application of a to b, where both a and b may depend on the variable. In this case, the function is written "S f g" where f is a function that gives a when applied to the variable and g is a function that gives b when applied to the variable. For any f, g and x, S f g x = f x (g x).

Any function can be represented by applications of I, K and S, or even only K and S, because I = S K K. This is called "combinatory logic".

But with this representation we obtain huge expressions difficult to read and understand. So we will introduce a notation to represent the function that, when applied to a variable x, gives a result M, where M represents an expression that may contain one or several occurences of x. Different notations are used, depending on typographic possibilities, for example :

- M with x replaced by x̂ (Principia Mathematica)
- x̂.M (original notation not very used)
- ˆx.M
- \x.M
- $\lambda x.M$ (probably the most used notation in lambda calculus)
- $(\lambda x M)$
- $\lambda x[M]$
- $[x].M$
- $x \mapsto M$
- $\lambda x \to M$ or \x -> M (in Haskell)

$\lambda x.\lambda y.\lambda z.M$ may be written $\lambda xyz.M$.
This is the lambda calculus notation.
The combinatory logic representation of a function can be retrieved from its lambda calculus representation using the following correspondence rules :

- $\lambda x.x = I$
- $\lambda x.y = Ky$ if y does not contain x
- $\lambda x.(ab) = S(\lambda x.a)(\lambda x.b)$

$(\lambda x.M)N$ is the result of the substitution of x by N in M.

The lambda calculus notation has a disadvantage for example $\lambda x.x$ and $\lambda y.y$ represent the same function although they are different expressions. To avoid this disadvantage, we can use De Bruijn index. With this notation, this function is written $\lambda 1$. Each occurence of a variable is replaced by a natural number n which means the variable corresponding to the n-th *lambda* in which it is nested, starting from the innermost.
With this notation we have :

- $I = \lambda 1$
- $K = \lambda\lambda 2$
- $S = \lambda\lambda\lambda 31(21)$

I will sometime use the notation $[\ldots * \ldots]$ or $[\ldots \bullet \ldots]$ for $\lambda \ldots 1 \ldots$.

See also https://ryanflannery.net/research/logic-notes/Church-CalculiOfLambdaConversion.pdf for more information about combinatory logic and lambda calculus.

2.2 Natural numbers

Natural numbers are defined by Peano axioms :

- 0 is a natural number.
- Every natural number has a successor.
- 0 is not the successor of any natural number.
- If two natural numbers have the same successor, they are equal.
- If 0 has a property, and if the fact that some natural number has this property implies that its successor also has this property, then every natural number has this property.

Arithmetical operations are defined as follow, where suc(n) represents the successor of the natural number n :

- addition : $a + 0 = a; a + suc(b) = suc(a + b)$
- multiplication : $a \cdot 0 = 0; a \cdot suc(b) = (a \cdot b) + a$
- exponentiation : $a^0 = 1; a^{suc(b)} = a^b \cdot a$

For natural numbers, the addition and the multiplication are commutative : $a + b = b + a; a \cdot b = b \cdot a$, but not the exponentiation : generally $a^b \neq b^a$.
We shall see later that the addition and the multiplication of transfinite ordinal numbers are not commutative.

2.3 Composition and iteration of functions

The composition of two functions f and g, written B f g or $f \circ g$ is a function satisfying $(f \circ g)x = f(gx)$.
The composition of a function with itself $f \circ f$ can be written f^2.
More generally, the n-th iterate f^n of the function f is defined by :

- $f^0 = I$
- $f^{suc(n)} = f \circ f^n$

and has the following properties :

- $f^{a+b} = f^b \circ f^a$
- $f^{a \cdot b} = (f^a)^b$

2.4 Different ways of representing finite sequences

A finite sequence of length n of elements of a given set S can be considered as a function which, to each natural number less than n, associates an element of S.
For example, we can define a finite sequence of length 4 of natural numbers by the function f defined by :

- f(0) = 4
- f(1) = 3
- f(2) = 0
- f(3) = 8

There are different ways to represent such a sequence :

- Comma separated list, from left to right : 4,3,0,8
- Comma separated list, from right to left : 8,0,3,4
- Matrix with values and ranks : $\begin{pmatrix} 4 & 3 & 0 & 8 \\ 0 & 1 & 2 & 3 \end{pmatrix}$
- Matrix with values and ranks, omitting null values : $\begin{pmatrix} 4 & 3 & 8 \\ 0 & 1 & 3 \end{pmatrix}$
- Polynom : $8x^3 + 3x + 4$
- A number whose representation in base n is the considered sequence, where n in any number greater than all numbers of the sequences, for example for n = 10, the number 8034. This is also the value of the polynom for x = n.

Some representations of ordinals use finite sequences of ordinals. Different ways of representing sequences are used by these representations, for example comma separated list for Veblen function with finitely many variables, matrix with values and ranks for Schütte bracket or Klammersymbol, or base Ω representation for collapsing functions, where Ω is an ordinal which is greater than all ordinals of the sequence, for example if we want to represent sequences of countable ordinals, we can use for Ω the least uncountable ordinal ω_1.

3 Defining transfinite ordinal numbers

Natural numbers can be represented by sets. Each number is represented by the set of all numbers smaller than it.

- $0 = \{\}$ (the empty set)
- $1 = \{0\} = \{\{\}\}$
- $2 = \{0, 1\} = \{\{\}, \{\{\}\}\}$
- $3 = \{0, 1, 2\} = \{\{\}, \{\{\}\}, \{\{\}, \{\{\}\}\}\}$
- ...

The successor of a natural number can be defined by $suc(n) = n + 1 = n \cup \{n\}$.

We have $n \leq p$ if and only if $n \subseteq p$.

$\mathbb{N}$ is the set of all natural numbers : $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ The natural numbers can be generalized to what is called "transfinite ordinal numbers", or simply "ordinal numbers" or "ordinals", by considering that infinite sets represent ordinal numbers. $\mathbb{N}$ considered as an ordinal number is written ω. ω is the least ordinal which is greater than all the numbers 0, 1, 2, 3, ... We say that ω is a limit ordinal and 0, 1, 2, 3, ... is a fundamental sequence of ω. This is written : $\omega = sup\{0, 1, 2, 3, \ldots\}$ or $\omega = lim(n \mapsto n)$ because the n-th element (starting with 0) of the sequence is n. An ordinal does not have a unique fundamental sequence, for example 1, 2, 3, 4, ... is also a fundamental sequence of ω, because the least ordinal that is greater than all ordinals of this sequence is also ω (more generally the limit ordinal is the same if any number of the least items of a sequence are removed), and the same stands for the sequence 0, 2, 4, 6, ...

Any ordinal can be defined as the least ordinal strictly greater than all ordinals of a set : the empty set for 0, $\{\alpha\}$ for the successor of α, $\{\alpha_0, \alpha_1, \alpha_2, \ldots\}$ for an ordinal with fundamental sequence $\alpha_0, \alpha_1, \alpha_2, \ldots$

The successor can be generalized to transfinite ordinal numbers : $suc(\omega) = \omega + 1 = \omega \cup \{\omega\} = \{0, 1, 2, 3, \ldots, \omega\}$; $suc(suc(\omega)) = \omega + 2 = \{0, 1, 2, 3, \ldots, \omega, \omega + 1\}$ and so on.

Then we can consider the set $\{0, 1, 2, 3, \ldots, \omega, \omega + 1, \omega + 2, \omega + 3, \ldots\}$ which is a limit ordinal, and $\omega, \omega + 1, \omega + 2, \omega + 3, \ldots$ is a fundamental sequence of this ordinal. This ordinal is $\omega + \omega = \omega \cdot 2$ or $\omega \times 2$ or $\omega 2$.

Then we can go on building greater and greater ordinals : $\omega \cdot 3, \ldots, \omega \cdot \omega = \omega^2, \omega^3, \ldots, \omega^\omega, \omega^{\omega^\omega}, \ldots$.

For natural numbers, arithmetical operations are defined as follows :

- addition : $a + 0 = a; a + suc(b) = suc(a + b)$
- multiplication : $a \cdot 0 = 0; a \cdot suc(b) = (a \cdot b) + a$
- exponentiation : $a^0 = 1; a^{suc(b)} = a^b \cdot a$

The definitions of arithmetical operations can be generalized to ordinals by adding canonical rules for limit ordinals :

- addition : $\alpha + 0 = \alpha; \alpha + suc(\beta) = suc(\alpha + \beta); \alpha + lim(f) = lim(n \mapsto \alpha + f(n))$
- multiplication : $\alpha \cdot 0 = 0; \alpha \cdot suc(\beta) = (\alpha \cdot \beta) + \alpha; \alpha \cdot lim(f) = lim(n \mapsto \alpha \cdot f(n))$
- exponentiation : $\alpha^0 = 1; \alpha^{suc(\beta)} = \alpha^\beta \cdot \alpha; \alpha^{lim(f)} = lim(n \mapsto \alpha^{f(n)})$

Note that addition and multiplication are not commutative, for example $1 + \omega = \omega \neq \omega + 1$, because if we take 0, 1, 2, 3, ... as fundamental sequence of ω, then a fundamental sequence of $1 + \omega$ is 1+0, 1+1, 1+2, 1+3, ... = 1, 2, 3, 4, ... and the least ordinal which is greater than all ordinals of this sequence is ω. We will say that "1+" is "absorbed" by ω. More generally, we have $1 + \alpha = \alpha$ for any ordinal $\alpha \geq \omega$.

For tutorial introductions to transfinite ordinal numbers, see also :

- Madore's introduction in French :
  http://www.madore.org/%7Edavid/weblog/2011-09-18-nombres-ordinaux-intro.html
- Pointless Gigantic List of Infinite Numbers :
  https://sites.google.com/site/pointlesslargenumberstuff/home/l/pglin?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F
- Sbiis Saibian's !!! FORBIDDEN LIST !!! of Infinite Numbers :
  https://sites.google.com/site/largenumbers/home/appendix/a/infinite_numbers

4 Veblen functions

The next step is the limit or least upper bound of $\omega, \omega^\omega, \omega^{\omega^\omega}, \ldots$, written $sup\{\omega, \omega^\omega, \omega^{\omega^\omega}, \ldots\}$ which is called ε_0. Note that we have $\omega^{\varepsilon_0} = \varepsilon_0$. We say that ε_0 is a fixed point (the least one) of the function $\alpha \mapsto \omega^\alpha$.

Then we can go on with $\varepsilon_0 + 1, \varepsilon_0 + 2, \ldots, \varepsilon_0 + \varepsilon_0 = \varepsilon_0 \cdot 2, \ldots, \varepsilon_0 \cdot \varepsilon_0 = \varepsilon_0{}^2, \varepsilon_0^{\varepsilon_0}, \ldots$

The limit of $\varepsilon_0, \varepsilon_0^{\varepsilon_0}, \varepsilon_0^{\varepsilon_0^{\varepsilon_0}}, \ldots$ is called ε_1. It can be shown that it is also the limit of $\varepsilon_0 + 1, \omega^{\varepsilon_0+1}, \omega^{\omega^{\varepsilon_0+1}}, \ldots$ (see proof below). These two fundamental sequences are examples of two ways of ascending ordinals :

- Build greater ordinals from known ones by increasing them using operations like successor, addition, multiplication, exponentiation, ... This method is used by the RSH0 notation which we will study later.
- When we have found a function that, when applied to a given ordinal, gives a greater one (for example $\alpha \mapsto \omega^\alpha$), use it ad infinitum starting for example with 0, and then to go further use it ad infinitum starting with the successor of the previous result, and so on. This is called an enumeration of the fixed points of this function. A fixed point of a function f is a value (for example an ordinal) α with $f(\alpha) = \alpha$. Under some conditions (see below), the least fixed point of f is the limit of 0, f(0), f(f(0)), f(f(f(0))), ... If it is called α, the next fixed point is the limit of $\alpha + 1, f(\alpha+1), f(f(\alpha+1)), f(f(f(\alpha+1))), \ldots$. More generally, the least fixed point of f that is greater or equal to ζ is the limit of $\zeta, f(\zeta), f(f(\zeta)), \ldots$. The Veblen functions use this method.

The required conditions are described for example in http://www.cs.man.ac.uk/ hsimmons/ORDINAL-NOTATIONS/Fruitful.pdf page 8 lemma 3.9 :
For each fruitful function f and each ordinal ζ, $f^\omega(\zeta + 1)$ is the least ordinal ν such that $\zeta < \nu = f(\nu)$, or the least fixed point of f that is strictly greater than ζ (or greater than or equal to $\zeta + 1$).
$f^\omega(\zeta + 1)$ is the limit of $\zeta + 1, f(\zeta + 1), f(f(\zeta + 1)), \ldots$.
A fruitful function is a function that is inflationary, monotone, big, and continuous.
A function f is inflationary if $\alpha \leq f(\alpha)$, monotone if $\alpha \leq \beta \Rightarrow f(\alpha) \leq f(\beta)$, big if $\omega^\alpha \leq f(\alpha)$ except possibly for $\alpha = 0$, continuous if f(VA) = Vf[A] where VA is the pointwise supremum of the set A.

We will now prove by induction the equivalence of the two fundamental sequences above.

We will use the notation $\alpha^{\cdot^{\cdot^{\cdot^{\alpha^\beta}}}}$ for an an "exponential tower" with α repeated n times.

Note that the ordinals respectively limits of the fondamental sequence whose n-th term is $\varepsilon_0^{\varepsilon_0^{\cdot^{\cdot^{\varepsilon_0^\omega}}}}$ and the one whose n-th term is $\varepsilon_0^{\varepsilon_0^{\cdot^{\cdot^{\varepsilon_0^{\varepsilon_0}}}}}$ is the same, the least fixed point of the function $\alpha \mapsto \varepsilon_0{}^\alpha$, which is greater than ω and also than ε_0.

So we have proved what we want if we prove that, for any n, we have $\omega^{\omega^{\omega^{\cdot^{\cdot^{\omega^{\omega^{\varepsilon_0+1}}}}}}} = \varepsilon_0^{\varepsilon_0^{\cdot^{\cdot^{\varepsilon_0^\omega}}}}$.

For n = 0, we have $\omega^{\omega^{\varepsilon_0+1}} = \omega^{\omega^{\varepsilon_0} \cdot \omega} = \omega^{\varepsilon_0 \cdot \omega} = (\omega^{\varepsilon_0})^\omega = \varepsilon_0{}^\omega$.

Now suppose we have $\omega^{\omega^{\cdot^{\cdot^{\omega^{\omega^{\varepsilon_0+1}}}}}} = \varepsilon_0^{\varepsilon_0^{\cdot^{\cdot^{\varepsilon_0^\omega}}}}$.

We must prove the equality for n+1, which can be written $\omega^{\omega^{\omega^{\cdot^{\cdot^{\omega^{\omega^{\varepsilon_0+1}}}}}}} = \varepsilon_0^{\varepsilon_0^{\cdot^{\cdot^{\varepsilon_0^\omega}}}}$.

We have $\omega^{\omega^{\omega^{\cdot^{\cdot^{\omega^{\omega^{\varepsilon_0+1}}}}}}} = \omega^{\varepsilon_0^{\varepsilon_0^{\cdot^{\cdot^{\varepsilon_0^\omega}}}}}$ (by our hypothesis) $= \omega^{\varepsilon_0^{1+\varepsilon_0^{\cdot^{\cdot^{\varepsilon_0^\omega}}}}}$ (for the same reason than $1 + \omega = \omega$, see above) $= \omega^{\varepsilon_0 \cdot \varepsilon_0^{\cdot^{\cdot^{\varepsilon_0^\omega}}}} =$

$(\omega^{\varepsilon_0})^{\varepsilon_0^{\cdot^{\cdot^{\varepsilon_0^\omega}}}} = \varepsilon_0^{\varepsilon_0^{\cdot^{\cdot^{\varepsilon_0^\omega}}}}$. QED.

In a similar way, the limit of $\varepsilon_1, \varepsilon_1^{\varepsilon_1}, \varepsilon_1^{\varepsilon_1^{\varepsilon_1}}, \ldots$ is called ε_2 and is also the limit of $\varepsilon_1 + 1, \omega^{\varepsilon_1+1}, \omega^{\omega^{\varepsilon_1+1}}, \ldots$.

We can define the same way ε_n for any natural number n. Then ε_ω is defined as the limit of $\varepsilon_0, \varepsilon_1, \varepsilon_2, \varepsilon_3, \ldots$, and $\varepsilon_{\omega+1}$ as the limit of $\varepsilon_\omega, \varepsilon_\omega^{\varepsilon_\omega}, \varepsilon_\omega^{\varepsilon_\omega^{\varepsilon_\omega}}, \ldots$ or $\varepsilon_\omega + 1, \omega^{\varepsilon_\omega+1}, \omega^{\omega^{\varepsilon_\omega+1}}, \ldots$.

For any ordinal $\alpha, \varepsilon_{\alpha+1}$ is the limit of $\varepsilon_\alpha, \varepsilon_\alpha^{\varepsilon_\alpha}, \varepsilon_\alpha^{\varepsilon_\alpha^{\varepsilon_\alpha}}, \ldots$ and is also the limit of $\varepsilon_\alpha + 1, \omega^{\varepsilon_\alpha+1}, \omega^{\omega^{\varepsilon_\alpha+1}}, \ldots$.

After comes $\varepsilon_{\varepsilon_0}$, and the limit of $\varepsilon_0, \varepsilon_{\varepsilon_0}, \varepsilon_{\varepsilon_{\varepsilon_0}}, \ldots$ which is called ζ_0. This is the least fixed point of $\alpha \mapsto \varepsilon_\alpha$. The next one is ζ_1 which is the limit of $\zeta_0 + 1, \varepsilon_{\zeta_0+1}, \varepsilon_{\varepsilon_{\zeta_0+1}}, \ldots$. Then we get $\zeta_2, \zeta_3, \ldots, \zeta_\omega, \zeta_{\omega+1}, \ldots, \zeta_{\varepsilon_0}, \ldots, \zeta_{\zeta_0}, \ldots, \zeta_{\zeta_{\zeta_0}}, \ldots$. The limit of $0, \zeta_0, \zeta_{\zeta_0}, \zeta_{\zeta_{\zeta_0}}, \ldots$ is called η_0.
We can go on using successively different greek letters, or we can use functions indiced by numbers

- $\varphi_0(\alpha) = \omega^\alpha$
- $\varphi_1(\alpha) = \varepsilon_\alpha$
- $\varphi_2(\alpha) = \zeta_\alpha$
- $\varphi_3(\alpha) = \eta_\alpha$
- $\varphi_{\alpha+1}(\beta)$ is the $(1+\beta)$-th fixed point of $\xi \mapsto \varphi_\alpha(\xi)$.

or a function with two variables :

- $\varphi(0, \alpha) = \omega^\alpha$
- $\varphi(1, \alpha) = \varepsilon_\alpha$
- $\varphi(2, \alpha) = \zeta_\alpha$
- $\varphi(3, \alpha) = \eta_\alpha$
- $\varphi(\alpha + 1, \beta)$ is the $(1+\beta)$-th fixed point of $\xi \mapsto \varphi(\alpha, \xi)$.

Then we can enumerate the fixed points of the function $\alpha \mapsto \varphi(\alpha, 0)$ and define Γ_α as the $(1+\alpha)$-th fixed point of this function, or add another variable to the φ function and define $\varphi_{1,0}(\alpha)$ or $\varphi(1, 0, \alpha)$ as the $(1+\alpha)$-th fixed point of this function. So we have $\Gamma_\alpha = \varphi_{1,0}(\alpha) = \varphi(1, 0, \alpha)$.
More generally, we can define a function with any (finite) number of variables $\varphi_{\alpha_n, \alpha_{n-1}, ..., \alpha_1, \alpha_0}(\beta) = \varphi(\alpha_n, \alpha_{n-1}, \ldots, \alpha_1, \alpha_0, \beta)$, with $\varphi(\alpha) = \varphi_0(\alpha) = \varphi(0, \alpha) = \omega^\alpha$.
The notation $\varphi_{\alpha_n, \alpha_{n-1}, ..., \alpha_1, \alpha_0}(\beta)$ has the advantage of highlighting the different role played by the last variable β.

For a complete definition of this Veblen function with finitely many variables, see for example :
https://en.wikipedia.org/wiki/Veblen_function :
"Let z be an empty string or a string consisting of one or more comma-separated zeros $0, 0, ..., 0$ and s be an empty string or a string consisting of one or more comma-separated ordinals $\alpha_1, \alpha_2, ..., \alpha_n$ with $\alpha_1 > 0$. The binary function $\varphi(\beta, \gamma)$ can be written as $\varphi(s, \beta, z, \gamma)$ where both s and z are empty strings.
The finitary Veblen functions are defined as follows:

- $\varphi(\gamma) = \omega^\gamma$
- $\varphi(z, s, \gamma) = \varphi(s, \gamma)$
- if $\beta > 0$, then $\varphi(s, \beta, z, \gamma)$ denotes the $(1+\gamma)$-th common fixed point of the functions $\xi \mapsto \varphi(s, \delta, \xi, z)$ for each $\delta < \beta$

(...)
The limit of the $\varphi(1, 0, ..., 0)$ where the number of zeroes ranges over ω, is sometimes known as the "small" Veblen ordinal.
Every non-zero ordinal α less than the small Veblen ordinal (SVO) can be uniquely written in normal form for the finitary Veblen function:
$\alpha = \varphi(s_1) + \varphi(s_2) + \cdots + \varphi(s_k)$
where

- k is a positive integer
- $\varphi(s_1) \geq \varphi(s_2) \geq \cdots \geq \varphi(s_k)$
- s_m is a string consisting of one or more comma-separated ordinals $\alpha_{m,1}, \alpha_{m,2}, ..., \alpha_{m,n_m}$ where $\alpha_{m,1} > 0$ and each $\alpha_{m,i} < \varphi(s_m)$

For limit ordinals $\alpha < SVO$, written in normal form for the finitary Veblen function:

- $(\varphi(s_1) + \varphi(s_2) + \cdots + \varphi(s_k))[n] = \varphi(s_1) + \varphi(s_2) + \cdots + \varphi(s_k)[n]$,
- $\varphi(\gamma)[n] =$
  - n if $\gamma = 1$
  - $\varphi(\gamma - 1) \cdot n$ if γ is a successor ordinal
  - $\varphi(\gamma[n])$ if γ is a limit ordinal
- $\varphi(s, \beta, z, \gamma)[0] = 0$ and $\varphi(s, \beta, z, \gamma)[n+1] = \varphi(s, \beta - 1, \varphi(s, \beta, z, \gamma)[n], z)$ if $\gamma = 0$ and β is a successor ordinal,
- $\varphi(s, \beta, z, \gamma)[0] = \varphi(s, \beta, z, \gamma - 1) + 1$ and $\varphi(s, \beta, z, \gamma)[n+1] = \varphi(s, \beta - 1, \varphi(s, \beta, z, \gamma)[n], z)$ if γ and β are successor ordinals,
- $\varphi(s, \beta, z, \gamma)[n] = \varphi(s, \beta, z, \gamma[n])$ if γ is a limit ordinal,
- $\varphi(s, \beta, z, \gamma)[n] = \varphi(s, \beta[n], z, \gamma)$ if $\gamma = 0$ and β is a limit ordinal,
- $\varphi(s, \beta, z, \gamma)[n] = \varphi(s, \beta[n], \varphi(s, \beta, z, \gamma - 1) + 1, z)$ if γ is a successor ordinal and β is a limit ordinal. "

The Veblen function can be generalized to transfinitely many variables with a finite number different from 0. Instead of writing the list of all the variable of the Veblen function, we can write only the non zero variables with position as indice, for example $\varphi(\alpha, 0, \beta, \gamma) = \varphi(\alpha_3, \beta_1, \gamma_0)$. We can then generalize the Veblen function by allowing any ordinal as indices, writing for example $SVO = \varphi(1_\omega)$. The limit of the ordinals that can be written with this notation is called the large Veblen ordinal (LVO).

According to Wikipedia, "The definition can be given as follows: let α be a transfinite sequence of ordinals (i.e., an ordinal function with finite support) which ends in zero (i.e., such that $\alpha_0 = 0$), and let $\alpha[0 \mapsto \gamma]$ denote the same function where the final 0 has been replaced by γ. Then $\gamma \mapsto \varphi(\alpha[0 \mapsto \gamma])$ is defined as the function enumerating the common fixed points of all functions $\xi \mapsto \varphi(\beta)$ where β ranges over all sequences which are obtained by decreasing the smallest-indexed nonzero value of α and replacing some smaller-indexed value with the indeterminate ξ (i.e., $\beta = \alpha[\iota_0 \mapsto \zeta, \iota \mapsto \xi]$ meaning that for the smallest index ι_0 such that α_{ι_0} is nonzero the latter has been replaced by some value $\zeta < \alpha_{\iota_0}$ and that for some smaller index $\iota < \iota_0$, the value $\alpha_\iota = 0$ has been replaced with ξ)."

Schütte brackets or Klammersymbols are another way to write Veblen fuctions with transfinitely many variables. A Schütte bracket consists in a matrix with two lines, with the positions of the variables in the second line in increasing order, and the corresponding values in the first line. This matrix is preceded by the function $\xi \mapsto \varphi(\xi)$. If we take $\xi \mapsto \omega^\xi$, we get the equivalent of the Veblen function. With this notation, the previous example is written :

$$(\xi \mapsto \omega^\xi) \begin{pmatrix} \gamma & \beta & \alpha \\ 0 & 1 & 3 \end{pmatrix}$$

In some of his papers, Harold Simmons puts the function after the matrix, which is more logical, the matrix being considered as a function which, when applied to a function, gives an ordinal :

$$\begin{pmatrix} \gamma & \beta & \alpha \\ 0 & 1 & 3 \end{pmatrix} (\xi \mapsto \omega^\xi)$$

When the function at the left of the matrix is $\xi \mapsto \omega^\xi$, it is sometimes omitted. Example :

$$\begin{pmatrix} \gamma & \beta & \alpha \\ 0 & 1 & 3 \end{pmatrix}$$

The corresponding fundamental sequences can be found in https://sites.google.com/site/travelingtotheinfinity/fundamental-sequences-for-extended-veblen-function .

Another possible notation is to represent the parameters of the φ function by a polynom of variable Ω where the exponent corresponds to the position of the variable, for example $\varphi(\alpha, 0, \beta, \gamma) = \varphi(\gamma_0, \beta_1, \alpha_3) = (\xi \mapsto \omega^\xi) \begin{pmatrix} \gamma & \beta & \alpha \\ 0 & 1 & 3 \end{pmatrix} = \varphi(\Omega^3 \cdot \alpha + \Omega \cdot \beta + \gamma)$.

