Hyperoperator analysis

by Dmitrii V. Guryanov

dmitriigur76@gmail.com

Abstract. The purpose of this article as a continuation of development of the Multiplical concept is to give an answer to the earlier raised question of why the place of the operator in the function **y = e↗^{x}** was taken by the operator - a power tower with left associativity, and not with the generally accepted right associativity (the Tetration). Answering on this question required to conduct an hyperoperator analyze. The hyperoperator nature is considered, definition is made and an alternative way of its development is proposed in the present analysis.

Keyword: Right associativity, Left associativity, Commutativity, Hyperoperator, Hyperoperation, Hyperoperator order, Hyper-root, Hyper-logarithm, Neutral element, Forward operator, Inverse operator, Main inverse hyperoperator, Hyper-function, Forward hyper-functions, Inverse hyper-functions, Flipped inverse hyper-functions, Perfectly monotonous function, Successor, Addition, Multiplication, Exponentiation, Acceleration, Tetration.

The preceding related article: Multiplical concept https://vixra.org/pdf/2205.0150v1.pdf

As we know increasing the hyperoperator order causes an increase of quantity of directions and branches of hyperoperator's further development. In fact, at 4th order there is associativity direction bifurcation. Both of the versions has right to exist, however if it is possible then we should clarify what direction to consider as a general, primary, central and what direction to consider as a specific, secondary, marginal.

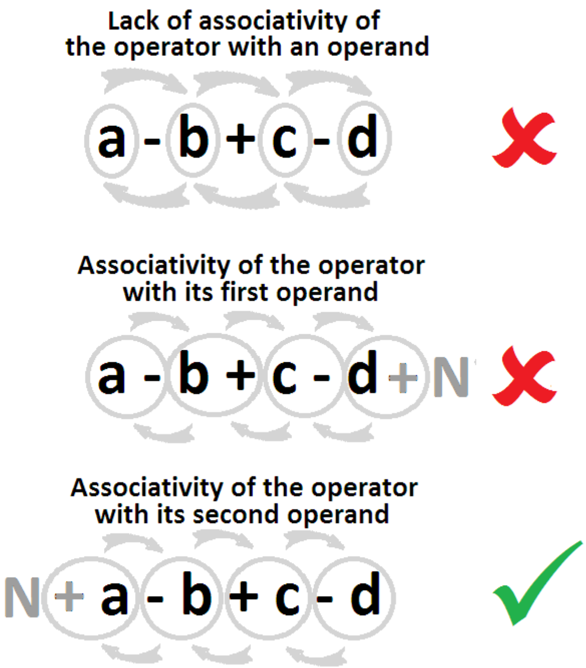
According to the generally accepted definition, a hyperoperator is the repeated execution of operations using one order lower hyperoperator for a sequence of numbers equal to the first operand, and in an amount equal to the second operand. At the same time, nothing is said about such a defining characteristic as the number of operations executed, because it depends on the hyperoperator order. So for summation, the number of successor iterations equals to the value of the second operand, for the higher hyperoperators it is 1 less according to the general idea of binary operators.

The consequence of this is some confusion in the verbal description of the actions performed. For instance, in order to multiply a number by **n** this number must be added to itself n number of times as people say. Well, let's multiply a number by 1, as it is suggested we add the number to the number only 1 time and we get two times of the number. In order to raise a number to power of **n** this number must be multiplied by itself n number of times as people say. Well, let's raise the number to power of 1, as it is suggested we multiply the number by the number only 1 time and we get the square of the number. Our vocabulary implicitly requires the resolution of this contradiction.

Let's presume that in the general formulation, the number does not need to be added to or multiplied by itself, but simply added or multiplied **n** number of times, where **n** is the second operand of the multiplication or the exponentiation operation respectively. In order to bring the sequence into compliance with the new wording, we need to add one more presumably a forward operator to one of the sequence ends,. Since the operators are binary adding another operator we have to finish what has been started and to add some closing quasi-operand or neutral element N to the sequence. Taking into account the obvious fact that the neutral element is not surrounded by two operators, as other operands are, we assume by contrast that the operator and operand that are close to each other, with the exception of the added neutral element, are functionally mutually associated and form one pair.

An answer to the question of what operand the operator is associated with, either the first (the left) or the second (the right) will determine the neutral element position in the sequence, so will it be at the beginning (at the left) or at the ending (at the right).

The generally accepted execution of addition / subtraction and multiplication / division operations in the direction from left to right for a sequence of arbitrarily chosen numbers and in case of presence of operators of both type in the sequence, the forward: the addition or the multiplication, and the inverse: subtraction or division, respectively, predetermines the associativity of the binary operator with the second (the right) operand, because only the associativity presence and only the presence of the associativity of the operator with its second (its right) close operand allows an unlimited rearrangement of pairs (operator and operand) in the sequence without a change of the result (see the illustration). As a consequence the neutral element position is at the sequence beginning (the leftmost). The proposed general approach to a hyperoperator is such that in fact each numeric element of the sequence, excepting the neutral element, is not just a number, but a second operand, an addend or a subtrahend to something, a multiplier or a divisor of something and so no.



The traditional view on an operator commutativity test suggest to do a rearrangement of operands separately and without operators, which is the same as a rearrangement of falsely associative pair of an operator and its first operand breaks its real associative relation with the second operand and therefore causes well known so called non-commutativity of inverse hyperoperators of the first and the second order. In some sense their non-commutativity problem is solved. Their rearrangement is possible, but only as a rearrangement of separate operations (an operator and operand pair) in one piece. We can name this property as conditional commutativity, that is a commutativity with a certain condition.

