Web PDF version -2004-09-15

ACKERMANN'S FUNCTION AND NEW ARITHMETICAL OPERATIONS

CONSTANTIN A. RUBTSOV – GIOVANNI F. ROMERIO

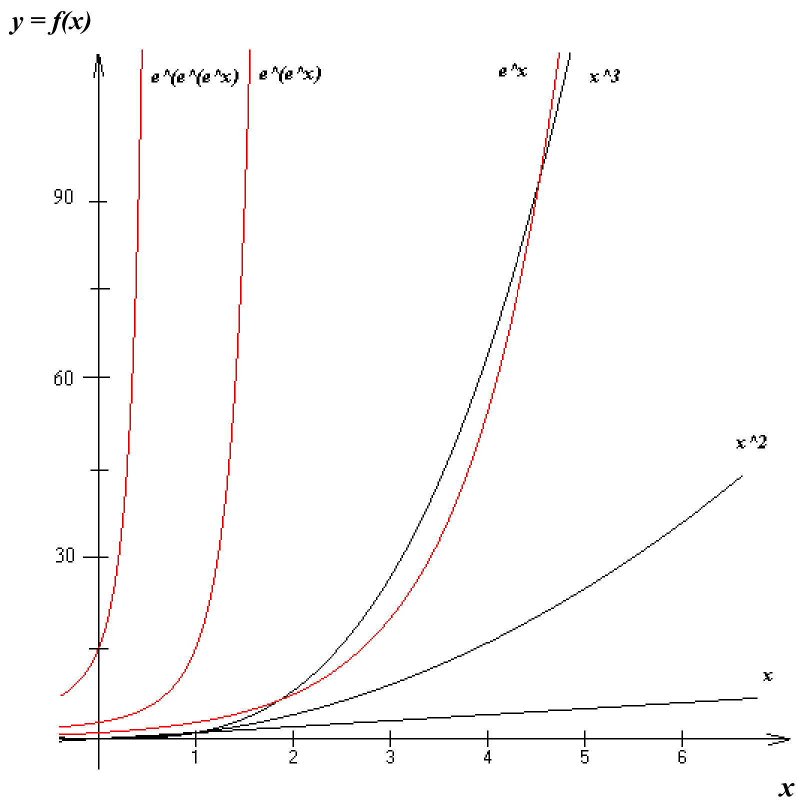
Abstract. Ackermann's function implies the existence of an infinite spectrum of new arithmetical operations (*hyperoperations*) belonging to the Grzegorczyk Hierarchy. Two of these hyperoperations, tetration and zeration, together with their inverses are given special attention. Tetration (power tower, or superpower) has recently been described in the scientific literature and has attracted the attention of both career and amateur mathematicians; zeration is less well known. However, the new arithmetical operation, called zeration, follows naturally from generalizing formulas used in iterative calculations of both known and new inverse operations. Zeration expands our approach to mathematical concepts such as infinity and continuity and its application for describing discontinuities is shown in practical examples. The possibility of using tetration to represent very large numbers is outlined. Questions requiring further research are also raised.

Mathematics Subject Classification (2000): 20k30 homomorphisms; 03D99 computability and recursion theory; 11A25 arithmetic functions; 11H99 computational number theory; 68Q15 complexity classes.

1. INTRODUCTION

1.1. *Algorithmic Complexity Classes*. In the framework of artificial intelligence (AI) and problem solving sciences, algorithm efficiency is analyzed by measuring how much the duration of a process, generated by an algorithm, grows as a function of the amount of data on which the algorithm itself operates. Let the amount of data (measured, for instance, in bits) be called x and the "length" of the process (i.e. the algorithm execution time, or the amount of process steps, or the total number of bits describing the process) be called y. Then, various levels of *algorithm complexity* can be assessed according to the types of expressions $y = f(x)$, as seen in *problems' complexity theory.* Normally, if an algorithm operates on a larger number of data, the corresponding process requires a larger number of steps to be executed, i.e. the process length y also increases. In practice, we have situations such as those shown in the following plots of the $y = f(x)$ functions.

Constantin A. Rubtsov. pr. Valutina, d. 12 - Belgorod – Russian Federation
Giovanni F. Romerio. via Torino, 48 - 12037 Saluzzo (CN) - Italy



In particular, the diagram shows variations of linear type (***y*** = ***x***), as well as of the quadratic (***y*** = ***x***^{2}) or cubic (***y*** = ***x***^{3}) types. The diagram also shows a series of exponential (or iterated exponential) variations, growing much faster than the "power" plots. In principle, we can have the following major classes of variations:

- *Class P* – The *process/data* plots are of type ***y = k x***, ***y = k x***^{2}, ***y = k x***^{3}, …. , or of generalized polynomial type ***y = a x***^{n} + ***bx***$^{n-1}$+ ***cx***$^{n-2}$***…*** + ***k***. Algorithms of this type are said to belong to the class of *polynomial* type, corresponding to relatively easily solvable problems.
- *Class EXP* – The *process/data* plots are not of the polynomial type but they can be described by exponential functions, such as ***y = exp(x)***. In practice, process length ***y*** can be represented by using the powers of ***2*** of the algorithm length ***x***, such as ***y = 2***^{x}, or by using the powers of ***e***. This class represents *exponential* or *intrinsically complex* algorithms, corresponding to the class of "*intrinsically difficult problems*", which normally strongly resist computer solutions.
- *Class H-EXP* - AI specialists have found algorithms identified by the ***y = 2^(2^x)*** plot (or bi-exponential), or even by tri-exponential or tetra-exponential or, generally, ***k***-exponential expressions. Some plots of this kind are indicated in the diagram. We may observe that these plots increase much more rapidly than any plot of polynomial or of simple exponential type. A detailed classification of *H-EXP* (hyper-exponential) algorithms could help in defining the ranks of intrinsical *problem complexity*.
- *Class 0* - A class of algorithms in which the process length (corresponding to the execution time) is always infinite. In this case the computer will never stop and, therefore, it will never deliver the solution. This is the class of "*unsolvable problems*", i.e. of problems for which the solution, in the framework of the existing and admitted hypotheses, is considered to be "non-existent".

1.2. *Iterated exponentiations.* It is interesting to note that, in order to define the subclasses of the *H-EXP* class, some expressions take the form:

(1) ***y = e ^(e ^(e ^(e ^ …..(e ^ x))))*** [***e*** ^ iterated ***k*** times, on ***x***]

and these are outside the range of normally accepted elementary arithmetical operations, as defined in classical Algebra. These kinds of expressions (i.e. iterated exponentiations) have been identified, in the literature, as "*power towers*", "*superpowers*" or, simply, "*towers*". L. Stockmeyer and A. Chandra, of the Thomas Watson IBM Research Centre, have analyzed expressions of this type.

They also reported, in a paper published in 1989 [1], that Albert Meyer [2] proved, in 1972, that the ***S1S*** algorithm has a complexity greater than exponential. In particular, he has shown that the algorithm deciding about the truth of a statement with length $\boldsymbol{x}$ of the ***S1S*** language has a complexity that increases more than the $\boldsymbol{k}$-exponential function of $\boldsymbol{x}$ (for any $\boldsymbol{k}$).

1.3. *Algorithmic Information*. As clearly stated by Stephen Wolfram [3], it is generally accepted that the description of a piece of data can always be done with a program for reproducing it. Andrei Kolmogorov analyzed the problem of defining the information carried by a sequence of bits, representing a process, with a different approach from that of Shannon's information theory, where the information contents of a signal is given by the logarithm, to the base 2, of the reciprocal of the configuration probability. Shannon's "*information quantity*", thus defined, supposes the existence of a communication channel, linking the *Source* of information with the *Receiver* and takes into account the probability that the *Receiver* can detect the signals sent by the *Source.* Kolmogorov's objective was, however, to try and define another "*information quantity*" that could possibly measure the absolute internal "order" of a process, independently of any transmission channel. For this, he had the idea of connecting a process with the shortest possible algorithm which could reproduce it. If no such algorithm shorter than the process exists, then we must resort to a "random" process. The shortest possible algorithm, associated with a process, is called its *Algorithmic Information*, which, in a way, highlights the internal redundancy of the process itself. Charles Bennet called it the process' *Logical Depth*. It is interesting to note that the definition of the Algorithmic Information of a process coincides with the definition of the Algorithmic Complexity of the program that has generated it (see **1.1.**).

1.4. *Ackermann's Function (AF)*. Ackermann's Function [4], a recursive function that, while Turing computable, grows faster than any primitive recursive functions, can be used to establish a link between classical elementary arithmetical operations (addition, multiplication, exponentiation) and to analyze operations such as:

(2) $\quad \boldsymbol{y = a\ \hat{}(a\ \hat{}(a\ \hat{}(a\ \hat{}\ .....(a\ \hat{}\ a))))}$ $\quad$ [$\boldsymbol{a}$ ^ iterated $\boldsymbol{k}$ times, on $\boldsymbol{a}$]

These are rather similar to expressions (1), except for the fact that iterated exponentiation, in this case, always operates on the same magnitude $\boldsymbol{a}$ which is the base of the exponential. AF itself demonstrates the existence of an infinite series of *hyperoperations*, including the classical arithmetical operations, as well as *tower power* and other iterated exponentiations, each one indexed by a positive integer value of a parameter $\boldsymbol{s}$, called the *hyperoperations rank.* This series is known as the Grzegorczyk Hierarchy [3].

1.5. *Ω-Mapping*. The first author (C. A. Rubtsov), while analyzing AF, had the idea of defining the properties of a new hyperoperation of rank zero ($\boldsymbol{s = 0}$), i.e. with a rank lower than the rank of addition, because it appeared to be automatically implied by some AF values. This new operation was named the *zero-rank operation"* or *"zeration"*. The properties of *zeration* have been presented in paper [5], concerning "New Mathematical Objects", following some ideas presented since 1989 [6]. Paper [5] analyses the properties of zeration and of the new set of numbers that it generates. This, together with the outline of the basis concepts of the homomorphisms' theory, expands our representation of the spectrum of existing mathematical objects (http://numbers.newmail.ru). Ω-mapping formalism can be used to discover the properties of hyperoperations such as *zeration* and *tetration*, together with those of their inverse operations.

1.6. *Extension of Tower to the Reals*. The second author (G. F. Romerio) has, since 1986, been attracted to the problem of analyzing the properties of the *tower* operation, defined within the set **N** of natural numbers, in order to extend it to the set **R** of real numbers. This problem is similar to that of extending the classical factorial operation:

$$\boldsymbol{n\,!} \qquad \text{with } \boldsymbol{n} \in \mathbf{N}$$

to any number $\boldsymbol{x}$, belonging to the set of real numbers. This question was solved by the definition of the *Gamma Function*, such as:

$$\boldsymbol{x\,! = \Gamma(x+1)} \qquad \text{with } \boldsymbol{x} \in \mathbf{R}.$$

Furthermore, the solution to our problem of extending the *tower* operation (also called *tetration*) to the reals is also similar to the classical question of extending the validity of expression $\boldsymbol{a^n}$ ($\boldsymbol{n} \in \mathbf{N}$) to $\boldsymbol{a^x}$, where we have $x \in \mathbf{R}$. This problem has also been solved a long time ago, starting by identifying an expression with rational exponent, such as $a^{½}$ (with $a^0 < a^{1/2} < a^1$; i.e.: $1 < a^{1/2} < a$), with the square root of a ($\sqrt{a}$).

Unfortunately, the homologue operation, at the *tower/tetration* operation rank, was never given a meaning until now. Therefore, there is still a need, for instance, to clarify the exact meaning of "*a-tetrated-½* " which, incidentally, must not be confused with what we might call the "*super-squareroot of **a*** " (see paragraphs 6.3 and 6.4).

1.7. *Objectives*. The immediate objective of this study is to draw the attention of mathematicians to the possibility of defining new elementary functions based on the properties of hyperoperations and their inverses, as well as to suggest possible practical applications and strategies for future research.

2. Iterative Extensions of Traditional Algebraic Operations

2.1. *Iterations of classical operations*. We wish to examine the possibility of introducing an extensible series of algebraic operations, by using the constructive iterative application of appropriate operators, each one corresponding to one of the traditional arithmetical operations (addition and multiplication), together with exponentiation. For this purpose, let us define the following operators:

$\boxed{a+}$, such that: $\boxed{a+}\, x = a + x$ *Operator a+* (Addition)

$\boxed{a\times}$, such that: $\boxed{a\times}\, x = a \times x = a \,.\, x$ *Operator a×* (Multiplication)

$\boxed{a\,\wedge}$, such that: $\boxed{a\,\wedge}\, x = a \wedge x = a^x$ *Operator a^* (Exponentiation)

These three operators act on the operand situated to the right and can be iterated, as follows:

$$\boxed{a+}\,\boxed{a+}\,\boxed{a+}\,x = \boxed{a+}\,\boxed{a+}\,(a + x) = \boxed{a+}\,(a + (a + x)) = a + (a + (a + x)) = a{+}a{+}a{+}x$$

$$\boxed{a\times}\,\boxed{a\times}\,\boxed{a\times}\,x = \boxed{a\times}\,\boxed{a\times}\,(a \,.\, x) = \boxed{a\times}\,(a \,.\, (a \,.\, x)) = a \,.\, (a \,.\, (a \,.\, x)) = a.a.a.x$$

$$\boxed{a\,\wedge}\,\boxed{a\,\wedge}\,\boxed{a\,\wedge}\,x = \boxed{a\,\wedge}\,\boxed{a\,\wedge}\,(a \wedge x) = \boxed{a\,\wedge}\,(a \wedge (a \wedge x)) = \mathbf{a \wedge (a \wedge (a \wedge x))} = a^{a^{a^x}}$$

As we know, in the first two cases (addition and multiplication), the parentheses can be removed in the final result, because of the associativity of the two operations. In the case of exponentiation, this cannot be done and the order defined by the parentheses remains unaltered.

Suppose we now apply the $\boxed{a+}$ operator, by repeated iterations, to the same numerical variable a. The result of each iteration of addition, which is an operation identified as the first rank in the hierarchy (s = 1), will be represented by multiplication, which is an operation of the second rank (s = 2):

<u>Rank s = 1</u>, *iterated additions*, represented by *multiplications* (s = 2):

	$\boldsymbol{a + a} =$	$\boldsymbol{a \,.\, 2}$
(3)	$\boldsymbol{a + a + a} =$	$\boldsymbol{a \,.\, 3}$
	....	....
	$\boldsymbol{a + a + a + a} + ... + \boldsymbol{a}$ *(n times)* =	***a . n*** (***a****-times-****n***)

We now go on to describe the second iterated operation, in which we use operator $\boxed{a\times}$. This consists of the iteration of multiplication (s = 2), represented by exponentiation (s = 3):

<u>Rank s = 2</u>, *iterated multiplications*, represented by *exponentiations* (s = 3):

	$\boldsymbol{a \,.\, a} =$	$\boldsymbol{a \wedge 2}$
(4)	$\boldsymbol{a \,.\, a \,.\, a} =$	$\boldsymbol{a \wedge 3}$
	....	....
	a . a . a . a . ..... ***a*** *(n times)* =	***a ^ n*** (***a****-power-****n***)

The iterations referred to above cover and represent the three classical algebraic operations (addition, multiplication and exponentiation).