For Ω, we can choose an ordinal which is greater than all the ordinals we want to produce. Since they all are countable, we can take for example $\Omega = \omega_1$ which is the least uncountable ordinal. The method consisting in using uncountable ordinals to define countable ordinals is called "collapsing". We will see later other examples of notations using this method.
Note that $\varphi(1, 0) = \varphi(\Omega)$ is the least α such that $\alpha = \varphi(\alpha) = \omega^\alpha$ (the least fixed point of $\alpha \mapsto \omega^\alpha$); $\varphi(1, 0, 0) = \varphi(\Omega^2) = \varphi(\Omega \cdot \Omega)$ is the least α such that $\alpha = \varphi(\alpha, 0) = \varphi(\Omega \cdot \alpha)$. Generally speaking, we can see that $f(\Omega)$ is the least fixed point of f. We shall see other examples of this equality later concerning ordinal collapsing functions. Note also that "Ω" can be replaced by "1,0" in the formulas.

If we want to distinguish the last variable, we can also use collapsing with the notation $\varphi_{\alpha_n, \ldots, \alpha_0}(\beta)$, writing for example $\varphi_{\alpha, \beta, \gamma}(\delta) = \varphi_{\Omega^2 \cdot \alpha + \Omega \cdot \beta + \gamma}(\delta)$, or $\varphi(\alpha, \beta, \gamma, \delta) = \varphi(\Omega^2 \cdot \alpha + \Omega \cdot \beta + \gamma, \delta)$.

See Veblen's article "Continuous Increasing Functions of Finite and Transfinite Ordinals" (http://www.ams.org/journals/tran/1908-009-03/S0002-9947-1908-1500814-9/S0002-9947-1908-1500814-9.pdf) for more information.

Here is an Agda implementation of the Veblen function with transfinitely many variables :

```
{-
    A definition of the large Veblen ordinal in Agda
    by Jacques Bailhache, March 2016
```

See https://en.wikipedia.org/wiki/Veblen_function

   (1) phi(a)=w**a for a single variable,

   (2) phi(0,an-1,...,a0)=phi(an-1,...,a0), and

   (3) for a>0, c->phi(an,...,ai+1,a,0,...,0,c) is the function enumerating the common fixed points of the
       functions x->phi(an,...,ai+1,b,x,0,...,0) for all b<a.

   (4) Let a be a transfinite sequence of ordinals (i.e., an ordinal function with finite support) which ends i
       zero (i.e., such that a0=0), and let a[0->c] denote the same function where the final 0 has been replac
       by c.
       Then c->phi(a[0->c]) is defined as the function enumerating the common fixed points of all functions
       x->phi(b) where b ranges over all sequences which are obtained by decreasing the smallest-indexed nonzer
       value of a and replacing some smaller-indexed value with the indeterminate x (i.e., b=a[i0->z,i->x]
       meaning that for the smallest index i0 such that ai0 is nonzero the latter has been replaced by some val
       z<ai0 and that for some smaller index i<i0, the value ai=0 has been replaced with x).

-}

module LargeVeblen where

```
 data Nat : Set where
  O : Nat
  1+ : Nat -> Nat

 data Ord : Set where
  Zero : Ord
  Suc : Ord -> Ord
  Lim : (Nat -> Ord) -> Ord

 -- rpt n f x = f^n(x)
 rpt : {t : Set} -> Nat -> (t -> t) -> t -> t
 rpt O f x = x
 rpt (1+ n) f x = rpt n f (f x)

 -- smallest fixed point of f greater than x, limit of x, f x, f (f x), ...
 fix : (Ord -> Ord) -> Ord -> Ord
 fix f x = Lim (\n -> rpt n f x)

 w = fix Suc Zero -- not a fixed point in this case !

 -- cantor a b = b + w^a
 cantor : Ord -> Ord -> Ord
 cantor Zero a = Suc a
 cantor (Suc b) a = fix (cantor b) a
 cantor (Lim f) a = Lim (\n -> cantor (f n) a)

 -- phi0 a = w^a
 phi0 : Ord -> Ord
 phi0 a = cantor a Zero

 -- Another possibility is to use phi'0 instead of phi0 in the definition of phi,
 -- this gives a phi function which grows slower
```

8

```
 phi'0 : Ord -> Ord
 phi'0 Zero = Suc Zero
 phi'0 (Suc a) = Suc (phi'0 a)
 phi'0 (Lim f) = Lim (\n -> phi'0 (f n))


 -- Associative list of ordinals
 infixr 40 _=>_&_
 data OrdAList : Set where
  Zeros : OrdAList
  _=>_&_ : Ord -> Ord -> OrdAList -> OrdAList


 -- Usage : phi al, where al is the associative list of couples index => value ordered by increasing values,
 -- absent indexes corresponding to Zero values


 phi : OrdAList -> Ord
 phi              Zeros  = phi0 Zero -- (1) phi(0) = w**0 = 1
 phi (Zero => a & Zeros) = phi0 a    -- (1) phi(a) = w**a
 phi (             k => Zero & al) = phi al -- eliminate unnecessary Zero value
 phi (Zero => a & k => Zero & al) = phi (Zero => a & al) -- idem
 phi (Zero => a & Zero => b & al) = phi (Zero => a & al) -- should not appear but necessary for completeness
 phi (Zero => Lim f & al) = Lim (\n -> phi (Zero => f n & al)) -- canonical treatment of limit
 phi (               Suc k => Suc b & al) = fix (\x -> phi (k => x & Suc k => b & al)) Zero
  -- (3) least fixed point
 phi (Zero => Suc a & Suc k => Suc b & al) = fix (\x -> phi (k => x & Suc k => b & al)) (Suc (phi (Zero => a &
Suc k => Suc b & al))) -- (3) following fixed points
 phi (               Suc k => Lim f & al) = Lim (\n -> phi (Suc k => f n & al)) -- idem
 phi (Zero => Suc a & Suc k => Lim f & al) = Lim (\n -> phi (k => Suc (phi (Zero => a & Suc k => Lim f & al)) &
Suc k => f n & al)) -- idem
 phi (               Lim f => Suc b & al) = Lim (\n -> phi (f n => (Suc Zero) & Lim f => b & al))
 phi (Zero => Suc a & Lim f => Suc b & al) = Lim (\n -> phi (f n => phi (Zero => a & Lim f => Suc b & al) & Lim
f => b & al))
 phi (               Lim f => Lim g & al) = Lim (\n -> phi (Lim f => g n & al))
 phi (Zero => Suc a & Lim f => Lim g & al) = Lim (\n -> phi (f n => phi (Zero => a & Lim f => Lim g & al) & Lim
f => g n & al))


 SmallVeblen = phi (w => Suc Zero & Zeros)

 LargeVeblen = fix (\x -> phi (x => Suc Zero & Zeros)) (Suc Zero)

{-
Normally it should terminate because the parameter of phi lexicographically decreases, but Agda is not clever en
so it must be called with no termination check option :

$ agda -I --no-termination-check LargeVeblen.agda

                     _
   ____            | |
  / __ \           | |
 | |__| |___   __| | ___
 |  __  / _ \/ _  |/ __\      Agda Interactive
 | |  |/ /_\ \/_| / /_| \
 |_|  |\___  /_____/     Type :? for help.
        __/ /
        \__/


The interactive mode is no longer supported. Don't complain if it doesn't work.
Checking LargeVeblen (/perso/ord/LargeVeblen.agda).
```

```
Finished LargeVeblen.
Main> phi Zeros
Suc Zero
Main> :typeOf LargeVeblen
Ord
Main>
```

5   Going beyond Veblen function with transfinitely many variables

We start with the large Veblen ordinal which is the least fixed point of the function $\alpha \mapsto \varphi(1_\alpha)$. Then we consider a function F which enumerates the fixed points of $\alpha \mapsto \varphi(1_\alpha)$. So we have LVO = F(0). The next fixed point F(1) is the limit of $LVO + 1, \varphi(1_{LVO+1}), \varphi(1_{\varphi(1_{LVO+1})}), ...$

Then we can consider the fixed points of the function F and define a function G which enumerates these fixed points, then a function H which enumerates the fixed points of G, and so on.

This construction is similar to ε which enumerates the fixed points of $\alpha \mapsto \omega^\alpha$, ζ which enumerates the fixed points of ε, η which enumerates the fixed points of ζ.

Like we have defined :

- $\varphi_0(\alpha) = \omega^\alpha$
- $\varphi_1(\alpha) = \varepsilon(\alpha)$
- $\varphi_2(\alpha) = \zeta(\alpha)$

...

we can define :

- $\varphi_0^+(\alpha) = F(\alpha)$
- $\varphi_1^+(\alpha) = G(\alpha)$
- $\varphi_2^+[\alpha) = H(\alpha)$

...

With this notation we can write $LVO = \varphi_0^+(0)$.

Then $\varphi_\alpha^+(\beta)$ can be written as a binary function $\varphi^+(\alpha, \beta)$ which can be generalized to finitely many variables like $\varphi^+(\alpha, \beta, \gamma)$ and transfinitely many variables like $\varphi^+(1_\omega)$.

Then we can consider the fixed points of the function $\alpha \mapsto \varphi^+(1_\alpha)$ and define a function φ_0^{++} which enumerates these fixed points.

The same way we can define φ^{+++}, φ^{++++}, ...

We can then define a new notation :

- $\Phi_0 = \varphi$
- $\Phi_1 = \varphi^+$
- $\Phi_2 = \varphi^{++}$

...

There is another way to express this construction.

There are different conventions for $\varphi_0(x)$, like ω^x or ε_x. We can write explicitely the convention chosen for φ_0 by writing ''$\varphi_f(\alpha, \beta)$'' for ''$\varphi_\alpha(\beta)$ with function f used for φ_0''. With this notation we have:

- $\varphi_f(0, \beta) = f(\beta)$
- $\varphi_f(\alpha + 1, \beta) = (1 + \beta)$th fixed point of the function $\beta \mapsto \varphi_f(\alpha, \beta)$
- $\varphi_f(\lambda, \beta) = (1 + \beta)$th common fixed point of the function $\beta \mapsto \varphi_f(\alpha, \beta)$ for all $\alpha < \lambda$, if λ is a limit ordinal.

(See http://www.cs.man.ac.uk/ hsimmons/TEMP/OrdNotes.pdf)

Then we generalize the binary function $\varphi_f(\alpha, \beta)$ to finitely many variables: for example $\varphi_f(1, 0, \alpha) = (1 + \alpha)$th common fixed point of the function $\xi \mapsto \varphi(\xi, 0)$ (see https://en.wikipedia.org/wiki/Veblen_function) and to infinitely many variables with a finite number of them different from 0, for example $\varphi_f(1_\omega)$.

Then we can define new φ functions by taking for φ_0 the function $\xi \mapsto \varphi_f(1_\xi)$ and define functions $\varphi_{\xi \mapsto \varphi_f(1_\xi)}$ with 2 variables, with finitely many variables and with transfinitely many variables.

To make a correspondence with my previous construction, if f is the function $\xi \mapsto \omega^\xi$, then $\varphi_f(\alpha, \beta)$ corresponds to what I wrote $\varphi_\alpha(\beta)$, and $\varphi_{\xi \mapsto \varphi_f(1_\xi)}(\alpha, \beta)$ to $\varphi_\alpha^+(\beta)$.

If we define the function S by $S(f)(\xi) = \varphi_f(1_\xi)$, then $\varphi_{\xi \mapsto \varphi_f(1_\xi)}$ can be written $\varphi_{S(f)}$. We can then consider $\varphi_{S(S(f))}$ and so on.

Given an ordinal α, we can iterate transfinitely "α times" the application of S to an initial function f_0, for example $f_0(\xi) = \omega^\xi$, to obtain a function which I will write $S^\alpha(f_0)$. We can use this function to define a function $\varphi_{S^\alpha(f_0)}$ which permits to construct big ordinals.

6 Simmons notation

6.1 Presentation

Harold Simmons defined a notation (see http://www.cs.man.ac.uk/ hsimmons/ORDINAL-NOTATIONS/ordinal-notations.html) based on fixed points enumeration which "contains" Veblen functions and permits to go further.

He uses the lambda calculus formalism, in which f x represents the application of function f to x, and f x y = (f x) y the application of function f to x which gives another function which is applied to y giving the final result. He uses tho notation $x \mapsto y$ to represent the function which, when applied to x, gives y (instead of the traditional lambda calculus notation $\lambda x.y$).

He also uses the notation $\omega^\bullet$ for $\alpha \mapsto \omega^\alpha$.

$f \circ g$ represents the composition of functions f and g : $(f \circ g)\alpha = f(g\alpha)$.

f^α is a canonical generalization of of exponentiation of a function to an ordinal power : f^n represents $f \circ f \circ \ldots \circ f$ with f repeated n times, $f^\omega \zeta$ is the limit of $\zeta, f\ \zeta, f(f\ \zeta), \ldots$, $f^{\omega+1}\zeta = f(f^\omega \zeta)$ and so on.

More precisely, Simmons gives the following definitions in http://www.cs.man.ac.uk/ hsimmons/TEMP/OrdNotes.pdf page 11 :

- $g^0\zeta = \zeta$
- $g^{\alpha+1}\zeta = g(g^\alpha\zeta)$
- $g^\lambda\zeta = V\{g^\alpha\zeta | \alpha < \lambda\}$ (if λ is a limit ordinal, where V denotes the poinwise supremum)

and the following equivalent definitions in http://www.cs.man.ac.uk/ hsimmons/ORDINAL-NOTATIONS/Fruitful.pdf page 4 :

- $g^0 = id$
- $g^{\alpha+1} = g \circ g^\alpha$
- $g^\lambda = V\{g^\alpha | \alpha < \lambda\}$

and he generalizes these definitions to higher order functions.

Then Simmons defines the following functions :

$Fix\ f\zeta = f^\omega(\zeta + 1) =$ limit of $\zeta + 1, f(\zeta + 1), f(f(\zeta + 1)), \ldots$ is the least fixed point of the function f which is strictly greater than ζ, which means the least ordinal ν satisfying $f\ \nu = \nu$ and $\nu > \zeta$.

$Next = Fix\ \omega^\bullet = Fix(\alpha \mapsto \omega^\alpha)$; $Next\ \zeta$ is the next ε_α after ζ.

$[0]h = Fix(\alpha \mapsto h^\alpha 0)$

$[1]hg = Fix(\alpha \mapsto h^\alpha g0)$

$[2]hgf = Fix(\alpha \mapsto h^\alpha gf0)$

... and so on ...

In http://www.cs.man.ac.uk/ hsimmons/ORDINAL-NOTATIONS/OrdSlides.pdf Simmons gives another equivalent definition :

$[0]h = Fix(\alpha \mapsto h^\alpha \omega)$

$[1]hg = Fix(\alpha \mapsto h^\alpha g\omega)$

$[2]hgf = Fix(\alpha \mapsto h^\alpha gf\omega)$

Simmons also defines :

$Veb\ f\ \zeta = (Fix\ f)^{1+\zeta}0$ is the $(1 + \zeta)$-th fixed point of f

$Enm\ h\ \alpha = h^{1+\alpha}0$

$Veb = Enm \circ Fix$

$[0] = Fix \circ Enm$

$Fix \circ Veb = Fix \circ Enm \circ Fix = [0] \circ Fix$

$Fix \circ Veb^\alpha = [0]^\alpha \circ Fix$

$\Delta[0] = \omega$

$\Delta[1] = Next\ \omega = \varepsilon_0$

$\Delta[2] = [0]Next\ \omega =$ least ν with $\nu = Next^\nu\omega = \zeta_0$

$\Delta[3] = [1][0]Next\ \omega =$ least ν with $\nu = [0]^\nu Next\ \omega = \Gamma_0$

$\Delta[4] = [2][1][0]Next\ \omega =$ least ν with $\nu = [1]^\nu[0]Next\ \omega = LVO$ (large Veblen ordinal)

... and so on ...

6.2 Implementation

Here is an implementation of the Simmons hierarchy in Haskell :

```haskell
module Simmons where

 -- Natural numbers
 data Nat
  = ZeroN
  | SucN Nat

 -- Ordinals
 data Ord
  = Zero
  | Suc Ord
  | Lim (Nat -> Ord)

 -- Ordinal corresponding to a given natural
 ordOfNat ZeroN = Zero
 ordOfNat (SucN n) = Suc (ordOfNat n)

 -- omega
 w = Lim ordOfNat

 lim0 s = Lim s
 lim1 f x = lim0 (\n -> f n x)
 lim2 f x = lim1 (\n -> f n x)

 -- this does not work :
 -- lim ZeroN s = Lim s
 -- lim (SucN p) f = \x -> lim p (\n -> f n x)

 -- f^a(x)
 fpower0 f Zero x = x
 fpower0 f (Suc a) x = f (fpower0 f a x)
 fpower0 f (Lim s) x = Lim (\n -> fpower0 f (s n) x)

 fpower l f Zero x = x
 fpower l f (Suc a) x = f (fpower l f a x)
 fpower l f (Lim s) x = l (\n -> fpower l f (s n) x)

 -- fix f z = least fixed point of f which is > z
 fix f z = fpower lim0 f w (Suc z) -- Lim (\n -> fpower0 f (ordOfNat n) (Suc z))

 -- cantor b a = a + w^b
 cantor Zero a = Suc a
 cantor (Suc b) a = fix (cantor b) a
 cantor (Lim s) a = Lim (\n -> cantor (s n) a)

 -- expw a = w^a
 expw a = cantor a Zero

 -- next a = least epsilon_b > a
 next = fix expw

 -- [0]
```

```
 simmons0 h = fix (\a -> fpower lim0 h a Zero)

 -- [1]
 simmons1 h1 h0 = fix (\a -> fpower lim1 h1 a h0 Zero)

 -- [2]
 simmons2 h2 h1 h0 = fix (\a -> fpower lim2 h2 a h1 h0 Zero)

 -- Large Veblen ordinal
  lvo = simmons2 simmons1 simmons0 next w
```

```
$ hugs

__   __ __  __ __  __ __ __            ---------------------------------------
||   || ||  || || || ||__             Hugs 98: Based on the Haskell 98 standard
||___|| ||__|| ||__||  __||           Copyright (c) 1994-2005
||---||          ___||                World Wide Web: http://haskell.org/hugs
||   ||                               Bugs: http://hackage.haskell.org/trac/hugs
||   || || Version: September 2006 _____

Haskell 98 mode: Restart with command line option -98 to enable extensions

Type :? for help
Hugs> :load simmons
Simmons> lvo
ERROR - Cannot find "show" function for:
*** Expression : lvo
*** Of type    : Ord

Simmons>
```

6.3   Correspondence with Veblen functions

ε_0 is the next ε_α after 0 (or after ω, or after any ordinal less than ε_0, so we have $\varepsilon_0 = Next\ 0 = Next\ \omega$.
ε_1 is the next ε_α after ε_0, so we have $\varepsilon_1 = Next\ \varepsilon_0 = Next\ (Next\ 0) = Next^2 0 = Next\ (Next\ \omega) = Next^2 \omega$.
ε_2 is the next ε_α after ε_1, so we have $\varepsilon_2 = Next\ \varepsilon_1 = Next\ (Next\ (Next\ 0)) = Next^3 0 = Next\ (Next\ (Next\ \omega)) = Next^3 \omega$.
...
ε_ω is the limit of $\varepsilon_0, \varepsilon_1, \varepsilon_2, \ldots$. It is the limit of $Next^1 0, Next^2 0, Next^3 0, \ldots$ which is $Next^\omega 0$.
More generally, we have $\varepsilon_\alpha = \varphi(1, \alpha) = Next^{1+\alpha} 0 = Next^{1+\alpha} \omega$.

$\zeta_0 = \varphi(2, 0)$ is the least fixed point of $\alpha \mapsto \varepsilon_\alpha$ (greater than 0), so $\zeta_0 = Fix(\alpha \mapsto \varepsilon_\alpha)0 = Fix(\alpha \mapsto Next^{1+\alpha}0)0 = Fix(\alpha \mapsto Next^\alpha 0)0$ (because the "1+" is "absorbed" after a few iterations) $= [0]Next\ 0$. Since ζ_0 is also greater than ω, it is also $[0]Next\ \omega$ according to a similar computation.
$\zeta_1 = \varphi(2, 1)$ is the next fixed point of $\alpha \mapsto \varepsilon_\alpha$, the least one which is strictly greater than ζ_0, so $\zeta_1 = Fix(\alpha \mapsto \varepsilon_\alpha)\zeta_0 = Fix(\alpha \mapsto Next^\alpha 0)\zeta_0 = [0]Next\ \zeta_0 = [0]Next([0]Next\ 0) = ([0]Next)^2 0 = [0]Next([0]Next\ \omega) = ([0]Next)^2 \omega$.
More generally, $\zeta_\alpha = ([0]Next)^{1+\alpha} 0$.
Similar computations give $\eta_0 = \varphi(3, 0) = [0]^2 Next\ 0$ and $\eta_\alpha = ([0]^2 Next)^{1+\alpha} 0$.
More generally, $\varphi(1 + \beta, \alpha) = ([0]^\beta Next)^{1+\alpha} 0$ or $([0]^\beta Next)^{1+\alpha} \omega$.

$\Gamma_0 = \varphi(1, 0, 0)$ is the least fixed point (greater than 0) of the function $\alpha \mapsto \varphi(\alpha, 0)$ or $\alpha \mapsto \varphi(1 + \alpha, 0)$ (for the same reason of "absorbsion" of "1+" than previously), so $\Gamma_0 = Fix(\alpha \mapsto \varphi(1 + \alpha, 0)0 = Fix(\alpha \mapsto ([0]^\alpha Next)^{(1 + 0)}0)0 = Fix(\alpha \mapsto [0]^\alpha Next\ 0)0 = [1][0]Next\ 0$.

13

$\Gamma_1 = \varphi(1,0,1)$ is the next fixed point : $\Gamma_1 = Fix(\alpha \mapsto [0]^\alpha Next\ 0)\Gamma_0 = [1][0]Next\ \Gamma_0 = [1][0]Next\ ([1][0]Next\ 0) = ([1][0]Next)^2 0$.

More generally, we have $\varphi(1,0,\alpha) = ([1][0]Next)^{1+\alpha}0$.

$\varphi(1,1,0)$ is the least fixed point (greater than 0) of the function $\alpha \mapsto \varphi(1,0,\alpha)$, so it is $Fix(\alpha \mapsto \varphi(1,0,\alpha))0 = Fix(\alpha \mapsto ([1][0]Next)^{1+\alpha}0)0 = Fix(\alpha \mapsto ([1][0]Next)^\alpha 0)0$ (absorbsion of $1+$) $= [0]([1][0]Next)0$.

$\varphi(1,1,1)$ is the next fixed point $Fix(\alpha \mapsto ([1][0]Next)^\alpha 0)\varphi(1,1,0) = ([0]([1][0]Next)([0]([1][0]Next)0) = ([0]([1][0]Next))^2 0$.

More generally, $\varphi(1,1,\alpha) = ([0]([1][0]Next))^{1+\alpha}0$.

$\varphi(1,2,0)$ is the least fixed point (greater than 0) of the function $\alpha \mapsto \varphi(1,1,\alpha), Fix(\alpha \mapsto \varphi(1,1,\alpha))0 = Fix([0]([1][0]Next)^{1+\alpha}0)0 = Fix(\alpha \mapsto ([0]([1][0]Next))^\alpha 0)0 = [0]([0]([1][0]Next))0 = [0]^2([1][0]Next)0$.

Like previously, $\varphi(1,2,\alpha)$ is the $(1+\alpha)$-th fixed point of the previous function, which is $([0]^2([1][0]Next))^{1+\alpha}0$.

More generally, $\varphi(1,\beta,\alpha) = ([0]^\beta([1][0]Next))^{1+\alpha}0$.

$\varphi(2,0,0)$ is the least fixed point (greater than 0) of the function $\beta \mapsto \varphi(1,\beta,0)$, which is $Fix(\alpha \mapsto \varphi(1,\beta,0))0 = Fix(\beta \mapsto ([0]^\beta([1][0]Next))^{1+0}0)0 = Fix(\beta \mapsto [0]^\beta([1][0]Next)0)0 = [1][0]([1][0]Next)0 = ([1][0])^2 Next\ 0$.

The $(1+\alpha)$-th fixed point of the previous function is $\varphi(2,0,\alpha) = (([1][0])^2 Next)^{1+\alpha}0$.

The least fixed point of the function $\alpha \mapsto \varphi(2,0,\alpha)$ is $\varphi(2,1,0) = Fix(\alpha \mapsto \varphi(2,0,\alpha))0 = Fix(\alpha \mapsto (([1][0])^2 Next)^{(}1+\alpha)0)0 = Fix(\alpha \mapsto (([1][0])^2 Next)^\alpha 0) = [0]((([1][0])^2 Next)0$ and its $(1+\alpha)$-th fixed point is $\varphi(2,1,\alpha) = ([0](([1][0])^2 Next))^{1+\alpha}0$.

More generally, we have $\varphi(2,\beta,\alpha) = ([0]^\beta(([1][0])^2 Next))^{1+\alpha}0$.

The general formula with three variables (with $\gamma \neq 0$) is $\varphi(\gamma,\beta,\alpha) = ([0]^\beta(([1][0])^\gamma Next))^{1+\alpha}0$.

In particular, we have $\varphi(\gamma,0,0) = ([1][0])^\gamma Next0$.

Using collapsing, we can write $\varphi(\gamma,\beta,\alpha) = \varphi_{\gamma,\beta}(\alpha) = \varphi_{\Omega \cdot \gamma + \beta}(\alpha) = \varphi(\Omega \cdot \gamma + \beta, \alpha) = \varphi(1 + \Omega \cdot \gamma + \beta, \alpha) = ([0]^{\Omega \cdot \gamma + \beta}Next)^{1+\alpha}0 = ([0]^\beta(([0]^\Omega)^\gamma Next))^{1+\alpha}0 = ([0]^\beta(([1][0])^\gamma Next))^{1+\alpha}0$ if we consider that $[0]^\Omega = [1][0]$.

$\varphi(1,0,0,0)$ is the least fixed point of the function $\gamma \mapsto \varphi(\gamma,0,0), Fix(\gamma \mapsto \varphi(\gamma,0,0))0 = Fix(\gamma \mapsto ([1][0])^\gamma Next\ 0)0 = [1]([1][0])Next\ 0 = [1]^2[0]Next\ 0$.