The pair of an operator and its second operand represent an operation, which can be considered as a capsule, where the operator is responsible for the operation quality property (the operation inner logic that depends on the operator order and the operator direction: the sole forward, and two versions of inverse) and the second operand is responsible for the quantity property of the operation. The operation is a subject of action, and the object is the first operand of the operator. Exactly as it is said, as for example, when we multiply a consisting of myriads of atoms apple by 3, we image 3 separated identical apples as the operation result as consisting of myriads of atoms identical combinations, and not myriads of separated 3-atom molecules each having one respective atom of each of 3 apples, and what could be imagined if we would multiply a 3 by an apple. And this is despite the fact that the overall quantity of atoms in both of variants of the operation result imagination is the same.

From the school desk we all know well that the result doesn't depend on places of operands in the addition and multiplication operations, neither the numerical quantity nor the value measurement. Therefore there is no any use to search for operands' functional differences, so to say, to search for "truth". But I think that for the purpose of conducting a hyperoperator analysis it is needed to be abstracted from this stereotype and to approach critically to happening inside a hyperoperation.

It is obvious that the first operand of each binary operator of a sequence of single-order operators is not just a number, but it is an operation result where this fist operand is the second one in a capsule with its own operator. In order to deal with any first operand in the sequence as with an operation object one have to preliminary execute an operation where this first operator is the second one, and to repeat this moving back to the causes through the chain of operations. Any first operand is the first because it is received as the previously executed operation result in the logic of causes and consequences. The chain of ascent to causes is ceased by the neutral element that leads the sequence and does not have its own operator. On the following example each operation object is placed into a pair of brackets, and each operation subject – the operation itself consisting of operator and its second operand is out of brackets, but highlighted in color (forward operations in red, inverse operations in blue):

$$\left(\left(\left(\left((N \curlyvee a) \curlywedge b\right) \curlyvee c\right) \curlywedge d\right) \curlyvee e\right) \curlywedge f, \quad (49.1)$$

where $\curlywedge$ and $\curlyvee$ - respectively forward and inverse operators of the 1st or the 2nd order.

This is also an argument in favor of performing operations in a sequence of single-order operators, but of any quality (forward and inverse) in the direction from left to right by default as general rule, since, on the contrary, the execution of such operations from right to left, with or without any of the options for associating the operand and operator, does not provide the possibility of rearrangement either operands or operations while maintaining the calculation result for hyperoperator orders up to the second inclusive. Execution from right to left breaks the actual relation between the operator and its second operand, replacing the latter with the result of the previous calculation, which is equivalent to the "anomaly-producing" technique of using parentheses, the expansion of which can cause the operators to be inverted to their opposites. The harmony and beauty of arithmetic operations are violated. In addition, inside the brackets, each individual expression must begin with an operation on a neutral element:

In the case of parentheses, the subjects of a possible permutation are associated pairs consisting of operators and parenthesized expressions as the second operand for their operators. Carrying out permutations of all expressions in brackets along with their associated operators, leads to a visual change in the order of the operands in the sequence to the reverse and seems to be like performing a sequence of operations from left to right, however, this does not save us from the need to open brackets and from the need to perform operations with neutral element, which itself always has the highest execution priority, each time before the execution of the next operation in the sequence:

$$N \curlyvee (N \curlywedge (N \curlyvee (N \curlywedge (N \curlyvee (N \curlywedge \mathrm{f} \curlywedge \mathrm{e}) \curlywedge \mathrm{d}) \curlywedge \mathrm{c}) \curlywedge \mathrm{b}) \curlywedge \mathrm{a}). \quad (49.3)$$

Calculating a sequence consisting of operations of the same order and the same quality (either forward or inverse) is a specific case relative to the case described above, therefore, the result of calculating such a sequence can't be an argument to justify the arbitrariness of the choice or the opposite choice of the direction of performing operations in the sequence in the general case. Here it should be stated that the accepted direction of calculation of a power tower from right to left, which is the basis of the Tetration operator, violates the general rule.

A calculation direction of a sequence of single-order operation (the generally accepted is from left to right) and positions of the 1^{st} and the 2^{nd} operands (as it is generally accepted the 1^{st} is at the left and the 2^{nd} is at the right) are mutually conditioned. So if the calculation would be conducted in the direction from right to left then the right operand would be considered as the 1^{st}, or if the right operator would be the 1^{st} one then the calculation direction would be conducted in the direction from right to left.

Here it is important to follow the uniformity of the direction of calculation for operations of all orders, choosing one (the generally accepted) of two for operators of the lower orders the chosen one should be applied to operators of the higher orders. As it is above mentioned switching the direction means an effective mutual exchange of places for the 1^{st} and the 2^{nd} operands, but since the operands carry completely different functions, the mutual exchange of operand places changes the calculation logic completely. For this reason switching the calculation direction in a process of the transition from lower to higher operator order is not forbidden but requires an explanation of the made decision, a proof of its necessity.

As it is well known two inverse binary hyperoperators exist for one forward binary hyperoperator of n^{th} order, namely those are the hyper-root and the hyper-logarithm. The latter of orders below the 3^{rd} are functionally indistinguishable and represent operators of subtraction and division for the 1^{st} and the 2^{nd} orders respectively. For the higher orders it is possible to methodologically select the primary and the secondary inverse binary hyperoperators out of the respective pair.

The primary inverse binary hyperoperator is that one which operation will bring the result of the previous operation conducted with the forward hyperoperator of the same order to its initial state using the same length interval measured in operator quality units, because the interval length sameness indicates the “180 degree” opposition of used the primary inverse hyperoperator quality unit relative to the quality unit used in forward hyperoperator, and that makes the primary hyperoperator to be an exceptional. Being the operation quantity property the second operand is responsible for the operation interval length, and this means that the second operands of the forward and presumably the primary inverse hyperoperator both have to have the same value. In circumstances of performing operations in the direction from left to right the hyper-root meets this condition and therefore it is accepted as primary inverse binary hyperoperator. Further in the context this inverse hyperoperator is meant as a inverse hyperoperator by the default. Below it is also explained why the separation to the primary and to the secondary inverse hyperoperators is critical in the scope of the conducted hyperoperator analysis.