2.2. *Tetration.* However, the iteration process can be continued, by defining an additional hierarchical level (s = 4), which represents iterated exponentiation. In this case we get:

<u>Rank s = 3</u>, *iterated exponentiations*, represented by a <u>new operation</u> called ***tetration*** (s = 4) or *power tower*, *or tower*, for which we choose operator symbol "#":

(5) $\boldsymbol{a \wedge a} =$ $\boldsymbol{a \# 2}$

$\boldsymbol{a \wedge a} \wedge a =$ $\boldsymbol{a \# 3}$

....

$\boldsymbol{a \wedge a \wedge a \wedge a \wedge \ldots\wedge a}$ *(n times)* = $\boldsymbol{a \# n}$ (***a***-<u>*tower*</u>-***n*** *or* ***a***-*tetrated*-***n***)

In both *exponentiation* and *tetration*, the associative property is not fulfilled. These operations are, therefore, intended to be executed with "priority to the right".

"*Tetration*" derives its name from the fact that it appears at the 4th rank in the hyperoperations hierarchy , in which "*addition*", "*multiplication*" and "*exponentiation*" are identified by ranks 1, 2 and 3, respectively. The first super-exponential operation (*a*-tetrated-*n* or *a*-tower-*n*) can be represented in various ways, such as: $a \# n = {}^{n}a$ or also as $a\uparrow\uparrow n$. The latter is derived from the representation of standard exponentiation by one arrow i.e.: $a \wedge n = a^n = a\uparrow n$, as proposed by Knut (1976), and it has the advantage of being applicable to all the hyperoperations with ranks > 3. The "arrows" symbolism for s < 3 fails. Notation ${}^{n}a$ was proposed by Rucker in 1995. We shall see that the iteration process can continue, for ranks s > 4, and that we can define a series of operations (hyperoperations) with no upper limit to the value of the rank.

2.3. *Zeration*. The problem of the existence of operations with a hierarchical rank s < 1, and particularly for s = 0 (the zero-rank operation, or ***zeration***) was studied by one of us [5], in 1986. Leaving the justification for it until a later stage, we shall at this point simply present here the scheme of this new operation, the iteration of which can be represented by the *addition* operation. The process requires the definition of a new operator, with rank "zero":

$\boxed{a^\circ}$, such that: $\boxed{a^\circ}\, x = a \circ x$ *Operator* a° (*zeration*)

The scheme so obtained is as follows (with priority to the right):

(6) $\boldsymbol{a \circ a} =$ $\boldsymbol{a + 2}$

$\boldsymbol{a \circ a \circ a} =$ $\boldsymbol{a + 3}$

....

$\boldsymbol{a \circ a \circ a \circ a \circ \ldots \circ a}$ *(n times)* = $\boldsymbol{a + n}$ (***a***-*plus*-***n***)

This operator $\boxed{a^\circ}$ defines a <u>new algebraic operation</u>, with hierarchical rank s = 0, for which the associative property is also not fulfilled. An appropriate name for this operation is *zeration*.

2.4. *The Grzegorczyk Hierarchy*. The family of operations mentioned in the previous paragraphs of this section (zeration, addition, multiplication, exponentiation and tetration) belong to an infinite series of hyperoperations, called the "*Grzegorczyk Hierarchy*", named after prof. Grzegorczyk, a distinguished Polish philosopher of Logic (See also [7]).

The hierarchy can be globally described by introducing a generalized hyperoperator of rank s $\boxed{a\boxed{\text{s}}}$ that will operate on its right in the following scheme:

$\boxed{a\boxed{\text{s}}}$, such as: $\boxed{a\boxed{\text{s}}}\, x = a \boxed{\text{s}}\, x$ hyper-operator of rank s: $\boxed{\text{s}}$

Here, $\boxed{\text{s}}$ is an infixed form of the hyper-operator of rank "s", that generates the following scheme:

for s=0 $\boldsymbol{a \boxed{\text{s}}\, b = a \boxed{0}\, b = a \circ b}$ <u>**zeration**</u>

for s=1 $\boldsymbol{a \boxed{\text{s}}\, b = a \boxed{1}\, b = a + b}$ **addition** (sum)

(7) for s=2 $\boldsymbol{a \boxed{\text{s}}\, b = a \boxed{2}\, b = a\,.\,b}$ **multiplication** (product)

for s=3 $\boldsymbol{a \boxed{\text{s}}\, b = a \boxed{3}\, b = a \wedge b}$ **exponentiation** (power)

for s=4 $\boldsymbol{a \boxed{\text{s}}\, b = a \boxed{4}\, b = a \# b}$ <u>**tetration**</u> (tower, super-power)

We shall see that rank "s" can assume the values of all natural numbers, thus completing an infinite series of hyperoperations. Indeed hyperoperation " $\boldsymbol{a}\boxed{\text{s}}\, \boldsymbol{b}$ " could probably also be defined for negative or a "non-integer" values of rank "s". So, we can foresee the mathematical "existence" of operations, not only with a negative "s", but also with an exotic rank "s = 1,5", between addition (s = 1) and multiplication (s = 2), or with ranks "s = π", or "s = e", or even "s = - i " (!!). But this may well require a lot of additional research work.

3. Iteration and Recursion

3.1. *Iterations*. In the mathematical procedures applied to computer science and artificial intelligence (A.I.) there are formalisms using concepts such as iteration and recursion [11]. We intend to take this into account in our analysis of the problems described in the previous section. We define the ***m***-fold application of function $y = f(x)$, and we denote it as $y^{[m]} = f^{[m]}(x)$, the iterative application, m times, of function $f(x)$ to the variable x (or $m - 1$ times to function $f(x)$).

For instance, if:	$y = f(x)$
we have:	$y^{[2]} = f^{[2]}(x) = f[f(x)]$
and:	$y^{[m]} = f^{[m]}(x) = f[f(\ldots.f(x))]$ applied m times to x.

It is very interesting to see, then, what happens if, by definition, we choose for $f(n)$ a series of functions $\Phi_s(n)$, such as:

$$\Phi_1(n) = 2 + n$$
$$\Phi_2(n) = 2\ .\ n$$
$$\Phi_3(n) = 2\ \wedge\ n$$
$$\Phi_4(n) = 2\ \#\ n \quad \text{(iterated exponentiation, or tetration).}$$

3.2. *An example of nested addition*. Concerning $\Phi_1(n)$, we can easily see that, if:

	$\Phi_1^{[1]}(n)$	$= 2 + n$	$= 2\ .\ 1 + n$
we have:	$\Phi_1^{[2]}(n)$	$= 2 + (2 + n)$	$= 2\ .\ 2 + n$
and:	$\Phi_1^{[3]}(n)$	$= 2 + (2 + (2 + n))$	$= 2\ .\ 3 + n$
and, in general:	$\Phi_1^{[k]}(n)$	$= \ldots\ldots$	$= 2\ .\ k + n$
or:	$\Phi_1^{[m-1]}(n)$	$= 2\ (m - 1) + n$	
and, finally, for $n = 2$:	$\Phi_1^{[m-1]}(2)$	$= 2\ (m - 1)\ + 2$	$= 2\ .\ m$

Therefore:

(8) $\boxed{\Phi_1^{[m-1]}(2)\ =\ \Phi_2(m) = 2\ .\ m}$

We can therefore see that the nested iteration of function $\Phi_1(n) = 2 + n$ on itself, m-1 times, produces (for n=2) a new function $\Phi_2(m) = 2\ .\ m$.

3.3. *An example of nested multiplication*. In a similar way, we can proceed with function $\Phi_2(n) = 2\ .\ n$.

In fact, if:	$\Phi_2^{[1]}(n)$	$= 2\ .\ n$	$= 2^1\ .\ n$
we have:	$\Phi_2^{[2]}(n)$	$= 2\ .\ (2\ .\ n)$	$= 2^2\ .\ n$
and:	$\Phi_2^{[3]}(n)$	$= 2\ .\ (2\ .\ (2\ .\ n))$	$= 2^3\ .\ n$
and, in general:	$\Phi_2^{[k]}(n)$	$= \ldots\ldots$	$= 2^k\ .\ n$
or:	$\Phi_2^{[m-1]}(n)$	$= 2^{m-1}.\ n$	
and, finally, for $n = 2$:	$\Phi_2^{[m-1]}(2)$	$= 2^{m-1}.\ 2$	$= 2^m$

In conclusion:

(9) $\boxed{\Phi_2^{[m-1]}(2)\ =\ \Phi_3(m) = 2^m = 2\ \wedge\ m}$

Again, the iterative application of function $\Phi_2(n) = 2\ .\ n$ on itself, m-1 times, produces (for n=2) a new function $\Phi_3(n) = 2\ \wedge\ n$.

3.4 *Recursive extensions*. We can easily verify that it must also be:

(10) $\boxed{\Phi_3^{[m-1]}(2)\ =\ \Phi_4(m) = {}^m2 = 2\ \#\ m}$

And, as foreseen, the process can continue, as we have also previously foreseen. It is possible to put together these various results by writing, in general:

(11) $\boxed{\Phi_s(n) = \Phi_{s-1}^{[n-1]}(2)}$

This expression can be considered both as the result of the iterative application of an operator and as a recursive function, since it is a function that refers to (and operates on) itself.

A celebrated example of a two variable recursive function is *Ackermann's Function* (William Ackermann, 1886-1992, see [4]), which has great importance in theoretical computer sciences and is very relevant to the subjects we are examining in this paper.

4. Ackermann's Function

4.1. *Ackermann's function*. Ackermann's Function $A(s,n)$ is described by the following recursive table, the elements of which (written in matrix "boxes") can be built up as follows:

- the first box of the table, with co-ordinates $s=0$ and $n=0$, contains value $A(0,0) = 1$;
- all the boxes in row $s=0$ contain numbers obtained by stating: $A(0,n) = n + 1$, in other words, they contain values: $A(0,0) = 1$, $A(0,1) = 2$, $A(0,2) = 3$, etc.;
- in row $s=1$, and for $n \geq 1$, each box $A(1,n)$ of the row contains a number found in column n of the top row ($s=0$), where n is equal to the contents of box $A(1,n-1)$, i.e. at the immediate left of $A(1,n)$; e.g. the contents of box $A(1,2)$ is equal to the content of box $A(0,3) = 4$, which is equivalent to saying that: $A(1,n) = A(0,n+1)$;
- for all the boxes in column $n=0$, we have $A(s,0) = A(s-1,1)$, as highlighted in the table, for example, concerning elements $A(1,0) = A(0,1) = 2;$
- in all the other boxes, as indicated for box $A(2,2)$, we have: $A(s,n) = A(s-1,(A(s,n-1))$.

A(s,n)	n=0	n=1	n=2	n=3	n=4	n=5	n=6
s=0	A(0,0) 1=2°3 - 3	A(0,1) 2=2°4 - 3	A(0,2) 3=2°5 - 3	A(0,3) 4=2°6 - 3	A(0,4) 5=2°7 - 3	A(0,5) 6=2°8 - 3	A(0,6) 7=2°9 - 3
s=1	A(1,0)=A(0,1) 2=2+3 - 3	A(1,1)=A(0,2) 3=2+4 - 3	A(1,2)=A(0,3) 4=2+5 - 3	A(1,3)=A(0,4) 5=2+6 - 3	A(1,4)=A(0,5) 6=2+7 - 3	A(1,5)=A(0,6) 7=2+8 - 3	A(1,6)=A(0,7) 8=2+9 - 3
s=2	A(2,0)=A(1,1) 3=2.3 - 3	A(2,1)=A(1,3) 5=2.4 - 3	A(2,2)=A(1,5) 7=2.5 - 3	A(2,3)=A(1,7) 9=2.6 - 3	A(2,4)=A(1,9) 11=2.7 - 3	A(2,5)=A(1,11) 13=2.8 - 3	A(2,6)=A(1,13) 15=2.9 - 3
s=3	A(3,0)=A(2,1) 5=2^3 - 3	A(3,1)=A(2,5) 13=2^4 - 3	A(3,2)=A(2,13) 29=2^5 - 3	A(3,3)=A(2,29) 61=2^6 - 3	A(3,4)=A(2,61) 125=2^7 - 3	A(3,5)=A(2,125) 253=2^8 - 3	A(3,6)=A(2,253) 509=2^9 - 3
s=4	A(4,0)=A(3,1) 13=2#3 - 3	A(4,1)=A(3,A(4,0)) 65.533=2#4 - 3	A(4,2)=A(3,A(4,1)) 2#5 - 3	A(4,3)=A(3,A(4,2)) 2#6 - 3	A(4,4)=A(3,A(4,3)) 2#7 - 3	A(4,5)=A(3,A(4,4)) 2#8 - 3	A(4,6)=A(3,A(4,5)) 2#9 - 3
s=5	A(5,0)=A(4,1) 65.533=2$3 - 3	A(5,1)=A(4,A(5,0)) 2$4 - 3	A(5,2)=A(4,A(5,1)) 2$5 - 3	A(5,3)=A(4,A(5,2)) 2$6 - 3	A(5,4)=A(4,A(5,3)) 2$7 - 3	A(5,5)=A(4,A(5,4)) 2$8 - 3	A(5,6)=A(4,A(5,5)) 2$9 - 3

The definition of Ackermann's Function can be summarised as follows:

(12)
$$\begin{aligned} A(0,n) &= n + 1 \\ A(s,0) &= A(s-1,1) \\ A(s,n) &= A(s-1,A(s,n-1)) \end{aligned}$$

4.2. *Hyperoperations progression.* When we analyze the table, we find extremely interesting structures in the elements contained in the various boxes. For instance, it is surprising to find the following pattern:

(13)
- o in row $s=0$, by definition: $A(0,n) = n + 1$
- o in row $s=1$: $A(1,n) = 2 + (n+3) - 3 = n + 2$;
- o in row $s=2$: $A(2,n) = 2 \,.\, (n+3) - 3 = 2n + 3$;
- o in row $s=3$: $A(3,n) = 2 \wedge (n+3) - 3 = 2^{n+3} - 3$;
- o in row $s=4$: $A(4,n) = 2 \# (n+3) - 3 = {}^{n+3}2 - 3$;
- o in row $s=5$: $A(5,n) = 2 \$ (n+3) - 3$, etc..