All of these computations could be done with ω instead of 0 at the end of the formulas so we also have $\varphi(\gamma,\beta,\alpha) = ([0]^\beta(([1][0])^\gamma Next))^{1+\alpha}$

In a similar way, we can obtain the formula with 4 variables :

$\varphi(1,0,0,\alpha) = ([1]^2[0]Next)^{1+\alpha}0$

$\varphi(1,0,1,0) = Fix(\alpha \mapsto ([1]^2[0]Next)^\alpha 0)0 = [0]([1]^2[0])0$

$\varphi(1,0,1,\alpha) = ([0]([1]^2[0]Next))^{1+\alpha}0$

$\varphi(1,0,\beta,\alpha) = ([0]^\beta([1]^2[0]Next))^{1+\alpha}0$

$\varphi(1,1,0,0) = Fix(\alpha \mapsto \varphi(1,0,\alpha,0)]0 = Fix(\alpha \mapsto [0]^\alpha([1]^2[0]Next)0]0 = [1][0]([1]^2[0]Next)0$

$\varphi(1,1,0,\alpha) = ([1][0]([1]^2[0]Next))^{1+\alpha}0$

$\varphi(1,1,1,0) = Fix(\alpha \mapsto \varphi(1,1,0,\alpha))0 = Fix(\alpha \mapsto ([1][0]([1]^2[0]Next))^\alpha 0)0 = [0]([1][0]([1]^2[0]Next))0$

$\varphi(1,1,1,\alpha) = ([0]([1][0]([1]^2[0]next)))^{1+\alpha}0$

$\varphi(1,1,\beta,\alpha) = ([0]^\beta([1][0]([1]^2[0]Next)))^{1+\alpha}0$

$\varphi(1,2,0,0) = Fix(\alpha \mapsto \varphi(1,1,\alpha,0))0 = Fix(\alpha \mapsto [0]^\alpha([1][0]([1]^2[0]next))0)0 = [1][0]([1][0]([1]^2[0]Next))0 = ([1][0])^2([1]^2[0]Next)0$

$\varphi(1,0,0,0) = [1]^2[0]Next0$

$\varphi(1,1,0,0) = [1][0]([1]^2[0]Next)0$

$\varphi(1,2,0,0) = ([1][0])^2([1]^2[0]Next)0$

$\varphi(1,\gamma,0,0) = ([1][0])^\gamma([1]^2[0]Next)0$

$\varphi(1,\gamma,\beta,\alpha) = ([0]^\beta(([1][0])^\gamma([1]^2[0]Next)))^{1+\alpha}0$

$\varphi(2,0,0,0) = Fix(\alpha \mapsto \varphi(1,\alpha,0,0)]0 = Fix(\alpha \mapsto ([1][0])^\alpha([1]^2[0]Next)0]0 = [1]([1][0])([1]^2[0]Next)0 = [1]^2[0]([1]^2[0]Next)0 = ([1]^2[0])^2 Next0$

$\varphi(\delta,0,0,0) = ([1]^2[0])^\delta Next\ 0$

The general formula with four variables is :

$\varphi(\delta,\gamma,\beta,\alpha) = ([0]^\beta(([1][0])^\gamma(([1]^2[0])^\delta Next)))^{1+\alpha}0 = ([0]^\beta(([1][0])^\gamma(([1]^2[0])^\delta Next)))^{1+\alpha}\omega$

and so on.

Using collapsing, we can write $\varphi(\delta,\gamma,\beta,\alpha) = \varphi_{\delta,\gamma,\beta}(\alpha) = \varphi_{\Omega^2 \cdot \delta + \Omega \cdot \gamma + \beta}(\alpha) = \varphi(\Omega^2 \cdot \delta + \Omega \cdot \gamma + \beta, \alpha) = \varphi(1 + \Omega^2 \cdot \delta + \Omega \cdot \gamma + \beta, \alpha) = ([0]^{\Omega^2 \cdot \delta + \Omega \cdot \gamma + \beta}Next)^{1+\alpha}0 = ([0]^\beta(([0]^\Omega)^\gamma(([0]^{\Omega^2})^\delta Next)))^{1+\alpha}0 = ([0]^\beta(([1][0])^\gamma(([1]^2[0])^\delta Next)))^{1+\alpha}0$ if we consider that $[0]^\Omega = [1][0]$ and $[0]^{\Omega^2} = ([0]^\Omega)^\Omega = ([1][0])^\Omega = [1]([1][0]) = [1]^2[0]$.

The small Veblen ordinal is the limit of :

$\varphi(1) = \omega, \varphi(1,0) = Next\ \omega, \varphi(1,0,0) = [1][0]Next\ \omega, \varphi(1,0,0,0) = [1]^2[0]Next\ \omega, \varphi(1,0,0,0,0) = [1]^3[0]Next\ \omega, \ldots$.

This limit is $[1]^\omega[0]Next\ \omega = [1]^\omega[0]Next\ 0$.

Allowing variables at any finite or transfinite positions (which is equivalent to Schütte brackets or Klammersymbols) gives ordinals smaller than the large Veblen ordinal which is the least fixed point of the function $\alpha \mapsto \varphi(1_\alpha)$. It is $Fix(\alpha \mapsto \varphi(1_\alpha))0 = Fix(\alpha \mapsto [1]^\alpha[0]Next\ 0)0 = [2][1][0]Next\ 0$
$[2][1][0]Next\ 0$ or $[2][1][0]Next\ \omega$.
The conversion rule from Schütte Klammersymbol to Simmons notation are described by Simmons in his paper : http://www.cs.man.ac.u
NOTATIONS/FromBelow.pdf (Simmons also wrote other papers but it seems to me that they contain inaccuracies and maybe even errors).
In summary :
$Fix\ f\zeta = f^\omega(\zeta + 1)$
$Enm\ h\ \alpha = h^{1+\alpha}0$
$Next = Fix(\alpha \mapsto \omega^\alpha)$
$[0]h = Fix(\alpha \mapsto h^\alpha 0)$
$[1]hg = Fix(\alpha \mapsto h^\alpha g0)$
$\nabla \begin{bmatrix} \alpha + 1 \\ i + 1 \end{bmatrix} = ([1]^i[0])^{1+\alpha}$ if $i \neq 0$; $[0]^\alpha$ if $i = 0$
$\nabla \begin{bmatrix} \alpha_1 + 1 & \dots & \alpha_s + 1 \\ i_1 + 1 & \dots & 1_s + 1 \end{bmatrix} = \nabla \begin{bmatrix} \alpha_1 + 1 \\ i_1 + 1 \end{bmatrix} \circ [0] \circ \dots \circ [0] \circ \nabla \begin{bmatrix} \alpha_s + 1 \\ i_s + 1 \end{bmatrix}$
where $f \circ g$ is the composition of functions f and g : $(f \circ g)x = f\ (g\ x)$
$Sch \begin{bmatrix} 1 + \alpha_1 & \dots & 1 + \alpha_s \\ 1 + i_1 & \dots & 1 + i_s \end{bmatrix} = Enm\ \circ \nabla \begin{bmatrix} 1 + \alpha_1 & \dots & 1 + \alpha_s \\ 1 + i_1 & \dots & 1 + i_s \end{bmatrix} o\ Fix$
f may be any function but it is usually $\alpha \mapsto \omega^\alpha$.
$f \begin{pmatrix} \zeta & 1 + \alpha_1 & \dots & 1 + \alpha_s \\ 0 & 1 + i_1 & \dots & 1 + i_s \end{pmatrix}$
$= Sch \begin{bmatrix} 1 + \alpha_1 & \dots & 1 + \alpha_s \\ 1 + i_1 & \dots & 1 + i_s \end{bmatrix} f\zeta$
$= (Enm \circ \nabla \begin{bmatrix} 1 + \alpha_1 & \dots & 1 + \alpha_s \\ 1 + i_1 & \dots & 1 + i_s \end{bmatrix} \circ Fix)f\zeta$
$= (Enm \circ \nabla \begin{bmatrix} \alpha_1 + 1 \\ i_1 + 1 \end{bmatrix} \circ [0] \circ \dots \circ [0] \circ \nabla \begin{bmatrix} \alpha_s + 1 \\ i_s + 1 \end{bmatrix} \circ Fix)f\zeta$
$= Enm((\nabla \begin{bmatrix} \alpha_1 + 1 \\ i_1 + 1 \end{bmatrix} \circ [0] \circ \dots \circ [0] \circ \nabla \begin{bmatrix} \alpha_s + 1 \\ i_s + 1 \end{bmatrix})(Fixf))\zeta$
$= (\nabla \begin{bmatrix} \alpha_1 + 1 \\ i_1 + 1 \end{bmatrix} \circ [0] \circ \dots \circ [0] \circ \nabla \begin{bmatrix} \alpha_s + 1 \\ i_s + 1 \end{bmatrix})(Fixf))^{1+\zeta}0$

If $f = \alpha \mapsto \omega^\alpha$, then Fix f = Next and
$f \begin{pmatrix} \zeta & 1 + \alpha_1 & \dots & 1 + \alpha_s \\ 0 & 1 + i_1 & \dots & 1 + i_s \end{pmatrix} = (\nabla \begin{bmatrix} \alpha_1 + 1 \\ i_1 + 1 \end{bmatrix} \circ [0] \circ \dots \circ [0] \circ \nabla \begin{bmatrix} \alpha_s + 1 \\ i_s + 1 \end{bmatrix})Next)^{1+\zeta}0$

Examples :

$\varphi(1 + \beta, \alpha)$
$= (\xi \mapsto \omega^\xi) \begin{pmatrix} \alpha & 1 + \beta \\ 0 & 1 \end{pmatrix}$
$= ((\nabla \begin{bmatrix} \beta + 1 \\ 1 \end{bmatrix})(Fix(\xi \mapsto \omega^\xi)))^{1+\alpha}0$
$= ((\nabla \begin{bmatrix} \beta + 1 \\ 1 \end{bmatrix})Next)^{1+\alpha}0$
$= ([0]^\beta Next)^{1+\alpha}0$

$\varphi(1 + \gamma, 1 + \beta, \alpha)$
$= (\xi \mapsto \omega^\xi) \begin{pmatrix} \alpha & 1 + \beta & 1 + \gamma \\ 0 & 1 & 2 \end{pmatrix}$
$= ((\nabla \begin{bmatrix} \beta + 1 \\ 1 \end{bmatrix} \circ [0] \circ \nabla \begin{bmatrix} \gamma + 1 \\ 2 \end{bmatrix})(Fix(\xi \mapsto \omega^\xi)))^{1+\alpha}0$

$$= ((\nabla \begin{bmatrix} \beta+1 \\ 1 \end{bmatrix} \circ [0] \circ \nabla \begin{bmatrix} \gamma+1 \\ 2 \end{bmatrix}) Next)^{1+\alpha} 0$$
$$= (([0]^\beta \circ [0] \circ ([1][0])^{1+\gamma}) Next)^{1+\alpha} 0$$
$$= ([0]^{1+\beta} (([1][0])^{1+\gamma} Next))^{1+\alpha} 0$$

Compare with the previously found formula :

if $\gamma > 0, \varphi(\gamma, \beta, \alpha) = ([0]^\beta (([1][0])^\gamma Next))^{1+\alpha} 0$

and note the "round trip" $1 + \gamma \to \gamma + 1 \to 1 + \gamma$.

$$\varphi(1+\delta, 1+\gamma, 1+\beta, \alpha)$$
$$= (\xi \mapsto \omega^\xi) \begin{pmatrix} \alpha & 1+\beta & 1+\gamma & 1+\delta \\ 0 & 1 & 2 & 3 \end{pmatrix}$$
$$= ((\nabla \begin{bmatrix} \beta+1 \\ 1 \end{bmatrix} \circ [0] \circ \nabla \begin{bmatrix} \gamma+1 \\ 2 \end{bmatrix} \circ [0] \circ \nabla \begin{bmatrix} \delta+1 \\ 3 \end{bmatrix}) (Fix(\xi \mapsto \omega^\xi)))^{1+\alpha} 0$$
$$= ((\nabla \begin{bmatrix} \beta+1 \\ 1 \end{bmatrix} \circ [0] \circ \nabla \begin{bmatrix} \gamma+1 \\ 2 \end{bmatrix} \circ [0] \circ \nabla \begin{bmatrix} \delta+1 \\ 3 \end{bmatrix}) Next)^{1+\alpha} 0$$
$$= (([0]^\beta \circ [0] \circ ([1][0])^{1+\gamma} \circ [0] \circ ([1]^2[0])^{1+\delta}) Next)^{1+\alpha} 0$$
$$= ([0]^{1+\beta} (([1][0])^{1+\gamma} ([0] ((([1]^2[0])^{1+\delta} Next))))^{1+\alpha} 0$$
$$= ([0]^{1+\beta} (([1][0])^{1+\gamma} ((([1]^2[0])^{1+\delta} Next)))^{1+\alpha} 0$$

because [0] is absorbed by the following operator (see http://www.cs.man.ac.uk/ hsimmons/ORDINAL-NOTATIONS/FromBelow.pdf p 33, 6.7)

Compare with the previously mentioned formula :
$$\varphi(\delta, \gamma, \beta, \alpha) = ([0]^\beta (([1][0])^\gamma (([1]^2[0])^\delta Next)))^{1+\alpha} 0$$

The equality
$$(\xi \mapsto \omega^\xi) \begin{pmatrix} \zeta & 1+\alpha_1 & \ldots & 1+\alpha_s \\ 0 & 1+i_1 & \ldots & 1+i_s \end{pmatrix} = (\nabla \begin{bmatrix} \alpha_1+1 \\ i_1+1 \end{bmatrix} \circ [0] \circ \ldots \circ [0] \circ \nabla \begin{bmatrix} \alpha_s+1 \\ i_s+1 \end{bmatrix}) Next)^{1+\zeta} 0$$
can be reformulated, distinguishing four cases :

- $(\xi \mapsto \omega^\xi) \begin{pmatrix} \zeta \\ 0 \end{pmatrix} = \varphi(0, \zeta) = \omega^\zeta$

- $(\xi \mapsto \omega^\xi) \begin{pmatrix} \zeta & 1+\alpha \\ 0 & 1 \end{pmatrix} = \varphi(1+\alpha, \zeta) = (\nabla \begin{bmatrix} \alpha+1 \\ 1 \end{bmatrix} Next)^{1+\zeta} 0 = ([0]^\alpha Next)^{1+\zeta} 0$

- $(\xi \mapsto \omega^\xi) \begin{pmatrix} \zeta & 1+\alpha_1 & 1+\alpha_2 & \ldots & 1+\alpha_s \\ 0 & 1 & 1+i_2 & \ldots & 1+i_s \end{pmatrix}$

  $= ((\nabla \begin{bmatrix} \alpha_1+1 \\ 1 \end{bmatrix} \circ [0] \circ \nabla \begin{pmatrix} \alpha_2+1 \\ i_2+1 \end{pmatrix} \circ [0] \circ \ldots \circ [0] \circ \nabla \begin{bmatrix} \alpha_s+1 \\ i_s+1 \end{bmatrix}) Next)^{1+\zeta} 0$

  $= (([0]^{\alpha_1} \circ [0] \circ ([1]^{i_2}[0])^{1+\alpha_2} \circ [0] \circ \ldots \circ [0] \circ ([1]^{i_s}[0])^{1+\alpha_s}) Next)^{1+\zeta} 0$

  $= (([0]^{1+\alpha_1} \circ ([1]^{i_2}[0])^{1+\alpha_2} \circ [0] \circ \ldots \circ [0] \circ ([1]^{i_s}[0])^{1+\alpha_s}) Next)^{1+\zeta} 0$

  $= (([0]^{1+\alpha_1} \circ ([1]^{i_2}[0])^{1+\alpha_2} \circ \ldots \circ ([1]^{i_s}[0])^{1+\alpha_s}) Next)^{1+\zeta} 0$

  The first separating [0] is combined with $[0]^{\alpha_1}$ giving $[0]^{1+\alpha_1}$ and the other are absorbed.

- $(\xi \mapsto \omega^\xi) \begin{pmatrix} \zeta & 1+\alpha_1 & \ldots & 1+\alpha_s \\ 0 & 1+i_1 & \ldots & 1+i_s \end{pmatrix}$ with $i_1 \neq 0$

  $= ((\nabla \begin{bmatrix} \alpha_1+1 \\ i_1+1 \end{bmatrix} \circ [0] \circ \ldots \circ [0] \circ \nabla \begin{bmatrix} \alpha_s+1 \\ i_s+1 \end{bmatrix}) Next)^{1+\zeta} 0$

  $= ((([1]^{i_1}[0])^{1+\alpha_1} \circ [0] \circ \ldots \circ [0] \circ ([1]^{i_s}[0])^{1+\alpha_s}) Next)^{1+\zeta} 0$

  $= ((([1]^{i_1}[0])^{1+\alpha_1} \circ \ldots \circ ([1]^{i_s}[0])^{1+\alpha_s}) Next)^{1+\zeta} 0$

  The separating [0] are absorbed.

We can see that the third case is contained in the fourth one if we remove the restriction $i_1 \neq 0$ because if $i_1 = 0$ we have $([1]^{i_1}[0])^{1+\alpha_1} = [0]^{1+\alpha_1}$ like in the third case.

For more information concerning the correspondence between Simmons notation and Schütte Klammersymbols, see :

The Simmons notation can also be used to represent the notation going beyond Veblen functions that we saw previously.

As we saw previously, the large Veblen ordinal is the least fixed point of the function $\alpha \mapsto \varphi(1_\alpha)$ or $\alpha \mapsto (\xi \mapsto \omega^\xi) \binom{1}{\alpha}$. It is $Fix(\alpha \mapsto \varphi(1_\alpha))0 = Fix(\alpha \mapsto [1]^\alpha[0]Next\ 0)0 = [2][1][0]Next\ 0$.

Using collapsing, we can write it $\varphi(1_\Omega) = [1]^\Omega[0]Next\ 0$. Compare with the previously obtained equality $[0]^{\Omega^2} = ([0]^\Omega)^\Omega = ([1][0])^\Omega = [1]([1][0]) = [1]^2[0]$ which can be generalized to $[0]^{\Omega^\alpha} = [1]^\alpha[0]$. We can also write $LVO = \varphi_{\Omega^\Omega}(0) = [0]^{\Omega^\Omega}Next\ 0 = [1]^\Omega[0]Next\ 0 = [2][1][0]Next\ 0$ with $[1]^\Omega = [2][1]$.

The fixed points of this function $\alpha \mapsto \varphi(1_\alpha)$ are enumerated by the function F, so we have LVO = F(0). More generally, the $(1 + \alpha$-th fixed point of $\alpha \mapsto \varphi(1_\alpha)$ is $F(\alpha) = \varphi_1^+(\alpha) = ([2][1][0]Next)^{1+\alpha}0$.

Then the fixed points of $F = \varphi_1^+$ are enumerated by $G = \varphi_2^+$. The least fixed point of F is $G(0) = \varphi_2^+(0) = Fix(\alpha \mapsto ([2][1][0]Next)^{1+\alpha}0)0 = [0]([2][1][0]Next)0$ (because of the absorbsion of "1+") and its $(1+\alpha)$-th fixed point is $G(\alpha) = \varphi_2^+(\alpha) = ([0]([2][1][0]Next))^{1+\alpha}0$.

Then the fixed points of $G = \varphi_2^+$ are enumerated by $H = \varphi_3^+$. The least fixed point of H is $H(0) = \varphi_3^+(0) = Fix(\alpha \mapsto ([0]([2][1][0]Next)^{1+\alpha}0)0 = [0]([0]([2][1][0]Next))0 = [0]^2([2][1][0]Next)0$ and its $(1 + \alpha)$-th fixed point is $H(\alpha) = \varphi_3^+(\alpha) = ([0]^2([2][1][0]Next))^{1+\alpha}0$.

More generally, we have $\varphi_{1+\alpha}^+(0) = [0]^\alpha([2][1][0]Next)0$ and $\varphi_{1+\alpha}^+(\beta) = ([0]^\alpha([2][1][0]Next))^{1+\beta}0$.

Then we generalize the function φ^+ to any number of variables :

$\varphi^+(\alpha, \beta) = \varphi_\alpha^+(\beta)$

$\varphi^+(1, 0, 0)$ is the least fixed point of the function $\alpha \mapsto \varphi^+(\alpha, 0) = \alpha \mapsto [0]^\alpha([2][1][0]Next)0$. It is $Fix(\alpha \mapsto [0]^\alpha([2][1][0]Next)0)0 = [1][0]([2][1][0]Next)0$.

Compare with $\varphi(1, 0, 0) = [1][0]Next0$.

More generally, like we found $\varphi(\gamma, \beta, \alpha) = ([0]^\beta(([1][0])^\gamma Next))^{1+\alpha}0$, we have $\varphi^+(\gamma, \beta, \alpha) = ([0]^\beta(([1][0])^\gamma([2][1][0]Next)))^{1+\alpha}0$.

Like we generalized the φ function to transfinitely many variables reaching all ordinals less than $LVO = [2][1][0]Next\ 0$, we can generalize the φ^+ function to transfinitely many variables and reach all ordinals less than a new limit which we will call $LVO^+ = [2][1][0]([2][1][0]Next)0$ which is the least fixed point of $\alpha \mapsto [1]^\alpha[0]([2][1][0]Next)0$.

Then we can do the same with $\varphi^{++} = \Phi_2$ and we shall get similar results with $([2][1][0])^2 Next$, and generally with Φ_α, getting formulas with $([2][1][0])^\alpha Next$.

The limit of Next 0, [0] Next 0, [1] [0] Next 0, [2] [1] [0] Next 0, [3] [2] [1] [0] Next 0, ...
or $Next\ \omega, [0]Next\ \omega, [1][0]Next\ \omega, [2][1][0]Next\ \omega, [3][2][1][0]Next\ \omega, \ldots$
is called the Bachmann-Howard ordinal (BHO).
It could be written $[\omega \ldots 0]Next\ 0$ or $[\omega \ldots 0]Next\ \omega$.

7 Rationalization of the Veblen functions

When we have defined the different notations, we have arbitrarily chosen some conventions, for example the limit of $\omega, \omega^\omega, \omega^{\omega^\omega}, \ldots$ have been called ε_0. We could have called it ε_1. In this case, ε_α would have been the α-th fixed point of $\xi \mapsto \omega^\xi$ instead of the the $(1 + \alpha)$-th one. Also we chose to define $\varphi(0, \alpha) = \omega^\alpha$. We could have chosen to define $\varphi(0, \alpha) = \varepsilon_\alpha$. The "1+" which appear in the correspondence between Simmons and Veblen notations may be due to the fact that the choices that have been made are not the most logical.

We will define a rationalized variant of the Veblen notations which simplifies the correspondence with the Simmons notation :

- $\varepsilon_\alpha = \varphi(1, \alpha) = \varepsilon'_{1+\alpha} = \varphi'(0, 1 + \alpha)$

- $\zeta_\alpha = \varphi(2, \alpha) = \zeta'_{1+\alpha} = \varphi'(1, 1 + \alpha)$

- $\eta_\alpha = \varphi(3, \alpha) = \eta'_{1+\alpha} = \varphi'(2, 1 + \alpha)$

- Generally, $\varphi(1 + \beta, \alpha) = \varphi'(\beta, 1 + \alpha)$

- $\Gamma_0 = \varphi(1, 0, 0) = \varphi'(1, 0, 1)$

- Generally, if $\gamma \neq 0, \varphi(\gamma, \beta, \alpha) = \varphi'(\gamma, \beta, 1 + \alpha)$

- In a similar way, if $\gamma \neq 0$ or $\delta \neq 0, \varphi(\delta, \gamma, \beta, \alpha) = \varphi'(\delta, \gamma, \beta, 1 + \alpha)$ and so on.

17

With this notation, the correspondence with Simmons notation becomes simpler, for example we have :

- $\varepsilon'_\alpha = Next^\alpha 0$ instead of $\varepsilon_\alpha = Next^{1+\alpha} 0$

- $\varphi'(\beta, \alpha) = ([0]^\beta Next)^\alpha 0$ instead of $\varphi(1 + \beta, \alpha) = ([0]^\beta Next)^{1+\alpha} 0$

- $\varphi'(\gamma, \beta, \alpha) = ([0]^\beta (([1][0])^\gamma Next))^\alpha 0$ instead of $\varphi(\gamma, \beta, \alpha) = ([0]^\beta (([1][0])^\gamma Next))^{1+\alpha} 0$

- $\varphi'(\delta, \gamma, \beta, \alpha) = ([0]^\beta (([1][0])^\gamma (([1]^2[0])^\delta Next)))^{1+\alpha} 0$ instead of $\varphi(\delta, \gamma, \beta, \alpha) = ([0]^\beta (([1][0])^\gamma (([1]^2[0])^\delta Next)))^{1+\alpha} 0$

It appears that the last variable (α in the previous examples) plays a different role from the other variables, so it could be more logical to write for example $\varphi'_{\delta,\gamma,\beta}(\alpha)$ instead of $\varphi'(\delta, \gamma, \beta, \alpha)$ and to consider that β is at position 0, γ at position 1 and δ at position 2. In this case, we see that the position corresponds to the exponent of [1] in the Simmons representation.
We can also use collapsing to represent the index list, writing for example :

- $\Gamma_0 = \varphi'_{1,0}(1) = \varphi'_\Omega(1)$
- Ackermann ordinal $= \varphi'_{1,0,0}(1) = \varphi'_{\Omega^2}(1)$
- $SVO = \varphi'_{\Omega^\omega}(1)$

This notation even permits writing ordinals that are out of range of Veblen notation like :

- $LVO = \varphi'_{\Omega^\Omega}(1)$
- $\varphi'_{\Omega^{\Omega^\Omega}}(1)$
- $\ldots$

8   RHS0 notation

8.1   Basic principles

Like Simmons notation, the RHS0 notation uses lambda calculus formalism.
The basic method consists in :

- Start from 0
- If we don't see any regularity, take the successor (add 1)
- If we see a regularity and we don't have a notation for it, invent it and jump to the limit
- If we see a regularity and we already have a notation for it, use it and jump to the limit.