The hyper-logarithm does not use the quantity property, but returns it as the operation result. The hyper-logarithm could be primary inverse operator in case if in the forward operator operands would exchange their places what exactly happens at the opposite direction of the operations calculation. The power tower calculation from right to left is performed as if the operands of the exponentiation operator exchanged their places however it is false according to the exponentiation operator definition. There is a contradiction. Down below it is demonstrated how the hyper-logarithm plays the primary inverse operator role in case of performing operations in the direction from right to left.

The forward binary hyperoperator is a sequence consisting of equal to the defined hyperoperator second operand absolute value quantity of operations which are performed by binary hyperoperators of one order lower than the defined hyperoperator and started by a neutral element of one order lower than the defined hyperoperator which takes place of the first operand of the first operation of the sequence and where copies of the defined hyperoperator first operand take places of all second operands of those operations and where hyperoperator quality (forward or inverse) depends on the defined hyperoperator second operand arithmetical sign as follows: if the sign is positive then the quality is forward else the quality is inverse. The hyperoperator general definition is following:

$$HO(-1,n,a,b) := HO_{|b|}\Big(S(b), n-1, \ldots HO_2\big(S(b), n-1, HO_1\big(S(b), n-1, N(n-1, a), a\big), a\big) \ldots, a\Big), \quad (50.1)$$

$$HO(-1,n,a,b) := HO_b\Big(1, n-1, \ldots HO_2\big(1, n-1, HO_1\big(1, n-1, N(n-1, a), a\big), a\big) \ldots, a\Big) \text{ at } b \geq 0, \quad (50.2),$$

$$HO(-1,n,a,b) := HO_{-b}\Big(-1, n-1, \ldots HO_2\big(-1, n-1, HO_1\big(-1, n-1, N(n-1, a), a\big), a\big) \ldots, a\Big) \text{ at } b \leq 0, \quad (50.3)$$

$$a^{\uparrow(n)}b = N(n-1, a)^{\uparrow(n-1)}a_1{}^{\uparrow(n-1)}a_2 \ldots {}^{\uparrow(n-1)}a_b \quad \text{at } b \geq 0,\ a_1 = a_2 = a_3 = \ldots = a_n = a \quad (51.1)$$

$$a^{\uparrow(n)}b = N(n-1, a)^{\downarrow(n-1)}a_1{}^{\downarrow(n-1)}a_2 \ldots {}^{\downarrow(n-1)}a_{-b} \quad \text{at } b \leq 0,\ a_1 = a_2 = a_3 = \ldots = a_n = a \quad (51.2)$$

where **HO(d, n, a, b)** – hyperoperator; **d** – hyperoperator quality (forward or inverse), accepted following values: **+1** – for the forward, **–1** – for the inverse; **n** – hyperoperator order, a natural positive number which is above zero; **a** - the **1st** hyperoperator operand; **b** – the **2nd** hyperoperator operand; **N(n, a)** – neutral element of $\mathbf{n^{th}}$ order for number a; **S(x)** - sign function that return **1** in case if **x** is a positive and returns **–1** in case if **x** is a negative; $\mathbf{HO_i(d, n, a, b)_i}$ – hyperoperator of $\mathbf{i^{th}}$ calculating iteration; $\mathbf{a^{\uparrow(n)}b}$ – forward hyperoperator of $\mathbf{n^{th}}$ order; $\mathbf{a^{\downarrow(n)}b}$ - inverse hyperoperator of $\mathbf{n^{th}}$ order.

If the second operand is positive then the forward hyperoperator of the lower order is applied, otherwise the inverse hyperoperator of the lower order is applied. Operations quantity is equal to the second operand absolute value.

The usage of inverse hyperoperators of lower order in the hyperoperator general definition reveals an obvious but critical dependence of the latter to the primary inverse hyperoperator choice and therefore justifies the decision to conserve the sequence calculation direction in the transition from lower to higher orders of the hyperoperators.

In the specific 0^{th} order hyperoperator (successor/predecessor) definition the second operand is ignored (the operator is unary) and the quality of the hyperoperator (forward or inverse) and therefore the operator result is defined by **d** value:

$$HO(d,0,a,b) := a + d. \quad (52)$$

The neutral element is a function of the first sequence operation where the neutral element is the first operand, more precisely this is a function of the operator order **n** and the second operator operand **a**. Therefore in a sequence of single-order operations the 1^{st} operation determines its fist operand – the neutral element:

N(n, a)⮟a⮝b⮟c⮝d.   (53)

In this case, the first operation in the sequence, the one performed on the neutral element, is what we mean by the operation using the unary operator. Thus, we can conclude that the neutral element is a function of a unary operation. The value of the neutral element does not depend on the quality of the operator (forward⮝/inverse⮟).

The neutral element function is defined as a single inverse binary operation of n^{th} order where both operands are value **a**:

$$N(n, a) = HO(-1, n, a, a) \text{ if } n > 0, \quad (54)$$

where **N(n, a)** – neutral element function; **n** –hyperoperator order; **a** – arbitrary real number; **HO(–1, n, a, b)** – inverse binary hyperoperator of $\mathbf{n}^{th}$ order.

It is obvious that a neutral element value does not depend on the operation quality (forward or inverse) and this is fundamental.

