Reihenalgebra: What comes beyond exponentiation?

M. Müller, mueller@math.tu-berlin.de

Abstract

Addition, multiplication and exponentiation are classical operations, successively defined by iteration. Continuing the iteration process, one gets an infinite hierarchy of higher-order operations, the first one sometimes called *tetration*

$$a \uparrow b = \left. a^{a^{a^{a\cdots}}} \right\} b \text{ terms} ,$$

followed by pentation, hexation, etc. This paper gives a survey on some ideas, definitions and methods that the author has developed as a young pupil for the German *Jugend forscht* science fair. It is meant to be a collection of ideas, rather than a serious formal paper.

In particular, a definition for negative integer exponents b is given for all higher-order operations, and a method is proposed that gives a very natural (but non-trivial) interpolation to *real* (and even complex) integers b for pentation and hexation and many other operations. It is an open question if this method can also be applied to tetration.

1 Introduction

Multiplication of natural numbers is nothing but repeated addition,

$$\underbrace{a + a + a + \ldots + a}_{b \text{ terms}} = a \cdot b \ . \tag{1}$$

Iterating multiplication, one gets another operation, namely exponentiation:

$$\underbrace{a \cdot a \cdot a \cdot \ldots \cdot a}_{b \text{ terms}} = a^b =: a\hat{\ }b \ . \tag{2}$$

Classically, this is it, and one stops here. But what if one continues the iteration process? One could define something like

$$\underbrace{a\hat{\ }a\hat{\ }a\hat{\ }\ldots\hat{\ }a}_{b \text{ terms}} = a \uparrow b \ .$$

But, wait a minute, in contrast to eq. (1) and (6), this definition will depend on the way we set brackets, i.e. on the order of exponentiation! Thus, we have to distinguish between the two canonical possibilities

$$a \uparrow b := \underbrace{a\hat{\ }(\ldots\hat{\ }(a\hat{\ }(a\hat{\ }a)))}_{b \text{ terms}} = a^{a^{a^{a\cdots}}} \tag{3}$$

and

$$a \downarrow b := \underbrace{(((a\hat{\ }a)\hat{\ }a)\hat{\ }\ldots)\hat{\ }a}_{b\text{ terms}} = ((a^a)^a)^{a\cdots} \quad . \tag{4}$$

The "power tower" $a \uparrow b$ is sometimes called "tetration". It is more interesting than the operation $a \downarrow b$, because we can simplify the latter to $a \downarrow b = a^{\left(a^{b-1}\right)}$.

Why do we only consider the two possibilities (3) and (4), and not other orders of exponentiation where the brackets are set in a more random order? The reason for this is that (3) and (4) have very simple *recursion laws*, namely

$$a \uparrow (b+1) = a^{a \uparrow b} \quad \text{and} \quad a \downarrow (b+1) = (a \downarrow b)^a \quad .$$

While $a \downarrow b$ seems so uninteresting that we might forget about it, we should remember it for consistency reasons - we may well still continue the iteration process, and then, the corresponding $\downarrow$ operation will no more be uninteresting. The next step beyond tetration involves four different possible operations:

$$\begin{aligned}
\underbrace{(((a \uparrow a) \uparrow a) \uparrow a) \uparrow \ldots \uparrow a}_{b\text{ terms}} &=: \quad a \downarrow\uparrow b \ , \\
\underbrace{a \uparrow \ldots \uparrow (a \uparrow (a \uparrow (a \uparrow a)))}_{b\text{ terms}} &=: \quad a \uparrow\uparrow b \ , \\
\underbrace{(((a \downarrow a) \downarrow a) \downarrow a) \downarrow \ldots \downarrow a}_{b\text{ terms}} &=: \quad a \downarrow\downarrow b \ , \\
\underbrace{a \downarrow \ldots \downarrow (a \downarrow (a \downarrow (a \downarrow a)))}_{b\text{ terms}} &=: \quad a \uparrow\downarrow b \ .
\end{aligned}$$

2 Some examples

Exponentiation with one always gives the identity,

$$a \cdot 1 = a^1 = a \uparrow 1 = a \downarrow 1 = a \uparrow\uparrow 1 = \ldots = a \ .$$

Similarly,

$$2 + 2 = 2 \cdot 2 = 2^2 = 2 \uparrow 2 = 2 \downarrow 2 = 2 \uparrow\uparrow 2 = \ldots = 4 \ .$$