The difficulty, which requires intelligence, is to see the regularities. It gives the following sequence :

- 0 : no regularity, take the successor
- $suc\ 0$: no regularity, take the successor
- $suc(suc\ 0)$: regularity : suc repeatedly applied to 0. No notation, invent it : H f x = limit of x, f x, f (f x), ...
- $H suc\ 0$: no regularity, take the successor
- $suc(H suc\ 0)$: no regularity, take the successor
- $suc(suc(H suc\ 0))$: regularity : suc repeatedly applied to H suc 0, notation exists
- $H suc(H suc\ 0)$: regularity : H suc repeatedly applied to 0, notation exists
- $H(H suc)0$: regularity : H repeatedly applied to suc, notation exists
- $H H suc\ 0$: regularity (suc 0, ..., H suc 0, ... H H suc 0, ... H H H suc 0, ...), invent notation $R_1 H suc\ 0$ for the limit of this sequence
- $R_1 H suc\ 0$: no regularity, take the successor
- $suc(R_1 H suc\ 0)$
- $suc(suc(R_1 H suc\ 0))$
- $H suc(R_1 H suc\ 0)$
- $suc(H suc(R_1 H suc\ 0))$
- $suc(suc(H suc(R_1 H suc\ 0)))$
- $H suc(H suc(R_1 H suc\ 0))$
- $H(H suc)(R_1 H suc\ 0)$

- $HHsuc(R_1Hsuc\,0)$
- $R_1Hsuc(R_1Hsuc\,0)$
- $H(R_1Hsuc)0$
- $suc(H(R_1Hsuc)0)$
- $suc(suc(H(R_1Hsuc)0))$
- $Hsuc(H(R_1Hsuc)0)$
- $suc(suc(Hsuc(H(R_1Hsuc)0)))$
- $Hsuc(Hsuc(H(R_1Hsuc)0)))$
- $H(Hsuc)(H(R_1Hsuc)0)$
- $HHsuc(H(R_1Hsuc)0)$
- $R_1Hsuc(H(R_1Hsuc)0)$
- $suc(R_1Hsuc(H(R_1Hsuc)0))$
- $suc(suc(R_1Hsuc(H(R_1Hsuc)0)))$
- $Hsuc(R_1Hsuc(H(R_1Hsuc)0))$
- $suc(Hsuc(R_1Hsuc(H(R_1Hsuc)0)))$
- $suc(suc(Hsuc(R_1Hsuc(H(R_1Hsuc)0))))$
- $Hsuc(Hsuc(R_1Hsuc(H(R_1Hsuc)0)))$
- $H(Hsuc)(R_1Hsuc(H(R_1Hsuc)0))$
- $HHsuc(R_1Hsuc(H(R_1Hsuc)0))$
- $R_1Hsuc(R_1Hsuc(H(R_1Hsuc)0))$
- $H(R_1Hsuc)(H(R_1Hsuc)0)$
- $H(H(R_1Hsuc))0$
- $HH(R_1Hsuc)0$
- $R_1H(R_1Hsuc)0$
- $H(R_1H)suc\,0$
- ...
- $R_1H(R_1H)suc\,0$
- $R_1(R_1H)suc\,0$
- $HR_1Hsuc\,0$
- ...
- $R_1HR_1Hsuc\,0$: invent notation $R_2R_1Hsuc\,0 = $ limit of $suc\,0, R_1Hsuc\,0, R_1HR_1Hsuc\,0, ...$
- ...
- $R_3R_2R_1Hsuc\,0$: invent notation $R_{3...1}Hsuc\,0$ and jump to limit
- $R_{\omega...1}Hsuc\,0$
- ...
- $R_2R_{\omega...1}Hsuc\,0$: invent notation $R_{\omega+1...1}Hsuc\,0$
- ...

To progress faster, we can use the following rule :
If we have found an ordinal α, and later another ordinal β of the form $f(s(sz))$, we may produce an ordinal $\gamma = f([suc \to s, 0 \to z]\alpha)$ where $[suc \to s, 0 \to z]\alpha$ means the expression obtained by replacing suc by s and 0 by z in α.
For example :

- $\alpha = R_1Hsuc\,0$
- $\beta = R_1H(R_1Hsuc)0$
- $s = R_1H$
- $z = suc$
- $fx = x\,0$
- $[suc \to R_1H, 0 \to suc]\alpha = R_1H(R_1H)suc$
- $\gamma = f([suc \to R_1H, 0 \to suc]\alpha) = R1H(R1H)suc\,0$

With the following rules :

- $0 :\to 0$

- $suc : x \to suc\ x$
- $H : f(fx) -> Hfx$
- $R_1 : ff -> R_1 f$
- $R_2 : fgfg -> R_2 fg$
- $R_3 : fghfgh -> R_3 fgh$
- ...
- $Repl : \alpha, f(s(sz)) \to f([suc \to s, 0 \to z]\alpha)$

we can produce the following sequence of ordinals :

- $0 : 0 : 0$
- $1 : suc\ 0 : suc\ 0$
- $2 : suc\ 1 : suc\ (suc\ 0)$
- $3 : H2 : Hsuc\ 0$
- $4 : suc\ 3 : suc\ (Hsuc\ 0)$
- $5 : suc\ 4 : suc\ (suc\ (Hsuc\ 0))$
- $6 : H5 : Hsuc\ (Hsuc\ 0)$
- $7 : H6 : H(Hsuc)0$
- $8 : H7 : HHsuc\ 0$
- $9 : R_1 8 : R_1 Hsuc\ 0$
- $10 : suc\ 9 : suc\ (R_1 Hsuc\ 0)$
- $11 : suc\ 10 : suc\ (suc\ (R_1 Hsuc\ 0))$
- $12 : Repl\ 9\ 11[suc-> suc, 0-> R_1 Hsuc\ 0] : R_1 Hsuc\ (R_1 Hsuc\ 0)$
- $13 : Repl\ 9\ 12[suc-> R_1 Hsuc, 0-> 0] : R_1 H(R_1 Hsuc)0$
- $14 : Repl\ 9\ 13[suc-> R_1 H, 0-> suc] : R_1 H(R_1 H)suc\ 0$
- $15 : R_1 14 : R_1(R_1 H)suc\ 0$
- $16 : Repl\ 9\ 15[suc-> R_1, 0-> H] : R_1 HR_1 Hsuc\ 0$
- $17 : R_2 16 : R_2 R_1 Hsuc\ 0$

The rules $R_1, R_2, R_3, ...$ may be replaced by H or Repl if f1 ... fn ... f1 ... fn is reformulated in $< f_1, ..., f_n > (...(< f_1, ..., f_n > I)...)$ with $< f_1, ..., f_n > g = g\ f_1...f_n$:

- $0 : 0 : 0$
- 1 : suc 0 : suc 0
- 2 : suc 1 : suc (suc 0)
- 3 : H 2 : H suc 0
- 4 : suc 3 : suc (H suc 0)
- 5 : suc 4 : suc (suc (H suc 0))
- 6 : Repl 3 5 [suc->suc,0->H suc 0] : H suc (H suc 0)
- 7 : Repl 3 6 [suc->H suc,0->0] : H (H suc) 0
- 8 : Repl 3 7 [suc->H,0->suc] : H H suc 0 = <H> (<H> I) suc 0
- 9 : Repl 3 8 [suc-><H>,0->I] : H <H> I suc 0
- 10 : suc 9 : suc (H <H> I suc 0)
- 11 : suc 10 : suc (suc (H <H> I suc 0))
- 12 : Repl 9 10 [suc->suc,0->H <H> I suc 0] : H <H> I suc (H <H> I suc 0)
- 13 : Repl 9 12 [suc->H <H>: I suc,0->0] : H <H> I (H <H> I suc) 0
- 14 : Repl 9 13 [suc->H <H> I,0->suc] : H <H> I (H <H> I) suc 0 = <H <H> I> (<H <H> I> I) suc 0
- 15 : Repl 3 14 [suc-><H <H> I>,0->I] : H <H <H> I> I suc 0 = [H <*> I] ([H <*> I] H) suc 0
- 16 : Repl 9 15 [suc->[H <*> I],0->H] : H <H> I [H <*> I] H suc 0
- = [H <*> I] H [H <*> I] H suc 0 = <[H <*> I],H> (<[H <*> I], H> I) suc 0
- 17 : Repl 3 16 [suc-><[H <*> I],H>,0->I] : H <[H <*> I], H> I suc 0

More formally, the RHS0 notation uses lambda calculus with De Bruijn indexes. $\lambda.x$ is written [x] and variables are written *, **, ***..., or $\bullet, \bullet\bullet, \bullet\bullet\bullet$, ... for example [... * ...] = [... $\bullet$ ...] = $\lambda x(...x...)$

```
CI = C I is defined by CI x f = f x.
CI x = <x>
<x1,...,xn> f = f x1 ... xn
tuple n f x1 ... xn = f <x1,...,xn>
tuple 0 = <I>
tuple (n+1) f x = tuple n [ f (insert x *) ]
with insert x a f  = a (f x)


r 0 f x = x
r (n+1) f x = f (r n f x)
r (lim g) f x = lim [r * f x]


H f x represents the limit of x, f x, f (f x), ...
H f x = r w f x
```

$R_1 = [H < \bullet > I] = tuple\ 1[H \bullet I]$
$R_2 = [[H < \bullet\bullet, \bullet > I]] = tuple\ 2[H \bullet I]$
$R_3 = [[[H < \bullet\bullet\bullet, \bullet\bullet, \bullet > I]]] = tuple\ 3[H \bullet I]$
$R_n = tuple\ n[H \bullet I]$
$R_{n...1} = R_n \ldots R_1$
$S_{n...1} = [S_{\bullet...1}]n =< R_n, ..., R_1 >$
$R_{n...1} = S_{n...1}I$
$[S_{\bullet...1}]0 = I$
$[S_{\bullet...1}](n+1) = insert(tuple(n+1)[H \bullet I])([S_{\bullet...1}]n)$


```
L f = lim f 0, f 1, ...
L f x = L [f * x]
H = [[L [r * *** **]]]
or
L0 = lim f 0, f 1, ...
L n f = tuple n [ L0 [ ** (f *) ]]
L n = [ tuple n [ L0 [ ** (*** *) ]]]
L = [[ tuple ** [ L0 [ ** (*** *) ]]]
 = \n \f (tuple n \a (L0 \i (a (f i)) ) )
```


To represent the replacement $[suc \to s, 0 \to z]$ we can represent ordinals by ordinal functions which, when applied to suc and 0, give the considered ordinal. For example, $R_1Hsuc\ 0$ is represented by the ordinal function $s \mapsto z \mapsto R_1Hsz$, $R_1H(R_1Hsuc)0$ by $s \mapsto z \mapsto R_1H(R_1Hs)z$. From these ordinals, with the replacement $[suc \to R_1H, 0 \to suc]$ we can produce a new ordinal represented by $s \mapsto z \mapsto ((s \mapsto z \mapsto R_1Hsz)(R_1H)sz) = s \mapsto z \mapsto R_1H(R_1H)sz)$ which, when applied to suc and 0, gives $R_1H(R_1H)suc\ 0$.

Operations can be represented with replacements :

- $\alpha + \beta = [0 \to \alpha]\beta$
- $\alpha \cdot \beta = [suc \to [\bullet + \alpha]]\beta = [suc-> [[0 \to \bullet\bullet]\alpha]]\beta$
- $\alpha^\beta = [suc \to [\bullet \cdot \alpha], 0 \to 1]\beta = [suc \to [[suc \to [[0 \to \bullet\bullet] \bullet \bullet\bullet]]\alpha], 0 \to suc\ 0]\beta$
- $\omega^\alpha = [suc \to [suc \to Hsuc], 0 \to suc\ 0]\alpha = [suc \to H, 0 \to suc]\alpha\ 0$
- $\varepsilon_0{}^\alpha = [suc \to R_1H, 0 \to suc]\alpha 0$
- $\varepsilon_a = [suc \to R_1, 0 \to H](1 + \alpha)suc\ 0; 1 + \alpha = [0 \to suc\ 0]\alpha$

8.2   Correspondence with other notations

- $suc\ 0 = 0 + 1 = 1$
- $suc\ (suc\ 0) = 1 + 1 = 2$

- $H suc\ 0 = \omega$
- $suc\ (H suc\ 0) = \omega + 1$
- $H suc\ (H\ suc\ 0) = \omega + \omega = \omega \cdot 2$
- $H(H suc)\ 0 = \omega \cdot \omega = \omega^2$
- $HH suc\ 0 = \omega^\omega$
- $R_1 H suc\ 0 = \text{limit of } suc\ 0, H suc\ 0, HH suc\ 0, HHH suc\ 0, \ldots = \varepsilon_0 = \varphi(1,0) = \varphi'(0,1) = Next\ \omega$
- $suc(R_1 H suc\ 0) = \varepsilon_0 + 1$
- $R_1 H suc(R_1 H suc\ 0) = \varepsilon_0 + \varepsilon_0 = \varepsilon_0 \cdot 2$
- $R_1 H(R_1 H suc)0 = \varepsilon_0 \cdot \varepsilon_0 = {\varepsilon_0}^2$
- $R_1 H(R_1 H)suc\ 0 = {\varepsilon_0}^{\varepsilon_0}$
- $R_1 H(R_1 H)(R_1 H)suc\ 0 = {\varepsilon_0}^{{\varepsilon_0}^{\varepsilon_0}}$
- $R_1(R_1 H)suc\ 0 = \varepsilon_1 = \varphi(1,1) = \varphi'(0,2) = Next(Next\ \omega)$ (note again that the correspondence is clearer with the rationalized function φ'

We have previously seen that ε_1 is the limit of $\varepsilon_0, {\varepsilon_0}^{\varepsilon_0}, {\varepsilon_0}^{{\varepsilon_0}^{\varepsilon_0}}, \ldots$ and is also the limit of $\varepsilon_0 + 1, \omega^{\varepsilon_0+1}, \omega^{\omega^{\varepsilon_0+1}}, \ldots$ and we have proved the equivalence of these two fundamental sequences. We have seen that the first fundamental sequence is equivalent to $\omega, {\varepsilon_0}^\omega, {\varepsilon_0}^{{\varepsilon_0}^\omega}, \ldots$, so we proved the equivalence of the two fundamental sequences by proving that for any n, we have :

$$\omega^{\omega^{\cdot^{\cdot^{\omega^{\omega^{\varepsilon_0+1}}}}}} = {\varepsilon_0}^{{\varepsilon_0}^{\cdot^{\cdot^{{\varepsilon_0}^\omega}}}}$$

We will now see how we can prove it using RHS0 notation.
First we will write the two sides of this equality using RHS0 notation :
We will use the notation $X \ldots X$ for X repeated n times.

- $\varepsilon_0 = R_1 H suc\ 0$
- $\varepsilon_0 + 1 = suc(R_1 H suc\ 0)$
- $\omega^{\varepsilon_0+1} = [suc \to H, 0 \to suc](suc(R_1 H suc 0))0 = H(R_1 H H suc)0 = H(R_1 H suc)0$
- $\omega^{\omega^{\varepsilon_0+1}} = H(R_1 H H)suc\ 0 = H(R_1 H)suc\ 0$
- $\omega^{\omega^{\omega^{\varepsilon_0+1}}} = H(R_1 H)H suc\ 0$
- $\omega^{\omega^{\omega^{\omega^{\varepsilon_0+1}}}} = H(R_1 H)HH suc\ 0$
- $\ldots$
- $\omega^{\omega^{\cdot^{\cdot^{\omega^{\omega^{\varepsilon_0}}}}}} = H(R_1 H)H \ldots H suc\ 0$

- $\omega = H suc\ 0$
- $\varepsilon_0^\omega = [suc \to R_1 H, 0 \to suc]\omega 0 = H(R_1 H)suc\ 0$
- ${\varepsilon_0}^{{\varepsilon_0}^\omega} = H(R_1 H)(R_1 H)suc\ 0$
- ${\varepsilon_0}^{{\varepsilon_0}^{{\varepsilon_0}^\omega}} = H(R_1 H)(R_1 H)(R_1 H)suc\ 0$
- $\ldots$
- ${\varepsilon_0}^{{\varepsilon_0}^{\cdot^{\cdot^{{\varepsilon_0}^\omega}}}} = H(R_1 H) \ldots (R_1 H)(R_1 H)suc\ 0$

We will now prove the equality $H(R_1 H)H \ldots H suc\ 0 = H(R_1 H) \ldots (R_1 H)(R_1 H)suc\ 0$ for any n by induction.

For n = 0, the equality is trivial : $H(R_1 H)suc\ 0 = H(R_1 H)suc\ 0$.

We will now suppose $H(R_1 H)H \ldots H suc\ 0 = H(R_1 H) \ldots (R_1 H)(R_1 H)suc\ 0$ for a given n and prove it for n + 1 :
$H(R_1 H)H \ldots HH suc\ 0 = H(R_1 H) \ldots (R_1 H)(R_1 H)(R_1 H)suc\ 0$

By elevating ω at the power of each side of this equality, we get :
$[suc \to H, 0 \to suc](H(R_1 H)H \ldots H suc\ 0)0 = [suc \to H, 0 \to suc](H(R_1 H) \ldots (R_1 H)(R_1 H)suc\ 0)0$
(1) $H(R_1 H)H \ldots HH suc\ 0 = H(R_1 H) \ldots (R_1 H)(R_1 H)H suc\ 0$

We also have :

$H(R_1H)\ldots(R_1H)suc\ 0 = H(R_1H)\ldots(R_1H)suc(suc\ 0)$

which corresponds to the RHS0 notation for :

$$\varepsilon_0^{\iddots{\varepsilon_0{}^\omega}} = 1 + \varepsilon_0^{\iddots{\varepsilon_0{}^\omega}}$$

by absorpsion of "1+" and "suc".

Now we elevate ε_0 to the power of each side of this equality, which gives :

$[suc \to R_1H, 0 \to suc](H(R_1H)\ldots(R_1H)suc\ 0)0 = [suc \to R_1H, 0 \to suc](H(R_1H)\ldots(R_1H)suc(suc\ 0))0$

$H(R_1H)\ldots(R_1H)(R_1H)suc\ 0 = H(R_1H)\ldots(R_1H)(R_1H)(R_1Hsuc)0$

Then we elevate ω to the power of each side of this equality :

$[suc \to H, 0 \to suc](H(R_1H)\ldots(R_1H)(R_1H)suc\ 0)0 = [suc \to H, 0 \to suc](H(R_1H)\ldots(R_1H)(R_1H)(R_1Hsuc)0)0$

$H(R_1H)\ldots(R_1H)(R_1H)Hsuc\ 0 = H(R_1H)\ldots(R_1H)(R_1H)(R_1HH)suc\ 0$

which can be simplified to :

$H(R_1H)\ldots(R_1H)(R_1H)Hsuc\ 0 = H(R_1H)\ldots(R_1H)(R_1H)(R_1H)suc\ 0$

Noting that the left side of this last equality is the same as the right side of (1), we get by transitivity of equality, equating the left side of (1) with the right side of the last equality :

$H(R_1H)H\ldots HHsuc\ 0 = H(R_1H)\ldots(R_1H)(R_1H)(R_1H)suc\ 0$

which corresponds to the equality we wanted to prove for n+1.

Then the correspondence continues with :

- $R_1(R_1(R_1H))suc\ 0 = \varepsilon_2 = \varphi(1,2) = \varphi'(0,3) = Next(Next(Next\ \omega))$
- $HR_1Hsuc\ 0 = \varepsilon_\omega = \varphi(1,\omega) = \varphi'(0,\omega) = Next^\omega \omega$
- $R_1HR_1Hsuc\ 0 = \varepsilon_{\varepsilon_0}$
- $R_1HR_1HR_1Hsuc\ 0 = \varepsilon_{\varepsilon_{\varepsilon_0}}$
- $R_2R_1Hsuc\ 0 = \zeta_0 = \varphi(2,0) = \varphi'(1,1) = [0]Next\ \omega$

The next step is ζ_1 which is the next fixed point of the function $\alpha \mapsto \varepsilon_\alpha$, the limit of $\zeta_0 + 1, \varepsilon_{\zeta_0+1}, \varepsilon_{\varepsilon_{\zeta_0+1}}, \ldots$. ε_α is $[suc \to R_1, 0 \to H](1+\alpha)suc\ 0$, or $[suc \to R_1, 0 \to H]\alpha\ suc\ 0$ if $\alpha \geq \omega$ by absorbsion of "1+". This is the result of replacing suc by R_1 and 0 by H in α and applying the result to suc and 0. So by iterating this transformation we get that ζ_1 is the limit of :

- $\zeta_0 + 1 = suc(R_2R_1Hsuc\ 0)$
- $R_1(R_2R_1HR_1H)suc\ 0 = R_1(R_2R_1H)suc\ 0$
- $R_1(R_2R_1H)R_1Hsuc\ 0$
- $R_1(R_2R_1H)R_1HR_1Hsuc\ 0$
- $\ldots$

In the previous correnspondence formulas, we can see a correspondence between RHS0 and Simmons notations :

- $R_2 \leftrightarrow [0]$
- $R_1 \leftrightarrow Next$
- $H \leftrightarrow \omega$
- suc 0 at the end of the RHS0 notation

If we apply this correspondence to $\zeta_1 = [0]Next([0]Next\ \omega)$ (see "Simmons notation / Correspondence with Veblen functions") we get $\zeta_1 = R_2R_1(R_2R_1H)suc\ 0$.

This is the limit of :

- $R_1(R_2R_1H)suc\ 0$
- $R_1(R_2R_1H)R_1(R_2R_1H)suc\ 0$
- $R_1(R_2R_1H)R_1(R_2R_1H)R_1(R_2R_1H)suc\ 0$
- $\ldots$

Compare with what we found previously :

- $\zeta_0 + 1 = suc(R_2R_1Hsuc\ 0)$
- $R_1(R_2R_1HR_1H)suc\ 0 = R_1(R_2R_1H)suc\ 0$

- $R_1(R_2R_1H)R_1Hsuc0$
- $R_1(R_2R_1H)R_1HR_1Hsuc0$
- ...

and with the previously proven equality :
$H(R_1H)H\ldots Hsuc\ 0 = H(R_1H)\ldots(R_1H)(R_1H)suc\ 0$
which could also be written :
$H(R_1H)H\ldots Hsuc\ 0 = H(R_1H)(R_1H)\ldots(R_1H)suc\ 0$
There is a similar equality :
$R_1(R_2R_2H)R_1H\ldots R_1Hsuc\ 0 = R_1(R_2R_1H)R_1(R_2R_1H)\ldots R_1(R_2R_1H)suc\ 0$
which proves the equivalence of the two fundamental sequences.
We saw that ζ_1 is the limit (or least upper bound) of $\zeta_0 + 1, \varepsilon_{\zeta_0+1}, \varepsilon_{\varepsilon_{\zeta_0+1}}, \ldots$. But we have $\varepsilon_{\zeta_0+1} = \zeta_0 + \varepsilon_{\zeta_0+1}$ because ζ_0 is "absorbed" by ε_{ζ_0+1}, so $\varepsilon_{\varepsilon_{\zeta_0+1}} = \varepsilon_{\zeta_0+\varepsilon_{\zeta_0+1}}$, and similarily $\varepsilon_{\varepsilon_{\varepsilon_{\zeta_0+1}}} = \varepsilon_{\zeta_0+\varepsilon_{\zeta_0+\varepsilon_{\zeta_0+1}}}$, and so on.
So ζ_1 is also the limit of $1, \varepsilon_{\zeta_0+1}, \varepsilon_{\zeta_0+\varepsilon_{\zeta_0+1}}, \ldots$.
We start with 1 because at each step, α is replaced by $\varepsilon_{\zeta_0+\alpha}$, the initial value of the sequence having no importance for its limit.
Now let us write the RHS0 representations of the values of this sequence, using the formula $\varepsilon_\alpha = [suc \to R_1, 0 \to H](1+\alpha)suc\ 0$:

- $1 = suc\ 0$
- $\zeta_0 = R_2R_1Hsuc\ 0$
- $\zeta_0 + 1 = suc(R_2R_1Hsuc\ 0)$
- $\varepsilon_{\zeta_0+1} = R_1(R_2R_1HR_1H)suc\ 0 = R_1(R_2R_1H)suc\ 0$
- $\zeta_0 + \varepsilon_{\zeta_0+1} = R_1(R_2R_1H)suc\ (R_2R_1Hsuc\ 0)$
- $\varepsilon_{\zeta_0+\varepsilon_{\zeta_0+1}} = R_1(R_2R_1H)R_1(R_2R_1HR_1H)suc\ 0 = R_1(R_2R_1H)R_1(R_2R_1H)suc\ 0$
- ...

We see that the limit of this sequence is $R_2R_1(R_2R_1H)suc\ 0$.

So we can go on with our correspondences :

- $R_2R_1(R_2R_1H)suc\ 0 = \zeta_1 = \varphi(2,1) = \varphi'(1,2) = [0]Next([0]Next\ \omega)$
- $H(R_2R_1)Hsuc\ 0 = \zeta_\omega$
- $R_2R_1H(R_2R_1)Hsuc\ 0 = \zeta_{\zeta_0}$
- $R_2(R_2R_1)Hsuc\ 0 = \eta_0 = \varphi(3,0) = \varphi'(2,1) = [0]([0]Next)\omega$
- $HR_2R_1Hsuc\ 0 = \varphi(\omega,0) = \varphi'(\omega,1)$
- $R_1HR_2R_1Hsuc\ 0 = \varphi(\varepsilon_0,0) = \varphi(\varphi(1,0),0) = \varphi'(\varepsilon_0,1) = \varphi'(\varphi'(0,1),1)$
- $R_2R_1HR_2R_1Hsuc\ 0 = \varphi(\zeta_0,0) = \varphi(\varphi(2,0),0) = \varphi'(\zeta_0,1) = \varphi'(\varphi'(1,1),1)$
- $R_3R_2R_1Hsuc\ 0 = \Gamma_0 = \varphi(1,0,0) = \varphi'(1,0,1) = [1][0]Next\ \omega$

We may then extend our correspondence rule :

- $R_3 \leftrightarrow [1]$
- $R_2 \leftrightarrow [0]$
- $R_1 \leftrightarrow Next$
- $H \leftrightarrow \omega$
- suc 0 at the end of the RHS0 notation

It is likely that this correspondence can be generalized in a simple and logical way, and it seems to me that the simpler generalization is :

- $R_{n+2} \leftrightarrow [n]$
- $R_1 \leftrightarrow Next$
- $H \leftrightarrow \omega$
- suc 0 at the end of the RHS0 notation

I will call it the "Simmons - RHS0 correspondence conjecture".