Table of neutral element N for number “a” depending on hyperoperation order “n”

Hyperoperation order, n	Forward hyperoperator | Inverse hyperoperator	Operation for searching N	Result of N or Equation for N
0	Successor | Predecessor	-	N = a
1	Addition | Subtraction	a – a	0
2	Multiplication | Division	a / a	1
3	Exponentiation | Root extraction	$\sqrt[a]{a}$	N = $\sqrt[a]{a}$
4	Acceleration | Deceleration	$a \searrow_a$	N = $\sqrt[N^{(a-1)}]{a}$

The exception is the way of getting a neutral element for the zero order hyperoperator (Successor/ Predecessor). Due to the fact those are atomic operators the whole math is built on and also due to the fact those are not a binary operator, not having their second operands which determine the quantity property in general, the neutral element value is just the original number **a** - the sole hyperoperator operand.

It is obvious the hyper-root is used for a neutral element obtaining. If the sequence calculation direction would be from right to left then it would be fairly to use the hyper-logarithm for the purpose and in this case the neutral element would be always 1 for hyperoperator orders above 1. In my opinion this invariant can’t be a reason to use the hyper-logarithm as basic inverse hyperoperator because it doesn’t propagate to hyperoperator orders below 2 therefore is not a general.

Despite the fact the neutral element is an operation function and depends on the operation second operand, nevertheless a neutral element value could be an invariant for a definite or indefinite sequence of single order operations with arbitrary operator quality (forward or inverse) and arbitrary second operands. There is unconditional neutral element invariance for hyperoperator orders up to 2 inclusively. In fact for the 0^{th} order it is the original number **a**, and for the 1^{st} and 2^{nd} orders it’s even a constant (see the table).

If we approach mathematically rigorous, then the result of calculating the neutral element of the first order using zero as an operand has an undefined value, it can be any finite number. If you do not approach strictly and consider the division operator as a ratio, implying that the ratio of two equal values equals to one, then you can take 1 as a neutral element in this case, especially since 1 belongs to the set of numbers of possible results of dividing zero by zero. You can also consider a definition of the neutral element through the limit: The non-rigorous approach has wider practical application, in particular, the postulate used in the definition of the product operator is consistent with it that the product of the zero number of multipliers is equal to 1, otherwise it would be impossible to formulate the multiplical concept.

A consequence of the forward hyperoperator definition is that the latter returns a neutral element value if its second operand **b** equals to 0:

$$HO(1, n, a, 0) = N(n - 1, a). \quad (55)$$

Non-rigorously $0^0 = 1$, since the neutral element of the 2^{nd} order is a constant and equals to 1.

The inverse binary hyperoperator (the hyper-root) is defined recurrently through the solution of the equation to determine the forward hyperoperator, but with a mutual change of places of the returned result of the hyperoperation and its second operand in this equation as follows:

$$x = HO(-1, n, a, b), \quad (56.1)$$

$$a = HO_{|b|}\Big(S(b), n-1, .. HO_2\Big(S(b), n-1, HO_1\big(S(b), n-1, N(n-1, x), x\big), x\Big)\ .. , x\Big) \text{ if } b \neq 0, \quad (56.2)$$

$$a = HO_b\Big(1, n-1, .. HO_2\Big(1, n-1, HO_1\big(1, n-1, N(n-1, x), x\big), x\Big)\ .. , x\Big) \text{ at } b > 0, \quad (56.3)$$

$$a = HO_{-b}\Big(-1, n-1, .. HO_2\Big(-1, n-1, HO_1\big(-1, n-1, N(n-1, x), x\big), x\Big)\ .. , x\Big) \text{ at } b < 0, \quad (56.4)$$

$$x = a^{\downarrow(n)}b, \quad (57.1)$$

$$a = N(n-1, x)^{\uparrow(n-1)}x_1{}^{\uparrow(n-1)}x_2 \ ... \ {}^{\uparrow(n-1)}x_b \quad \text{at } b > 0, \quad x_1 = x_2 = x_3 = .. = x_n = x \qquad (57.2)$$

$$a = N(n-1, x)^{\downarrow(n-1)}x_1{}^{\downarrow(n-1)}x_2 \ ... \ {}^{\downarrow(n-1)}x_{-b} \quad \text{at } b < 0, \quad x_1 = x_2 = x_3 = .. = x_n = x \qquad (57.3)$$

where **x** – pointer to return value of the inverse binary hyperoperator.

The hyper-root also can be expressed compactly through the forward hyperoperator of the same order with substitution in place of the first operand of the forward hyperoperator:

$$a = HO(1, n, HO(-1, n, a, b), b). \quad (58)$$

In general it is not allowed to supply 0 as the second operand to a inverse hyperoperator other than the inverse hyperoperator of the 1st order (the subtraction) . Obviously, as an exception, only zero can be allowed to divide by zero, but rigorously the result of this operation can be any number, as can be seen from the following equation derived from the previous:

$$0 = N(1, x). \quad (59.1)$$

Also, as an exception, it is possible to extract the root of zero degree only from 1, and the result of this operation can also be any number, but except for zero with a rigor, like the previous:

$$1 = N(2, x). \quad (59.2)$$

There is another general recurrent and at the same time explicit solution for the hyper-root, which, on the contrary, is not applicable to defining the operator as a sequence of operations with left associativity in the general case, but is applicable for it as a sequence with right associativity where the hyper-root is not a main inverse hyperoperator:

$$x = HO(-1, n, a, b), \quad (60.1)$$

$$x = HO\big(-1, n-1, a, HO(1, n, x, b-1)\big) \text{ if } n \geq 2, \quad (60.2)$$

In the sequence of single-order operations of mixed quality (forward$\curlywedge$/inverse$\curlyvee$) with identical operands "a" (with an exception of the neutral element as the 1st operand of the 1st operation) the conditional commutativity is possible, which is observed regardless of the hyperoperation order (see fig.). At the same time, this rule checks the correctness of the choice of internal logic of the inverse hyperoperator as opposition of the forward, and what fundamentally distinguishes the inverse hyperoperator (the hyper-root) from an alternative (the hyper-logarithm).