With the provisional exception of row s=0, we could re-define Ackermann's Function as follows:

(14) $$\boxed{A(s,n) = 2\boxed{s}(n+3) - 3}$$

In other words, by a change of variable, we have:

$$A(s,n-3) = 2\boxed{s}\ n - 3$$

or: $$2\ \boxed{s}\ n = A(s,n-3) + 3$$

and, remembering (10), we can also write:

(15) $$\boxed{\Phi_s(n) = 2\boxed{s}\ n = A(s,n-3) + 3 = \Phi_{s-1}^{[n-1]}(2)}$$

This formula, verified for $s>0$, puts each hyperoperation $2 \boxed{s}\ n$ into a binary relation with element $A(s,n-3)$. of Ackermann's Function.

4.3. *Zeration.* The case of the values in row $s=0$ deserves separate consideration. In fact, to keep the same pattern, we should also have a similar formula in the (13) system. By putting $s=0$ in expression (14) and remembering (13) we again obtain what we have called *zeration* (see [5] and [6]):

$$A(0,n) = 2 \boxed{0} (n+3) - 3 = n + 1 \qquad \text{(zeration)}$$

which gives:

(16) $$\boxed{2 \circ (n+3) = n + 4}$$

and, finally:

(17) $$2 \circ n = n + 1 \qquad \text{(for: } n \geq 3\text{)}$$

But, since we already know that:

(18) $$2 \circ 2 = 2 + 2$$

and that: $$n \circ n = n + 2$$

we can start using these expressions in order to find out the first properties of the "*zeration*" operation.

4.4. *Recursive properties.* We may conclude, in fact, that *zeration*, *addition*, *multiplication*, *exponentiation*, *tetration* (etc.) are generalized "natural" hyperoperations that belong to the class of *recursive functions*, for which we know the following relations (the first line being presented as an homomorphic extension of the others):

s = 1	*Addition*:	$x + 0 = x$	;	$x + (y + 1) = \underline{x \circ (x + y)}$
s = 2	*Multiplication*:	$x\ .\ 0 = 0$	;	$x\ .\ (y + 1) = x + (x\ .\ y)$
s = 3	*Exponentiation*:	$x \wedge 0 = 1$	;	$x \wedge (y + 1) = x\ .\ (x \wedge y)$
s = 4	*Tetration*:	$x \# 0 = 1$	;	$x \# (y + 1) = x \wedge (x \# y)$

or, generally:

(19) $$\boxed{x \boxed{s} (y + 1) = x \boxed{s\text{-}1} (x \boxed{s} y)}$$

4.5. *The Grzegorczyk Hierarchy.* Therefore, from the definition of Ackermann's function, new arithmetical operations automatically follow. In particular, we should like to stress the following two:

$$A(0,n) = 2 \circ (n+3) - 3 = n + 1 \qquad \text{(zeration)}$$
$$A(4,n) = 2 \# (n+3) - 3 = {}^{n+3}2 - 3 \qquad \text{(tetration)}$$

The terms tetration and zeration and their operator symbols ("#" and "°") are proposed by the authors. W. Ackermann has actually established the existence of an infinite spectrum of arithmetical operations, sometimes referred to as the Grzegorczyk Hierarchy. Conventional mathematics notation breaks down at this point and something new needs to be specifically devised, as we shall see, in order to represent very big numbers. We shall describe in detail the tetration and zeration operations, in the following sections.

5. TETRATION

5.1. *Construction of a Tetration operation.* As we have seen in an elementary way in section 2, the "construction" principles governing operations such as *multiplication* and *exponentiation* are well known:

(20) $$a\ .\ n = \underbrace{a + a + a + \ldots\ldots + a}_{n \text{ times}}$$

(21) $$a \wedge n = \underbrace{a\ .\ a\ .\ a\ .\ \ldots\ldots\ .\ a}_{n \text{ times}}$$

where: $n \in \mathbf{N}$ (Set of natural numbers).

By analogy with these formulas and recalling (5), we can write a similar formula for *tetration*:

(22) $$a \# n = \underbrace{a \wedge (a \wedge (a\ .\ \ldots\ldots (a \wedge a))}_{n \text{ times}}$$

with: $n \in \mathbf{N}$.

We must however observe that both *exponentiation* and *tetration* are not commutative, i.e. that:

(23) $\quad \boldsymbol{a \wedge n = a^n \neq n \wedge a = n^a}$

$\boldsymbol{a \# n = {}^n a \neq n \# a = {}^a n}$

We shall see that they don't satisfy the associative property either and we can anticipate that all operations with ranks different from those of *addition* and *multiplication* have properties which are different from those valid for the two first basic elementary operations.

5.2. *Right and Left Towers*. The question to be examined here is how we can represent what we can call the "compaction" of an operation such as:

$\boldsymbol{c = a \wedge a}$ **.**

It is a power "tower" built up by elevating a number to itself. As we have seen, we can define a new operation that we have called "*tower*" or "*tetration*", that we shall write as follows:

(24) $\quad \boxed{\boldsymbol{c = a \# 2 = a \wedge a = a^a}}$

with $\boldsymbol{a}$: real number

and that we could also show as:

$\boldsymbol{c = a \# 2 = a \uparrow\uparrow 2 = {}^2 a}$, to be read: "***a***-tower-***2***".

As we have already said, symbol ***a # 2*** is used in this paper. It is the first thing that comes to mind, if we need a clear and "free" symbol. As we have also seen, symbol $\boldsymbol{{}^2 a}$ was proposed by Rucker (1995) and ↑↑ was created by Knut (1976). The word *tower* means precisely that, but it is also inspired by an assonance with *power*, since "*tower*" is hierarchically following the "*power*" operation. The meaning of the word *tetration* has already been clarified. The result of a tetration operation such as ***c = a # 2*** is evidently a real number "to the power of itself", which gives an exponential tower at two levels. If the floor levels (the extension, or the height of the tower) are only two, there are no problems. When this is not the case, we have to devise a way of performing this operation. Indeed, given the non commutativity of exponentiation, the two following expressions are different:

$$a^{(a^a)} \neq (a^a)^a .$$

(*Please note! - For **a = 2**, we exceptionally have **2 ^ (2 ^ 2) = (2 ^ 2) ^ 2 = 16**, but this is only a coincidence*).

By using the sequential notation of standard pocket calculators, we may also write:

(25) $\quad \boldsymbol{a \wedge (a \wedge a) \neq (a \wedge a) \wedge a}$

But the non commutativity also implies the non associativity of the operation. In order to give meaning to an expression such as: $\quad z = a \wedge a \wedge a \wedge a = a^{a^{a^a}}$

we have to choose a procedural rule, otherwise the operation is not defined. We may in fact have:

priority to the right: $\quad z = a \wedge (a \wedge (a \wedge a))$ *or,*

priority to the left: $\quad z = ((a \wedge a) \wedge a) \wedge a$

(but other configurations are also possible).

5.3. *Definitions.* As we see in the following example, starting from a power tower with any number of levels and with priority to the left, we finally obtain an inhomogeneous tower, with only three levels, but with priority to the right. It is "inhomogeneous", or "incomplete", because the last exponent of the tower is different from ***a***.

This, however, is generally the case:

since, if (***n*** levels): $\quad z = ((((a \wedge a) \wedge a) \wedge a) \wedge \ldots.) \wedge a = ((((a^a)^a)^a)^{\cdots})^a$ [***n*** times]

we always have:

(26) $\quad \boxed{z = a^{a.a.a.....a[n-1.times]} = a^{(a^{n-1})}}$

with $\boldsymbol{n}$: positive integer number

Which means that all the "left" homogeneous towers (with any number of levels) are collapsible to become "right" (inhomogeneous) towers, with only three levels. This suggest that only the "right" towers are actually new fundamental elementary operations, not automatically reducible to others, whereas all the "left" towers are to be considered as banal cases of reducible expressions. We are therefore justified, as shown in section 2, in going on to use an "exponential" operator that acts on its right, as follows: $\quad \boldsymbol{z = \boxed{a \wedge} x = a \wedge x = a^x}$

with $\boldsymbol{a}$: positive real number.

(The of ***a > 0*** condition is chosen for sake of simplicity and to stay in touch with … reality!)

By continuing the iteration of $\boxed{a\ \hat{}\ }$, equivalent to the use of operator ***exp$_a$*(...)**, exponential to the base ***a*** of operand ***(...)***, we are able to create complete homogeneous towers (with identical elements) with priority to the right, for instance, as in the following example (with ***5*** levels):

$$z = \boxed{a\ \hat{}\ }\ \boxed{a\ \hat{}\ }\ \boxed{a\ \hat{}\ }\ \boxed{a\ \hat{}\ }\ a = a \wedge (a \wedge (a \wedge (a \wedge a))) = a^{(a^{(a^{(a^a)})})}$$

i,e.: $z = exp_a\,(exp_a\,(exp_a\,(exp_a(a))))$

and we can agree that:

(27) $$\boxed{z = exp_a\,(exp_a\,(exp_a\,(exp_a(a)))) = a^{a^{a^{a^a}}}}$$

(*without parenthesis and with priority to the right*)

In general, for the "right" towers, the following expression:

(28) $$z = \underbrace{exp_a\,(\,...\,(exp_a(a)))}_{n \text{ times}} = a^{\cdot^{\cdot^{a}}} \quad [\boldsymbol{n} \text{ times}]$$

is not reducible. We can therefore establish that this is the true "*tower*", "*superpower*", or "*tetration*" operation (with priority to the right), which is shown as follows (in the case of ***n*** iterations):

(29) $\boxed{\boldsymbol{z = a\ \#\ n}}$ [***a***-tower-***n***]

(*with **a**: positive real and **n**: positive integer*).

And we have:

a:	base of the tower
n:	height of the tower or super-exponent.

5.4. *The super-exponential function*. From expression ***z = x # y***, we have two main situations, depending on to which of the ***x, y*** values is the independent variable (the other one being a parameter). We can have:

- ***z = a # y*** (with ***a > 0*** and a real number, the *super-exponential* function), or
- ***z = x # n*** (with integer ***n ≥ 0***, the *super-power* or *tower* function).

From expression (27), with $y = m$ (natural), we have:

$$z(m) = \underbrace{exp_a\,(\,...\,(exp_a(a)))}_{m \text{ times}} = a^{\cdot^{\cdot^{a}}} \quad [m \text{ times}] = a\ \#\ m$$

we may write: $z(m+1) = a^{\cdot^{\cdot^{a}}}\ [m+1 \text{ times}] = a^{a\ \#\ m} = a\ \#\ (m+1)$

and also: $z(m-1) = a^{\cdot^{\cdot^{a}}}\ [m-1 \text{ times}] = log_a\,(a\ \#\ m) = a\ \#\ (m-1).$

In other words, we can say that operator ***exp$_a$*(...)** elevates the tower's height by one unit and that operator ***log$_a$*(...)** lowers the tower's height by one unit; i.e. we have the following important recursive properties:

(30) $$\boldsymbol{a\ \#\ (m+1) = a^{a\ \#\ m} = a \wedge (a\ \#\ m)}$$
$$\boldsymbol{a\ \#\ (m-1) = log_a\,(a\ \#\ m)}$$

As far as the "*tetration*" operation is concerned, with base ***a*** = constant, we can easily calculate the values of ***z = a # y***, for small values of ***y***, starting from the fact that we must always have: ***z(2) = a # 2 = a^a***.

Therefore:

(31) $$a\ \#\ 1 = a$$
$$a\ \#\ 2 = a^a$$
$$a\ \#\ 3 = a^{a^a} \quad ... \text{ etc.}$$

but also: $a\ \#\ 0 = log_a\,(a\ \#\ 1) = log_a\, a = 1$

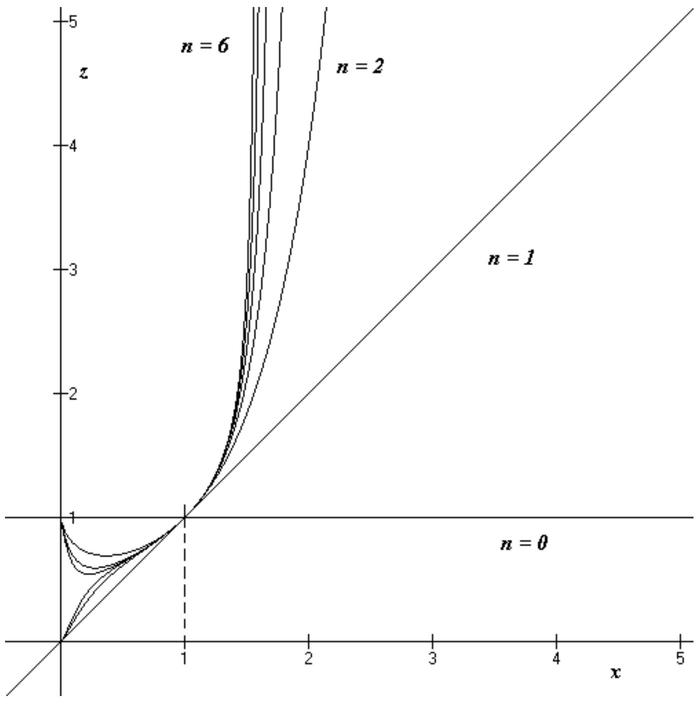
and: $a\ \#\ {-1} = log_a\,(a\ \#\ 0) = log_a\, 1 = 0$

and, finally: $a\ \#\ {-2} = log_a\,(a\ \#\ {-1}) \rightarrow -\infty.$

The fact remains that function ***z(y) = a # y*** (the *super-exponential* function) can be calculated only for integer values of ***y*** (and for ***y ≥ -2***). The dependent variable ***z(y)*** is therefore a quantified discontinuous entity. Any effort to obtain something like a continuous plot faces a serious problem when it comes to extending *tetration* to real numbers. Another important matter is that all the values of ***z***, for ***y > 5***, are finite (for ***y*** finite). This is almost unbelievable, because the value that we can calculate for ***2 # 5 = 2 ^ 65.536*** is already virtually unimaginable.

Let us think of the immensity of a number such as ***10.000 # 10.000*** ! This fact suggests that *tetration* could be an excellent tool for representing very large numbers (See § 7.4).

5.5. *The Tower function.* Nevertheless, the plot of function $z(x) = x \# n$ (with n positive integer), i.e. the *tower* function with super-degree n, can easily be obtained by using a very simple pocket calculator, at least for $x \geq 0$. For the values of $x < 0$, the dependent variable $z(x)$ assumes complex values.
In the next figure, we plot the values of functions $z = x \# n$, that can be called "super-power functions", for integer values of the super-degree parameter $n = 0, 1, 2, 3, 4, 5, 6$ ... and for $x > 0$.