Then, if the correspondence conjecture is true, the correspondence goes on with :

- $R_3 R_2 R_1 H suc\ 0 = \Gamma_0 = \varphi(1,0,0) = \varphi'(1,0,1) = [1][0] Next\ \omega$
- $R_3(R_3 R_2) R_1 H suc\ 0 = \varphi(1,0,0,0) = \varphi'(1,0,0,1) = [1]([1][0]) Next\ 0$ (Note that in the φ and φ' functions, the last variable plays a different role than the others, as mentioned previously, so the most logical representation should probably be $\varphi'_{1,0,0}(1)$ where the first 1 should be considered at position 2 and not 3, in this case its position corresponds to the number of occurences (or the exponent) of R_3 and [1])
- $H R_3 R_2 R_1 H suc\ 0 = SVO = [1]^\omega [0] Next\ \omega$
- $R_4 R_3 R_2 R_1 H suc\ 0 = LVO = [2][1][0] Next\ \omega$
- $R_{\omega \ldots 1} H suc\ 0 = BHO$
- $\ldots$

Note the importance of using logical notations to make correct conjectures : if, instead of φ', we use the less logical function φ, we have the correspondence :

- $R_1 H suc\ 0 = \varepsilon_0 = \varphi(1,0)$
- $R_2 R_1 H suc\ 0 = \zeta_0 = \varphi(2,0)$

and we could think that it continues with :

- $R_3 R_2 R_1 H suc\ 0 = \eta_0 = \varphi(3,0)$
- $R_{\omega \ldots 1} H suc\ 0 = \varphi(\omega,0)$

Like with the Veblen functions, we can use collapsing with RHS0 notation, writing for example :
$\Gamma_0 = \varphi_{1,0}(0) = \varphi'_{1,0}(1) = \varphi_\Omega(0) = \varphi'_\Omega(1) = [0]^\Omega Next\ \omega = [1][0] Next\ \omega = (R_2)^\Omega R_1 H suc\ 0 = R_3 R_2 R_1 H suc\ 0$
which gives $(R_2)^\Omega = R_3 R_2$
to be compared with $[0]^\Omega = [1][0]$.
Γ_0 is also the limit of the following sequence :

- $\zeta_0 = \varphi'_1(1) = [0] Next\ \omega = R_2 R_1 H suc\ 0$
- $\varphi'_{\zeta_0}(1) = \varphi'_{\varphi'_1(1)}(1) = [0]^{[0] Next\ \omega} Next\ \omega = (R_2)^{R_2 R_1 H suc\ 0} R_1 H suc\ 0 = R_2 R_1 H R_2 R_1 H suc\ 0$
- $\varphi'_{\varphi'_{\varphi'_1(1)}(1)}(1) = [0]^{[0]^{[0] Next\ \omega} Next\ \omega} Next\ \omega = R_2 R_1 H R_2 R_1 H R_2 R_1 H suc\ 0$

This limit is $R_3 R_2 R_1 H suc\ 0$.

8.3 Going further with RHS0 notation and collapsing

The Bachmann-Howard ordinal (BHO) is the limit of $R_1 H suc\ 0, R_2 R_1 H suc\ 0, R_3 R_2 R_1 H suc\ 0, \ldots$ which we will write $R_{\omega \ldots 1} H suc\ 0$. We can go on ascending ordinals after BHO :

- $BHO = R_{\omega \ldots 1} H suc\ 0$
- $suc(R_{\omega \ldots 1} H suc\ 0)$
- $R_{\omega \ldots 1} H suc\ (R_{\omega \ldots 1} H suc\ 0)$
- $R_{\omega \ldots 1} H (R_{\omega \ldots 1} H suc) 0$
- $R_{\omega \ldots 1} H (R_{\omega \ldots 1} H) suc\ 0$
- $R_1 (R_{\omega \ldots 1} H) suc\ 0$
- $R_{\omega \ldots 1} (R_{\omega \ldots 1} H) suc\ 0$
- $H R_{\omega \ldots 1} H suc\ 0$
- $R_2 R_{\omega \ldots 1} H suc\ 0$
- $R_3 R_2 R_{\omega \ldots 1} H suc\ 0$
- $R_{\omega \ldots 2} R_{\omega \ldots 1} H suc\ 0$ which we will write $R_{\omega \cdot 2 \ldots 1} H suc\ 0$
- $R_{\omega \ldots 3} R_{\omega \ldots 2} R_{\omega \ldots 1} H suc\ 0$ which we will write $R_{\omega \cdot 3 \ldots 1} H suc\ 0$
- $R_{\omega^2 \ldots 1} H suc\ 0$
- $R_{\varepsilon_0 \ldots 1} H suc\ 0 = R_{R_1 H suc\ 0 \ldots 1} H suc\ 0$

Then we can take the least fixed point of the function $\alpha \mapsto R_{\alpha \ldots 1} H suc\ 0$ which we can also write $[R_{\bullet \ldots 1} H suc\ 0]$. This fixed point is $H[R_{\bullet \ldots 1} H suc\ 0]0$ which we may also write $R_1^1 H suc\ 0$ if we define $R_1^1 x_1 x_2 x_3 = H[R_{\bullet \ldots 1} x_1 x_2 x_3]0$. Then the ascension goes on with :

- $H[R_{\bullet\ldots1}Hsuc\ 0]0 = R_1^1 Hsuc\ 0$
- $R_2 R_1^1 Hsuc\ 0$
- $R_3 R_2 R_1^1 Hsuc\ 0$
- $R_{\omega\ldots2} R_1^1 Hsuc\ 0$
- $H[R_{\bullet\ldots2} R_1^1 Hsuc\ 0]0 = R_2^1 R_1^1 Hsuc\ 0 = R_{2\ldots1}^1 Hsuc\ 0$ with $R_2^1 x_1 x_2 x_3 x_4 = H[R_{\bullet\ldots2} x_1 x_2 x_3 x_4]0$
- $R_3^1 R_2^1 R_1^1 Hsuc\ 0 = R_{3\ldots1}^1 Hsuc\ 0$
- $R_{\omega\ldots1}^1 Hsuc\ 0$
- $H[R_{\bullet\ldots1}^1 Hsuc\ 0]0 = R_1^2 Hsuc\ 0$ with $R_1^2 x_1 x_2 x_3 = H[R_{\bullet\ldots1}^1 x_1 x_2 x_3]0$
- $R_1^3 Hsuc\ 0$
- $R_1^\omega Hsuc\ 0$
- $H[R_1^\bullet Hsuc\ 0]0 = R_1^{1,0} Hsuc\ 0$ with $R_1^{1,0} x_1 x_2 x_3 = H[R_1^\bullet x_1 x_2 x_3]0$
- $R_1^{1,0,0} Hsuc\ 0$

We can number the positions in the list of upper indices of R or introduce collapsing to write

- $R_1^1 = R_1^{1_0} = R_1^{0:1}$
- $R_1^{1,0} = R_1^{1_1} = R_1^{1:1} = R_1^\Omega$
- $R_1^{1,0,0} = R_1^{1_2} = R_1^{2:1} = R_1^{\Omega^2}$
- ...

We also need a notation for uncountable ordinals. We can take $\Omega = \omega_1$ the least uncountable ordinal and use a notation similar to the one we used for countable ordinals, replacing H by H_1 when ω is replaced by $\Omega = \omega_1$, writing for example :

- $\Omega = \omega_1 = H_1 suc\ 0$
- $\Omega^2 = H_1(H_1 suc)0$
- $\Omega^\omega = H H_1 suc\ 0$
- $\Omega^\Omega = H_1 H_1 suc\ 0$
- $\Omega^{\Omega^\Omega} = H_1 H_1 H_1 suc\ 0$
- ...

Then we can go on ascending ordinals by using greater and greater uncountable ordinals as upper indices of R, for example :
$R_1^{H[R_1^\bullet H_1 suc\ 0]0} Hsuc\ 0$

9 Extending Simmons notation

The limit of the Simmons notation is the limit of :

- $Next\ \omega = \varepsilon_0$
- $[0]Next\ \omega = \zeta_0$
- $[1][0]Next\ \omega = \Gamma_0$
- $[2][1][0]Next\ \omega = LVO$
- $[3][2][1][0]Next\ \omega$
- ...

which is BHO, the Bachmann-Howard ordinal.
Using RHS0 notation, it corresponds to :

- $R_1 Hsuc\ 0 = \varepsilon_0$
- $R_2 R_1 Hsuc\ 0 = \zeta_0$
- $R_3 R_2 R_1 Hsuc\ 0 = \Gamma_0$
- $R_4 R_3 R_2 R_1 Hsuc\ 0 = LVO$
- $R_5 R_4 R_3 R_2 R_1 Hsuc\ 0$
- ...

which can be written $R_{\omega\ldots1}Hsuc\ 0$ in RHS0 notation.

And we just saw that the RHS0 notation goes much further.

So, using the correspondence, we can entend the Simmons notation in a similar way the RHS0 extends beyond BHO.

Using similar notations, we can write $[\omega\ldots0]Next\omega$ for the BHO.

Then we can go on :

- $R_{\omega\ldots1}Hsuc\ 0 = R_{\omega\ldots2}R_1Hsuc\ 0 = [\omega\ldots0]Next\omega$
- $R_2R_{\omega\ldots1}Hsuc\ 0 = R_2(R_{\omega\ldots2}R_1)Hsuc\ 0 = [0]([\omega\ldots0]Next)\omega$
- $R_3R_2R_{\omega\ldots1}Hsuc\ 0 = R_3R_2(R_{\omega\ldots2}R_1)Hsuc\ 0 = [1][0]([\omega\ldots0]Next)\omega$
- $R_{\omega\cdot2\ldots1}Hsuc\ 0 = R_{\omega\ldots2}R_{\omega\ldots1}Hsuc\ 0 = R_{\omega\ldots2}(R_{\omega\ldots2}R_1)Hsuc\ 0 = [\omega\ldots0]([\omega\ldots0]Next)\omega$
- $HR_{\omega\ldots2}R_1Hsuc\ 0 = [\omega\ldots0]^\omega Next\ \omega$
- $R_1HR_{\omega\ldots2}R_1Hsuc\ 0 = [\omega\ldots0]^{\varepsilon_0}Next\ \omega$
- $R_3R_{\omega\ldots2}R_1Hsuc\ 0 = Fix(\alpha\mapsto[\omega\ldots0]^\alpha Next\ \omega)\omega = [1][\omega\ldots0]Next\ \omega = [\omega+1\ldots0]Next\ \omega$
- $R_{\omega\cdot2+1\ldots1}Hsuc\ 0$
  $= R_3R_{\omega\ldots2}R_{\omega\ldots1}Hsuc\ 0$
  $= R_3R_{\omega\ldots2}(R_{\omega\ldots2}R_1)Hsuc\ 0$
  $= [1][\omega\ldots0]([\omega\ldots0]Next)\omega$
  $= Fix(\alpha\mapsto[\omega\ldots0]^\alpha([\omega\ldots0]Next)\omega)\omega$
  $= Fix(\alpha\mapsto[\omega\ldots0]^\alpha Next\omega)\omega$ (absorbsion of $[\omega\ldots0]$)
  $= [1][\omega\ldots0]Next\ \omega$
  $= [\omega+1\ldots0]Next\ \omega$
- $\ldots$

10 Ordinal collapsing functions

Remember that collapsing consists in using uncountable ordinals to define countable ordinals.

We already saw how collapsing can be used to define ordinals, for example :

- $\varphi(\alpha,0,\beta,\gamma) = \varphi(\gamma_0,\beta_1,\alpha_3) = (\xi\mapsto\omega^\xi)\begin{pmatrix}\gamma & \beta & \alpha\\ 0 & 1 & 3\end{pmatrix} = \varphi(\Omega^3\cdot\alpha+\Omega\cdot\beta+\gamma)$.

- $\varphi_{\alpha,\beta,\gamma}(\delta) = \varphi_{\Omega^2\cdot\alpha+\Omega\cdot\beta+\gamma}(\delta)$

- $\varphi(\gamma,\beta,\alpha) = \varphi_{\gamma,\beta}(\alpha) = \varphi_{\Omega\cdot\gamma+\beta}(\alpha) = \varphi(\Omega\cdot\gamma+\beta,\alpha) = \varphi(1+\Omega\cdot\gamma+\beta,\alpha) = ([0]^{\Omega\cdot\gamma+\beta}Next)^{1+\alpha}0 = ([0]^\beta(([0]^\Omega)^\gamma Next))^{1+\alpha}0 = ([0]^\beta(([1][0])^\gamma Next))^{1+\alpha}0$ with $[0]^\Omega = [1][0]$

- $\varphi(\delta,\gamma,\beta,\alpha) = \varphi_{\delta,\gamma,\beta}(\alpha) = \varphi_{\Omega^2\cdot\delta+\Omega\cdot\gamma+\beta}(\alpha) = \varphi(\Omega^2\cdot\delta+\Omega\cdot\gamma+\beta,\alpha) = \varphi(1+\Omega^2\cdot\delta+\Omega\cdot\gamma+\beta,\alpha) = ([0]^{\Omega^2\cdot\delta+\Omega\cdot\gamma+\beta}Next)^{1+\alpha}0 = ([0]^\beta(([0]^\Omega)^\gamma(([0]^{\Omega^2})^\delta Next)))^{1+\alpha}0 = ([0]^\beta(([1][0])^\gamma(([1]^2[0])^\delta Next)))^{1+\alpha}0$ with $[0]^\Omega = [1][0]$ and $[0]^{\Omega^2} = ([0]^\Omega)^\Omega = ([1][0])^\Omega = [1]([1][0]) = [1]^2[0]$

- $\varphi(1_\Omega) = [1]^\Omega[0]Next\ 0$.

- $LVO = \varphi_{\Omega^\Omega}(0) = [0]^{\Omega^\Omega}Next\ 0 = [1]^\Omega[0]Next\ 0 = [2][1][0]Next\ 0$ with $[1]^\Omega = [2][1]$

- $\Gamma_0 = \varphi'_{1,0}(1) = \varphi'_\Omega(1)$

- Ackermann ordinal $= \varphi'_{1,0,0}(1) = \varphi'_{\Omega^2}(1)$

- $SVO = \varphi'_{\Omega^\omega}(1)$

- $LVO = \varphi'_{\Omega^\Omega}(1)$

- $\varphi'_{\Omega^{\Omega^\Omega}}(1)$

- $\Gamma_0 = \varphi_{1,0}(0) = \varphi'_{1,0}(1) = \varphi_\Omega(0) = \varphi'_\Omega(1) = [0]^\Omega Next\ \omega = [1][0]Next\ \omega = (R_2)^\Omega R_1Hsuc\ 0 = R_3R_2R_1Hsuc\ 0$

- $(R_2)^\Omega = R_3R_2$

- $[0]^\Omega = [1][0]$.

We will now see another way to use collapsing : ordinal collapsing functions.

An ordinal collapsing function is a function which, when applied to an uncountable ordinal, gives a countable ordinal.

The general idea is to define a set of ordinals C(a) or C(a,b) where a and b are ordinals, which contains all ordinals that can be built using an initial set of ordinals and some operations or functions, and then define $\psi(a)$ or $\psi(a,b)$ as the smallest ordinal which is not in C(a) or C(a,b), or the least ordinal which is greater than than all countable ordinals of C(a) or C(a,b).

Different ordinal collapsing functions are described in http://googology.wikia.com/wiki/Ordinal_notation .

These functions are extensions of functions on countable ordinals, whose fixed points can be reached by applying them to an uncountable ordinal.

Here is a correspondence between basic notation systems and their collapsing extensions based on formula : least fixed point of $f = f(\Omega)$:

Basic notation	Formula	Limit	Extension	Correspondence	Crossing
Cantor	$cantor(\alpha, \beta)$ $= \beta + \omega^\alpha$	least $\alpha = cantor(\alpha, 0)$ $= \omega^\alpha = \varepsilon_0$	Taranovsky's C	$C(\alpha, \beta) = \beta + \omega^\alpha$ iff $C(\alpha, \beta) \geq \alpha$	$C(\Omega, 0) = \varepsilon_0$
	ω^α	least $\alpha = \omega^\alpha$ $= \varepsilon_0$	Buchholz ψ_0	$\psi_0(\alpha) = \omega^\alpha$ if $\alpha < \varepsilon_0$	$\psi_0(\Omega) = \varepsilon_0$
Epsilon	ε_α	least $\alpha = \varepsilon_\alpha$ $= \zeta_0$	Madore's ψ	$\psi(\alpha) = \varepsilon_\alpha$ for all $\alpha < \zeta_0$	$\psi(\Omega) = \zeta_0$
Binary Veblen	$\varphi_\alpha(\beta)$ or $\varphi(\alpha, \beta)$	least $\alpha = \varphi(\alpha, 0)$ $= \Gamma_0$	θ	$\theta(\alpha, \beta) = \varphi(\alpha, \beta)$ below Γ_0	$\theta(\Omega, 0) = \Gamma_0$

10.1 Buchholz ψ_ν functions

Buchholz's psi-functions are a hierarchy of single-argument ordinal functions $\psi_\nu(\alpha)$ introduced by german mathematician Wilfried Buchholz in 1986.[1] These functions are a simplified version of the θ-functions, but nevertheless have the same strength as those.

10.1.1 Definition

Buchholz defined his functions as follows :

- $C_\nu^0(\alpha) = \Omega_\nu$,
- $C_\nu^{n+1}(\alpha) = C_\nu^n(\alpha) \cup \{\gamma | P(\gamma) \subseteq C_\nu^n(\alpha)\} \cup \{\psi_\mu(\xi) | \xi \in \alpha \cap C_\nu^n(\alpha) \wedge \xi \in C_\mu(\xi) \wedge \mu \leq \omega\}$,
- $C_\nu(\alpha) = \bigcup_{n<\omega} C_\nu^n(\alpha)$,
- $\psi_\nu(\alpha) = \min\{\gamma | \gamma \notin C_\nu(\alpha)\}$,

where
$$\Omega_\nu = \begin{cases} 1 & \text{if } \nu = 0 \\ \aleph_\nu & \text{if } \nu > 0 \end{cases}$$
and $P(\gamma) = \{\gamma_1, ..., \gamma_k\}$ is the set of additive principal numbers in form ω^ξ,
$P = \{\alpha \in On : 0 < \alpha \wedge \forall \xi, \eta < \alpha(\xi + \eta < \alpha)\} = \{\omega^\xi : \xi \in On\}$,
the sum of which gives this ordinal γ:
$\gamma = \alpha_1 + \alpha_2 + \cdots + \alpha_k$ where $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_k$ and $\alpha_1, \alpha_2, ..., \alpha_k \in P(\gamma)$.
Note: Greek letters always denotes ordinals. On denotes the class of all ordinals.
The limit of this notation is Takeuti-Feferman-Buchholz ordinal.

10.1.2 Properties

Buchholz showed following properties of those functions:

- $\psi_\nu(0) = \Omega_\nu$,
- $\psi_\nu(\alpha) \in P$,
- $\psi_\nu(\alpha + 1) = \min\{\gamma \in P : \psi_\nu(\alpha) < \gamma\}$ if $\alpha \in C_\nu(\alpha)$,
- $\Omega_\nu \leq \psi_\nu(\alpha) < \Omega_{\nu+1}$,
- $\alpha \leq \beta \Rightarrow \psi_\nu(\alpha) \leq \psi_\nu(\beta)$,
- $\psi_0(\alpha) = \omega^\alpha$ if $\alpha < \varepsilon_0$,
- $\psi_\nu(\alpha) = \omega^{\Omega_\nu + \alpha}$ if $\alpha < \varepsilon_{\Omega_\nu + 1}$ and $\nu \neq 0$,
- $\theta(\varepsilon_{\Omega_\nu+1}, 0) = \psi_0(\varepsilon_{\Omega_\nu+1})$ for $0 < \nu \leq \omega$.

10.1.3 Normal form and fundamental sequences

Normal form :
The normal form for 0 is 0. If α is a nonzero ordinal number $\alpha < \Lambda = \min\{\beta | \psi_\beta(0) = \beta\}$ then the normal form for α is $\alpha = \psi_{\nu_1}(\beta_1) + \psi_{\nu_2}(\beta_2) + \cdots + \psi_{\nu_k}(\beta_k)$ where k is a positive integer and $\psi_{\nu_1}(\beta_1) \geq \psi_{\nu_2}(\beta_2) \geq \cdots \geq \psi_{\nu_k}(\beta_k)$ and each ν_i, β_i are also written in normal form.
Fundamental sequences :

The fundamental sequence for an ordinal number α with cofinality $\mathrm{cof}(\alpha) = \beta$ is a strictly increasing sequence $(\alpha[\eta])_{\eta<\beta}$ with length β and with limit α, where $\alpha[\eta]$ is the η-th element of this sequence. If α is a successor ordinal then $\mathrm{cof}(\alpha) = 1$ and the fundamental sequence has only one element $\alpha[0] = \alpha - 1$. If α is a limit ordinal then $\mathrm{cof}(\alpha) \in \{\omega\} \cup \{\Omega_{\mu+1}|\mu \geq 0\}$.

For nonzero ordinals $\alpha < \Lambda$, written in normal form, fundamental sequences are defined as follows:

- If $\alpha = \psi_{\nu_1}(\beta_1) + \psi_{\nu_2}(\beta_2) + \cdots + \psi_{\nu_k}(\beta_k)$ where $k \geq 2$ then $\mathrm{cof}(\alpha) = \mathrm{cof}(\psi_{\nu_k}(\beta_k))$ and $\alpha[\eta] = \psi_{\nu_1}(\beta_1) + \cdots + \psi_{\nu_{k-1}}(\beta_{k-1}) + (\psi_{\nu_k}(\beta_k)[\eta])$,
- If $\alpha = \psi_0(0) = 1$, then $\mathrm{cof}(\alpha) = 1$ and $\alpha[0] = 0$,
- If $\alpha = \psi_{\nu+1}(0)$, then $\mathrm{cof}(\alpha) = \Omega_{\nu+1}$ and $\alpha[\eta] = \Omega_{\nu+1}[\eta] = \eta$,
- If $\alpha = \psi_\nu(0)$ and $\mathrm{cof}(\nu) \in \{\omega\} \cup \{\Omega_{\mu+1}|\mu \geq 0\}$, then $\mathrm{cof}(\alpha) = \mathrm{cof}(\nu)$ and $\alpha[\eta] = \psi_{\nu[\eta]}(0) = \Omega_{\nu[\eta]}$,
- If $\alpha = \psi_\nu(\beta + 1)$ then $\mathrm{cof}(\alpha) = \omega$ and $\alpha[\eta] = \psi_\nu(\beta) \cdot \eta$ (and note: $\psi_\nu(0) = \Omega_\nu$),
- If $\alpha = \psi_\nu(\beta)$ and $\mathrm{cof}(\beta) \in \{\omega\} \cup \{\Omega_{\mu+1}|\mu < \nu\}$ then $\mathrm{cof}(\alpha) = \mathrm{cof}(\beta)$ and $\alpha[\eta] = \psi_\nu(\beta[\eta])$,
- If $\alpha = \psi_\nu(\beta)$ and $\mathrm{cof}(\beta) \in \{\Omega_{\mu+1}|\mu \geq \nu\}$ then $\mathrm{cof}(\alpha) = \omega$ and $\alpha[\eta] = \psi_\nu(\beta[\gamma[\eta]])$ where $\begin{cases} \gamma[0] = \Omega_\mu \\ \gamma[\eta+1] = \psi_\mu(\beta[\gamma[\eta]]) \end{cases}$.
- If $\alpha = \Lambda$ then $\mathrm{cof}(\alpha) = \omega$ and $\alpha[0] = 0$ and $\alpha[\eta+1] = \psi_{\alpha[\eta]}(0) = \Omega_{\alpha[\eta]}$.

10.1.4 Explanation

Buchholz is working in Zermelo–Fraenkel set theory, that means every ordinal α is equal to set $\{\beta|\beta < \alpha\}$. Then condition $C_\nu^0(\alpha) = \Omega_\nu$ means that set $C_\nu^0(\alpha)$ includes all ordinals less than Ω_ν in other words $C_\nu^0(\alpha) = \{\beta|\beta < \Omega_\nu\}$.

The condition $C_\nu^{n+1}(\alpha) = C_\nu^n(\alpha) \cup \{\gamma|P(\gamma) \subseteq C_\nu^n(\alpha)\} \cup \{\psi_\mu(\xi)|\xi \in \alpha \cap C_\nu^n(\alpha) \wedge \mu \leq \omega\}$ means that set $C_\nu^{n+1}(\alpha)$ includes:

- all ordinals from previous set $C_\nu^n(\alpha)$,
- all ordinals that can be obtained by summation the additively principal ordinals from previous set $C_\nu^n(\alpha)$,
- all ordinals that can be obtained by applying ordinals less than α from the previous set $C_\nu^n(\alpha)$ as arguments of functions ψ_μ, where $\mu \leq \omega$.