Checking equality of the left and right sequences, depending on the direction of the calculation (the associativity direction):

$$a \curlyvee a \curlywedge a \;=\; a \curlywedge a \curlyvee a, \quad (61.1)$$

$$(\mathrm{a} \curlyvee \mathrm{a}) \curlywedge \mathrm{a} \;=\; (\mathrm{a} \curlywedge \mathrm{a}) \curlyvee \mathrm{a} \Leftrightarrow \left(\sqrt[\mathrm{a}]{\mathrm{a}}\right)^{\mathrm{a}} = \sqrt[\mathrm{a}]{(\mathrm{a}^{\mathrm{a}})} \text{ - left associativity,} \quad (61.2)$$

$$\mathrm{a} \curlyvee (\mathrm{a} \curlywedge \mathrm{a}) \neq \mathrm{a} \curlywedge (\mathrm{a} \curlyvee \mathrm{a}) \;\Leftrightarrow \sqrt[\mathrm{a}]{(\mathrm{a}^{\mathrm{a}})} \neq \mathrm{a}^{\left(\sqrt[\mathrm{a}]{\mathrm{a}}\right)} \text{ – right associativity,} \quad (61.3)$$

where $\curlyvee$ - root extraction, $\curlywedge$ - exponentiation.

But if $\curlyvee$represents a logarithm then the equality is not respected for the left associativity and is respected for the right associativity:

$$(\mathrm{a} \curlyvee \mathrm{a}) \curlywedge \mathrm{a} \;\neq (\mathrm{a} \curlywedge \mathrm{a}) \curlyvee \mathrm{a} \Leftrightarrow (\log_{\mathrm{a}} \mathrm{a})^{\mathrm{a}} \neq \log_{\mathrm{a}}(\mathrm{a}^{\mathrm{a}}) \text{ - left associativity,} \quad (61.4)$$

$$\mathrm{a} \curlyvee (\mathrm{a} \curlywedge \mathrm{a}) = \mathrm{a} \curlywedge (\mathrm{a} \curlyvee \mathrm{a}) \;\Leftrightarrow \log_{\mathrm{a}}(\mathrm{a}^{\mathrm{a}}) = \mathrm{a}^{\log_{\mathrm{a}} \mathrm{a}} \text{ – right associativity.} \quad (61.5)$$

For any hyperoperator order observance of this equality, checked by a permutation of mixed (forward or inverse) operations performed with one operand **a** depends on the choice of the main inverse hyperoperator, which in turn is predetermined by the direction of calculation the sequence.

As a consequence of the method of determining the neutral element the hyperoperator of the 2nd order and above returns its first operand as the result in case if the second operand equals to 1:

$$\mathrm{HO}(d, n, a, 1) = \mathrm{HO}(-d, n, a, 1) = \mathrm{HO}_1\big(1, n-1, \mathrm{N}(n-1, a), a\big) = a \text{ if } n \geq 2. \quad (62)$$

The last statement is valid regardless to the hyperoperator quality (forward or inverse), because interchanged in the hyperoperator definition equation, the result of the hyperoperator and its second operand **a** are equal to each other based on the last equation.

Due to the binaryity of hyperoperators, a remarkable common property of all forward hyperoperators is the return of 4 if both of its operands are equal to 2, which is observed regardless of hyperoperator order, but with the exception of the hyperoperator of the 0th order due to its non-binarity:

$$\mathrm{HO}(1, n, 2, 2) = \mathrm{HO}(1, n-1, 2, 2) = 4 \text{ if } n \geq 1. \quad (63)$$

Another interesting observable property of forward hyperoperators of the 4th and higher orders is that they tend to 1 when the second operand tends to $-\infty$:

$$HO(1, n, a, -\infty) = 1 \text{ if } n \geq 4. \quad (64)$$

Results of inverse hyperoperators of the 4^{th} and higher orders belong to the set of transcendental numbers according to the set definition. In general a inverse hyperoperator result belongs to the set of countable numbers since the set of such numbers is a result of combining numbers of two countable sets to replace the first and second operands of the inverse hyperoperator respectively. According to the hyperoperator order, its result can be attributed to a countable subset of the corresponding order if its same result cannot be obtained by hyperoperators of lower orders. So the integer numbers as result of the subtraction are countable of the 1^{st} order, the rational numbers as result of the division are countable of the 2^{nd} order, the results of the root extracting, while being algebraic, are countable of the 3^{rd} order, the results of the deceleration are countable of the 4^{th} order, and so on. Despite the fact that results of the deceleration and results of the root extracting are irrational and for someone may seem similar, they are different in their essence. Just as a rational number cannot be represented through an integer, and an algebraic irrational cannot be represented as a fraction of rational numbers, the result of the deceleration cannot be represented as an algebraic number root of an algebraic number. As far as integers differ from rational ones, so rational ones differ from algebraic irrational ones, so far the latter differ from the result of the deceleration; they all differ from each other in just one hyperoperator order. But a common property of all countable numbers of the 2^{nd} and higher orders is their non-integrity, which automatically means the practical inaccuracy of the value that algebraically obtained only by means of passage to the limit when the number of recursions of the inverse hyperoperator, with the use of which these numbers are obtained, tends to infinity.

The hyper-logarithm can be expressed recurrently and compactly through the forward hyperoperator:

$$c = HO(1, n, a, HL(n, a, c)), \quad (65)$$

where **HL(n,a,c)** – base **a** hyper-logarithm of **n^th^** order.

As it is seen unlike the solution for the inverse hyperoperator in the solution for the hyper-logarithm, the latter is substituted for the second operand of the forward hyperoperator, which determines its applicability as the main inverse hyperoperator when calculating the sequence from right to left.