6. THE SUPER-ROOT

6.1. *Inversions.* As already stated in expressions (23), *exponentiation* and *tetration* are not commutative operations, i.e.:

$$x \wedge y = x^y \neq y \wedge x = y^x \qquad \text{(exponentiation)}$$
$$x \# y = {}^{y}x \neq y \# x = {}^{x}y \qquad \text{(tetration).}$$

Let us now compare the two above-mentioned z functions (of variables x, y) such that, by analogy with:

$z = x \wedge y = x^y$ [*power or exponential*]

we have: $z = x \# y = {}^{y}x$ [*tower or super-exponential*].

Let us then try to "extract" the values of x and y, i.e. to find the <u>inverse functions</u> of z. We have:

- in the case of the <u>power-exponential</u>:

(32) $x = \sqrt[y]{z} = y\text{-}rt(z)$ [y-th root of z]

$y = log_x(z)$ [logarithm|base x *of* z]

- in the case of the <u>tower-super-exponential</u>:

(33) $x = {}^{y}\big)z = y\text{-}srt(z)$ [y-th super-root of z]

$y = slog_x(z)$ [superlog|base x of z]

Therefore, we have the following inverse functions :

- <u>for $y = n$ (constant):</u>

(34) $z = x \wedge n = x^n$ n-th power of x => $x = {}^{n}\sqrt{z}$ n-th root of z

$z = x \# n = {}^{n}x$ n-th tower of x => $x = {}^{n}\big)z$ n-th super-root of z

- <u>for $x = a$ (constant)</u>:

(35) $z = a \wedge y = a^y$ exp, base a, of y => $y = log_a\, z$ logarithm, base a, of z

$z = a \# y = {}^{y}a$ superexp, base a, of y => $y = slog_a\, z$ superlog, base a, of z

Consequently, the non-commutativity of the operations implies that there must be <u>*two*</u> different *inverse tower* operations, designated as "***super-root***" and "***super-log***".

6.2. *The Square-root.* In this section we propose to investigate the possibility of finding some algorithms for calculating the super-root (order n) of a number z, defined as a number x that satisfies the following expression:

$$x \# n = z \qquad \Rightarrow \qquad x = {}^{n}\!\!\Big)\overline{z} = \textit{nth-srt}\ z \qquad \text{(nth-super-root)}$$

In particular, we shall try to solve the problem where $n = 2$, i.e. in the case where:

$$x \# 2 = z \qquad \Rightarrow \qquad x = {}^{2}\!\!\Big)\overline{z} = \textit{s-sq-rt}\ z \qquad \text{(super-square-root)}$$

The analysis of this problem started in 1986 [5] with the review of an iterative formula used since ancient time for calculating the classical square root of a number. In fact, if z is the square of x, i.e.:

$$z = x\,.\,x = x \wedge 2 = x^{2}$$

then, x can be obtained with:

(36) $$\boxed{x = \textit{sqrt}\ z \approx (p + z/p)/2}$$

where number p is a first approximated value for the square-root of z, with $z > 0$. The result is obtained by an iterated application of the formula, by systematically stating $x \rightarrow p$. It can be verified with a pocket calculator that the iterations converge very rapidly to the value $x = \textit{sqrt}\ z$. This formula can be proved as follows. Supposing p to be an approximate solution, we shall certainly have:

$$z = x^{2} = x\,.\,x = z/p\,.\,p$$

In fact, a more approximate value for solution x could be obtained using the arithmetic mean between the two magnitudes z/p and p. After putting the arithmetic mean as the new approximate value p and after a certain number of iterations, we get $z/p = p = x$, the required value for the square-root of z. This iterative formula has been known for more than two thousand years and was apparently used in ancient Greece.

6.3. *The Super Square-root (First algorithm for calculating the "ssqrt").* If we consider the operations of immediately higher rank, i.e. the rank of "*exponentiation*" instead of "*multiplication*", we can suppose that the following expressions would be valid:

If: $$z = x \wedge x = x \# 2 = {}^{2}x$$

then:

(37) $$\boxed{x = \textit{ssqrt}\ z \approx \sqrt{(p\,.\,\log_{p} z)} \quad \text{or:} \quad x = \textit{ssqrt}\ z \approx \sqrt{(p\,.\,\sqrt[p]{z})}}$$

where p, again, is a first approximate value for x, super-square-root of z.

The procedure is indeed similar to that used for finding the square-root of z, by increasing all the operations ranks by one unit and by using the geometric, instead of the arithmetic mean. In this case, we observe that $\log_{p} z$ and $\sqrt[p]{z}$ are the inverse operations of $z = p \wedge x$ (with $x = \log_{p} z$) and of $z = x \wedge p$ (with $x = \sqrt[p]{z}$), both corresponding to expression z/p, used for finding the square-root of z. Here, again, it can easily be verified that the application of one of the last two formulas (after repeatedly exchanging $x \rightarrow p$) for the calculation of the geometric mean (of either $\log_{p} z$ or $\sqrt[p]{z}$ with p), gives a result rapidly converging to the super-square-root of z, i.e. to $x = \textit{ssqrt}\ z$. Actually, formula (37) was tentatively admitted as a working hypothesis, by using a homo-morphical mapping between expressions (36) and (37). It can easily be verified that formula (37) rapidly converges, for values of the argument $z > \approx 1{,}7$ ($p > \approx 1{,}6$).

6.4. *A second algorithm for calculating the ssqrt.* Another algorithm for calculating the super-square root of a number is based on a different principle. In fact, a very important relation can be found in an extreme case, in which n is unlimited, i.e. where the value of $n \rightarrow \infty$.

Let us consider the following expression:

(38) $$z_{\infty} = \lim_{n\to\infty} x \# n = x^{x^{\cdot^{x}}} = h(x)$$

(∞ times !)

We propose to examine the problem of the values of function $z_\infty = h(x)$ for all possible values of variable x (real > 0). Let us take the logarithm (base e) of z_∞ in (38):

$$\ln z_\infty = \ln (\lim_{n\to\infty} x \# n) = \ln \underbrace{x^{x^{\cdot^{x}}}}_{(\infty \text{ times})} = x^{x^{\cdot^{x}}} . \ln x = h(x). \ln x = \ln h(x)$$

i.e.:

(39) $\ln z_\infty = h(x) . \ln x = \ln h(x)$

which gives: $\ln x = (\ln h(x))/ h(x) = (\ln z_\infty)/ z_\infty = \ln (z_\infty \wedge (1/ z_\infty))$

and, then:

(40) $x = \sqrt[z_\infty]{z_\infty} = g(z_\infty)$

But, as we remember, from expression (38):

$$z_\infty = \lim_{n\to\infty} x \# n = x \# \infty = h(x)$$

we have: $x = g(z_\infty) = \lim_{n\to\infty} {}^{n}\!\!\left.\right) z_\infty = {}^{\infty}\!\!\left.\right) z_\infty$

Therefore:

(41) $\boxed{{}^{\infty}\!\!\left.\right) z_\infty = \sqrt[z_\infty]{z_\infty}}$

In other words, the ∞-th super-root (s = 4) of a quantity z is equal to the z-th (classical, s = 3) root of z. This is an important result, that deserves further careful analysis, since it appears that a similar formula is also valid for the other hyperoperation levels. From formula (41) it follows that, in general, we can write:

$${}^{\infty}\!\!\left.\right) x = \sqrt[x]{x}$$

or, that: $\lim_{n\to\infty} {}^{n}\!\!\left.\right) x = \sqrt[x]{x}$

Formula (41) allows us to calculate the value of the super-square root of a number, for $z < 1.7$.

In fact, let us put: ${}^{2}\!\!\left.\right) x = y \quad => \quad {}^{2}y = x$

Then, by applying formula (41) to argument $1/y$, we have:

$${}^{\infty}\!\!\left.\right) 1/y = \sqrt[1/y]{1/y} = (1/y)^y = 1 / y^y = 1 / {}^{2}y$$

which implies: ${}^{\infty}(1 / {}^{2}y) = 1 / y$

or, to be more precise: $\lim_{n\to\infty} [{}^{n}(1 / {}^{2}y)] = 1 / y$

i.e.: $\lim_{n\to\infty} [{}^{n}(1 / x)] = 1 / y$

or : $y = ssqrt(x) = 1 / \lim_{n\to\infty} [{}^{n}(1 / x)]$

and, in order to be consistent with the notation of (37), we can write:

(42) $\boxed{x = ssqrt(z) = 1 / \lim_{n\to\infty} [{}^{n}(1 / z)]}$

This is the second formula for calculating the super-square root of z (for $z < 1{,}7$).

6.5. *A more general formula*. Let us now reconsider expression (42), that we can write as:

$(: z) \# \infty = : ({}^{2}\!\!\left.\right) z)$ (with $: z = 1/z$, reciprocal of z)

which can also be written as: ${}^{2}\!\!\left.\right) z = 1 / [(1/z) \# \infty]$.

In other words, the <u>super-square-root of z</u> is the *reciprocal of the infinite tower* of $1/z$. This last formula can be further developed by expressing the infinite tower of "$1/z$" by using function "h($1/z$)", as defined by expression (38), e.g. by: $h(1/z) = (1/z) \# \infty$

and can be expressed as: $h(1/z) = - w[-\ln(1/z)] / \ln(1/z)$

where $w(u)$ is the solution of: $u = w \cdot e^w$

and it is called is the *Lambert function* or *Product Log* (see Appendix A-06).

Therefore we have that: ${}^{2}\!\!\left.\right) z = 1 / h(1/z) = - \ln(1/z) / w[-\ln(1/z)]$

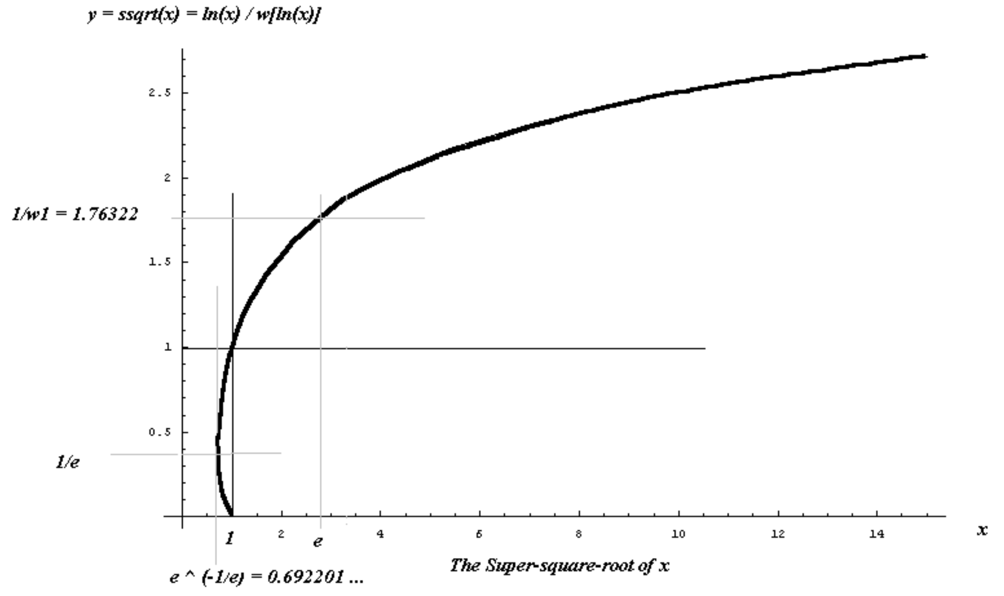
And, finally, by changing the variables, we obtain:

(43) $\boxed{{}^{2}\big)\overline{x} = \ln(x) / w[\ln(x)]}$ (with $x > e^{-1/e}$, $x \neq 1$)

This is the third (more general) formula can be adopted for the calculation of the super-square-root of number x. Condition $x \geq e^{-1/e}$ is also geometrically justified by the graphical inversion of function:

$$x = y \ \# 2 \ (y = {}^{2}\big)\overline{x}\ ).$$

The graph of $y = {}^{2}\big)\overline{x}$ is shown in the following figure.



The super-square-root of x for $x < e^{-1/e}$ (0,692 201 …) involves complex solutions. The unique real value of $ssqrt(e^{-1/e})$ is $1/e$. For $e^{-1/e} < x \leq 1$, the super-square root has two real values. For $x > 1$, it is a continuous increasing function and we have: $\lim\limits_{x\to\infty} ssqrt(x) = \infty$. In particular, value $w(1) = w_1 =$ 1,567 143 … is the "*omega constant*", obtained for $w \,.\, e^{w} = 1$. For $x = e$, we have: ${}^{2}\big)\overline{e} = 1/w_1 =$ 1,763 220 … .

7. THE SUPER-LOG

7.1. *The Super-logarithm*. The *tetration* operation is only well defined, in the field of real numbers, for integer values of the super-exponent, greater than ***-2***. In other words, in expression:

(44) $\boldsymbol{z = x \# y = {}^{y}x = x \uparrow\uparrow y}$ (***x***-tetrated-***y***)

tetration is only valid for "super-exponent" $\boldsymbol{y = n}$, with $\boldsymbol{n > -2}$ (for $\boldsymbol{n = -2}$, the value of $\boldsymbol{z}$ is unlimited, i.e. $\boldsymbol{z} \to -\infty$). For the domain of $\boldsymbol{y < -2}$, the estimation of $\boldsymbol{z}$ requires us to use the logarithms of negative numbers. We shall examine here the application of expression (44) to the case in which the "base" $\boldsymbol{x}$ is a real number $\boldsymbol{a}$, larger than ***1***, i.e., for:

$\boldsymbol{x = a}$, with $\boldsymbol{a > 1}$.

The above-mentioned expression will be transformed, in this case, as follows:

(45) $\boldsymbol{z = a \# y = {}^{y}a = a \uparrow\uparrow y}$

From expression (45), and taking also into account definitions (33) and supposing we know the value of $\boldsymbol{z}$, we can extract $\boldsymbol{y}$, as follows:

(46) $\boxed{\boldsymbol{y = slog_{a}\, z}}$ (the super-log, base $\boldsymbol{a}$, of $\boldsymbol{z}$)

The "***super-log***, base a , of z " is the value, or the "*height* ", of the "*super-exponent* " y that should be assigned to number a , for obtaining number z . From expression (44) and bearing in mind its restrictions, we must accept that quantity y has been defined only for values of z for which y is an integer number > -2. But, since z is supposed to be any real number, the problem posed here is how to extend the validity of the ***super-log*** definition so as to eliminate this restriction, if possible.
We shall, in particular, see that the extension of the *superlog* to the reals is closely connected to the solution of the problem of extending *tetration* itself to the reals (§ 7.3).