That is why we can rewrite this condition as:
$C_\nu^{n+1}(\alpha) = \{\beta + \gamma, \psi_\mu(\eta)|\beta, \gamma, \eta \in C_\nu^n(\alpha) \wedge \eta < \alpha \wedge \mu \leq \omega\}$.
Thus union of all sets $C_\nu^n(\alpha)$ with $n < \omega$ i.e. $C_\nu(\alpha) = \bigcup_{n<\omega} C_\nu^n(\alpha)$ denotes the set of all ordinals which can be generated from ordinals $< \aleph_\nu$ by the functions $+$ (addition) and $\psi_\mu(\xi)$, where $\mu \leq \omega$ and $\xi < \alpha$.
Then $\psi_\nu(\alpha) = \min\{\gamma|\gamma \notin C_\nu(\alpha)\}$ is the smallest ordinal that does not belong to this set.
Examples :
Consider the following examples:
$C_0^0(\alpha) = \{0\} = \{\beta : \beta < 1\}$,
$C_0(0) = \{0\}$ (since no functions $\psi(\eta < 0)$ and 0+0=0).
Then $\psi_0(0) = 1$.
$C_0(1)$ includes $\psi_0(0) = 1$ and all possible sums of natural numbers:
$C_0(1) = \{0, 1, 2, ..., \mathrm{googol}, ..., \mathrm{TREE}(\mathrm{googol}), ...\}$.
Then $\psi_0(1) = \omega$ - first transfinite ordinal, which is greater than all natural numbers by its definition.
$C_0(2)$ includes $\psi_0(0) = 1, \psi_0(1) = \omega$ and all possible sums of them.
Then $\psi_0(2) = \omega^2$.
For $C_0(\omega)$ we have set $C_0(\omega) = \{0, \psi(0) = 1, ..., \psi(1) = \omega, ..., \psi(2) = \omega^2, ..., \psi(3) = \omega^3, ...\}$.
Then $\psi_0(\omega) = \omega^\omega$.
For $C_0(\Omega)$ we have set $C_0(\Omega) = \{0, \psi(0) = 1, ..., \psi(1) = \omega, ..., \psi(\omega) = \omega^\omega, ..., \psi(\omega^\omega) = \omega^{\omega^\omega}, ...\}$.
Then $\psi_0(\Omega) = \varepsilon_0$.
For $C_0(\Omega + 1)$ we have set $C_0(\Omega) = \{0, 1, ..., \psi_0(\Omega) = \varepsilon_0, ..., \varepsilon_0 + \varepsilon_0, ...\psi_1(0) = \Omega, ...\}$.
Then $\psi_0(\Omega + 1) = \varepsilon_0\omega = \omega^{\varepsilon_0+1}$.
$\psi_0(\Omega 2) = \varepsilon_1$,
$\psi_0(\Omega^2) = \zeta_0$,
$\varphi(\alpha, 1 + \beta) = \psi_0(\Omega^\alpha \beta)$,
$\psi_0(\Omega^\Omega) = \Gamma_0 = \theta(\Omega, 0)$, using Feferman theta-function,
$\psi_0(\Omega^{\Omega^\Omega})$ is large Veblen ordinal,
$\psi_0(\Omega \uparrow\uparrow \omega) = \psi_0(\varepsilon_{\Omega+1}) = \theta(\varepsilon_{\Omega+1}, 0)$.
Now let's research how ψ_1 works:
$C_1^0(\alpha) = \{\beta : \beta < \Omega_1\} = \{0, \psi(0) = 1, 2, ...\mathrm{googol}, ..., \psi_0(1) = \omega, ..., \psi_0(\Omega) = \varepsilon_0, ...$
$..., \psi_0(\Omega^\Omega) = \Gamma_0, ..., \psi(\Omega^{\Omega^\Omega+\Omega^2}), ...\}$ i.e. includes all countable ordinals.

$C_1(\alpha)$ includes all possible sums of all countable ordinals. Then
$\psi_1(0) = \Omega_1$ first uncountable ordinal which is greater than all countable ordinal by its definition i.e. smallest number with cardinality $\aleph_1$.
$C_1(1) = \{0, ..., \psi_0(0) = \omega, ..., \psi_1(0) = \Omega, ..., \Omega + \omega, ..., \Omega + \Omega, ...\}$
Then $\psi_1(1) = \Omega\omega = \omega^{\Omega+1}$.
Then $\psi_1(2) = \Omega\omega^2 = \omega^{\Omega+2}$,
$\psi_1(\psi_0(\Omega)) = \Omega\varepsilon_0 = \omega^{\Omega+\varepsilon_0}$,
$\psi_1(\psi_0(\Omega^\Omega)) = \Omega\Gamma_0 = \omega^{\Omega+\Gamma_0}$,
$\psi_1(\psi_1(0)) = \psi_1(\Omega) = \Omega^2 = \omega^{\Omega+\Omega}$,
$\psi_1(\psi_1(\psi_1(0))) = \omega^{\Omega+\omega^{\Omega+\Omega}} = \omega^{\Omega\cdot\Omega} = (\omega^\Omega)^\Omega = \Omega^\Omega$,
$\psi_1^4(0) = \Omega^{\Omega^\Omega}$,
$\psi_1(\Omega_2) = \psi_1^\omega(0) = \Omega \uparrow\uparrow \omega = \varepsilon_{\Omega+1}$.
For case $\psi(\Omega_2)$ the set $C_0(\Omega_2)$ includes functions ψ_0 with all arguments less than Ω_2 i.e. such arguments as $0, \psi_1(0), \psi_1(\psi_1(0)), \psi_1^3(0), ..., \psi$
and then $\psi_0(\Omega_2) = \psi_0(\psi_1(\Omega_2)) = \psi_0(\varepsilon_{\Omega+1})$.
In general case: $\psi_0(\Omega_{\nu+1}) = \psi_0(\psi_\nu(\Omega_{\nu+1})) = \psi_0(\varepsilon_{\Omega_\nu+1}) = \theta(\varepsilon_{\Omega_\nu+1}, 0)$.
We also can write:
$\theta(\Omega_\nu, 0) = \psi_0(\Omega_\nu^{\Omega_\nu})$ (for $1 \le \nu < \omega$).

10.1.5   Extension

We rewrite Buchholz's definition as follows[2]:

- $C_\nu^0(\alpha) = \{\beta|\beta < \Omega_\nu\}$,
- $C_\nu^{n+1}(\alpha) = \{\beta + \gamma, \psi_\mu(\eta)|\mu, \beta, \gamma, \eta \in C_\nu^n(\alpha) \wedge \eta < \alpha\}$,
- $C_\nu(\alpha) = \bigcup_{n<\omega} C_\nu^n(\alpha)$,
- $\psi_\nu(\alpha) = \min\{\gamma|\gamma \notin C_\nu(\alpha)\}$,

where
$\Omega_\nu = \begin{cases} 1 \text{ if } \nu = 0 \\ \text{smallest ordinal with cardinality } \aleph_\nu \text{ if } \nu > 0 \end{cases}$
and ω is the smallest infinite ordinal.
There is only one little detail difference with original Buchholz definition: ordinal μ is not limited by ω, now ordinal μ belongs to previous set C_n.
For example if $C_0^0(1) = \{0\}$ then $C_0^1(1) = \{0, \psi_0(0) = 1\}$ and $C_0^2(1) = \{0, ..., \psi_1(0) = \Omega\}$ and $C_0^3(1) = \{0, ..., \psi_\Omega(0) = \Omega_\Omega\}$ and so on.
Limit of this notation must be omega fixed point $\psi(\Omega_{\Omega_{\dots}}) = \psi(\psi_{\psi_{\dots}(0)}(0))$, where ψ without subscript denotes ψ_0.

10.1.6   Sources

Buchholz, W. "A New System of Proof-Theoretic Ordinal Functions". Annals of Pure and Applied Logic, vol. 32. Retrieved 2017-05-13.
Maksudov, Denis. The extension of Buchholz's function. Traveling To The Infinity. Retrieved 2017-05-18.
http://googology.wikia.com/wiki/Buchholz
Jaiger, G (1984). "P-inaccessible ordinals, collapsing functions, and a recursive notation system". Archiv f. math. Logik und Grundlagenf. pp. 49–62.
Buchholz, W.; Schiitte, K. (1983). "Ein Ordinalzahlensystem ftir die beweistheoretische Abgrenzung der H -Separation und Bar-Induktion". Sitzungsberichte der Bayerischen Akademie der Wissenschaften, Math.-Naturw. Klasse.
https://en.wikipedia.org/wiki/Buchholz_psi_functions

10.2   Madore's ψ

This ordinal collapsing function is described in https://en.wikipedia.org/wiki/Ordinal_collapsing_function and http://quibb.blogspot.fr/ impredicative-ordinals.html .
The definition of this function uses the ordinal Ω which is the least uncountable ordinal.
$C(\alpha)$ is the set of all ordinals constructible using only $0, 1, \omega, \Omega$ and addition, multiplication, exponentiation, and the function ψ (which will be defined later) restricted to ordinals smaller than α.
$\psi(\alpha)$ is the smallest ordinal not in $C(\alpha)$.

The smallest ordinal not in C(0) is the limit of $\omega, \omega^\omega, \omega^{\omega^\omega}, \ldots$ which is ε_0, so $\psi(0) = \varepsilon_0$. More generally, $\psi(\alpha) = \varepsilon_\alpha$ for all $\alpha < \zeta_0, \psi(\alpha) = \zeta_0$ for $\zeta_0 \le a \le \Omega$, and $\psi(\Omega + \alpha) = \varepsilon_(\zeta_0 + \alpha)$ for $\alpha <= \zeta_1$.

Note that $\psi(\Omega) = \zeta_0$ is the least fixed point of $\alpha \mapsto \varepsilon_\alpha$; we already saw such an equality when we introduced collapsing in the Veblen function.

Some examples of fundamental sequences (FS) are :

A FS of ω is 0, 1, 2, 3, ...

A FS of $\psi(0)$ is $\omega, \omega^\omega, \omega^{\omega^\omega}, \ldots$.

A FS of $\psi(\alpha + 1)$ is $\psi(\alpha), \psi(\alpha)^{\psi(\alpha)}, \psi(\alpha)^{\psi(\alpha)^{\psi(\alpha)}}, \ldots$.

A FS of $\psi(f(\Omega))$ is $\psi(0), \psi(f(\psi(0))), \psi(f(\psi(f(\psi(0))))), \ldots$.

For example :

A FS of $\psi(\Omega)$ is $\psi(0), \psi(\psi(0)), \psi(\psi(\psi(0))), \ldots$.

A FS of $\psi(\Omega \cdot 2)$ is $\psi(0), \psi(\Omega + \psi(0)), \psi(\Omega + \psi(\Omega + \psi(0))), \ldots$.

A FS of $\psi(\Omega^\Omega \cdot 3)$ is $\psi(0), \psi(\Omega^\Omega \cdot 2 + \Omega^{\psi(0)}), \psi(\Omega^\Omega \cdot 2 + \Omega^{\Omega^\Omega \cdot 2 + \Omega^{\psi(0)}}), \cdots$.

The limit $\psi(\varepsilon_{\Omega+1})$ of $\psi(\Omega), \psi(\Omega^\Omega), \psi(\Omega^{\Omega^\Omega}), \ldots$ is the Bachmann-Howard ordinal.

But $\varepsilon_{\Omega+1}$ cannot be expressed in this system, because $[\varepsilon_\bullet]$ does not belong to the functions used to define $C(\alpha)$. We could add it, but that would not bring us very far. A better idea is to define a new function ψ_1 :

> Let $\psi_1(\alpha)$ be the smallest ordinal which cannot be expressed from all countable ordinals, Ω and Ω_2 using sums, products, exponentials, and the ψ_1 function itself (to previously constructed ordinals less than α), where Ω_2 is an ordinal which is greater than all the ordinals which will be constructed using ψ_1, for example we can take $\Omega = \omega_1$ (the least uncountable ordinal) and $\Omega_2 = \omega_2$, the least ordinal whose cardinal is strictly greater than the cardinal of ω_1.

With this definition, we have $\psi_1(0) = \varepsilon_{\Omega+1}, \psi_1(1) = \varepsilon_{\Omega+2}$, and more generally $\psi_1(\alpha) = \varepsilon_{\Omega+1+\alpha}$.

10.3 Correspondence between Madore's ψ and other notations

To distinguish between the different Veblen functions, let us call φ_F the Veblen function with finitely many variables, and φ_T the Veblen function with transfinitely many variables.

φ_F is a function which, when applied to a list of countable ordinals, gives a countable ordinal. A list of countable ordinals can be seen as a function which, when applied to a natural number, gives a countable ordinal, with the restriction that the result differs from 0 for finitely many integers. If we denote ω the set of natural numbers and Ω the set of countable ordinals, then this can be written : $\varphi_F : (\omega \to \Omega) \to \Omega$. If we replace $\alpha \to \beta$ by β^α, we get Ω^{Ω^ω}, and if we apply ψ to it, we get $\psi(\Omega^{\Omega^\omega})$, which is the small Veblen ordinal, the least ordinal that cannot be reached using φ_F.

For φ_T, the position of a variable is represented by a countable ordinal instead of a natural number, also with the restriction that finitely many variables differ from 0, so we have $\varphi_T : (\Omega \to \Omega) \to \Omega$. If we replace $\alpha \to \beta$ by β^α, we get Ω^{Ω^Ω}, and if we apply ψ to it, we get $\psi(\Omega^{\Omega^\Omega})$, which is the large Veblen ordinal, the least ordinal that cannot be reached using φ_T.

A correspondence between Madore's ψ and other notations can be established by starting from $\psi(0) = \varepsilon_0$ and using the properties $\psi(\alpha + 1) = sup\{\psi(\alpha), \psi(\alpha)^{\psi(\alpha)}, \psi(\alpha)^{\psi(\alpha)^{\psi(\alpha)}}, \ldots\}$ and $\psi(f(\Omega)) = $ least fixed point of $\alpha \mapsto \psi(f(\alpha))$.

This method gives the following correspondence :

- $\psi(0) = \varepsilon_0 = R_1 H suc\, 0$
- $\psi(1) = sup\{\psi(0), \psi(0)^{\psi(0)}, \psi(0)^{\psi(0)^{\psi(0)}}, \ldots\} = sup\{\varepsilon_0, \varepsilon_0^{\varepsilon_0}, \varepsilon_0^{\varepsilon_0^{\varepsilon_0}}, \ldots\} = \varepsilon_1 = R_1(R_1 H) suc\, 0$
- $\psi(\alpha) = \varepsilon_\alpha = [suc \to R_1, 0 \to H](1 + \alpha) suc\, 0$
- $\psi(\Omega) = \psi(H_1 suc\, 0) = sup\{0, \varepsilon_0, \varepsilon_{\varepsilon_0}, \ldots\} = \zeta_0 = H[[suc \to R_1, 0 \to H]\bullet suc\, 0](suc\, 0) = sup\{suc\, 0, R_1 H suc\, 0, R_1 H R_1 H suc\, 0, \ldots\}$
  $R_2 R_1 H suc\, 0$
- $\psi(\Omega + 1) = \psi(suc\, (H_1 suc\, 0)) = sup\{\zeta_0, \zeta_0^{\zeta_0}, \zeta_0^{\zeta_0^{\zeta_0}}, \ldots\} = \varepsilon_{\zeta_0+1}$ (see proof below) $= R_1(R_2 R_1 H) suc\, 0$
- $\psi(\Omega + 2) = \psi(suc\, (suc\, (H_1 suc\, 0))) = \varepsilon_{\zeta_0+2} = R_1(R_1(R_2 R_1 H)) suc\, 0$
- $\psi(\Omega + \alpha) = \varepsilon_{\zeta_0+\alpha} = [suc \to R1, 0 \to R_2 R_1 H] \alpha suc\, 0$
- $\psi(\Omega + \Omega) = \psi(\Omega \times 2) = \psi(H_1 suc\, (H_1 suc\, 0)) = \zeta_1 = H[[suc \to R1, 0 \to R_2 R_1 H] \bullet suc\, 0](suc\, 0) = R_2 R_1(R_2 R_1 H) suc\, 0$
- $\psi(\Omega \times (1 + \alpha)) = \zeta_\alpha$
- $\psi(\Omega \times \alpha) = \zeta'_\alpha = [suc \to R_2 R_1, 0 \to H] \alpha suc\, 0$
- $\psi(\Omega \times \Omega) = \psi(\Omega^2) = \psi(H_1(H_1 suc)0) = sup\{0, \zeta_0, \zeta_{\zeta_0}, \ldots\} = \eta_0 = \varphi(3, 0) = \varphi'(2, 1) = H[[suc \to R_2 R_1, 0 \to H] \bullet suc\, 0](suc\, 0) = R_2(R_2 R_1) H suc\, 0$
- $\psi(\Omega^\alpha) = \varphi(1 + \alpha, 0) = \varphi'(\alpha, 1) = [suc \to R_2, 0 \to R_1] \alpha suc\, 0$

- $\psi(\Omega^\alpha \times \beta) = \varphi'(\alpha, \beta)$
- $\psi(\Omega^\Omega) = \psi(H_1 H_1 suc\, 0) = \Gamma_0 = \varphi(1,0,0) = \varphi'(1,0,1) = H[[suc \to R_2, 0 \to R_1] \bullet suc\, 0](suc\, 0) = R_3 R_2 R_1 H suc\, 0$ (Note that this confirms the Simmons - RHS0 correspondence conjecture ; note also that since $\psi(\Omega)$ and before this point we have $\psi(\alpha) = [H_1 \to R_2, suc \to R_1, 0 \to H]\, \alpha\, suc\, 0$, but this does not work anymore from this point)
- $\psi(\Omega^{\Omega \times \alpha + \beta} \times \gamma) = \varphi'(\alpha, \beta, \gamma) = \varphi'_{\alpha, \beta}(\gamma)$
- $\psi(\Omega^{\Omega^\Omega}) = LVO = R_4 R_3 R_2 R_1 H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}) = BHO = R_{\omega \dots 1} H suc\, 0$ with $\varepsilon_{\Omega+1} = sup\{\Omega, \Omega^\Omega, \Omega^{\Omega^\Omega}, \dots\}$
- $\psi(\varepsilon_{\Omega+1} + 1) = sup\{BHO, BHO^{BHO}, BHO^{BHO^{BHO}}, \dots\} = R_1(R_{\omega \dots 1} H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} + 2) = R_1(R_1(R_{\omega \dots 1} H)) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} + \alpha) = [suc \to R_1, 0 \to R_{\omega \dots 1} H]\alpha suc\, 0$
- $\psi(\varepsilon_{\Omega+1} + \Omega) = H[[suc \to R_1, 0 \to R_{\omega \dots 1} H] \bullet suc\, 0](suc\, 0) = R_2 R_1(R_{\omega \dots 1} H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} + \Omega + 1) = R_1(R_2 R_1(R_{\omega \dots 1} H)) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} + \Omega + 2) = R_1(R_1(R_2 R_1(R_{\omega \dots 1} H))) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} + \Omega + \alpha) = [suc \to R_1, 0 \to R_2 R_1(R_{\omega \dots 1} H)]\alpha suc\, 0$
- $\psi(\varepsilon_{\Omega+1} + \Omega + \Omega) = \psi(\varepsilon_{\Omega+1} + \Omega \times 2) = H[[suc \to R_1, 0 \to R_2 R_1(R_{\omega \dots 1} H)] \bullet suc\, 0](suc0) = R_2 R_1(R_2 R_1(R_{\omega \dots 1} H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} + \Omega \times \alpha) = [suc \to R_2 R_1, 0 \to R_{\omega \dots 1} H]\alpha suc\, 0$
- $\psi(\varepsilon_{\Omega+1} + \Omega \times \Omega) = \psi(\varepsilon_{\Omega+1} + \Omega^2) = H[[suc \to R_2 R_1, 0 \to R_{\omega \dots 1} H] \bullet suc\, 0](suc\, 0) = R_2(R_2 R_1)(R_{\omega \dots 1} H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} + \Omega^\alpha) = [suc \to R_2, 0 \to R_1]\alpha(R_{\omega \dots 1} H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} + \Omega^\Omega) = H[[suc \to R_2, 0 \to R_1] \bullet (R_{\omega \dots 1} H) suc\, 0](suc\, 0) = R_3 R_2 R_1(R_{\omega \dots 1} H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} + \Omega^{\Omega^\Omega}) = R_4 R_3 R_2 R_1(R_{\omega \dots 1} H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} + \varepsilon_{\Omega+1}) = \psi(\varepsilon_{\Omega+1} \times 2) = R_{\omega \dots 1}(R_{\omega \dots 1} H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \alpha) = [suc \to R_{\omega \dots 1}, 0 \to H]\alpha suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega) = H[[suc \to R_{\omega \dots 1}, 0 \to H] \bullet suc\, 0](suc\, 0) = R_2 R_{\omega \dots 1} H suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega + 1) = R_1(R_2 R_{\omega \dots 1} H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega + \alpha) = [suc \to R_1, 0 \to R_2 R_{\omega \dots 1} H]\alpha suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega + \Omega) = H[[suc \to R_1, 0 \to R_2 R_{\omega \dots 1} H] \bullet suc\, 0](suc\, 0) = R_2 R_1(R_2 R_{\omega \dots 1} H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega + \Omega + 1) = R_1(R_2 R_1(R_2 R_{\omega \dots 1} H)) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega + \Omega + \alpha) = [suc \to R_1, 0 \to R_2 R_1(R_2 R_{\omega \dots 1} H)]\alpha suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega + \Omega + \Omega) = \psi(\varepsilon_{\Omega+1} \times \Omega + \Omega \times 2) = H[[suc \to R_1, 0 \to R_2 R_1(R_2 R_{\omega \dots 1} H)] \bullet suc\, 0](suc\, 0) = R_2 R_1(R_2 R_1(R_2 R_{\omega \dots 1} H)) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega + \Omega \times \alpha) = [suc \to R_2 R_1, 0 \to R_2 R_{\omega \dots 1} H]\alpha suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega + \Omega \times \Omega) = \psi(\varepsilon_{\Omega+1} \times \Omega + \Omega^2) = H[[suc \to R_2 R_1, 0 \to R_2 R_{\omega \dots 1} H] \bullet suc\, 0](suc\, 0) = R_2(R_2 R_1)(R_2 R_{\omega \dots 1} H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega + \Omega^\alpha) = [suc \to R_2, 0 \to R_1]\alpha(R_2 R_{\omega \dots 1} H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega + \Omega^\Omega) = H[[suc \to R_2, 0 \to R_1] \bullet (R_2 R_{\omega \dots 1} H) suc\, 0](suc\, 0) = R_3 R_2 R_1(R_2 R_{\omega \dots 1} H) suc_0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega + \Omega^{\Omega^\Omega}) = R_4 R_3 R_2 R_1(R_2 R_{\omega \dots 1} H) suc_0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega + \varepsilon_{\Omega+1}) = \psi(\varepsilon_{\Omega+1} \times (\Omega + 1)) = R_{\omega \dots 1}(R_2 R_{\omega \dots 1} H) suc_0$
- $\psi(\varepsilon_{\Omega+1} \times (\Omega + 2)) = R_{\omega \dots 1}(R_{\omega \dots 1}(R_2 R_{\omega \dots 1} H)) suc_0$
- $\psi(\varepsilon_{\Omega+1} \times (\Omega + \alpha)) = [suc \to R_{\omega \dots 1}, 0 \to R_2 R_{\omega \dots 1} H]\alpha suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times (\Omega + \Omega)) = \psi(\varepsilon_{\Omega+1} \times \Omega \times 2) = H[[suc \to R_{\omega \dots 1}, 0 \to R_2 R_{\omega \dots 1} H] \bullet suc\, 0](suc\, 0) = R_2 R_{\omega \dots 1}(R_2 R_{\omega \dots 1} H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega \times \alpha) = [suc \to R_2 R_{\omega \dots 1}, 0 \to H]\alpha suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega \times \Omega) = \psi(\varepsilon_{\Omega+1} \times \Omega^2) = H[[suc \to R_2 R_{\omega \dots 1}, 0 \to H] \bullet suc\, 0](suc\, 0) = R_2(R_2 R_{\omega \dots 1}) H suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega^\alpha) = [suc \to R_2, 0 \to R_{\omega \dots 1}]\alpha H suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega^\Omega) = H[[suc \to R_2, 0 \to R_{\omega \dots 1}] \bullet H suc\, 0](suc\, 0) = R_3 R_2 R_{\omega \dots 1} H suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \Omega^{\Omega^\Omega}) = R_4 R_3 R_2 R_{\omega \dots 1} H suc\, 0$
- $\psi(\varepsilon_{\Omega+1} \times \varepsilon_{\Omega+1}) = \psi(\varepsilon_{\Omega+1}{}^2) = R_{\omega \dots 2} R_{\omega \dots 1} H suc\, 0 = R_{\omega \times 2 \dots 1} H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^\alpha) = R_{\omega \times \alpha \dots 1} H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^\Omega) = H[R_{\omega \times \bullet \dots 1} H suc\, 0]0 = H[R_{\bullet \dots 1} H suc\, 0]0 = R_1^1 H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^\Omega + 1) = R_1(R_1^1) H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^\Omega + \alpha) = [suc \to R_1, 0 \to R_1^1 H]\alpha suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^\Omega + \Omega) = H[[suc \to R_1, 0 \to R_1^1 H] \bullet suc\, 0](suc\, 0) = R_2 R_1(R_1^1 H) suc\, 0$
- $\dots$
- $\psi(\varepsilon_{\Omega+1}{}^\Omega + \Omega^\Omega) = R_3 R_2 R_1(R_1^1 H) suc\, 0$

- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} + \varepsilon_{\Omega+1}) = R_{\omega\ldots 1}(R_1^1 H) suc\, 0$
- ...
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} + \varepsilon_{\Omega+1} \times 2) = R_{\omega\ldots 1}(R_{\omega\ldots 1}(R_1^1 H)) suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} + \varepsilon_{\Omega+1} \times \alpha) = [suc \to R_{\omega\ldots 1}, 0 \to R_1^1 H]\alpha suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} + \varepsilon_{\Omega+1} \times \Omega) = H[[suc \to R_{\omega\ldots 1}, 0 \to R_1^1 H] \bullet suc\, 0](suc\, 0) = R_2 R_{\omega\ldots 1}(R_1^1 H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} + \varepsilon_{\Omega+1} \times \Omega + \varepsilon_{\Omega+1}) = \psi(\varepsilon_{\Omega+1}{}^{\Omega} + \varepsilon_{\Omega+1} \times (\Omega+1)) = R_{\omega\ldots 1}(R_2 R_{\omega\ldots 1}(R_1^1 H)) suc\, 0$
- ...
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} + \varepsilon_{\Omega+1} \times \Omega^{\Omega}) = R_3 R_2 R_{\omega\ldots 1}(R_1^1 H) suc\, 0$
- ...
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} + \varepsilon_{\Omega+1}^2) = R_{\omega\ldots 2} R_{\omega\ldots 1}(R_1^1 H) suc\, 0 = R_{\omega\times 2\ldots 1}(R_1^1 H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} + \varepsilon_{\Omega+1}^{\alpha}) = R_{\omega\times\alpha\ldots 1}(R_1^1 H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} + \varepsilon_{\Omega+1}^{\Omega}) = \psi(\varepsilon_{\Omega+1}{}^{\Omega} \times 2) = H[R_{\bullet\ldots 1}(R_1^1 H) suc\, 0]0 = R_1^1(R_1^1 H) suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} \times \alpha) = [suc \to R_1^1, 0 \to H]\alpha suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} \times \Omega) = H[[suc \to R_1^1, 0 \to H] \bullet suc\, 0](suc\, 0) = R_2 R_1^1 H suc\, 0$
- ...
- $\psi(\ldots \times \Omega) = R_2 \ldots H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} \times \Omega^2) = R_2(R_2 R_1^1) H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} \times \Omega^{\alpha}) = [suc \to R_2, 0 \to R_1^1]\alpha H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} \times \Omega^{\Omega}) = H[[suc \to R_2, 0 \to R_1^1] \bullet H suc\, 0](suc\, 0) = R_3 R_2 R_1^1 H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} \times \varepsilon_{\Omega+1}) = \psi(\varepsilon_{\Omega+1}{}^{\Omega+1}) = R_{\omega\ldots 2} R_1^1 H suc\, 0$
- ...
- $\psi(\ldots \times \varepsilon_{\Omega+1}) = R_{\Omega\ldots\ldots} \ldots H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} \times \varepsilon_{\Omega+1}^2) = R_{\omega\ldots 3} R_{\omega\ldots 2} R_1^1 H suc\, 0 = R_{\omega\times 2\ldots 2} R_1^1 H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega} \times \varepsilon_{\Omega+1}^{\Omega}) = \psi((\varepsilon_{\Omega+1}{}^{\Omega})^2) = \psi(\varepsilon_{\Omega+1}^{\Omega\times 2}) = R_2^2 R_1^1 H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega\times\alpha}) = R_{\alpha\ldots 1}^1 H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega^2} = H[R_{\bullet\ldots 1}^1 H suc\, 0]0 = R_1^2 H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega^{\alpha}} = R_1^{\alpha} H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\Omega^{\Omega}} = H[R_1^{\bullet} H suc\, 0]0 = R_1^{1,0} H suc\, 0 = R_1^{\Omega} H suc\, 0 = R_{\Omega^{\Omega}} H suc\, 0 = R_1^{H_1 suc\, 0} H suc\, 0 = R_{H_1 H_1 suc\, 0} H suc\, 0$
- $\psi(\varepsilon_{\Omega+1}{}^{\varepsilon_{\Omega+1}}) = R_{\varepsilon_{\Omega+1}} H suc\, 0 = R_{R_1 H_1 suc\, 0} H suc\, 0$
- $\psi(\varepsilon_{\Omega+2}) = R_1^{H[R_1^{\bullet} H_1 suc\, 0]0} H suc\, 0$
- ...