It is not possible to express the hyper-logarithm not compactly, namely, in the form of an equation for the sequence of hyperoperations of the lowest order as it is possible for the inverse hyperoperator since such operations quantity cannot be non-integer. This logical obstacle discredits the hyper-logarithm as the alternative main inverse hyperoperator, therefore it discredits the direction of performing operations in sequence from right to left (the right associativity) for which the hyper-logarithm would have to be such.

Table of forward hyperoperator results for different argument values

Hyperoperator own name	Addition/ Subtraction	Multiplication / Division	Raising to power / Root extracting	Acceleration/ Deceleration	The hyperoperator of 5^{th} order	Tetration
Hyperoperator order	1	2	3	4	5	-
HO(1,n,a, $-\infty$)	$-\infty$	$-\infty$	0	1	1	?
HO(1,n,a, 0)	a	0	1	$\sqrt[a]{a}$	$a \searrow_a$	1
HO(1,n,a, 1)	a + 1	a	a	a	a	a
HO(1,n,2, 2)	4	4	4	4	4	4

Hyperoperator synonyms can be defined in order to shorten the notation:

$$H(n, a, b) = HO(1,n,a,b), \quad (66.1)$$

$$HR(n, a, b) = HO(-1,n,a,b). \quad (66.2)$$

The hyper-power function is a hyper-function that represents a single forward binary hyperoperator, where the function argument takes place of the hyperoperator 1^{st} operand and an arbitrary real number takes place of the hyperoperator 2^{nd} operand:

$$y = H(n, x, a). \quad (67.1)$$

The hyper-exponential function is a hyper-function that represents a single forward binary hyperoperator, where the function argument takes place of the hyperoperator 2^{nd} operand and an arbitrary real number takes place of the hyperoperator 1^{st} operand:

$$y = H(n, a, x). \quad (67.2)$$

The hyper-root function is a hyper-function that represents a single inverse binary hyperoperator, where the function argument takes place of the hyperoperator 1^{st} operand and an arbitrary real number takes place of the hyperoperator 2^{nd} operand:

$$y = HR(n, x, a). \quad (67.3)$$

The hyper-logarithmic function is a hyper-function that represents a hyper-logarithm, where the base is an arbitrary positive real number:

$$y = HL(n, a, x). \quad (67.4)$$

A function built on the basis of a binary hyperoperator, but in which an arbitrary constant and a function argument are interchanged as the first and second operands of a binary operator, can be called “flipped” in relation to the function where the mutual change of places is not performed. Strictly speaking, this definition is relative. But with regard to the conducted hyperoperator analysis and the analysis of functions which are built with the use of the hyperoperators, we can agree that we will consider the function flipped, where the place of the first operand - the object of the operation is occupied by an arbitrary constant, and the place of the second operand, which is responsible for the quantitative change of the object of the operation, is occupied by the argument of the function. This way the hyper-exponential function is the flipped hyper-power function. The 1st operand of the hyperoperator - forward with respect to the hyper-logarithm, takes the place of an arbitrary constant in the hyper-logarithm itself, but not the argument of the function, for this reason the hyper-logarithm is not referred to flipped functions.

Having three binary hyperoperators (one forward and two inverse) it is possible to compile three hyper-functions and to compile a one flipped function out of each of them three, all together six hyper-functions of $\mathbf{n}^{\mathbf{th}}$ order can be compiled. Therefore it is possible to define two more hyper-functions except those above mentioned. I don't know generalizing names of those functions, but we can refer them to flipped inverse hyper-functions and name them as follows:

The flipped hyper-root function:

$$y = HR(n, a, x). \quad (67.5)$$

The flipped hyper-logarithmic function:

$$y = HL(n, x, a). \quad (67.6)$$

The hyper-function table for hyperoperator orders from 1st to 3rd

Function group	Generalizing function name	Generalizing definition	Specific definition depending on the order		
			1	2	3
Forward functions	Hyper-power	H(n, x, a)	x + a ⇔ a + x	x · a ⇔ a · x	x^a
	Hyper-exponential	H(n, a, x)			a^x
Inverse functions	Hyper-root	HR(n, x, a)	x – a	x / a	$\sqrt[a]{x} \Leftrightarrow x^{1/a}$
	Hyper-logarithmic	HL(n, a, x)			$\log_a x$
Flipped inverse functions	Flipped hyper-root	HR(n, a, x)	a – x	a / x	$\sqrt[x]{a} \Leftrightarrow a^{1/x}$
	Flipped hyper-logarithmic	HL(n, x, a)			$\log_x a \Leftrightarrow \frac{1}{\log_a x}$

The three groups of hyper-functions are formed by their growth rate characteristic at arbitrary constant above 1 and positive argument and also by the fact that for hyperoperator orders below the 3rd functions of one formed group are functionally identical.

For forward functions their growth rate is not less than the **y=x** growth rate, for inverse functions their growth rate is not higher than the **y=x** growth rate, and for flipped inverse functions their growth is negative.

All hyper-function are different in their math essence, however for the 1st and 2nd hyperoperator orders functions that belongs to one group are functionally identical and return identical results for respective orders. Also the forward hyper-function of the 1st and 2nd orders are identical to the inverse hyper-functions of respective orders with the only difference that for inverse hyper-functions an arbitrary constant is opposite to that for forward hyper-functions with respect to the value of the neutral element for the corresponding order.

Despite the fact that all hyper-functions of the 1st and 2nd orders, except for the function **y=a/x**, are linear, I did not combine them into one group, since they have essential differences.