7.2. *Incomplete Towers*. Furthermore, in the recent scientific literature, we can find expressions such as:

(47) $$z = a^{a^p} = a \wedge (a \wedge p).$$

This (generalized) *tetration* appears as an "*incomplete tower operation*", incomplete because the "last" exponent of the iterated exponentiation is different from base "a ". However, this kind of expressions is commonly found in scientific texts concerning, for instance, problem-complexity theory, game theory or in some AI developments. How can we incorporate this *incomplete tower* into a kind of *generalized tetration*?
In order to investigate the possibility of doing this, let us again use the following "exponentiation" operator:

(48) $$\boxed{a \wedge}\, p = a \wedge p$$

and: $$\boxed{a \wedge}^{\,n}\, p = a \wedge (a \wedge (a \wedge (a \wedge \ldots\ldots p))) \qquad \text{with } n \text{ iterations of } a \wedge$$

Let us now suppose that quantity p is a real number larger than 1 and smaller than a. This is often the case in expressions similar to what we have called incomplete towers. If this is not so, we can transform these expressions into a canonical form, with $1 < p < a$.
Let us now define another new tool, the "logarithm" operator, complementary to $\boxed{a \wedge}$, i.e.:

(49) $$\boxed{\log_a}\, p = \log_a p$$

and: $$\boxed{\log_a}^{\,n}\, z = \log_a(\log_a(\log_a(\log_a \ldots\ldots z))) \qquad \text{with } n \text{ iterations}$$

Now, if we have: $z = \boxed{a \wedge}^{\,n}\, p$
we could also put: $z = a \,\#\, (n + q)$. with: $0 < q < 1$
In fact, the incomplete tower has an "extension" (its super-exponent) that must be higher than n, otherwise we would have had $p = 1$ ($q = 0$), and must be less than $n + 1$, otherwise we would have had $p = a$ ($q = 1$).
In other words, we can write:

(50) $$\boxed{z = \boxed{a \wedge}^{\,n}\, p = a \,\#\, (n + q).}$$

But, by iteratively applying the $\log_a$ operator, we also have:

$$\boxed{\log_a}^{\,n}\, z = p = \boxed{\log_a}^{\,n}\, [a \,\#\, (n + q)] = \quad a \,\#\, (n + q - n)$$

i.e.:

(51) $$p = a \,\#\, q \qquad \text{with } 0 < q < 1 \text{ and } 1 < p < a$$
$$q = slog_a\, p \qquad \text{for any } n.$$

7.3. *Strategy for extension to the reals*. Formulas (50) and (51) provide a strategy for defining a generalized tetration operation $z = a \,\#\, y$ (as well as a continuous ***super-log*** function), extending its validity to all the real values of its super-exponent argument y and, so, including the results of the incomplete tower operation. Once a value for p is chosen, depending upon the canonical form of the incomplete tower, variable q is calculated as the super-log, base a, of p. Knowing p and q, if they can be estimated, would therefore be essential for the extension of tetration to the reals. Recalling the tetration expression $z = a \,\#\, y$, variable p will be used to define the value of z and variable q will fix the y coordinate.
The problem is therefore to calculate q as the super-log, base a, of a quantity p (with $1 < p < a$), bearing in mind that the result of the operation for q should be such that $0 < q < 1$.

In some cases these difficult calculations (assuming that they are actually possible) are not necessary, as in the following examples of super-logs, base 2, concerning certain known integer quantities:

$$\mathbf{slog_2\ 16 = 3} \qquad \text{because: } \mathbf{2 \# 3 = 16}$$

$$\mathbf{slog_2\ 65.536 = 4} \qquad \text{because: } \mathbf{2 \# 4 = 65.536}$$

However, there is the further possibility of simply indicating the super-exponent extension, without actually performing the calculation of the non integer value of the super-exponent $\boldsymbol{n + q}$ of a *tower*, or of a *super-log*, i.e. by exactly estimating the value of $\boldsymbol{q}$ (with $\boldsymbol{n}$ integer and $\boldsymbol{0 < q < 1}$). This can be done, very precisely, by using an operator that would merely highlight the value of $\boldsymbol{p}$ (with $\boldsymbol{1 < p < a}$). This operator is the so-called "concatenation operator", from now on shown as an asterisk (*), with reference to the second formula of the (49) set and to (50), as follows:

(52) $$\boxed{\boxed{a\ \hat{}}\ ^{n}\, \boldsymbol{p} = (\boldsymbol{a \# n}) \, {*}\, \boldsymbol{p} = \boldsymbol{a} \# (\boldsymbol{n + q}).}$$

Formula (52) shows an incomplete tower built up from iterated exponentiations of a number $\boldsymbol{a}$, on itself, and terminated by a last exponent (the *super-exponent extension*) equal to $\boldsymbol{p}$. In other words:

$$(\boldsymbol{a \# n}) \, {*}\, \boldsymbol{p} = \boldsymbol{a}\ \hat{}\ (\boldsymbol{a}\ \hat{}\ (\boldsymbol{a}\ \hat{}\ (\boldsymbol{a}\ \hat{}\ \ldots\ldots \boldsymbol{p})))$$

or: $$(\boldsymbol{a \# n}) \, {*}\, \boldsymbol{p} = \boldsymbol{a}\ \hat{}\ \boldsymbol{a}\ \hat{}\ \boldsymbol{a}\ \hat{}\ \ldots\ldots\ \hat{}\ \boldsymbol{p}$$

(with $\boldsymbol{n}$ times $\boldsymbol{a}$ and … priority to the right!)

7.4. *Tetradic Representation of numbers*. The few examples shown in Appendix A-03, are sufficient to give an idea of how *tetration* can be used to represent very big numbers in what we could call a "*Tetradic Representation*". D. W. Lozier and P. R. Turner have published papers ([8], [9], [10])describing a number format called *Symmetric Level-Index* (SLI), in which numbers are stored in the form:

$$\boldsymbol{N = e\ \hat{}\ (e\ \hat{}\ \ldots(e\ \hat{}\ p))} = e^{e^{\cdot^{\cdot^{P}}}} \ ,$$

where $\boldsymbol{p}$ is a fraction from ***0,000*** to ***0,999***... and there are as many $\boldsymbol{e}$'s as necessary.

For example: $$\boldsymbol{10 = e\ \hat{}\ (e\ \hat{}\ 0{,}834\ 032 \ldots) = {}^{2}e * 0{,}834\ 032 \ldots}$$

$$\boldsymbol{143 = e\ \hat{}\ (e\ \hat{}\ (e\ \hat{}\ 0{,}471\ 239 \ldots) = {}^{3}e * 0{,}471\ 239 \ldots}$$

The advantage of this proposed system is that there will not be any computing overflow or underflow if we perform a finite number of operations such as + - × and /. In one of their articles, Lozier and Turner proposed a format that uses a 3-bit level field with 2 "sign" bits and the remaining bits (59, if it is a 64-bit word) for the fraction $\boldsymbol{p}$. This lets us represent numbers as high as the number represented by a tower of seven $\boldsymbol{e}$'s. This is the highest known number that can be handled by a computer number-representation system. However, there is also a computing software called "*Hypercalc*" that goes even higher.

8. ZERATION

8.1. *Homomorphisms of inverse operations*. In section 2 we introduced *zeration*, the new operation with a rank lower than that *addition*, in the hyperoperation hierarchy. The basic characteristics of such operation are described in formulas (16), (17) and (18) and should be consistent with table (6). This means that we must have:

- from (6): $\boldsymbol{a \circ a = a + 2}$

  $\boldsymbol{a \circ a \circ \ldots \circ a}$ ($\boldsymbol{n}$ times) $=$ $\boldsymbol{a + n}$
- from (17): $\boldsymbol{2 \circ a = a + 1}$ (for $\boldsymbol{a > 2}$)

In order to discover the properties that *zeration* must have, for it to be compatible with known operations, let us consider expression (42):

(53) $$\lim_{n\to\infty} {}^{n}\big)z = \sqrt[z]{z} \qquad [= \boldsymbol{g(z)}]$$

This formula can easily be compared with similar expressions used in other mathematical fields, such as:

(54) $$\lim_{n\to\infty} \sqrt[n]{z} = \boldsymbol{z / z} \qquad [= \boldsymbol{1}]$$

and: $$\lim_{n\to\infty} \boldsymbol{z / n} = \boldsymbol{z - z} \qquad [= \boldsymbol{0}]$$

This comparison is made because all the (53) and (54) expressions involve the inverse operations (of the "root" type) of the homologue hyperoperations of three pairs of contiguous ranks (s=4, s=3), (s=3, s=2), (s=2, s=1). Therefore, as a working hypothesis, it can be accepted that an equivalent formula might be valid involving inverse operations of the (s=1, s=0) rank pair.

Indeed, the inverse operations, of the "root" type (left-inverse operations) of the hyperoperations of ranks 4, 3, 2, 1 and 0 can be shown as follows:

(55)	s = 4	$z = x \# n$	=>	$x = {}^{n}\!\!\big)\overline{z}$
	s = 3	$z = x \wedge n$	=>	$x = \sqrt[n]{z}$
	s = 2	$z = x\ .\ n$	=>	$x = z / n$
	s = 1	$z = x + n$	=>	$x = z - n$
	s = 0	$z = x \circ n$	=>	$x = z \,\Delta\, n$

In the last line, we implicitly defined the inverse operation of *zeration*, indicated with the "Delta" operator.

First of all, we observe that expressions (53) and (54) imply the calculations of "$\lim_{n\to\infty} x$" and, then, that in expression (54) the same calculation, for the rank s=0, is not there. In other words (and respecting the homomorphic mappings) we should have:

(56) $$\boxed{\lim_{n\to\infty} z - n = z \,\Delta\, z \qquad [= -\infty]}$$

Expression (56) means, if it is acceptable, that "$-\infty$" should operate as the unit element for the *zeration* operation, i.e. we should have:

$$a\ .\ 1 \quad = a$$
$$a + 0 \quad = a$$
(57) $$\boxed{a \circ (-\infty) = a} \qquad \text{(unit element)}$$

We must observe that, if we have $x = z \,\Delta\, z = -\infty$, then we must also have $z \circ (-\infty) = z$, i.e. quantity $-\infty$ indeed acts as the unit element for the zeration operation.

8.2. *Homomorphisms of algorithms*. At this point, let us recall the formulas used to calculate, by iterative attempts, the super-square root and the square root of a number (36 and 37).

From:	$z = x \wedge x = {}^{2}x$	
we have (from 37):	$x = ssqrt\, z \approx \sqrt{(p\ .\ \log_p z)}$	[with $x \to p$],
or:	$x = ssqrt\, z \approx \sqrt{(p\ .\ \sqrt[p]{z})}$	[with $x \to p$].
Also, from:	$z = x\ .\ x = x^2$	
we have (from 36):	$x = sqrt\, z \approx (p + z/p) / 2$	[with $x \to p$].

As we have already noted, formulas (37) and (36), the validity of which we can verify, are linked together through a homomorphic mapping that relates operations of contiguous ranks, by decreasing their rank by one unit. In fact, tetration corresponds to exponentiation, multiplication to addition and, as far as the inverse operations are concerned, logarithm and root correspond to the division operation.

At this point, we can tentatively apply the same mapping model, by decreasing the operations rank by one unit (with ***sqrt*** corresponding to ½, addition to zeration, division to subtraction), and so obtaining:

from: $z = x + x = x\ .\ 2$

by analogy with (37, 36), if z is an even integer number, the following formula is valid:

(58) $$\boxed{x = z / 2 \quad \approx (p \circ (z - p)) - 2} \qquad [\text{with } x \to p].$$

with $p \in \mathbf{Z}$ ($\mathbf{Z}$: set of integers)

By a change of variables ($x\text{-}p \leftrightarrow p$), we can assume that the following relation is also verified:

$$x = z / 2 \quad \approx ((z - p) \circ p) - 2$$

and conclude that, within the validity of (58) *zeration* must be commutative. However, it is not associative.

8.3. *Properties of Zeration*. We can therefore put together the operational properties that, in our working hypothesis, should be satisfied by *zeration*, grouped into a new set of formulas:

(59)	- from (6):	$a \circ a = a + 2$	
		$a \circ a \circ \ldots \circ a$ (n times) $= a + n$	
	- from (17):	$2 \circ a = a + 1$	(for $a > 2$)
	- from (58):	$a \circ b = b \circ a$	(commutativity)
	- therefore:	$a \circ 2 = a + 1$	(for $a > 2$)
	- from (57):	$a \circ (-\infty) = a$	($-\infty$: unit element)

We must also remember expression (19), for s=1:

$$x + (y + 1) = x \circ (x + y) \qquad (y > 0)$$

and, by putting $x = a$ and remembering (59):

$$a \circ (a + y) = (a + y) \circ a = a + y + 1 = (a + y) + 1$$

<u>for any y</u>, by putting $a + y = d > a$, we must have:

$$a \circ d = d \circ a = d + 1 \qquad (d > a)$$

From these formulas we conclude that:

(60) $a \circ b = \max(a,b) + 1$ (for $a \neq b$)

and: $a \circ b = a + 2 = b + 2$ (for $a = b$)

and: $a \circ (-\infty) = (-\infty) \circ a = a$

In conclusion, the properties of zeration ($a \circ b$) should be those shown in the following table ($a \circ b = b \circ a$):

(61)

$a \circ b = a + 1$	for:	$a > b$
$a \circ b = b + 1$	for:	$a < b$
$a \circ b = a + 2 = b + 2$	for:	$a = b$
$a \circ b = a$	for:	$b = -\infty$
$a \circ b = b$	for:	$a = -\infty$

8.4. *Graph of a Zeration function*. As an example, the graph of a *zeration* function, defined as:

$$y(x) = a \circ x = x \circ a$$

can be shown, for $a = 2$, in the diagram of the next figure ($y = 2 \circ x$).

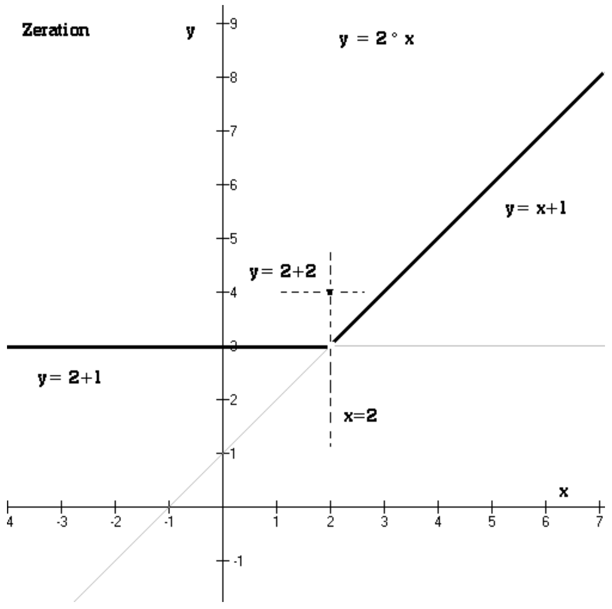
The diagram shows the following characteristics of the *zeration* plot:

- the zeration graph shows a constant branch, $y = 3$, for $x < 2$. In the general case, i.e. for $y = a \circ x$, this branch is <u>$y = a + 1$</u>, for $x < a$;
- the zeration graph is linear (with a 45° slope), $y = x + 1$, for $x > 2$. In the general case, if we put $y = a \circ x$, there is a second branch, <u>$y = x + 1$</u>, for $x > a$, with a discontinuity in the tangent in $x = a$;
- there is a **discontinuity** of "y", for the value $x = 2$, where we have $y = 2 + 2 = 4$.
  In general, for $y = a \circ x$, the value of "y" for $x = a$ is $y = a + 2$ and the limit of "y", for $x \rightarrow a$, (i.e.:
  $y = a + 1$), is different from $y(a) = a + 2$;
- the $y = x + 1$ straight line is a *characteristic* of all the zeration operations of type $y = a \circ x = x \circ a$, disregarding the value that constant "a" can assume; it could be considered as the "support" straight line for all the zeration functions, defining the <u>second branch</u> of all the graphs (the first branch is defined by the value of the constant "a");
- the value of $y = a \circ x$, for $x = -\infty$, i.e. $y = a \circ (-\infty)$, is **another discontinuity** of the zeration function, <u>not shown by the graph</u>, since we have: <u>$y = a \circ (-\infty) = a$</u> (instead of $a + 1$), and <u>$(-\infty) \circ x = x$</u> (instead of $x + 1$). This fact, as derived from Rubtsov's demonstrations (see [5],[6]), must be carefully emphasized.