Proof of $sup\{\zeta_0, \zeta_0{}^{\zeta_0}, \zeta_0{}^{\zeta_0{}^{\zeta_0}}, \ldots\} = \varepsilon_{\zeta_0+1}$:
We have $\varepsilon_{\alpha+1} = sup\{\varepsilon_\alpha + 1, \omega^{\varepsilon_\alpha+1}, \omega^{\omega^{\varepsilon_\alpha+1}}, \ldots\}$, which gives for ζ_0 : $\varepsilon_{\zeta_0+1} = sup\{\varepsilon_{\zeta_0} + 1, \omega^{\varepsilon_{\zeta_0}+1}, \omega^{\omega^{\varepsilon_{\zeta_0}+1}}, \ldots\}$ so we have to prove
$sup\{\varepsilon_{\zeta_0} + 1, \omega^{\varepsilon_{\zeta_0}+1}, \omega^{\omega^{\varepsilon_{\zeta_0}+1}}, \ldots\} = sup\{\zeta_0, \zeta_0{}^{\zeta_0}, \zeta_0{}^{\zeta_0{}^{\zeta_0}}, \ldots\}$.
We have already seen concerning Veblen functions that the ordinals respectively limits of the fondamental sequence whose n-th

term is $\varepsilon_0^{\varepsilon_0^{\cdot^{\cdot^{\cdot^{\varepsilon_0^{\omega}}}}}}$ and the one whose n-th term is $\varepsilon_0^{\varepsilon_0^{\cdot^{\cdot^{\cdot^{\varepsilon_0^{\varepsilon_0}}}}}}$ is the same, the least fixed point of the function $\alpha \mapsto \varepsilon_0^{\alpha}$, which is greater than ω and also than ε_0.
For a similar reason, we also have $sup\{\zeta_0, \zeta_0{}^{\zeta_0}, \zeta_0{}^{\zeta_0{}^{\zeta_0}}, \ldots\} = sup\{\zeta_0{}^{\omega}, \zeta_0{}^{\zeta_0{}^{\omega}}, \zeta_0{}^{\zeta_0{}^{\zeta_0{}^{\omega}}}, \ldots\}$.
So we have to prove $sup\{\varepsilon_{\zeta_0} + 1, \omega^{\varepsilon_{\zeta_0}+1}, \omega^{\omega^{\varepsilon_{\zeta_0}+1}}, \ldots\} = sup\{\zeta_0{}^{\omega}, \zeta_0{}^{\zeta_0{}^{\omega}}, \zeta_0{}^{\zeta_0{}^{\zeta_0{}^{\omega}}}, \ldots\}$.

We can prove this by proving $\omega^{\omega^{\cdot^{\cdot^{\cdot^{\omega^{\omega^{\varepsilon_{\zeta_0}+1}}}}}}} = \zeta_0^{\zeta_0^{\cdot^{\cdot^{\cdot^{\zeta_0^{\omega}}}}}}$ in a similar way we proved $\omega^{\omega^{\cdot^{\cdot^{\cdot^{\omega^{\omega^{\varepsilon_0+1}}}}}}} = \varepsilon_0^{\varepsilon_0^{\cdot^{\cdot^{\cdot^{\varepsilon_0^{\omega}}}}}}$, where $\alpha^{\cdot^{\cdot^{\cdot^{\alpha^{\beta}}}}}$ represents an "exponential tower" with α repeated n times.
For n = 0, we have :
$\omega^{\omega^{\varepsilon_{\zeta_0}+1}} = \omega^{\omega^{\zeta_0+1}} = \omega^{\omega^{\zeta_0}\times\omega} = \omega^{\zeta_0\times\omega} = (\omega^{\zeta_0})^{\omega} = \zeta_0{}^{\omega}$.

Suppose we have $\omega^{\omega^{\cdot^{\cdot^{\cdot^{\omega^{\omega^{\varepsilon_{\zeta_0}+1}}}}}}} = \zeta_0^{\zeta_0^{\cdot^{\cdot^{\cdot^{\zeta_0^{\omega}}}}}}$.

We must prove the equality for n+1, which can be written $\omega^{\omega^{\cdot^{\cdot^{\cdot^{\omega^{\omega^{\varepsilon_{\zeta_0}+1}}}}}}} = \zeta_0^{\zeta_0^{\zeta_0^{\cdot^{\cdot^{\cdot^{\zeta_0^{\omega}}}}}}}$.

We have $\omega^{\omega^{\omega^{\iddots^{\omega^{\omega^{\varepsilon_{\zeta_0}+1}}}}}}_{\zeta_0 \omega} = \omega^{\zeta_0^{\zeta_0^{\iddots^{\zeta_0 \omega}}}} $ (by our hypothesis) $= \omega^{\zeta_0^{1+\zeta_0^{\iddots^{\zeta_0 \omega}}}} $ (for the same reason than $1 + \omega = \omega$, see above) $=$

$\omega^{\zeta_0 \times \zeta_0^{\iddots^{\zeta_0 \omega}}} = (\omega^{\zeta_0})^{\zeta_0^{\iddots^{\zeta_0 \omega}}} = \zeta_0^{\zeta_0^{\iddots^{\zeta_0 \omega}}} $. QED.

In RHSZ notation, this corresponds to the equality $H(R_2 R_1 H) H \ldots H suc\ 0 = H(R_2 R_1 H)(R_2 R_1 H) \ldots (R_2 R_1 H) suc\ 0 = R_1(R_2 R_1 H) suc\ 0$.

10.4 A variant of Madore's ψ function

Another approach could consist, instead of defining ordinal collapsing functions by taking the least ordinal which cannot be constructed using a given set of ordinals and operations, in defining it recursively according to the value of the variable.

Let us call ψ' this new collapsing function.

First, to be consistent with the rationalized functions previously defined, we would like to have $\psi'(\alpha) = \varepsilon'_\alpha$ instead of $\psi(\alpha) = \varepsilon_\alpha$.

So $\psi'(0)$ must be equal to ε'_0. As we have $\varepsilon'_{\alpha+1} = sup\{\varepsilon'_\alpha, \varepsilon'_\alpha{}^{\varepsilon'_\alpha}, \varepsilon'_\alpha{}^{\varepsilon'_\alpha{}^{\varepsilon'_\alpha}}, \ldots\}$, it is consistent to define $\varepsilon'_0 = \omega$, because $\varepsilon'_1 = \varepsilon_0 = sup\{\omega, \omega^\omega, \omega^{\omega^\omega}, \ldots\}$.

Then we can define $\psi'(\alpha + 1)$ as the limit or least upper bound of $\psi'(\alpha), \psi'(\alpha)^{\psi'(\alpha)}, \psi'(\alpha)^{\psi'(\alpha)^{\psi'(\alpha)}}, \ldots$.

Then we can define canonically $\psi'(lim(f)) = lim(n \mapsto \psi'(f(n))$.

Then we must define ψ' for the case when it collapses an uncountable ordinal, in such a way that $\psi'(f(\Omega))$, for f correctly defined, is the least fixed point of $\psi' \circ f$, which can be written for example $sup\{1, \psi'(f(1)), \psi'(f(_psi'(f(1)))), \ldots\}$ as we previously saw. In RHS0 notation, if we write $\Omega = H_1 suc\ 0$, this gives : $\psi(H_1 xyz_1 \ldots z_n) = H[\psi([suc \to x, 0 \to y] \bullet z_1 \ldots z_n)](suc\ 0)$.

In summary, we can define the ψ' function by :

- $\psi'(0) = \omega$
- $\psi'(\alpha + 1) = sup\{\psi'(\alpha), \psi'(\alpha)^{\psi'(\alpha)}, \psi'(\alpha)^{\psi'(\alpha)^{\psi'(\alpha)}}, \ldots\}$
- $\psi'(lim(f)) = lim(n \mapsto \psi'(f(n))$
- $\psi(H_1 xyz_1 \ldots z_n) = H[\psi([suc \to x, 0 \to y] \bullet z_1 \ldots z_n)](suc\ 0)$

Note that this definition does not give exactly the same function as Madore's ψ, apart from the shift of one unit for finite values due to rationalization : for example, we have $\psi(\zeta_0 + 1) = \zeta_0$, but $\psi'(\zeta_0 + 1) = sup\{\psi'(\zeta_0), \psi'(\zeta_0)^{\psi'(\zeta_0)}, \psi'(\zeta_0)^{\psi'(\zeta_0)^{\psi'(\zeta_0)}}, \ldots\} = \varepsilon_{\zeta_0+1} = \varepsilon'_{\zeta'_1+1}$.

Scheme inplementation :

```
(define mp (lambda (x)
(if (pair? x)
(if (eq? (car x) ':) (list (mp (cdr x)))
 (cons (mp (car x)) (mp (cdr x))) )
x ) ))

(eval (mp '(begin

(define last : lambda (l) :
 if (not : pair? l) l :
 if (not : pair? : cdr l) l :
 last : cdr l)

(define butlast : lambda (l) :
 if (not : pair? l) '() :
 if (not : pair? : cdr l) '() :
 cons (car l) : butlast : cdr l)

(define length : lambda (l) :
 if (not : pair? l) 0 :
```

```scheme
    + 1 : length : cdr l)

(define r2a : lambda (l) :
 if (not : pair? l) l :
 if (pair? : car l) (r2 : append (car l) : cdr l) :
 if (eq? (car l) 'suc) (cdr l) :
 if (eq? (car l) 'H) (cons (cadr l) : cons (list (cadr l) (caddr l)) : cdddr l) :
 if (eq? (car l) 'R1) (cons (cadr l) : cons (cadr l) : cddr l) :
 if (eq? (car l) 'R2) (cons (cadr l) : cons (caddr l) : cons (cadr l) : cons (caddr l) : cdddr l)
 l)

(define r : lambda (l) :
 if (not : pair? l) l :
 let ((l1 (map r l))) :
 if (pair? : car l1) (append (car l1) (cdr l1))
 l1)

(define r2 : lambda (l) : r : r2a l)

(define loopr2 : lambda (n l) :
 begin (display n) (display " ") (display l) (newline) :
 if (equal? n 0) '() :
 loopr2 (- n 1) (r2 l))

(loopr2 10 '(R1 H suc 0))

(define simplif : lambda (x) :
 if (not : pair? x) x :
 if (not : pair? : cdr x) (car x)
 x)

(define subst : lambda (s z a) :
 if (equal? '0 a) z :
 if (equal? 'suc a) s :
 if (not : pair? a) a :
 cons (subst s z : car a) (subst s z : cdr a))

(define format : lambda (a) :
 if (not : pair? a) a :
 if (not : pair? : car a) (cons (car a) : map format : cdr a) :
 format : append (car a) (cdr a))

(define memo '())

(define find : lambda (a memo) :
 if (not : pair? memo) '#f :
 if (equal? a : caar memo) (cdar memo) :
 find a : cdr memo)

(define psi : lambda (a) :
 let ((m : find a memo)) :
 if m m :
 let ((b : psi1 a))
  (if (or (not : pair? a) (not : pair? : car a))
   (begin
    (display "psi ") (display : format a) (display " = ") (display : format b) (newline)
```

```
;    (read-char)
     ;(newline)
  ))
  (set! memo : cons (cons a b) memo)
  b)

(define psi1 : lambda (a) :
 if (not : pair? a) a :
 if (pair? : car a) (psi : myappend (car a) (cdr a)) :
 if (equal? (car a) '0) '(H suc 0) :
 if (equal? 'suc : car a)
  (let ((b : psi : cdr a)) :
   if (and (equal? '(0) : last b)
           (equal? '(suc) : last : butlast b))
    (list 'R1 (simplif : butlast : butlast b) 'suc '0)
    (list 'psi a)) :
 if (and (equal? 'H : car a) (>= (length a) 3))
  (limit (psi : cddr a)
         (psi : cdr a)
         (psi : cons (cadr a) : cons (list (cadr a) (caddr a)) : cdddr a)) :
 if (and (equal? 'R1 : car a) (>= (length a) 2))
  (limit (psi : cddr a)
         (psi : cdr a)
         (psi : cons (cadr a) : cdr a)) :
 if (and (equal? 'R2 : car a) (>= (length a) 3))
  (limit (psi : cdddr a)
         (psi : cdr a)
         (psi : cons (cadr a) : cons (caddr a) : cdr a)) :
 if (and (equal? 'H1 : car a) (>= (length a) 3))
  (let ((b : psi : cdr a)) :
   limit '(suc 0)
         b
         (psi : myappend (subst (cadr a) (caddr a) b) : cdddr a))
 a)

(define myappend : lambda (a b) :
 if (not : pair? a) (cons a b) :
 append a b)

(define commonstart : lambda (a b) :
 if (not : pair? a) (list '() a b) :
 if (not : pair? b) (list '() a b) :
 if (not : equal? (car a) (car b)) (list '() a b) :
 let ((c : commonstart (cdr a) (cdr b))) :
 let ((com : car c) (dif1 : cadr c) (dif2 : caddr c)) :
 list (cons (car a) com) dif1 dif2)

(define limit : lambda (a b c) :
 if (and (equal? (cdr a) (cddr b))
         (equal? (cdr a) (cddr c))
         (equal? a (cdr b))
         (equal? (car b) (car c))
         (equal? (cadr c) (list (car b) (cadr b))))
  (cons 'H b) :
 if (and (equal? a (myappend (cadr b) (cddr b)))
         (equal? (car b) (car c))
```

```
          (equal? (cadr c) (list (car b) (cadr b)))
          (equal? (cddr b) (cddr c)))
    (cons 'H b) :
  let ((d : commonstart b c)) :
  let ((com : car d) (difb : cadr d) (difc : caddr d)) :
  if (and (pair? com)
          (equal? com : butlast : car difc)
          (equal? (car difb) (car : last : car difc))
          (equal? a : myappend (car difb) (cdr difb))
          (equal? (cdr difb) (cdr difc)))
    (cons 'H : cons com difb) :
  if (and (equal? a : cdr b)
          (equal? b : cdr c)
          (equal? (car b) (car c))
          (equal? (car c) (cadr c)))
    (cons 'R1 b) :
  if (and (equal? a : myappend (cadr b) : cddr b)
          (equal? b : cdr c)
          (equal? (car c) (cadr c)))
    (cons 'R1 b) :
  if (and (equal? a difb)
          (equal? difc : cons com difb))
    (cons 'R1 : cons com difb) :
  if (and (equal? a : cddr b)
          (equal? b : cddr c)
          (equal? (car b) (car c))
          (equal? (cadr b) (cadr c))
          (equal? (car c) (caddr c))
          (equal? (cadr c) (cadddr c)))
    (cons 'R2 b) :
  list 'limit a b c)

;(display : psi '(H H suc 0))
(display : psi '(H1 suc 0))
;(display : psi '(H1 suc (H1 suc 0)))
(newline)

)))



C implementation :


#include <stdlib.h>
#include <stdio.h>
#include <string.h>

typedef long expr;

int ispair (expr x)
{
    return x < 0;
}

int isatom (expr x)
{
```

```c
        return x >= 0;
}

expr atom (char *s)
{
    expr x;
    x = *(int *)s;
    if (x < 0 || strlen(s) >= sizeof(expr))
    {
        printf ("Bad name for atom: %s\n", s);
        exit(0);
    }
    return x;
}

struct pair
{
    expr fst, snd;
};

#define SIZE 10000000

int npairs = 0;

struct pair mem[SIZE];

expr fst (expr x)
{
    if (!ispair(x))
    {
        printf ("fst of not pair 0x%X\n", x);
        exit(0);
    }
    return mem[-x-1].fst;
}

expr snd (expr x)
{
    if (!ispair(x))
    {
        printf ("fst of not pair 0x%X\n", x);
        exit(0);
    }
    return mem[-x-1].snd;
}

expr newpair (expr x, expr y)
{
    if (npairs >= SIZE)
    {
        printf("Overflow\n");
        exit(0);
    }
    mem[npairs].fst = x;
    mem[npairs].snd = y;
    npairs++;
```

```
        return -npairs;
}

expr findpair (expr x, expr y)
{
    int i;
    for (i=0; i<npairs; i++)
    {
        if (mem[i].fst == x && mem[i].snd == y)
            return -i-1;
    }
    return 0;
}

expr pair (expr x, expr y)
{
    expr z;
    z = findpair (x, y);
    if (z)
        return z;
    else
        return newpair (x, y);
}

expr eq (expr x, expr y)
{
    return x == y;
}

struct charwriter
{
    int (*f) (struct charwriter *, char);
};

int writechar (struct charwriter *cw, char c)
{
    return (*(cw->f))(cw,c);
}

void writeexpr (struct charwriter *cw, expr x)
{
    char s[sizeof(expr)+1];
    int i;
    if (isatom(x))
    {
        for (i=0; i<sizeof(s); i++)
            s[i] = 0;
        memcpy (s, &x, sizeof(s));
        for (i=0; s[i]; i++)
            writechar(cw,s[i]);
    }
    else
    {
        writechar(cw,'-');
        writeexpr(cw,fst(x));
        writechar(cw,' ');
```

```
        writeexpr(cw,snd(x));
    }

}

expr zero, suc, H, H1, R1, R2, R3, Psi, lim, w;

#define ap(x,y) pair(x,y)
#define fnc(x) fst(x)
#define arg(x) snd(x)
#define isap(x) ispair(x)
#define isnap(x) isatom(x)

init ()
{
    zero = atom("0");
    suc = atom("suc");
    H = atom("H");
    H1 = atom("H1");
    R1 = atom("R1");
    R2 = atom("R2");
    R3 = atom("R3");
    Psi = atom("psi");
    lim = atom("lim");
    w = ap(ap(H,suc),zero);
}

expr first (expr a)
{
    if (isnap(a))
        return ap(atom("fst"),a);
    if (isap(fnc(a)) && eq(H,fnc(fnc(a))))
        return arg(a);
    if (isap(fnc(a)) && eq(R1,fnc(fnc(a))))
        return arg(a);
    if (isap(fnc(a)) && isap(fnc(fnc(a))) && eq(R2,fnc(fnc(fnc(a)))))
        return arg(a);
    return ap(first(fnc(a)),arg(a));
}

expr next (expr a)
{
    if (isnap(a))
        return ap(atom("nxt"),a);
    if (isap(fnc(a)) && eq(H,fnc(fnc(a))))
        return ap(fnc(a),ap(arg(fnc(a)),arg(a)));
    if (eq(R1,fnc(a)))
        return ap(ap(R1,arg(a)),arg(a));
    if (isap(fnc(a)) && eq(R2,fnc(fnc(a))))
        return ap(ap(a,arg(fnc(a))),arg(a));
    return ap(next(fnc(a)),arg(a));
}

expr subst (expr s, expr z, expr a)
{
    if (eq(zero,a))
```

```
            return z;
        if (eq(suc,a))
            return s;
        if (isnap(a))
            return a;
        return ap(subst(s,z,fnc(a)),subst(s,z,arg(a)));
}

expr limit (expr a, expr b, expr c)
{
        if (isap(b) &&
            eq(a,arg(b)) &&
            isap(c) &&
            eq(fnc(b),fnc(c)) &&
            eq(b,arg(c)))
            return ap(ap(H,fnc(b)),a);
        if (isap(b) &&
            eq(a,arg(b)) &&
            isap(c) &&
            eq(a,arg(c)) &&
            isap(fnc(c)) &&
            eq(fnc(b),fnc(fnc(c))) &&
            eq(fnc(b),arg(fnc(c))))
            return ap(ap(R1,fnc(b)),a);
        if (isap(b) &&
            eq(a,arg(b)) &&
            isap(c) &&
            eq(a,arg(c)) &&
            isap(fnc(c)) &&
            isap(fnc(fnc(c))) &&
            eq(fnc(b),fnc(fnc(fnc(c)))) &&
            isap(fnc(b)) &&
            eq(arg(fnc(b)),arg(fnc(c))) &&
            eq(fnc(fnc(b)),arg(fnc(fnc(c)))))
            return ap(ap(ap(R2,fnc(fnc(b))),arg(fnc(b))),a);
        if (isap(a) && isap(b) && isap(c) &&
            eq(arg(a),arg(b)) && eq(arg(b),arg(c)))
            return ap(limit(fnc(a),fnc(b),fnc(c)),arg(a));
        return ap(ap(ap(lim,a),b),c);
}

#define MAXMEMO SIZE

int nmemo = 0;

struct item
{
    expr arg;
    expr val;
};

struct item memo[MAXMEMO];

int count = 0;

int cwf_putchar (struct charwriter *cw, char c)
```

```
{
    return putchar(c);
}

expr psi2 (int level, expr a);

expr psi (expr a)
{
    return psi2(0,a);
}

expr psi1 (int level, expr a);

expr psi2 (int level, expr a)
{
    expr b;
    struct charwriter cw;
    int i;
    for (i=0; i<nmemo; i++)
    {
        if (eq(a,memo[i].arg))
            return memo[i].val;
    }
    b = psi1(level,a);
    memo[nmemo].arg = a;
    memo[nmemo].val = b;
    nmemo++;
    cw.f = cwf_putchar;
    count++;

    printf ("\n %6d %3d ", count, level);
    writeexpr (&cw, a);
    printf ("\n                ");
    writeexpr (&cw, b);
    printf ("\n");

    return b;
}

expr psi1 (int level, expr a)
{
    expr b, c, d;
    if (eq(zero,a))
        return w;
    if (isnap(a))
        return ap(Psi,a);
    if (eq(suc,fnc(a)))
    {
        expr c;
        c = psi2(level+1,arg(a));
        if (isap(c) && eq(zero,arg(c)) && isap(fnc(c)) && eq(suc,arg(fnc(c))))
            return ap(ap(ap(R1,fnc(fnc(c))),suc),zero);
    }
    if (isnap(fnc(a)))
        return ap(Psi,a);
    if (eq(H1,fnc(fnc(a))))
```

```
    {
        expr b, c, d;
        b = ap(suc,zero);
        c = psi2(level+1,ap(arg(fnc(a)),arg(a)));
        d = psi2(level+1,subst(arg(fnc(a)),arg(a),c));
        return limit(b,c,d);
    }
    /*if (eq(H,fnc(fnc(a))))
        return limit (
            psi(arg(a)),
            psi(ap(arg(fnc(a)),arg(a))),
            psi(ap(arg(fnc(a)),ap(arg(fnc(a)),arg(a))))
            );*/
    //return limit (psi2(level+1,first(a)), psi2(level+1,first(next(a))), psi2(level+1,first(next(next(a)))));
    b = psi2(level+1,first(a));
    c = psi2(level+1,first(next(a)));
    d = psi2(level+1,first(next(next(a))));
    return limit(b,c,d);
    //return ap(Psi,a);
}

dump ()
{
    int i;
    for (i=0; i<npairs; i++)
    {
        printf(" %4d %08X : %08X %08X \n", i, -i-1, mem[i].fst, mem[i].snd);
    }
}

main ()
{
    struct charwriter cw;
    cw.f = cwf_putchar;

    printf(" %d ", sizeof(expr));
    expr x;
    x = pair (pair (atom("abc"), atom("def")), atom("ghi"));
    // x = pair (atom("abc"), atom("def"));
    printf("x = %d\n",x);
    printf("mem: %X %X\n", mem[0].fst, mem[0].snd);
    writeexpr(&cw,x);

    printf("\n");

    init();

    expr a, b, c, d, f;
    a = ap(ap(H,suc),zero);
    b = next(a);
    printf ("b = ");
    writeexpr(&cw,b);
    printf("\n");

    a = ap(ap(ap(R1,H),suc),zero);
    b = first(next(next(a)));
```

```
printf ("b = ");
writeexpr(&cw,b);
printf("\n");

a = ap(ap(ap(ap(R2,R1),H),suc),zero);
b = first(next(next(a)));
printf ("b = ");
writeexpr(&cw,b);
printf("\n");

x = atom("x");
f = atom("f");
a = x;
b = ap(f,a);
c = ap(f,b);
d = limit(a,b,c);
printf ("d = ");
writeexpr(&cw,d);
printf("\n");
//a = ap(suc,ap(suc,zero));
//a = ap(ap(ap(H,H),suc),zero);
//a = ap(ap(ap(R1,H),suc),zero);

a = ap(ap(H1,suc),zero);

//a = ap(ap(ap(R1,H),suc),ap(ap(H1,suc),zero));
//a = ap(ap(H1,suc),ap(ap(H1,suc),zero));
//a = ap(suc,ap(ap(H1,suc),zero));
//a = ap(ap(ap(ap(R2,R1),H),suc),ap(ap(H1,suc),zero));
//a = ap(ap(ap(R1,H),suc),ap(ap(H1,suc),zero));
//a = ap(ap(ap(ap(H,R1),H),suc),ap(ap(H1,suc),zero));
//a = ap(ap(H,ap(ap(ap(H,R1),H),suc)),ap(ap(H1,suc),zero));
//a = ap(ap(ap(R1,H),ap(ap(ap(H,R1),H),suc)),ap(ap(H1,suc),zero));

//a = ap(ap(ap(H,ap(R1,H)),ap(ap(ap(H,R1),H),suc)),ap(ap(H1,suc),zero));

//a = ap(ap(ap(H,ap(H,ap(R1,H))),ap(ap(ap(H,R1),H),suc)),ap(ap(H1,suc),zero));
//a = ap(ap(ap(ap(H,H),ap(R1,H)),ap(ap(ap(H,R1),H),suc)),ap(ap(H1,suc),zero));
//a = ap(ap(ap(ap(R1,H),ap(R1,H)),ap(ap(ap(H,R1),H),suc)),ap(ap(H1,suc),zero));
//a = ap(ap(ap(R1,ap(R1,H)),ap(ap(ap(H,R1),H),suc)),ap(ap(H1,suc),zero));
//a = ap(ap(ap(ap(H,R1),H),ap(ap(ap(H,R1),H),suc)),ap(ap(H1,suc),zero));
//a = ap(ap(ap(H,ap(ap(H,R1),H)),suc),ap(ap(H1,suc),zero));
//a = ap(ap(ap(ap(H,R1),H),ap(ap(H,R1),H)),suc),ap(ap(H1,suc),zero));
//a = ap(ap(ap(R1,ap(ap(H,R1),H)),suc),ap(ap(H1,suc),zero));
//a = ap(ap(ap(ap(H,R1),ap(ap(H,R1),H)),suc),ap(ap(H1,suc),zero));
//a = ap(ap(ap(ap(H,ap(H,R1)),H),suc),ap(ap(H1,suc),zero));
//a = ap(ap(ap(ap(ap(H,H),R1),H),suc),ap(ap(H1,suc),zero));
//a = ap(ap(ap(ap(ap(R1,H),R1),H),suc),ap(ap(H1,suc),zero));

//a = ap(ap(H1,suc),ap(ap(H1,suc),zero));

b = psi(a);
//dump();
printf ("\na=%X b=%X\n",a,b);
printf("psi ");
writeexpr(&cw,a);
```

```
        printf (" = ");
        writeexpr(&cw,b);
        printf("\n");

        printf("npairs = %d\n", npairs);

}
```

10.5 θ function

θ function is a binary function. It's defined as follows:

- $C_0(\alpha, \beta) = \{\gamma | \gamma < \beta\} \cup \{0\}$.