Table of returning values of the hyper-exponential function in base 2: y = H(n, 2, x) for different hyperoperator orders "n" and argument values "x"

N	X							
	−∞	**−2**	**−1**	**0**	**1**	**2**	**3**	**4**
1	−∞	0	1	2	3	4	5	6
2	−∞	-4	-2	0	2	4	6	8
3	0	¼	½	1	2	4	8	16
4	1	$\sqrt[2]{\sqrt[4]{2}} = \sqrt[8]{2}$ 1.09050773	$\sqrt[2]{\sqrt[2]{2}} = \sqrt[4]{2}$ 1.18920712	$\sqrt[2]{2}$ 1.41421356	2	4	$4^2 = 2^4 = 16$	$16^2 = 2^8 = 256$
5	1	$y = \sqrt[y]{z = \sqrt[z]{a = \sqrt[a]{2}}}$ 1.28486975	$y = \sqrt[y]{z = \sqrt[z]{2}}$ 1.37997040	$y = \sqrt[y]{2}$ 1.55961047	2	4	$4^4 = 2^8 = 256$	$256^{256} = 2^{2048}$
Tetration	?	−∞	0	1	2	4	$2^4 = 16$	$2^{16} = 65536$

With argument growth step equal to 1 each next value of the hyper-exponential function in base 2 is 2 raised to power of the previous function value for the 4th hyperoperator order, is the previous function value raised to power of the previous function value for the 5th hyperoperator order, and is the previous function value raised to power of 2 for the Tetration based hyper-function (see table). Based on this pattern, and by the criterion of growth rate I would attribute the Tetration to an order of 4½ as growing faster than the 4th order hyper-function, and slower than the 5th order hyper-function.

An interesting pattern of hyper-exponential functions in base 2 is also observed. The value of the hyper-function with an argument equal to 4 is equal to the value with an argument equal to 3, but for a hyper-function of one order up than the one under consideration, which is valid starting from the first order of the hyperoperator:

$$H(n, 2, 4) = H(n + 1, 2, 3) \text{ if } n \geq 1. \quad (68)$$

There is such a family of analytically defined functions that not the functions themselves, not one of their derivatives (the first, the second, and further without a restriction) do not have extrema and points of discontinuity for any values of the argument within the boundaries of its definition. For them, the values of functions and all their derivatives are either equal to zero, or are constant, or only decrease, or only increase. Unfortunately, I do not know the name of the family of such functions and if there is no known name then it could be a family of **perfectly monotonous functions**. This family includes hyper-exponential, hyper-logarithmic (the list is inclusive). The remaining analytically defined functions, by definition, have this characteristic in the interval of the argument between their own points of discontinuity and extremum points and such points for all their derivatives. These functions include all hyperoperator based functions, elliptical, trigonometric, inverse trigonometric, hyperbolic and many other functions. In the latter it is appropriate to talk about perfectly monotonous function scopes.

Different from all the other hyper-exponential functions in base 2, the hyper-exponential function built on the basis of Tetration passes through points: **(–1,0)**; **(0,1)**; **(1,2)** (see table) which a straight line can be drawn through. Consequently, in the interval of the argument from –1 to 1, the hypothetical hyper-exponential generalization function for a non-integer argument built on the basis of the operator Tetration has extrema of the derivative, so the function is not perfectly monotonous. Also the function obviously has a break at argument value of -2, and it is not known whether it has a definition for its argument to the left of this point. These circumstances are signs that this function is not hyper-exponential. Moreover, it has coincidences with two other hyper-exponential functions at three points of the argument: 0,1,2 with the hyper-exponential function of 3rd order – the Exponent and 1,2,3 with the hyper-exponential function of 4th order - the Accelent (see table), which would be an anomaly for a hyper-exponential function. At the same time, there are no three points of argument at which any hyper-exponential function built on basis of hyperoperator with left associativity, including Exponent and Accelent, would coincide at any combinations of power base applied and especially for Exponent and Accelent arbitrary multiplier applied for each of functions separately. All this discredits the right associativity with the use of which the Tetration operator is defined.

Computing the power tower from right to left is tantamount to applying parentheses that separate the second operand from its operator in sequence, which leads to an "anomaly" in the result, where normal result means the result obtained according to the definition for the hyperoperator, that is, with the so-called left associativity or the direction of evaluation from left to right.

I do not know what argument exactly determined the choice of right associativity in the Tetration operator. If the reason was the presumption that a higher-order operator should not be compactly described in terms of lower-order operators, that convenience of compact notation as a reason we introduce new operators into use, then I can comment that objective reality exists independently on our subjective idea of it, and mathematics exists independently on our way of describing it. Apparently, the hyperoperator of the third order – the raising to a power - is a rather “powerful” tool that has a special feature - the ability to compactly describe an operator of one order higher than itself. By the way, in English terminology it is called “power”. But apparently the latter was not taken into account. On my opinion, the hyperoperator of the 5th order can no longer be described compactly in terms of the exponentiation. Obviously, here comes the limit of the "power" of the latter.

Arguments of choice the sequence computation direction in the hyperoperator definition