In conclusion, the zeration function ($y = a \circ x$, with "a": real constant), apart from the discontinuity of tangent, is characterised by <u>two other major value discontinuities</u>:

- for $x = a$ where y jumps **<u>up</u>** by one unit;
- for $x = -\infty$ where y jumps **<u>down</u>** by one unit.

The *zeration* graph (***y*** = ***a*** ° ***x*** = ***x*** ° ***a***) takes the form of a "broken" straight line, with a strong discontinuity at point ***x*** = ***a***.



9. PRACTICAL APPLICATIONS OF ZERATION

9.1. *Discontinuities*. In the physical and technical fields, the majority of analyzed events are connected with modifications to the internal structure of a subject of research (submitted to the attention of an observer) and, as a corollary, with modifications to functional relations of the subject with its environment. In most cases these modifications involve sudden changes in the measured physical magnitudes. Modern mathematical methods are not perfectly suitable for describing digital processes. As a rule, discontinuous functions, impulse and step functions are approximated by Stieltjes integrals, Fourier series, generalized "by functions" and "differentials", as defined in the framework of the Laurentz theory.
Use is also made of various linearization models, spline approximations, etc., to describe these processes. However, all procedures for adapting such mathematical models still require rigorous demonstration [21]. But, in some cases, we need to use operations that allow us to express all the segments of a "broken" linear, or discontinuous, process with a single function. Among the existing tools, particularly used in digital signal analysis, there are the well-known Dirac's impulse function and Heaviside's unitary step function, which, however, are not defined by means of elementary operations. In the following paragraphs, we shall see how it is possible to introduce, in an elementary and logical way and without having to rely on a merely "ad-hoc" definition, a set of such functions, once zeration is accepted as one of the elementary hyperoperations.

9.2. *The Step function*. Heaviside's unitary step function ***z = H(x)*** is defined as follows:

$$z = H(x) = 0 \qquad \text{for: } x < 0$$
$$z = H(x) = 1 \qquad \text{for: } x \geq 0$$

Function *H(x)* is a well known tool used in digital circuit analysis and it has a very simple, continuous and regular Laplace transform, function 1 / *p*:

$$\int_0^\infty e^{-px} H(x)dx = 1/p \qquad (p\text{: complex variable})$$

In order to see how zeration can allow us to avoid an "ad-hoc" definition of *H(x)*, let us consider expressions (60) and (61), re-interpreted as follows, where an independent variable ***x*** is supposed to be "*zerated to*" a constant, real number, ***a***:

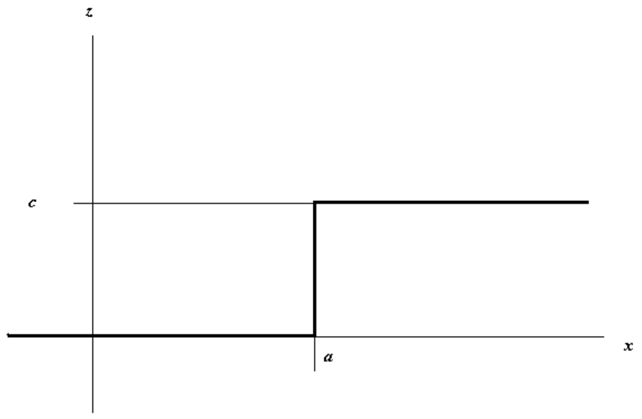
$$x \circ a = max\,(x, a) + 1 \qquad \text{if.} \qquad x \neq a$$
$$x \circ a = max\,(x, a) + 2 \qquad \text{if:} \qquad x = a.$$

We can then re-define Heaviside's function *H(x-a),* as follows:

(62) $\boxed{H(x-a) \;=\; Sign((x \circ a)-(a+1))}$

where *Sign(...)* is an operation that evaluates the sign of the operand *(...)* and gives, as a result, the values +1 or –1, according to whether the operand is positive or negative.

The graph of $z = c \cdot H(x-a)$ is shown in the following figure:



Function *H(x - a)* is also called the *"unitary step function"* and it is also written as follows:

$H(x-a) = Step(x, a) = On\,(x;a)$ (see also [6])

The link between Heaviside's function *H(x-a)* and Dirac's function *δ(x - a)* is given by the following formula:

$$H(x-a) = \int_{-\infty}^{x} \delta(x-a)dx$$

9.3. *Other examples*. The use of zeration also allows us to introduce a very peculiar function, having a unique single value for $x = a$. For this, let us start by defining the following auxiliary function, also implicitly appearing in paragraph 9.2, in the definition of *H(x – a)*:

$xsu(x-a) = (x \circ a) - (a+1)$

we then have: $xsu(x-a) = 0$; for: $x < a$

$xsu(x-a) = 1$; for: $x = a$

$xsu(x-a) = x-a$; for: $x > a$

We can now define function *Spnt(x - a)* as follows:

(63) $\boxed{Spnt(x-a) = xsu(x-a) \cdot [xsu(x-a)-(x-a)]}$

And we have: $Spnt(x-a) = 0$ for: $x < a$

$Spnt(x-a) = 1$ for: $x = a$

$Spnt(x-a) = 0$ for: $x > a$

Function *Spnt(x - a)*, the "*unitary single point function*", has the characteristic of always being equal to zero, except for $x = a$, where its value is 1 and where it describes a single point at the distance "1 " from the "*x* " axis. Function $z = c \cdot Spnt(x-a)$ is the same as function "- *Gas(x;a;c)* ", as defined in [5].

Let us now reconsider the following auxiliary function:

$z = xsu(x-a) = (x \circ a) - (a+1)$.

As we know, it is characterized by a single point discontinuity, for $x = a$, and we have:

$z(a) = 1$

$\lim_{x\to a} z(x) = 0$.

We can eliminate the value discontinuity, at $x = a$, by defining a new function $z = Rmp(x-a)$, such as:

(64) $\boxed{Rmp(x-a) = xsu(x-a) - Spnt(x-a)}$

and we have:

$Rmp(x-a) = 0$ for: $x \le a$

$Rmp(x-a) = x-a$ for: $x > a$.

Function *Rmp(x – a)* can be called the "*ramp function*". It is a continuous function, a broken straight-line with a tangent discontinuity at $x = a$.

9.4. *New elementary functions*. The set of functions such as:

H(x – a)	the *unitary step* function
Spnt(x – a)	the *unitary single point* function, and
Rmp(x – a)	the *ramp* function,

defined only using the new zeration operation, can be considered as a set of new elementary functions, the composition of which would be useful to describe discontinuous functions or continuous functions with tangent discontinuities. The following relations are also valid:

$$\int_{-\infty}^{x} \delta(x-a)dx = H(x-a)$$

$$\int_{-\infty}^{x} H(x-a)dx = Rmp(x-a)\ .$$

From the previous relations, we could say that:

$$H(x-a) = \frac{d}{dx}[Rmp(x-a)]$$

to which some authors [19] "dare" also to add:

$$\delta(x-a) = \frac{d}{dx}[H(x-a)]$$

Function *Spnt(x – a)* is not directly connected with *δ(x – a),* that has a more elaborate definition. It can however be used to extract a single value out of a function *f(x)*, since we have:

$$f(x) \cdot Spnt(x-a) = f(a).$$

But we also have:

$$\int_{-\infty}^{x} \delta(x-a)f(x)dx = f(a)$$

and this fact also suggests the need for some additional research.

10. CONCLUSIONS

10.1. *Zeration*. The inclusion of *zeration* among the set of basic arithmetical operations is based on the following facts:

1. *Zeration* follows naturally from generalization of the formulas used for iterative calculation of the results of <u>both</u> known <u>and</u> new inverse arithmetical operations. In particular, we have: (by repeated iteration of: $x \rightarrow p$):
   $z = x \wedge x = {}^2x \quad \Rightarrow \quad x = \mathrm{ssqrt}\, z \approx \sqrt{(p \,.\, \log_p z)} \qquad (z \geq\approx 1{,}7;\ p \geq 1{,}6)$
   $z = x \,.\, x = x^2 \quad \Rightarrow \quad x = \mathrm{sqrt}\, z \approx (p + z/p)/2$
   $z = x + x = x \,.\, 2 \quad \Rightarrow \quad x = z/2 \approx (p \circ (z - p)) - 2 \qquad (z_{even} \in \mathbb{Z};\ p \in \mathbb{Z})$.
2. *Zeration* also follows from analysis of common invariant formulas (see [5], § 2.2].
3. *Zeration* implicitly follows from Ackermann's functions.
4. *Zeration* is immediately connected to *addition* and, being its predecessor in the hyperoperations hierarchy, it defines its underlying mathematical structure, i.e.:
   [i.e.: $a \circ a \circ a \circ a \circ \ldots \circ a$ *(n times)* = $a + n$ *(a-plus-n)*].
5. *Zeration* is useful for the exact description of functions plotted as "broken" lines, i.e. by means of
   curves that have tangent discontinuity. Such plots are very commonly found in the modeling analysis of digital circuits and in quantum mechanics, but also in the study of ordinary classical physical systems.
6. *Zeration* (and its derived operations and functions) establishes a kind of bridge between discrete and
   continuous functions, as shown in the example of the graph of $y = 2 \circ x$ (see § 8.4, 9.1 and 9.2); it
   can be used for naturally introducing a set of elementary discontinuous functions, such as *H(x – a),*
   Spnt(x – a) and *Rmp(x – a),* without the need for any additional "*ad-hoc*" definitions.
7. *Zeration* is <u>commutative, but not associative</u>.

10.2. *Tetration*. Similarly, the *tetration* operation (s = 4) and the other *hyperoperations* for values of the rank s > 4 may considerably expand the ability of mathematics to solve practical problems. In particular, *tetration*, <u>which is neither commutative nor associative</u>, could make available the means for:

1. Providing specialists in artificial intelligence (**AI**) with a tool for very concise assessments relating to *problem complexity levels* for non polynomial (**NP**) and, especially, for hyperexponential problems (**H-EXP**), by using expressions such as:
   (65) $\boxed{e \wedge (e \wedge (e \wedge \ldots\ldots \wedge t)) = {}^n e * t}$.
2. Expressing very large numbers in a very compact way, by using *tetradic representation* (see Appendix A-03), or the similar *Level Index* representation (LI), such as $N = {}^n a * p$, thus avoiding any computing overflow, when handling extremely large numerical data . A number N would be represented as $N = {}^n a * p$, where n is the super-exponent of the *tetration* and p is the super-exponent extension [10].

10.2. *Fields for further research*. Problems remaining for further investigations are those concerning:

- the extension of hyperoperations to the whole range of relative integer numbers (i.e. also to negative integers, see also Appendix A-04);
- the full extension of such operations to the field of real numbers (including rational numbers, as well as positive and negative irrational and transcendent numbers) and, possibly, to the field of complex numbers. The extension to the reals would mean writing:
  (66) $\boxed{{}^n a * p = {}^{n+q} a \text{ (with: } 0 < q < 1)}$.

Concerning the problem of fully extending *tetration* to the "*reals*" (real numbers), so far not completely solved, we can then make the following observations. The possibility of using *tetradic representation*, as shown in paragraphs 7.3 and 7.4, as well as in Appendix A-03, suggests that **<u>this is possible</u>** and it should remain, therefore, one of the main subjects for research in this sector.
The problem will be finally solved when we are be able to find an algorithm for obtaining the superlog of a number, perhaps using iterative formulas (see also the attempts of Russel and Nelson, as shown in Appendix A-08).

23

APPENDICES

A-01 – *Values of a # 2*. Now, within the validity limits of expression $\log_a a = 1$, we can establish that all the possible plots of function $z(y) = a \# y$ will pass through points ($y = 0, z = 1$) and ($y = -1, z = 0$). For instance, for $a = 2$ we have:

$2 \# -2$	=	$-\infty$
$2 \# -1$	=	0
$2 \# 0$	=	1
$2 \# 1$	=	2
$2 \# 2$	=	4
$2 \# 3$	=	16
$2 \# 4$	=	65.536
$2 \# 5$	=	$2 \wedge 65.536$
……………………..		

A-02 – *Values of the super-squareroot*. By applying formula (37) we obtain, for example:

$ssqrt\ (1)$	=	1
$ssqrt\ (2)$	=	$1{,}559\ 610\ \ldots$
$ssqrt\ (3)$	=	$1{,}825\ 455\ \ldots$
$ssqrt\ (4)$	=	2
$ssqrt\ (27)$	=	3
$ssqrt\ (100)$	=	$3{,}597\ 285\ \ldots$
$ssqrt\ (1000)$	=	$4{,}555\ 537\ \ldots$
…………		……………..