- $C_{n+1}(\alpha, \beta) = \{\gamma + \delta | \gamma, \delta \in C_n(\alpha, \beta)\} \cup \{\theta(\gamma, \delta) | \gamma < \alpha \& \gamma, \delta \in C_n(\alpha, \beta)\} \cup \{\Omega_c | c \in C_n(\alpha, \beta)\}$.

- $C(\alpha, \beta) = \cup_{n<\omega} C_n(\alpha, \beta)$

- $\theta(\alpha, \beta) = min\{c| \ (c \in C(\alpha, \gamma) \& (\forall \delta < \beta : \gamma > \theta(\alpha, \delta))\}$

  where $\Omega_0 = 0$ and Ω_a represents the a-th uncountable ordinal.

It means that $\theta(\alpha, \beta)$ is the $(1+\beta)$-th ordinal such that it cannot be built from ordinals less than it by addition, applying $\theta(\delta, \ldots)$ where $\delta < \alpha$ and getting an uncountable cardinal.
It seems that $\theta(\alpha, \beta) = \varphi(\alpha, \beta)$ below Γ_0, making θ function an extension of φ function. Even $\theta(\Gamma_0, \beta) = \varphi(\Gamma_0, \beta)$ is true.

Other important values are :

- $\theta(\Omega, \alpha) = \Gamma_\alpha$

- $\theta(\Omega^\omega, 0) =$ small Veblen ordinal

- $\theta(\Omega^\Omega, 0) =$ large Veblen ordinal

- $\theta(\varepsilon_{\Omega+1}, 0) =$ Bachmann Howard ordinal

See https://stepstowardinfinity.wordpress.com/2015/05/04/ordinal2/ for more information.

10.6   Taranovsky's C

C(a,b) is the least element above b that has degree a.
Definition: A degree for a well-ordered set S is a binary relation on S such that :

- Every element $c \in S$ has degree 0_S (the least element of S). 0_S only has degree 0_S.

- For a limit a, c has degree a iff it has every degree less than a.

- For a successor a'=a+1, either of the following holds:

  - An element has degree a' iff it is a limit of elements of degree a.
  - There is a limit element d ¡= a such that for every c in S, c has degree a' iff it has degree a and either c ¡= d or c is a limit of elements of degree a (or both).

Note: The third condition can be equivalently written as $\forall a(C_{a+1} = lim(C_a) \lor \exists d \in lim(S) \cap (a+1) C_{a+1} = lim(C_a) \cup (C_a \cap (d+1)))$, where S is identified with an ordinal (so a+1 consists of ordinals $\le a$), C_a is the set of elements that have degree a, and lim is limit points.
In other terms : Let η be an ordinal, and let 0_S and let Ld(a,b) be the statement that a is a limit of ordinals c such that $(c, b) \in D$. Let D be the following binary relation over η :

- $\forall a < \eta : (a, 0) \in D$

- $\forall a < \eta : a \neq 0 \Rightarrow (0, a) \notin D$

- $\forall b \in Lim \cup \eta : (a, b) \in D \Leftrightarrow \forall c < b : (a, c) \in D$

- $\forall b : (a, b) \in D \Leftrightarrow Ld(a, b+1) \forall b : (a, b) \in D \Leftrightarrow Ld(a, b+1)$

- $\forall b : \exists d \in Lim \cup \eta : d <= b \Rightarrow \forall c : (c, a+1) \in D \Leftrightarrow (c <= d \vee Ld(c, b))$

Then $C(a, b) = minc : c \in \eta \wedge c > b \wedge (c, a) \in D$.
$C(a, b) = b + \omega^a$ iff $C(a, b) \geq a$.
For ordinals in the standard representation written in the postfix form, the comparison is done in the lexicographical order where $'C' <' 0' <' \Omega'$: For example, $C(C(0,0),0) < C(\Omega, 0)$ because $000CC < 0\omega C$. (This does not hold for non-standard representations of ordinals.)
The fundamental sequences of Taranovsky's notation can be easily defined.
Let $L(\alpha)$ be the amount of C's in standard representation of α, then $\alpha[n] = max\{\beta | \beta < \alpha \wedge L(\beta) \leq L(\alpha) + n\}$.
Here is a summary of the system by Taranovsky (see https://cs.nyu.edu/pipermail/fom/2012-March/016349.html) :

```
I discovered a conjectured ordinal notation system that I conjecture
reaches full second order arithmetic.  I implemented the system in a
python module/program:
http://web.mit.edu/dmytro/www/other/OrdinalArithmetic.py
along with ordinal arithmetic operations (addition, multiplication,
exponentiation, etc.) and other functions.  The ordinal arithmetic
functionality is useful even if you are only interested in ordinals
below epsilon_0.

The notation system is simple enough to be defined in full here.

Definition: An ordinal a is 0-built from below from b iff a<=b
a is n+1-built from below from b iff the standard representation of a
does not use ordinals above a except in the scope of an ordinal n-built
from below from b.

(Note: "in the scope of" means "as a subterm of".)

The nth (n is a positive integer) ordinal notation system is defined as
follows.

Syntax: Two constants (0, W_n) and a binary function C.
Comparison: For ordinals in the standard representation written in the
postfix form, the comparison is done in the lexicographical order where
'C' < '0' < 'W_n': For example, C(C(0,0),0) < C(W_n, 0) because 0 0 0 C
C < 0 W_n C.

Standard Form:
0, W_n are standard
"C(a, b)" is standard iff
1. "a" and "b" are standard,
2. b is 0 or W_n or "C(c, d)" with a<=c, and
3. a in n-built from below from b.

I conjecture that the strength of the nth ordinal notation system is
between Pi^1_{n-1}-CA and Pi^1_n-CA_0, and thus the sum of the order
types of these ordinal notation systems is the proof-theoretical ordinal
of second order arithmetic.

The full notation system is obtained by combining these notation systems
```

as follows:
Constants 0 and W_i (for every positive integer i), and a binary function C.
W_i = C(W_{i+1}, 0) and the standard form always uses W_i instead of
C(W_{i+1}, 0).
To check for standard form and compare ordinals use W_i = C(W_{i+1}, 0)
to convert each W to W_n for a single positive integer n (it does not
matter which n) and then use the nth ordinal notation system.

To make C a total function for a and b in the notation system (this is
not required for standard forms), let C(a, b) be the least ordinal (in
the notation system) of degree >=a above b, where the degree of W_i is
W_{i+1} and the degree of C(c,d) is c if "C(c,d)" is the standard form.
A polynomial time computation of C(a, b) (that I believe is correct) is
included in the program.

To complete ordinal analysis of second order arithmetic, one would need:
* A canonical assignment of notations to formulas that provably in
second order arithmetic denote an ordinal, and such that for every two
ordinals/formulas, comparison is provable in second order arithmetic.
The idea is that the notation system captures not only provably
recursive ordinals of second order arithmetic but all ordinals that have
a provable canonical definition in second order arithmetic. For
example, W_1 is best assigned to the least admissible ordinal above
omega. A partial assignment is in my paper. (It is because of such
assignment that I believe that the system reaches full second order
arithmetic.)
* Proof that the system is well-founded and that it has the right
strength, etc. (If you do not fully understand the notation system, or
if you think that it is not well-founded, let me know.)

Historical Note: In 2005, I discovered the right general form of C,
defined a notation system at the level of alpha-recursively inaccessible
ordinals (FOM postings in August 2005), and had an idea for reaching
second order arithmetic. In January 2006 (or possibly late 2005), I
defined the notation system with W_2 and in 2009 (June 29, 2009 FOM
posting) implemented it is a computer program. This year I defined the
key concept -- n-built from below -- that allowed me to complete the
full notation system.

Details about the ordinal notation system and its initial segments are
in my paper:
http://web.mit.edu/dmytro/www/other/OrdinalNotation.htm

Sincerely,
Dmytro Taranovsky

Here are some examples of representations of some ordinals :

- $0 = 0$

- $1 = 0 + \omega^0 = C(0,0)$

- $2 = 1 + \omega^0 = C(0,1) = C(0, C(0,0))$

- $\omega = 0 + \omega^1 = C(1,0)$

- $\omega + 1 = \omega + \omega^0 = C(0,\omega) = C(0, C(1,0))$

- $\omega \cdot 2 = \omega + \omega^1 = C(1, \omega) = C(1, C(1, 0))$

- $\omega^2 = 0 + \omega^2 = C(2, 0)$

- $\omega^\omega = 0 + \omega^\omega = C(\omega, 0) = C(C(1, 0), 0)$

- $\omega^{\omega^\omega} = 0 + \omega^{\omega^\omega} = C(\omega^\omega, 0) = C(C(C(1, 0), 0), 0)$

- $\varepsilon_0 = \varphi(1, 0) = \varphi'(0, 1) = C(\Omega_1, 0)$

- $\varepsilon_1 = \varphi(1, 1) = \varphi'(0, 2) = C(W, C(W, 0))$ (note that the correspondence with φ' is simpler than with φ)

- $\zeta_0 = \varphi(2, 0) = \varphi'(1, 1) = C(C(\Omega_1, \Omega_1), 0) = C(\Omega_1 \cdot 2, 0)$ with $\Omega_1 \cdot 2 = C(\Omega_1, \Omega_1)$

- $\zeta_1 = \varphi(2, 1) = \varphi'(1, 2) = C(\Omega_1 \cdot 2, C(\Omega_1 \cdot 2, 0))$

- $\eta_0 = \varphi(3, 0) = \varphi'(2, 1) = C(\Omega_1 \cdot 3, 0)$ with $\Omega_1 \cdot 3 = C(\Omega_1, C(\Omega_1, \Omega))$

- $\Gamma_0 = \varphi(1, 0, 0) = \varphi'(1, 0, 1) = C(C(\Omega_1 \cdot 2, \Omega_1), 0) = C(\Omega_1{}^2, 0)$ with $\Omega_1{}^2 = C(\Omega_1 \cdot 2, \Omega_1)$

- $\Gamma_1 = C(\Omega_1{}^2, C(\Omega_1{}^2, 0))$

- $\Gamma_\omega = C(\Omega_1{}^2 + 1, 0)$

- Small Veblen ordinal $= C(\Omega_1{}^\omega, 0)$

- Large Veblen ordinal $= C(\Omega_1{}^{\Omega_1}, 0)$

- Bachmann Howard ordinal $= C(C(\Omega_2, \Omega_1), 0)$

See http://web.mit.edu/dmytro/www/other/OrdinalNotation.htm and
https://stepstowardinfinity.wordpress.com/2015/06/22/ordinal3/ for more information.

11 Tree representations of ordinals

Ordinal can also be represented by trees. An example of such a representation is given in :
http://www.madore.org/ david/math/ordtrees.pdf .
In this representation, the order on finite rooted trees is recursively defined as follows : $A < B$ if and only if one of this conditions
is true :

- There is some mmediate subtree B' of B such that $A \leq B'$.
- Every child A' of A satisfies $A' < B$ and the list of children of A is lexicographically less than the list of children of B for
  the order $<$ with the leftmost children having the most weight.

Trees can also be represented by parenthesized expressions, for example :

- $0 = ()$
- $1 = (())$
- $2 = ((()))$
- $\omega = (()())$
- $\omega + 1 = ((()()))$
- $\omega \cdot 2 = (()(()))$
- $\omega \cdot 3 = (()((())))$
- $\omega^2 = (()(()()))$
- $\omega^2 + \omega = (()((()())))$
- $\omega^2 \cdot 2 = (()(()(())))$
- $\omega^2 \cdot 3 = (()(()((()))))$
- $\omega^3 = (()(()(()())))$
- $\omega^\omega = ((())())$
- $\omega^{\omega^\omega} = (((()))())$
- $\epsilon_0 = \varphi_1(0) = (()()())$

- $\varepsilon_1 = \varphi_1(1) = (()()(()))$
- $\varepsilon_2 = \varphi_1(2) = (()()(()))$
- $\varepsilon_\omega = \varphi_1(\omega) = (()()((())))$
- $\varepsilon_{\varepsilon_0} = (()()(()()()))$
- $\zeta_0 = \varphi_2(0) = (()(())())$
- $\Gamma_0 = \varphi(1,0,0) = (((())()())$
- $\varphi(1,0,0,0) = (()()()())$
- ...

Another example of tree representation is Takeuti ordinal diagrams, see :
https://projecteuclid.org/download/pdf_1/euclid.jmsj/1261153819 .

12 Summary

Any ordinal can be defined as the least ordinal strictly greater than all ordinals of a set : the empty set for 0, $\{\alpha\}$ for the successor of α, $\{\alpha_0, \alpha_1, \alpha_2, ...\}$ for an ordinal with fundamental sequence $\alpha_0, \alpha_1, \alpha_2, ...$

12.1 Algebraic notation

We define the following operations on ordinals :

- addition : $\alpha + 0 = \alpha; \alpha + suc(\beta) = suc(\alpha + \beta); \alpha + lim(f) = lim(n \mapsto \alpha + f(n))$
- multiplication : $\alpha \times 0 = 0; \alpha \times suc(\beta) = (\alpha \times \beta) + \alpha; \alpha \times lim(f) = lim(n \mapsto \alpha \times f(n))$
- exponentiation : $\alpha^0 = 1; \alpha^{suc(\beta)} = \alpha^\beta \times \alpha; \alpha^{lim(f)} = lim(n \mapsto \alpha^{f(n)})$

12.2 Veblen functions

These functions use fixed points enumaration : $\varphi(\ldots, \beta, 0, \ldots, 0, \gamma)$ represents the $(1+\gamma)^{th}$ common fixed point of the functions $\xi \mapsto \varphi(\ldots, \delta, \xi, 0, \ldots, 0)$ for all $\delta < \beta$.

12.3 Simmons notation

$Fix f z = f^w(z+1)$ = least fixed point of f strictly greater than z.
$Next = Fix(\alpha \mapsto \omega^\alpha)$
$[0]h = Fix(\alpha \mapsto h^\alpha \omega)$; $[1]hg = Fix(\alpha \mapsto h^\alpha g\omega)$; $[2]hgf = Fix(\alpha \mapsto h^\alpha gf\omega)$; etc...
Correspondence with Veblen's ϕ : $\phi(1+\alpha, \beta) = ([0]^\alpha Next)^{1+\beta}\omega; \phi(\alpha, \beta, \gamma) = ([0]^\beta(([1][0])^\alpha Next))^{1+\gamma}\omega$

12.4 RHS0 notation

We start from 0, if we don(t see any regularity we take the successor, if we see a regularity, if we have a notation for this regularity, we use it, else we invent it, then we jump to the limit.
$Hfx = lim\ x, fx, f(fx), \ldots; R_1 fgx = lim\ gx, fgx, ffgx, \ldots; R_2 fghx = lim\ hx, fghx, fgfghx, \ldots$
Correspondence with Simmons notation : $\ldots, [3] \to R5, [2] \to R4, [1] \to R3, [0] \to R2, Next \to R1, \omega \to Hsuc\ 0$

12.5 Ordinal collapsing functions

These functions use uncountable ordinals to define countable ordinals.
We define sets of ordinals that can be built from given ordinals and operations, then we take the least ordinal which is not in this set, or the least ordinal which is greater than all contable ordinals of this set.
These functions are extensions of functions on countable ordinals, whose fixed points can be reached by applying them to an uncountable ordinal.
Examples :

- Madore's ψ : $\psi(\alpha) = \varepsilon_\alpha$ if $\alpha < \zeta_0; \psi(\Omega) = \zeta_0$ which is the least fixed point of $\alpha \mapsto \varepsilon_\alpha$.
- Feferman's θ : $\theta(\alpha, \beta) = \varphi(\alpha, \beta)$ if $\alpha < \Gamma_0$ and $\beta < \Gamma_0; \theta(\Omega, 0) = \Gamma_0$ which is the least fixed point of $\alpha \mapsto \varphi(\alpha, 0)$.
- Taranovsky's C : $C(\alpha, \beta) = \beta + \omega^\alpha$ if α is countable; $C(\Omega_1, 0) = \varepsilon_0$ which is the least fixed point of $\alpha \mapsto \omega^\alpha$.

13   Comparison table

Name	Symbol	Algebraic	Veblen	Simmons	RHS0	Madore	Taranovsky
Zero	0	0			0		0
One	1	1	$\varphi(0,0)$		suc 0		C(0,0)
Two	2	2			suc (suc 0)		C(0,C(0,0))
Omega	ω	ω	$\varphi(0,1)$	ω	H suc 0		C(1,0)
		$\omega+1$			suc (H suc 0)		C(0,C(1,0))
		$\omega\times 2$			H suc (H suc 0)		C(1,C(1,0))
		ω^2	$\varphi(0,2)$		H (H suc) 0		C(C(0,C(0,0)),0)
		ω^ω	$\varphi(0,\omega)$		H H suc 0		C(C(1,0),0)
		ω^{ω^ω}	$\varphi(0,\omega^\omega)$		H H H suc 0		C(C(C(1,0),0),0)
Epsilon zero	ε_0	ε_0	$\varphi(1,0)$	$Next\ \omega$	$R_1 H suc\ 0$	$\psi(0)$	$C(\Omega_1,0)$
		ε_1	$\varphi(1,1)$	$Next^2\omega$	$R_1(R_1 H)suc\ 0$	$\psi(1)$	$C(\Omega_1,C(\Omega_1,0)$
		ε_ω	$\varphi(1,\omega)$	$Next^\omega\omega$	$HR_1 H suc\ 0$	$\psi(\omega)$	$C(C(0,\Omega_1),0)$
		$\varepsilon_{\varepsilon_0}$	$\varphi(1,\varphi(1,0))$	$Next^{Next\omega}\omega$	$R_1 H R_1 H suc\ 0$	$\psi(\psi(0))$	$C(C(C(\Omega_1,0),\Omega_1),0)$
Zeta zero	ζ_0	ζ_0	$\varphi(2,0)$	$[0]Next\ \omega$	$R_2 R_1 H suc\ 0$	$\psi(\Omega)$	$C(C(\Omega_1,\Omega_1),0)$
Eta zero	η_0	η_0	$\varphi(3,0)$	$[0]^2 Next\ \omega$	$R_2(R_2 R_1)H suc\ 0$		$C(C(\Omega,C(\Omega,\Omega)),0)$
			$\varphi(\omega,0)$	$[0]^\omega Next\ \omega$	$HR_2 R_1 H suc\ 0$		$C(C(C(0,\Omega_1),\Omega_1),0)$
Feferman -Schütte	Γ_0	Γ_0	$\varphi(1,0,0)$ $=\varphi(2\mapsto 1)$	$[1][0]Next\ \omega$	$R_3 R_2 R_1 H suc\ 0$ $=R_{3\dots 1}H suc\ 0$	$\psi(\Omega^\Omega)$	$C(C(C(\Omega_1,\Omega_1),$ $\Omega_1),0)$
Ackermann			$\varphi(1,0,0,0)$ $=\varphi(3\mapsto 1)$	$[1]^2[0]Next\ \omega$	$R_3(R_3 R_2)R_1 H suc\ 0$	$\psi(\Omega^{\Omega^2})$	
Small Veblen ordinal			$\varphi(\omega\mapsto 1)$	$[1]^\omega[0]Next\ \omega$	$HR_3 R_2 R_1 H suc\ 0$	$\psi(\Omega^{\Omega^\omega})$	$C(\Omega_1^\omega,0)$ $=C(C(C(C(0,\Omega_1),$ $\Omega_1),\Omega_1),0)$
Large Veblen ordinal			least ord. not rep.	$[2][1][0]Next\ \omega$	$R_4 R_3 R_2 R_1 H suc\ 0$ $=R_{4\dots 1}H suc\ 0$	$\psi(\Omega^{\Omega^\Omega})$	$C(\Omega_1^{\Omega_1},0)$ $=C(C(C(C(\Omega_1,\Omega_1),$ $\Omega_1),\Omega_1),0)$
Bachmann- Howard ordinal				least ord. not rep.	$R_{\omega\dots 1}H suc\ 0$	$\psi(\varepsilon_{\Omega+1})$	$C(C(\Omega_2,\Omega_1),0)$

14   Links

- http://www.madore.org/%7Edavid/weblog/2011-09-18-nombres-ordinaux-intro.html : Tutorial introduction to ordinal numbers in French
- https://sites.google.com/site/pointlesslargenumberstuff/home/l/pglin?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F : Pointless Gigantic List of Infinite Numbers
- https://sites.google.com/site/largenumbers/home/appendix/a/infinite_numbers : Sbiis Saibian's !!! FORBIDDEN LIST !!! of Infinite Numbers
- http://quibb.blogspot.fr/p/infinity-series-portal.html : Professor Quibb's Infinity Series Portal
- http://googology.wikia.com/wiki/Ordinal_notation : Ordinal notation
- https://sites.google.com/site/travelingtotheinfinity/ : Traveling to the infinity
- http://www.cs.man.ac.uk/ hsimmons/TEMP/OrdNotes.pdf : A short introduction to Ordinal Notations by Harold Simmons
- http://www.mathematik.uni-muenchen.de/ buchholz/articles/jaegerfestschr_buchholz3.pdf : A survey on ordinal notations around the Bachmann-Howard ordinal by Wilfried Buchholz
- http://web.mit.edu/dmytro/www/other/OrdinalNotation.htm : Ordinal Notation by Dmytro Taranovsky
- http://arxiv.org/html/1203.2270 : Higher Order Set Theory with Reflective Cardinals by Dmytro Taranovsky
- https://www1.maths.leeds.ac.uk/ rathjen/realm.pdf : The Realm of Ordinal Analysis by Michael Rathjen
- https://cage.ugent.be/ jvdm/Site/Research_files/DissertationJeroenVanderMeerenPrinted.pdf : Connecting the Two Worlds: Well-partial-orders and Ordinal Notation Systems¡/a¿ by Jeroen Van der Meeren¡br¿
- https://en.wikipedia.org/wiki/Veblen_function : Veblen function on Wikipedia

- http://www.ams.org/journals/tran/1908-009-03/S0002-9947-1908-1500814-9/S0002-9947-1908-1500814-9.pdf : Continuous increasing functions of finite and transfinite ordinals by Oswald Veblen
- http://en.wikipedia.org/wiki/Ordinal_collapsing_function : Ordinal collapsing function on Wikipedia
- https://en.wikipedia.org/wiki/Buchholz_psi_functions : Buchholz psi functions on Wikipedia
- http://www.madore.org/%7Edavid/math/ordtrees.pdf : Ordinal trees
- https://johncarlosbaez.wordpress.com/2016/06/29/large-countable-ordinals-part-1/ : Large Countable Ordinals by John Baez, Part 1
- https://johncarlosbaez.wordpress.com/2016/07/04/large-countable-ordinals-part-2/ : Large Countable Ordinals by John Baez, Part 2
- https://johncarlosbaez.wordpress.com/2016/07/07/large-countable-ordinals-part-3/ : Large Countable Ordinals by John Baez, Part 3
- https://medium.com/@joshkerr/mind-blown-the-fast-growing-hierarchy-for-laymen-aka-enormous-numbers-d9a865c6443b : Mind blown: the fast growing hierarchy for laymen — aka enormous numbers
- https://sites.google.com/site/largenumbers/home : Sbiis Saibian's Large Number Site
- https://www.youtube.com/playlist?list=PLUZ0A4xAf7nkaYHtnqVDbHnrXzVAOxYYC : Extremely large numbers (videos)
- https://www.youtube.com/playlist?list=PL3A50BB9C34AB36B3 : Ridiculously huge numbers (videos)
- http://forums.xkcd.com/viewtopic.php?f=14&t=7469 : My number is bigger !
- http://www.cl.cam.ac.uk/%7Ejrh13/papers/reflect.html : Metatheory and Reflection in Theorem Proving: A Survey and Critique by John Harrison
- http://math.stanford.edu/%7Efeferman/papers/penrose.pdf : Penrose's Gödelian argument by Solomon Feferman
- http://www.turingarchive.org/browse.php/B/15 : Systems of logic based on ordinals by Alan Turing
- https://coq.inria.fr/documentation : Coq documentation
- http://wiki.portal.chalmers.se/agda/pmwiki.php?n=Main.Documentation : Agda documentation