A less trivial example is

$$\begin{aligned}
2 \downarrow\uparrow 3 &= (2 \uparrow 2) \uparrow 2 = 4 \uparrow 2 = 4^4 = 256 \ , \\
2 \uparrow\uparrow 3 &= 2 \uparrow (2 \uparrow 2) = 2 \uparrow 4 = 2^{2^{2^2}} = 65536 \ , \\
2 \downarrow\downarrow\uparrow 3 &= (2 \downarrow\uparrow 2) \downarrow\uparrow 2 = 4 \downarrow\uparrow 2 = 4 \uparrow 4 = 4^{4^{4^4}} \\
&= 2.36 \cdot 10^{\left(\begin{array}{l} 8072304726028225379382630397085399030071367921738 74 \\ 3031867082828418414481568309149198911814701229483451 \\ 9815575747711564964572385352990874812449902613511 16 \end{array}\right)}
\end{aligned}$$

As one can see, the numbers soon get very, very large. One of the largest numbers that have ever shown up in a mathematical proof is Graham's number. It can easily be majorized by

$$\text{Graham's number} < 3 \uparrow\uparrow 129\ .$$

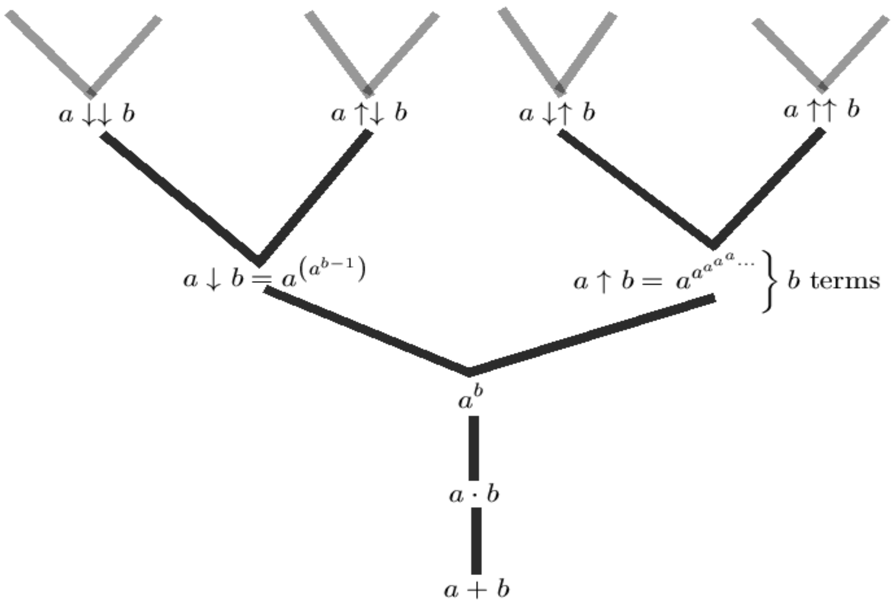

Figure 1: The tree of operations

As shown in Figure 1, left and right arrow generate an infinite family of operations beyond exponentiation. While the two different recursion laws (5) are identical for $\odot = +$ and $\odot = \cdot$, they are different for higher-order operations. This is why the number of operations is doubling every step above exponentiation.

3 General Definition for Positive Integer Exponents

To state the general definition in a simple way, we use the conventions

$$\uparrow\,\hat{}\;:=\;\uparrow \quad \text{and} \quad \downarrow\,\hat{}\;:=\;\downarrow\ .$$

We propose the following general definition:

Definition 3.1 (Hyperexponentiation)
If $\odot$ is any combination of up- and downarrows or (classical) exponentiation, i.e. $\odot \in \{\hat{}\} \cup \bigcup_{n=1}^{\infty}\{\uparrow,\downarrow\}^n$, then we set

$$a \uparrow \odot 1 = a \downarrow \odot 1 := a\ ,$$

and recursively

$$\begin{aligned} a \uparrow \odot (b+1) &:= a \odot (a \uparrow \odot b) \\ a \downarrow \odot (b+1) &:= (a \downarrow \odot b) \odot a\ . \end{aligned} \qquad (5)$$