A-03- *Tetradic representation of large numbers*. Towards 1940, the mathematician Edward Kasner asked his nine-year-old nephew to name an enormous number such as 10^{100}, and his nephew invented the name ***googol***. The same Kasner, sometime later, named 10^{googol} with the name ***googolplex***. We give some examples below of very large numbers represented by using tetration (bases $a = 2$ or $a = 10$), completed by a super-exponent extension p, with $1 < p < a$:

- $65.536 = 2 \# 4$	$= {}^{4}2$
- the Avogadro's constant = $6{,}022 \times 10^{23}$	$= {}^{4}2 * 1{,}409\ 361$
- 1 ***googol*** $= 10^{100}$	$= {}^{4}2 * 1{,}616\ 471$
- max precision of TI-85 and TI-92 calculators = $9{,}999 \times 10^{999}$	$= {}^{4}2 * 1{,}827\ 073$
- $2 \wedge 65.536 = 2 \# 5$	$= {}^{5}2$
- max precision of the "Mathematica" software = $1{,}440 \times 10^{369\ 693\ 099}$	$= {}^{5}2 * 1{,}200\ 088$
- 1 ***googolplex*** $= 10^{googol} = {}^{3}10 * 2$	$= {}^{5}2 * 1{,}617\ 078$
- $4 \$ 2 = 4 \# 4$	$= {}^{5}2 * 1{,}664\ 449$

A-04- *Extension of tetration to negative integers*. Concerning extension of tetration to the negative integers (or to the set of the relative numbers **Z**), let us start by recalling the following definitions (valid for $n \in$ **N**, i.e. positive integer):

$$a \circ a \circ a \circ a \circ \ldots\circ a \ (n\ times) = a + n\ (a\text{-}plus\text{-}n)$$
$$a + a + a + a + \ldots + a\ (n\ times) = a\ .\ n\ (a\text{-}times\text{-}n)$$
$$a\ .\ a\ .\ a\ .\ a\ .\ \ldots\ .\ a\ (n\ times) = a \wedge n\ (a\text{-}power\text{-}n)$$
$$a \wedge a \wedge a \wedge a \wedge \ldots \wedge a\ (n\ times) = a \# n\ (a\text{-}tower\text{-}n)$$

Then, let us consider what happens when n decreases, passing through zero, in one of the hyperoperations (for instance addition, implying, for negative values of n, iterated subtractions):

$$a + a = a\ .\ 2$$
$$a = a\ .\ 1$$
$$a - a = a\ .\ 0$$
$$a - a - a = a\ .\ (-1)$$
$$a - a - a - a = a\ .\ (-2)$$

In other words, when n decreases passing through zero, to formula:

$$a + a + a + a + \ldots + a\ \underline{(n\ times)} = a\ .\ n\ (a\text{-}times\text{-}n) \quad [\text{for } n > 0]$$

we have to substitute formula:

$$a - a - a - a - \ldots - a\ \underline{(2 - n\ times)} = a\ .\ n\ (a\text{-}times\text{-}n) \quad [\text{for } n \leq 0]$$

Similar formulas can be provided for all the ranks, for calculating the extension of hyperoperations to the negative integer values of n. In this case (for $n = 0, -1, -2, \ldots$) we obtain

(67)
$$a - a - a - a - \ldots - a \quad (2 - n\ times\ a) = a\ .\ n\ (a\text{-}times\text{-}n)$$
$$((((a / a) / a) / a) / \ldots) / a \quad (2 - n\ times\ a) = a \wedge n\ (a\text{-}power\text{-}n)$$
$$\log_a (\log_a \ldots \log_a(\log_a a)) \quad (2 - n\ times\ a) = a \# n\ (a\text{-}tower\text{-}n)$$

This is the starting point for any further research in this area, see [5] and [6]. We may also observe that in the first line the parentheses are not necessary and that, due to the "modus operandi" of the respective inverse operations, for the second and third line we must adopt priority *to the left* and *to the right*, respectively. From the third expression in (67), it is easy to calculate the values of a tower operation, for negative super-exponents.

For example, for $a > 1$, we have ($2 - n$ is the number of a's):

(68)
$${}^{0}a = \log_a a = 1 \qquad [n = 0, 2 - n = 2]$$
$${}^{-1}a = \log_a (\log_a a) = \log_a 1 = 0 \qquad [n = -1, 2 - n = 3]$$
$${}^{-2}a = \log_a (\log_a (\log_a a)) = \log_a 0 = -\infty \qquad [n = -2, 2 - n = 4]$$
$${}^{-3}a = \log_a (\log_a (\log_a (\log_a a))) = \log_a (-\infty) \qquad [n = -3, 2 - n = 5]$$

The results of the research, for $n \leq -3$, is given in [5]. In the same paper, examples of crucial relations are also given, e. g.:

$$(:a) \# (+\infty) = :(\sqrt[2]{a}) \qquad \text{[with: } :a = 1/a\text{]}$$

A-05 – *Delta Numbers* . Let us now designate as $\overline{\boxed{S}}$ the operator indicating the inverse of a commutative hyperoperation $\boxed{S}$, with rank s. From equation: $x \boxed{S} a = \varepsilon_s$ [ε_s: unit element of operation $\boxed{S}$]

it follows that: $x = \varepsilon_s \overline{\boxed{S}}\, a$ [x : inverse element of a, in respect of ε_s]

that we could write as: $\boxed{x = \overline{\boxed{S}}\, a}$ [implying the presence of ε_s, before $\boxed{S}$]

For example:	s=2	$x\,.\,a = 1$	=>	$x = 1/a$	$= :a$	$\varepsilon_2 = 1$
	s=1	$x + a = 0$	=>	$x = 0 - a$	$= -a$	$\varepsilon_1 = 0$
	s=0	$x \circ a = -\infty$	=>	$x = -\infty \Delta a$	$= \Delta a$	$\varepsilon_0 = -\infty$

With $a \in \mathbf{R}_+$ (a real > 0), $-a \in \mathbf{R}_-$ (a real < 0), $:a \in \mathbf{R}_f$ ($:a$ real, inverse of a). Similarly, the set of numbers $\{\Delta a\} = \mathbf{R}_\Delta$, where "Δ" stands for the operator of the inverse of *zeration*, can be considered as a new branch of numbers, corresponding to the logarithms of negative numbers. The definitions, axioms and theorems concerning these new "*Delta Numbers*" are presented in [5], where it is also shown that the following formula is valid:

(69) $\boxed{b^{\Delta a} = -(b^a)}$ [with $a, b \in \mathbf{R}$]

In order to justify the relationship between delta numbers and logarithms of negative numbers, which is outside of the scope of the present paper, we should simply remember that [5] also shows that a particular "rule of signs" is verified, concerning the "+" and "Δ" operators:

(70) $\boxed{a + (\Delta b) = \Delta(a + b)}$

similar to: $a \cdot (-b) = -(a \cdot b)$.

Actually, from (69) we have that: $-(b^a) = b^{\Delta a}$

but we know that: $:(b^a) = b^{-a}$ $\qquad (:b^a = 1/b^a)$.

Now, we also know that: $[:(b^a)] \cdot [-(b^a)] = -1$

which implies that: $[b^{-a}] \cdot [b^{\Delta a}] = -1$.

But, remembering (70) we have: $b \wedge [-a + (\Delta a)] = b \wedge [\Delta(-a + a)] = b^{\Delta 0} = -1$

and, therefore: $\boxed{\Delta 0 = \log_b(-1)}$ (with: $b \in \mathbf{R},\ b \neq 0$)

But, we can write: $\log_b(-1) + \log_b(b^a) = \log_b[-(b^a)] = \log_b(b^{\Delta a}) = \Delta a$

which means that: $\Delta 0 + a = \Delta a$

If we now consider that: $\log_b[-(b \wedge \log_b a)] = \log_b[b \wedge (\Delta \log_b a)] = \Delta \log_b a$

we are also allowed to write: $\boxed{\log_b(-a) = \Delta \log_b a}$.

This shows that "*Delta Numbers*" can be put in correspondence with the logarithms of negative numbers, which have multiple complex values.

As we have seen, in expression $z = x \# y = {}^y x = x \uparrow\uparrow y$ (x-tetrated-y) *tetration* is defined only for "super-exponent" $y = n$, with $n > -2$ (for $n = -2$, we have $z \to -\infty$). It has also been demonstrated by (30) that the extension of *tetration* to negative values of n (for $n < -2$) will also involve logarithms of negative numbers and, consequently, might also be approached by using *Delta Numbers*.

A-06. *Asymptotic values of the super-exponential function.* Concerning the super-exponential function:

$$z = x \# y \qquad \text{[with } x = a > 0\text{, constant parameter]}$$

it is important to observe that z is limited or unlimited (for $y \to +\infty$) depending on the constant values given to the argument $x = a$. In fact, from expression (40) of the "*∞-th super-root of* z_∞":

$$x = \sqrt[z_\infty]{z_\infty} = g(z_\infty) = \sqrt[\infty]{z_\infty}$$

we can determine, for each $x = a$, the corresponding asymptotic value of z (called z_∞) and assumed to be obtained as:

$$z_\infty = \lim_{y \to \infty} (a \# y)$$

Therefore, the study of function:

(71) $$x = g(z) = z \wedge (1/z) = \sqrt[z]{z}$$

is of extreme importance. The graph of this function is as follows (Fig. A1). Function $h(x) = w(-\ln x)/(-\ln x)$ is obtained by studying the *Lambert Function* $w(y)$, which is the inverse of function $y = w\,e^{w}$, also called, in the scientific literature, "*ProductLog*" [3][7].

Fig. A2 shows the inverse function (corresponding to *the infinite tower of* x):

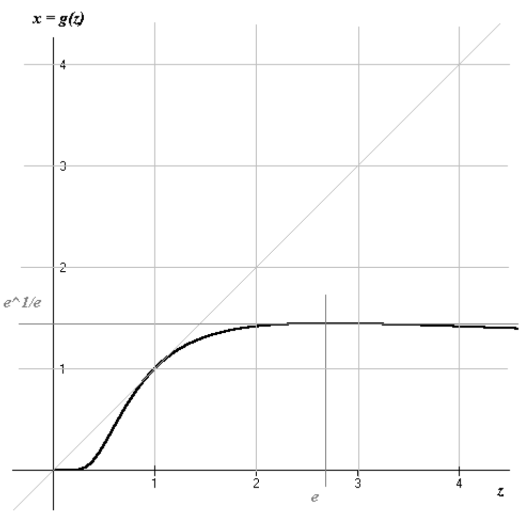
(72) $$z = h(x) = -w(-\ln x)/\ln x = {}^{\infty}x$$

Concerning $g(z)$, we can say that: $\lim_{z \to +0} g(z) = 0$

and that: $\max g(z) = g(e) = e^{1/e} = 1{,}444\ 667\ 861 \ldots$ for: $z = e = 2{,}718\ 281\ 829$ …

and, also, that: $\lim_{z \to +\infty} g(z) = 1$.

From fig A2, we could conclude that the "infinite towers", represented by function $z = h(x) = x \# \infty$, seem to assume limited values only for $0 < x < g(e)$ and that, outside this range of x, the values of $z = x \# \infty$ are unlimited. Function $h(x)$, given by the (71), is the inverse function of $x = g(z) = {}^{\infty}\!\!\!\!\!\!\!\!\!\!\;\;\;\;\;\;)z = \sqrt[z]{z}$, described by (70), for $x < g(e) = \sqrt[e]{e}$.



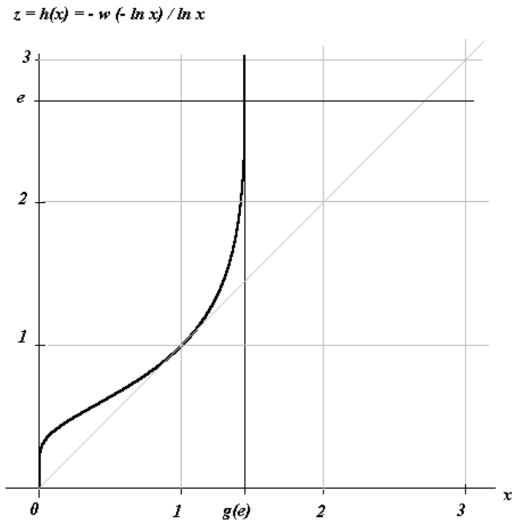


Fig. A1 Fig. A2

This can be demonstrated as follows. Let us consider the following function, that could be called (but there is no general agreement about this) "*Product-Exponential*", since it is obtained as the product of w by e^w:

(73) $y = w(y) \,.\, e^{\,w(y)}$

Its inverse function: $w = w(y)$

Is the "*Product-Logarithm*", "*ProductLog*" or "*Lambert's Function*". From (73) we have then (and let's call by z the result):

(74) $w(y) / y = e^{-w(y)} = z.$

Therefore, from (74), we may write: $1 / z = e^{\,w(y)}$, and : $\ln z = - w(y)$

and, therefore, from (73): $(1 / z) \,.\, \ln z = - w(y) \,.\, e^{\,w(y)} = - y$

and, finally: $\ln (z \wedge (1 / z)) = - y$

i.e.:

(75) $z \wedge (1/z) = \sqrt[z]{z} = e^{-y}$

If now we put : $y = - \ln x$

we obtain, from (75) and (74): $\boxed{x = \sqrt[z]{z} \;;\qquad z = - w(-\ln x) / \ln x}$

Sicut erat demonstrandum].

From the point of view of the graphs of function $z = x \# y$, it should be investigated how, from the plot of fig. A2, a second value of z should suddenly appear, for $1 < x < g(e)$, as it is also expected from a "*graphical inversion*" of the plot of fig. A1 ($x \rightleftarrows z$). Euler [15] and Eisenstein [13] have, however, also shown that all serial developments trying to approximate $h(x) = - w(- \ln x) / \ln x$, converge only for values of x such as:

$g(1/ e) < x < g(e)$

with: $x = g(1/ e) = e^{-e} = 0{,}065\ 988\ 804 \ldots$

for: $z = 1 / e = 0{,}367\ 879\ 441 \ldots$

which are normally considered as the range of convergence of the "infinite towers", i.e. for

$0{,}065\ 988 \ldots. < x < 1{,}444\ 667 \ldots$."

A-07. *"Mathematica" implementations of relevant functions*. In some popular advanced mathematical software packages (for instance *Mathematica* of the Wolfram Research, Inc., software created by Dr Stephen Wolfram in 1987, see [3]), some of the operations and functions mentioned in this paper are already available or can easily be implemented. Already listed among the standard basic operations, we can mention **ProductLog**.