<table>
<tr><th>From left to right</th><th>From right to left</th></tr>
<tr><td>Geometrical growth of presumable hyper-exponential function of the 4th order formulated on a basis of the 4th order hyperoperator defined as a power tower with the left associativity coincides with the function itself (argument of necessity)</td><td rowspan="2">For the order of hyperoperators above the 1st, the neutral element obtained by means of the hyper-logarithm as the main inverse hyperoperator for the direction of calculation from right to left is equal to one. The argument is dubious, because it simultaneously points to imperfection</td></tr>
<tr><td>Proven left-to-right directionality for hyperoperators of order below the third, and consistency in the choice of the computation direction when moving from lower to higher orders of hyperoperators, without the need to redefine them (argument of consistency)</td></tr>
<tr><td>Similarly to the forward hyperoperator and being the main inverse hyperoperator for the direction of calculation from left to right, the hyper-root through its second operand receives a quantitative characteristic of the operation, which results in the possibility of obtaining a hyper-root by means of a recursive equation consisting of a sequence of lower-order hyperoperators, which is impossible for a hyper-logarithm (argument of preservation of hyperoperators properties)</td><td>The impossibility of compact notation for a higher-order operator using lower-order operators as the basis for introducing a next-order operator. (argument of convenience)</td></tr>
<tr><td>Hyper-exponential functions of different orders, formulated on a basis of forward hyperoperators defined with left associativity, never coincide with each other at three points of the argument. They are perfectly monotonic functions over the entire interval of the argument definition (argument of perfection).</td><td>The existence of explicit recurrent solution for the hyper-root which is however not the main inverse hyperoperator for the right associativity</td></tr>
</table>

It is not proposed herein to change the existing power tower calculation rule implying right associativity, therefore it is not proposed to redefine Tetration, Pentation, Hexation, etc. But it is proposed to recognize the legitimacy of the hyperoperator definition as a sequence of operations with left associativity, with the status of the main hyperoperator definition assigned to it, and to separate the hyperoperators thus obtained into a separate set. It is also proposed to use own names of hyperoperators defined according to the proposed rule. So for the 4th order they are "Acceleration" and "Deceleration" as forward and inverse hyperoperators, respectively.

In order to distinguish the above-described hyperoperators defined with left associativity from those defined with right associativity, it is proposed to introduce a separate general shortened arrow notation for hyperoperators defined with left associativity, orders higher than 3rd, as follows from the table below.

Arrow notation of hyperoperators defined using left associativity

Hyperoperator quality	Hyperoperator order, n		
	4	5	6
H(n,a,b)	$a\nearrow^{b}$	$a\nearrow\nearrow^{b}$	$a\nearrow\nearrow\nearrow^{b}$
HR(n,a,b)	$a\searrow_{b}$	$a\searrow\searrow_{b}$	$a\searrow\searrow\searrow_{b}$
HL(n,a,c)	$c\nwarrow_{a}$	$c\nwarrow\nwarrow_{a}$	$c\nwarrow\nwarrow\nwarrow_{a}$

Sequences of operators are composed and calculated from left to right with the priority of computing a sequence of operations of a higher order over a sequence of operations of a lower order and with equal priority of operators of the same order, regardless of their quality. There are 3 equivalent entries of one example down below:

$$\mathrm{e} \nearrow\nearrow^{a\nearrow\nearrow\nearrow^{\mathrm{b}}\searrow\searrow\searrow_{\mathrm{c}}\nwarrow\nwarrow\nwarrow_{\mathrm{d}}} \searrow_{\mathrm{f}\nwarrow\nwarrow_{\mathrm{g}\nearrow\nearrow\nearrow^{\mathrm{i}}}} \nearrow^{\mathrm{j}\nearrow\nearrow^{\mathrm{k}}} \qquad (69.1)$$

$$\left(\mathrm{e} \nearrow\nearrow^{\left(\left(\left(a\nearrow\nearrow\nearrow^{\mathrm{b}}\right)\searrow\searrow\searrow_{\mathrm{c}}\right)\nwarrow\nwarrow\nwarrow_{\mathrm{d}}\right)}\right) \searrow_{\left(\mathrm{f}\nwarrow\nwarrow_{\left(\mathrm{g}\nearrow\nearrow\nearrow^{\mathrm{i}}\right)}\right)} \nearrow^{\left(\mathrm{j}\nearrow\nearrow^{\mathrm{k}}\right)} \qquad (69.2)$$

$$\mathrm{H}\Big(4, \mathrm{HR}\Big(4, \mathrm{H}\big(5, \mathrm{e}, \mathrm{HL}\big(6, \mathrm{HR}(6, \mathrm{H}(6, a, b), c), d\big)\big), \mathrm{HL}\big(5, \mathrm{H}(6, g, i), f\big)\Big), \mathrm{H}(5, j, k)\Big),$$

(69.3)

The unary operator is essentially a binary operator with a neutral element as the first operand - the object of the operation, since there is an obvious circumstance: the operation cannot be performed on anything. If a direct operation is meant, then for the sake of brevity, it is customary not to indicate this operator, which, without assessing the correctness of the rule itself, probably caused the aforementioned confusion in the definition of the hyperoperator. If the inverse operator is meant, then it must be indicated, but again, for the sake of brevity, if possible, without indicating the neutral element. However, since the inverse second-order unary hyperoperator (unary division operator) is not used, one has to write its full binary form with a neutral element: **1/a**. I personally had to explicitly use this form of notation, doing so for the sake of readability of the program code, when in set of code rows it is required to follow the sequence of certain operations without regard to their quality (quality of forward or inverse action) but keeping the same order of operands. As a result of this technique, the expression looked like this: **y=1/a*b/c**, although in my opinion the expression in the form **y=/a*b/c** is also well readable and intuitive as everyone understands which neutral element is meant by default in a factorial case. Also, a stand-alone **/a** or **:a** could be read as the reciprocal of **a**, in the same way that **–a** is considered as the opposite of **a**.

References:

[1] Hyperoperation https://en.wikipedia.org/wiki/Hyperoperation

[2] Successor_function https://en.wikipedia.org/wiki/Successor_function

[3] Addition https://en.wikipedia.org/wiki/Addition

[4] Multiplication https://en.wikipedia.org/wiki/Multiplication

[5] Exponentiation https://en.wikipedia.org/wiki/Exponentiation

[6] Tetration https://en.wikipedia.org/wiki/Tetration

[2] Dmitrii V. Guryanov, Multiplical concept https://vixra.org/pdf/2205.0150v1.pdf