4 Negative Integer Exponents

It is straight-forward to use equation (5) iteratively to compute values of tetration and other hyperexponentials at negative integer arguments. For example, since

$$a^{a\uparrow 0} = a \uparrow 1 = a\ ,$$

we have

$$a \uparrow 0 = \log_a a = 1\ .$$

Similarly, since

$$a^{a\uparrow -1} = a \uparrow 0 = 1\ ,$$

we get (up to a choice of the branch of the complex logarithm)

$$a \uparrow -1 = \log_a 1 = 0\ .$$

Definition 4.1 (Negative Integer Exponents)
Suppose that $0 \geq b \in \mathbb{Z}$. Then, for every $\odot \in \{\hat{}\} \cup \bigcup_{n=1}^{\infty}\{\uparrow,\downarrow\}^n$, we define

$$\begin{aligned} a \uparrow \odot b &:= \{z \in D : \underbrace{a \odot (a \odot (a \odot (\ldots a \odot z)))}_{1-b \text{ times } a} = a\}\ , \\ a \downarrow \odot b &:= \{z \in D : \underbrace{(((z \odot a) \odot \ldots \odot a) \odot a) \odot a}_{1-b \text{ times } a} = a\}\ , \end{aligned}$$

where $D \subset \mathbb{C}$ is the (arbitrary) domain of definition.

So $a \odot b$ is defined to be a *set* if $b < 0$ is a negative integer. If the set has only one element, we identify the set with its element.

To give another interesting example (which also prepares the later treatment of fractional exponents), we introduce a higher-order-analogue to the square root. Recall that the square root $x = \sqrt{a}$ for $a > 0$ is defined to be the only positive solution to $x^2 = a$, or to

$$x \cdot x = a \ .$$

Similarly, since the function $x \mapsto x^x$ is increasing for $x \geq \frac{1}{e}$ (where $e = 2.71828\ldots$ is Euler's number), we can define the *hyper square root* $x = \mathrm{hsqrt}_2(a)$ to be the only solution $x > e^{-\frac{1}{e}}$ to $x \uparrow 2 = a$, or to

$$x^x = a \ .$$

It is clear how to define $\mathrm{hsqrt}_n(a)$ for other integers $n \in \mathbb{N}$.

Let $D := [1, \infty)$ be our domain of definition. Since we have

$$(2 \downarrow\uparrow 0) \uparrow 2 = 2 \downarrow\uparrow 1 = 2 \ ,$$

it must hold that

$$2 \downarrow\uparrow 0 = \mathrm{hsqrt}_2(2) = 1.559610469\ldots$$

Analogously, it holds that

$$(2 \downarrow\uparrow -1) \uparrow 2 = 2 \downarrow\uparrow 0 \ ,$$

so it follows that

$$2 \downarrow\uparrow -1 = \mathrm{hsqrt}_2(\mathrm{hsqrt}_2(2)) = 1.3799703966\ldots$$

Thus, we can compute the *binary pentation function* $\mathfrak{P}_2(x) := 2 \downarrow\uparrow x$ (which is the same as $2 \downarrow\downarrow x$ due to the basis 2) at negative integer x by iteration of the hyper square root hsqrt_2. Summarizingly,

$$\begin{aligned} \mathfrak{P}_2(1) &= 2 \ , \\ \mathfrak{P}_2(x+1) &= \mathfrak{P}_2(x)^{\mathfrak{P}_2(x)} \ , \\ \mathfrak{P}_2(x-1) &= \mathrm{hsqrt}_2(\mathfrak{P}_2(x)) \ . \end{aligned}$$

It is interesting to note that, while $\mathfrak{P}_2(x)$ grows unbelievably quickly for $x \to \infty$, it holds that

$$\lim_{\mathbb{Z} \ni n \to -\infty} \mathfrak{P}_2(n) = 1 \ .$$

We are now going to evaluate this function at non-integer arguments.

5 Non-integer exponents: A General Interpolation Method

It is a very difficult task to find a natural definition for non-integer exponents in full generality for all higher-order operations beyond exponentiation. In fact, it is already amazingly difficult to do it for tetration, i.e. to define $a \uparrow x$ for $x \notin \mathbb{Z}$. Of course, it is possible to find an arbitrary interpolation, even one that is continuous, by letting, say, $2 \uparrow x := 2x$ for $x \in [1, 2]$, and then iteratively for every other $x \in \mathbb{R}$ via $2 \uparrow (x+1) = 2^{2 \uparrow x}$. But such a definition looks ugly (also the graph of this function does), and seems arbitrary.

Shouldn't there be a single natural definition that is in some sense "the" correct one, as it is for exponentiation? The definition $a^{\frac{1}{2}} := \sqrt{a}$ obviously is the "right" one. Why? Because it respects the identity

$$a^{b+c} = a^b a^c \ , \tag{6}$$

an identity that is first proven for integer b, c, and carries it over to the case of fractional exponents. Therefore, almost everybody who thinks about defining tetration for non-integer exponents first comes to the idea to use

$$a \uparrow \frac{1}{2} \overset{?}{:=} \mathrm{hsqrt}_2(a) \ .$$

Unfortunately, this definition cannot be justified, since there is *no* identity comparable to (6) for tetration (still, the graph of this interpolation looks ugly and does not seem to be differentiable).