Among those that can be implemented, we can consider the "tower function" ($z = x \# n,\ n \in \mathbf{N}$) (from: [22]):

```
Tower [x_, n]         : = Nest [Power [x, # ] &, 1, n]
```

Another form of implementation could involve the tetration operation, as a function of two variables ($z = x \# y$), which can be given by the following expression (from: [7], p. 2336):

```
Tetration [x_, y_]  : = Fold [Power [x, # ] &, 1, Table [x, (y)]].
```

Concerning the "super-exponential function, base e" ($z = e \# x, x \in \mathbf{N}$), the authors propose a very simple implementation:

```
SuperExp [x_]         : = Nest [Exp, 1, x]
```

Concerning "zeration" ($z = x \circ y$), the autors propose the following implementation (in two lines):

```
Zeration [x_, y_]   : = Max [x, y] + 1  / ;  x !=y
Zeration [x_, y_]   : = Max [x, y] + 2  / ;  x==y.
```

Concerning the super-square-root of a number ($y = {}^{2}\big)x$, $x \in \mathbf{R}$, $x > e^{-e}$), the authors propose:

```
SuperSqrt [x_]     := Log [x] / ProductLog [Log [x]].
```

A-08. *Algorithms for evaluating the superlog*. During a discussion, in the framework of an Internet User Group, in August 1999, Seth Russel and Clifford J. Nelson [23] proposed the following "*Mathematica*" program for evaluating some of what, in this paper, we called the "*right inverse hyperoperations*", particularly for ranks 1, 2 and 3:

```
Clear[p]
p[z_,z_]  :=1
p[0,_]    :=1
p[_,0]    :=1
p[z_,1]   :=z
p[z_,x_]  :=1+p[f[z, x],x]          /; z>=x
p[z_,x_]  :=1/p[x, z]               /; z<x
```

The "key" function is **f[z,x].** According to Russel and Nelson, we have the following situations:

- if: **f[z_,x_]:= z - x** then: **p[z, x] = z / x**
- if: **f[z_,x_]:= z / x** then: **p[z, x] = Log[z] / Log[x] = Log[x, z]**
- if: **f[z_,x_]:= Log[x, z]** then: **p[z, x] = Superlog[x, z].**

Therefore, if f[z,x] is Log[z,x], the program should allow us to calculate the superlog of z, base x. However, at the s=3 rank level, the formula does not normally converge and *Mathematica* requires a continuous increase of the "*MaxExtraprecision*".
NB: Prefix *hyper*-, in this paper, concerns each *hyperoperation* of the hierarchy (zeration, addition, multiplication, exponentiation, tetration, pentation, etc.), disregarding its rank. Prefix *super*- is reserved to the *hyperoperation* of rank s=4 (tetration).

A-09. *Extension of tetration to the rationals*. It is interesting to examine a strategy proposed for estimating the value of expressions such as $z = a \# (1/k)$.

Somebody, for example, suggested that since: $z = a \cdot (1/2) = a / 2$
is the unique solution of: $z + z = z \cdot 2 = a$,
and since everybody agrees that: $z = a \wedge (1/2) = \sqrt[2]{a}$
must be the unique solution of: $z \cdot z = z \wedge 2 = a$,
we should hypothetically suppose that: $z = a \# (1/2)$
could be the unique solution of: $z \wedge z = z \# 2 = a$. <u>(which cannot be taken for granted)</u>
Indeed, expression: $z \wedge z = z \# 2 = a$
actually implies that: $z = {}^{2}\big)a$ = ssqrt a,
but, unfortunately, we also have that: $\boxed{z = a \# (1/2) \neq {}^{2}\big)a = \text{ssqrt } a.}$

In other words, <u>it is not demonstrated</u> that **ssqrt *a*** should be equal to ***a*-tower-(½)** and, actually, this assumption is not acceptable.
In fact, if we consider: $z = a \# (1/n)$ (for $n \in \mathbf{N}$)
and, at the same time, <u>if we hypothetically assume</u> that it should also be:

$$z \# n = a \qquad \Rightarrow \qquad z = {}^{n}\big)a \; .$$

then we should have: $\lim\limits_{n\to\infty}[a \# (1/n)\,] = a \# (1/\infty) = a \# 0 = 1$

but, also: $\lim\limits_{n\to\infty} {}^{n}\big)a = {}^{\infty}\big)a = \sqrt[a]{a}$.

But, since we obviously know that: $\sqrt[a]{a} \neq 1$ for: $a \neq 1$ and $a \neq \infty$.

we must conclude that: $\lim\limits_{n\to\infty}[a \# (1/n)\,] \neq \lim\limits_{n\to\infty} {}^{n}\big)a$

and that, in general, we have: $\boxed{z = a \# (1/n) \neq {}^{n}\big)a}$ (for any $n \in \mathbf{N}$).

Therefore, the above-mentioned <u>hypothetical assumption cannot be accepted</u>. Let us now examine the following table, which shows the hyperoperations of rank s, together with their respective unit elements ε_s , the first (left) inverse operations, or hyperroots, as well as the auto-hyperroots, coinciding with the unit elements (or *unitary functions*, in case for instance of $\sqrt[z]{z} = g(z)$).

s	ε_s	**hyperoperation**	**hyperroot**	**auto-hyperroot**
0	$-\infty$	$z = x \circ n \Rightarrow$	$x = z \circ (-\infty \,\Delta\, n) \quad = z \,\Delta\, n$	$z \,\Delta\, z = -\infty$
1	0	$z = x + n \Rightarrow$	$x = z + (0 - n) \quad = z - n$	$z - z = 0$
2	1	$z = x \cdot n \Rightarrow$	$x = z \cdot (1 / n) \quad = z / n$	$z / z = 1$
3	$\sqrt[n]{n}$	$z = x \wedge n \Rightarrow$	$x = z \wedge (\log_n \sqrt[n]{n}\,) = \sqrt[n]{z}$	$\sqrt[z]{z} = g(z)$
4	${}^{n}\big)n$	$z = x \# n \Rightarrow$	$x = z \# (\text{slog}_n {}^{n}\big)n\,) = {}^{n}\big)z$	${}^{z}\big)z = \gamma(z)$

As stipulated by classical Algebra, the hyperroot expression $x = z \wedge (\log_n \sqrt[n]{n}) = z \wedge 1/n = \sqrt[n]{z}$, for s=3, holds **iff** we admit that we can write $\sqrt[n]{n} = n^{1/n}$. The equivalent (homomorphic) formula, in case of s=4, should then indeed be as follows:

$$\boxed{x = z \,\#\, (\mathrm{slog}_n\, {}^{n}\big)n\,) = {}^{n}\big)z}$$

As we have seen, however, this expression cannot be simplified.

A-10. *Left and right unit elements*. Hyperoperations such as $z = x \boxed{s}\ y$ normally have two different unit elements, which happen to be identical only in the case when the hyperoperation is commutative, e.g. for the ranks s=0, s=1 and s=2.

In general, we have: $z = \varepsilon_s \boxed{s}\ a = a$ with ε_s : <u>left</u> unit element

$z = a \boxed{s}\ \eta_s = a$ with η_s : <u>right</u> unit element

with: ε_s = s-rank, *a*-th hyperroot(*a*) $= \overset{(s)}{{}^{a}\lceil\overline{a}}$, (i.e.: $a \,\Delta\, a,\ a - a,\ a \,/\, a,\ \sqrt[a]{a},\ {}^{a}\big)a$, …)

and: η_s = s-rank, base *a*, hyperlog(*a*) $= \underset{(s)}{{}_{a}\lfloor\underline{a}}$, (i.e.: $a \,\Delta\, a,\ a - a,\ a \,/\, a,\ \log_a a,\ \mathrm{slog}_a a$, …).

The following table shows the values of the left (hyperroot) and right (hyperlog) unit elements (and functions), for the levels of rank s = 0, 1, 2, 3, 4.

s	ε_s	$\varepsilon_s \boxed{s}\ a$	η_s	$a \boxed{s}\ \eta_s$	Comments
0	$-\infty$	$(-\infty)\ \circ\ a = a$	$-\infty$	$a \circ (-\infty) = a$	$\varepsilon_0 = \eta_0$
1	0	$0 + a = a$	0	$a + 0 = a$	$\varepsilon_1 = \eta_1$
2	1	$1 \cdot a = a$	1	$a \cdot 1 = a$	$\varepsilon_2 = \eta_2$
3	$\sqrt[a]{a}$	$\sqrt[a]{a} \wedge a = a$	1	$a \wedge 1 = a$	$\varepsilon_3 \neq \eta_3$
4	${}^{a}\big)a$	${}^{a}\big)a \,\#\, a = a$	1	$a \,\#\, 1 = a$	$\varepsilon_4 \neq \eta_4$

The left and right unit elements are identical ($\varepsilon_s = \eta_s$) for $0 \le s \le 2$, because the respective hyperoperations are commutative. They are different ($\varepsilon_s \neq \eta_s$) for $s \ge 3$, when the hyperoperations are not commutative. Always for $s \ge 3$, ε_s (the unitary function, the *a*-th hyperroot of *a*) is variable and η_s (the hyperlog, base *a*, of *a*) is constant and always equal to 1. This kind of left-right asymmetry also depends on the adoption of the asymmetrical "priority to the right" rule of procedure.

A-11. *Some peculiarities.* Simple numerical calculations using hyperoperations allows us to highlight some interesting peculiarities.

In fact, for s≥2, we always have: $1 \cdot 1 = 1 \wedge 1 = 1 \,\#\, 1 = \ldots\ldots 1 \boxed{s}\ 1 = 1$

Instead, for s≤2, we have :

$1 \cdot 1 = 1$
$1 + 1 = 2$
$1 \circ 1 = 3$ (sic!)

Moreover, we always have : $2 \circ 2 = 2 + 2 = 2 \cdot 2 = 2 \wedge 2 = 2 \,\#\, 2 = \ldots.\ 2 \boxed{s}\ 2 = 4$.

REFERENCES

[1] L. Stockmeyer – A. Chandra, *Les problèmes intrinsequement difficiles*, in *Le Calcul Intensif* , pp. 6-21, Editions Pour la Science, 1989.

[2] A. Meyer, *Weak monadic second order theory of successor is not elementary-recursive*, in "*Lecture Notes in Mathematics*", no. **453**, Springer-Verlag, 1975.

[3] S. Wolfram, *A New Kind of Science*, Wolfram Media, Inc., (II ed.), Champaign, Illinois, USA, pp. 1187, 2002.

[4] D. Hilbert – W. Ackermann, *Grundzüge der theoretischen Logik*, Berlin, 1928.

[5] C. A. Rubtsov, *New mathematical objects*, BelGTASM, Belgorod, Russia; NPP-Informavtosim, Kiev, Ukraine; 1996, Monograph, 251 p. (In Russian). See also: (http://www.geocities.com/rubcov/pdf/book rus.zip), and: (http://numbers.newmail.ru/pdf/book rus.pdf).

[6] C. A. Rubtsov, *Algorithms ingredients in a set of algebraic operations*, Cybernetics **3**, pp. 111-112, 1989. (In Russian).

[7] E. W. Weisstein, *CRC Concise Encyclopedia of Mathematics,* Chapman & Hall / CRC, pp. 3242, 2002

[8] M. A. Anuta – D. W. Lozier – N. Schabanel – P. R. Turner, *Basic Linear Algebra Operations In SLI Arithmetic*, Euro-Par, Vol. II, pp. 193-202, 1996.

[9] C. W. Clenshaw – P. R. Turner, *The symmetric level-index (SLI) system*, IMA J. Numer. Anal. **8**, pp. 517-526, 1988.

[10] R. P. Munafo, *Large Numbers* **-** *Notable Properties of Specific Numbers,* The Internet, Prof. R. P. Munafo's homepage (http://home.earthlink.net/~mrob/pub/math/largenum.html).

[11] D. Hofstadter, *Gödel Escher, Bach, Les Brins d'une Guirlande Eternelle*, Inter Éditions, Paris, pp. 884, 1986.

[12] R. Penrose, *The Emperor's New Mind*, Oxford University Press, New York, 1990.

[13] G. Eisenstein, *Entwicklung von α ^ (α ^ (α ^ (α ^ ....)))*. J. reine angev. Math. **28**, pp 48-52, 1844.

[14] D. E. Knuth, *Mathematics and Computer Science. Coping with Finiteness.* Science **194**, pp. 1235-1242, 1976.

[15] L. Euler, *De serie Lambertina plurimisque eius insignibus proprietatibus.* Acta Acad. Scient. Petropol. **2,** pp. 29-51, 1783.

[16] H. Länger, *An elementary proof of the convergence of iterated exponentiations.* Elem. Math. **51**, pp. 75-77, 1986.

[17] R. A. Knoebel, *Exponentials Reiterated*, The American Mathematical Monthly, **88**, pp. 235-252, 1981.

[18] R. L. Goodstein, *Transfinite ordinals in recursive number theory*, Journal of Symbolic Logic **12**, 1947.

[19] R. Bracewell, *The Impulse Symbol*, Ch. 5, The Fourier Transform and its Applications, 3rd ed., New York,, McGraw-Hill, pp. 69-97, 1999.

[20] S. D. Chatterji - *Proceedings of the International Congress of Mathematicians. (Zürich, August 3-11, 1994).* (In English). Birkhäuser Verlag, Basel, 1995. Vol **1**, **2**, pp. 1605.

[21] G. Korn - T. Korn, *Mathematical Handbook for Scientists and Engineers*, McGraw-Hill Books Co., N. Y., 1968.

[22] Eric W. Weisstein. "Power Tower." From MathWorld--A Wolfram Web Resource. (http://mathworld.wolfram.com/PowerTower.html); © 1999 CRC Press LLC, © 1999-2004 Wolfram Res., Inc.

[23] S. Russel – C. J. Nelson, http://www.math.niu.edu/~rusin/known-math/99/iteratedexp.

The Authors

Constantin A. Rubtsov, Dr Eng. Sci., is a researcher of the State Technological University of Belgorod, Russia (BGTU, BelGTASM until 2003). He is the author of a monograph [5] containing the theoretical foundations of the present work and collaborates with the Cybernetics Institute of Kiev, Ukraine [6]. In 1994, after an assessment of his research works by the Russian Academy of Sciences (RAN) and during the International Congress of Mathematics (ICM-94), his name was included in a list of the world leading mathematicians [20].

Giovanni F. Romerio, Dr Eng. (Politecnico di Torino, Italy), has been involved in the design of information systems and networks, since 1962. He served as developer, researcher, expert and project director in international institutes and organizations, such as EURATOM (Brussels, Belgium), Battelle (Geneva Research Centre, Switzerland), ESRO/ESA (Paris, France), UN-ECA (Addis Ababa, Ethiopia) and UNESCO (Paris, France, and in various field posts, including the management of the "Bibliotheca Alexandrina" project, Alexandria, Egypt).

Acknowledgements

This collaboration between the authors began after a contact via the Internet, when the second author posted a question to a User Group. The first author, researcher in the BGT University in Belgorod, Russia, answered this question with great interest. This first contact was followed by a meeting in Kiev, Ukraine, in October, 2002, at which the plans for this a paper were discussed. The advice of prof. Letterio Gatto, Dept of Mathematics of the "Politecnico di Torino", Turin, Italy, was sought concerning the content of the text. The text itself was also made available for comments to the Battelle Institute, Geneva Research Centre, as well as to Dr Stephen Wolfram, Wolfram Research, Inc., Champaign, Ill., USA. The final text was linguistically revised by Mr. Alan Tulloch, Fontana Studies Center, Saluzzo, Italy, and, concerning the overall presentation, by prof. Ezio Racca (Dr Phys.), Saluzzo, Italy. The Web publication was ensured by the Rotary Club of Saluzzo, District 2030, Italy. The above-mentioned individuals and organizations are warmly thanked.

Constantin A. Rubtsov
Giovanni F. Romerio
Saluzzo (Italy), Belgorod (Russia) – 15th September 2004