Let us reformulate (6) in a more general setting. We can say that "adding $\frac{1}{2}$ to the exponent multiplies the expression with some constant", i.e.

$$\exists c : a^{b+\frac{1}{2}} = a^b \cdot c \ .$$

Already from this assertion, we can deduce that $a^{\frac{1}{2}} = \sqrt{a}$, since $a \cdot a^b = a^{b+\frac{1}{2}+\frac{1}{2}} = a^b \cdot c \cdot c$, so $c = \sqrt{a}$ and $a^{0+\frac{1}{2}} = \sqrt{a}$. We can still reformulate (6) by fixing a, and defining the function $b \mapsto c(b)$ by the equation

$$a^{b+\frac{1}{2}} = a^b \cdot c(b) \ . \tag{7}$$

Knowing that $c(b) \equiv c$ is constant again yields $a^{\frac{1}{2}} = \sqrt{a}$. Equation (7) can easily be extended to all higher-order operations. Fix a, and define $c(b)$ by

$$\begin{aligned} a \uparrow \odot \left(b + \frac{1}{2}\right) &= c(b) \odot (a \uparrow \odot b) \ , \\ a \downarrow \odot \left(b + \frac{1}{2}\right) &= (a \downarrow \odot b) \odot c(b) \ . \end{aligned}$$

If $c(b)$ is constant, then we can use these equations to compute $a \uparrow \odot b$ or $a \downarrow \odot b$ at half-integer arguments. Unfortunately, $c(b)$ cannot be constant in general. For example, assuming that $c(b) = c$ is constant for tetration with $a = 2$, we have on the one hand

$$c^{\left(c^{2\uparrow 1}\right)} = 2 \uparrow 2 \ ,$$

so $c = 1.6569\ldots$, but on the other hand

$$c^{\left(c^{2\uparrow 2}\right)} = 2 \uparrow 3 \ ,$$

so $c = 1.5729\ldots$, which is a contradiction. Thus, $c(b)$ cannot be constant. But maybe we can require it to be *decreasing*? In this case, it would follow that $c(1) > 1.6569\ldots$, and thus

$$2 \uparrow 1.5 > (1.6569\ldots)^2 = 2.745368\ldots$$

and since $c(1.5) > 1.5729\ldots$, it follows that

$$2 \uparrow 1.5 < \log_{1.5729\ldots}(2 \uparrow 2) = 3.0606\ldots$$

Unfortunately, there is no chance to get more information on $2 \uparrow 1.5$ from this method[1].

It is an interesting coincidence that the same idea gives a chance to compute pentation $\downarrow\downarrow$ and hexation $\downarrow\downarrow\downarrow$ at fractional exponents! We will illustrate it for the binary pentation function $\mathfrak{P}_2(x) = 2 \downarrow\downarrow x$.

First, we define $c(b)$ by

$$2 \downarrow\downarrow \left(b + \frac{1}{2}\right) = (2 \downarrow\downarrow b) \downarrow c(b) \ .$$

Recall that $x \downarrow y = x^{\left(x^{y-1}\right)}$. Again, it turns out that assuming $c(b)$ to be constant leads to a contradiction. But we can assume that $c(b)$ *is decreasing*! In particular, we require that $c\left(n + \frac{1}{2}\right) < c(n)$ for every n. Suppose $n \in \mathbb{Z}$. It follows that

$$2 \downarrow\downarrow (n+1) = 2 \downarrow\downarrow \left(n + \frac{1}{2}\right) \downarrow c\left(n + \frac{1}{2}\right) = 2 \downarrow\downarrow n \downarrow c(n) \downarrow c\left(n + \frac{1}{2}\right) \ .$$

Thus, if we define $\tilde{c}(n)$ as the solution of

$$2 \downarrow\downarrow (n+1) = 2 \downarrow\downarrow n \downarrow \tilde{c}(n) \downarrow \tilde{c}(n) \ ,$$

we get $c(n) > \tilde{c}(n)$, or

$$2 \downarrow\downarrow \left(n + \frac{1}{2}\right) > 2 \downarrow\downarrow n \downarrow \tilde{c}(n) \ . \qquad (8)$$

[1] The following ideas do *not* help: estimate c(3) or c(4) and iterate backwards; try smaller steps than $\frac{1}{2}$ and stick them together.

On the other hand, the monotonicity of $c(n)$ implies that $c(n+1) < \tilde{c}(n)$, so we get

$$2 \downarrow\downarrow \left(n+1+\frac{1}{2}\right) = 2 \downarrow\downarrow (n+1) \downarrow c(n+1) < 2 \downarrow\downarrow (n+1) \downarrow \tilde{c}(n) \ ,$$

and by iterating backwards, i.e. taking the hyper square root

$$2 \downarrow\downarrow \left(n+\frac{1}{2}\right) < \text{hsqrt}_2\left(2 \downarrow\downarrow (n+1) \downarrow \tilde{c}(n)\right) \ . \tag{9}$$

The function $\tilde{c}(n)$ can be computed numerically. For example, $\tilde{c}(1) = 1.42660839953973\ldots$, so according to (8),

$$2 \downarrow\downarrow 1.5 > 2 \downarrow\downarrow 1 \downarrow \tilde{c}(1) = 2.53867\ldots \ , \tag{10}$$

and according to (9),

$$2 \downarrow\downarrow 1.5 < \text{hsqrt}_2\left(2 \downarrow\downarrow 2 \downarrow \tilde{c}(1)\right) = 2.61024\ldots \tag{11}$$

If we do the same calculation for example for $n = -2$, we get

$$1.325825835\ldots < 2 \downarrow\downarrow -1.5 < 1.3264742627\ldots \ ,$$

and we can compute new inequalities (a new interval) for $2 \downarrow\downarrow 1.5$ just by iterating forwards (using that $2 \downarrow\downarrow (x+1) = (2 \downarrow\downarrow x)^{2\downarrow\downarrow x}$). We get

$$2.549442774\ldots < 2 \downarrow\downarrow 1.5 < 2.560742044\ldots \ ,$$

which is much better than (10) and (11)! Moreover, starting with the inequalities for $n = -15$ and iterating, one gets

$$2.551772168947734\ldots < 2 \downarrow\downarrow 1.5 < 2.55251901374805\ldots$$

It seems that the interval is getting smaller and smaller, if iteration and equations (8) and (9) are used for $n \to -\infty$! One gets

$$2 \downarrow\downarrow 1.5 = \mathfrak{P}_2(1.5) = 2.5518\ldots \ ,$$

and the method seems to be successful for other real exponents also. This way, the binary pentation function $\mathfrak{P}_2(x)$ can be computed to any desired accuracy[2] for every $x \in \mathbb{R}$ (even $x \in \mathbb{C}$).

Figure 3 shows a plot of $\mathfrak{P}_2(x)$ for $x \in \mathbb{R}$. It looks smooth and natural.

It is interesting to note that many different interpolation schemes all seem to give the same answer, if only they are based on

- interpolating the desired operation (exponentially, linearly or in any monotone way) for exponents in the interval $[n, n+1]$,

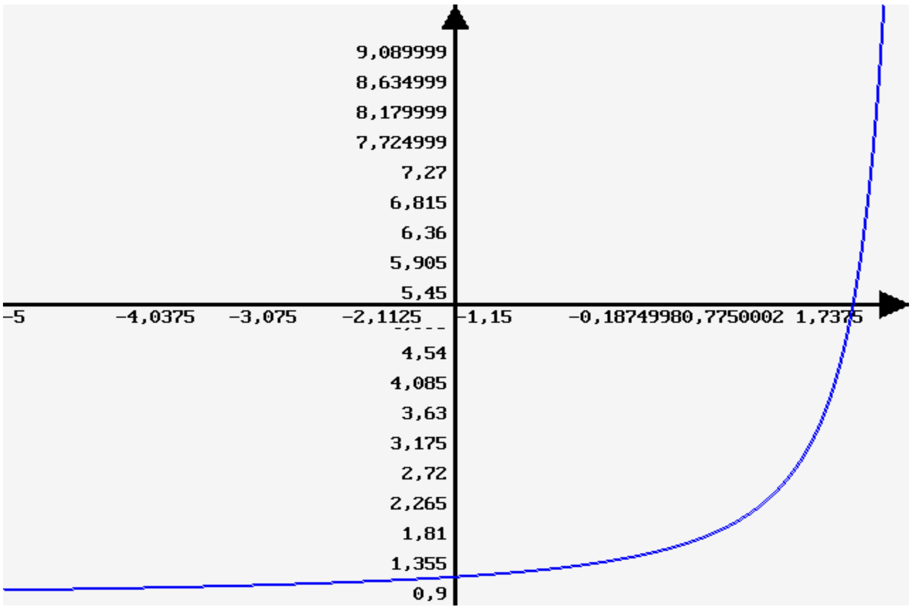

Figure 2: The binary pentation function $x \mapsto \mathfrak{P}_2(x) = 2 \downarrow\downarrow x$

- iterating up to the desired value (for example $b = 1.5$),
- taking the limit $n \to -\infty$ of both steps.

The same method can even be applied to hexation $\mathfrak{H}_2(x) := 2 \downarrow\downarrow\downarrow x$, which satisfies the recurrence relation $\mathfrak{H}_2(x+1) = \mathfrak{H}_2(x)^{\left(\mathfrak{H}_2(x)^{\mathfrak{H}_2(x)-1}\right)}$. This seems to be the "highest" non-standard operation that is accessible without too extensive numerical efforts.

6 A Last Observation

Since $2 \odot 1 = 2$ and $2 \odot 2 = 4$ for multiplication and every higher-order operation, the value of $2 \odot 1.5$ should be somewhere in between. It is interesting to see that $2 \odot 1.5$ seems to be decreasing in the "order" of operation:

$$\begin{aligned} 2 + 1.5 &= 3.5\ , \\ 2 \cdot 1.5 &= 3\ , \\ 2^{1.5} &= 2.828427\ldots\ , \\ 2 \downarrow 1.5 &= 2.66514\ldots \\ 2 \downarrow\downarrow 1.5 &= 2.5518\ldots\ , \\ 2 \downarrow\downarrow\downarrow 1.5 &= 2.4655\ldots \end{aligned}$$

[2] I do not have a proof that for $n \to -\infty$, the inequalites one gets become really sharp.

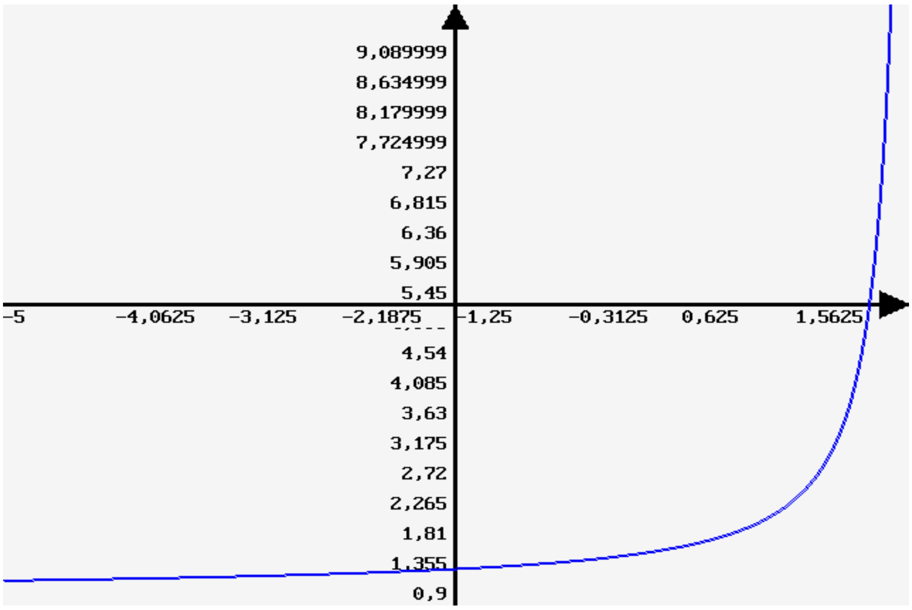

Figure 3: The binary hexation function $x \mapsto \mathfrak{H}_2(x) = 2 \downarrow\downarrow\downarrow x$

References

[!!!] Note that I do not claim mathematical rigour in this paper, and so I also do not claim in any way completeness of references. Maybe a good starting point for further reading can be the website *http://www.tetration.org*.

The following books are those that I used as a pupil for my *Jugend forscht* project. The first one was my reference for the "Verhulst" process and iteration in general, while the second one is reference for "Graham's number".

[B] R. Behr, *Ein Weg zur fraktalen Geometrie*, Ernst Klett Schulbuchverlag, Stuttgart 1989

[W] D. Wells, *Das Lexikon der Zahlen*, Fischer Taschenbuch Verlag, Frankfurt am Main 1990