Transitional Sequences and Nopt Structures

Alister Wilson

Abstract (Beta version)

Variations of Nopt structures. Computational pathways. Transitional sequences from exponentiation to tetration. A canonical transitional sequence that moves through the hyperoperator hierarchy. Laddered exponents. Complexity based definitions of finite and infinite. Some Bowers array numbers compared with Naropt structures.

Keywords: Natural numbers, hereditary base, transitional sequences, laddered exponents, Nopt structures, long repdigit numbers, binary sequences in other bases.

Contents

i Acknowledgements

Thanks very much to the hard work put in by the many contributors to Wikipedia and Wolfram Mathsworld. Thanks also to Jonathan Bowers for his colourful and thorough research into polyhedra and polychora.

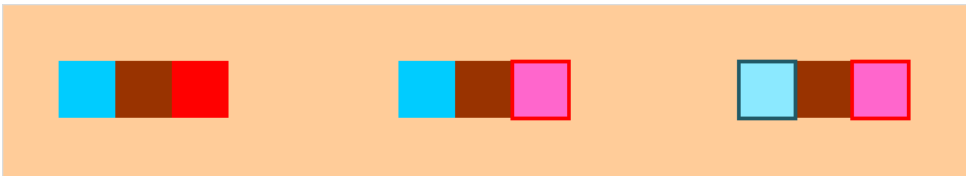
Part 0 Introduction

The definition of "first number larger than all the finite numbers" sounds reasonable, but is not a constructive definition, we don't know "all" the finite numbers, and as it is non-constructive in the most basic constructive part of maths, [the familiar counting numbers and pattern-numbers] it is silly if accurate reasoning is implied. Perhaps a more suitable proxy for omega would be "the first non-constructive number" and then it would be context-sensitive depending on available constructive techniques. As the pattern-numbers that take into account hyperoperations and bracketing patterns and sequences of these are a genuine mathematical reality, they shouldn't be ignored if the context would otherwise be neglectful. Pattern-numbers and their sequences could perhaps be compared in sensible ways with possible infinite ordinal counterparts, depending again, on some justifiable assumptions about where to place the key ordinals omega and epsilon-zero. Nopt structures are useful because they are visual and we now know their appearance for different ordertypes and with this geometry, recursion levels are clearer to see and so on. The other hierarchies (eg FGH) treat accumulation points differently. In some ways the FGH is more complicated to understand. At this stage, it seems quite difficult to compare FGH and nopt structures, but one thing is fairly clear, the accumulation points have correspondence with seed values in nopt structures. Nopt structures show the "whole structure" not just that the "latest stage" in the hierarchy. Also, an advantage of nopt structures is that they visually show how seed values are used at the different levels, and it may be possible to assign different seed values at different levels. Both structures show fast-growing increase, in some simple examples it may be interesting to compare. Nopt structures may be able to be adapted to focus more on computational pathways rather than increasing functions as in the standard definition.
In this paper, we also look at binary-indexed sequences and transitional sequences. It is noticeable, but rarely mentioned about the [induced-as-binary magnitude-reuse-sequences] from arbitrary finite bases. Also, various transitional sequences can be imagined in three natural realms. There is the transitional realm from exponentiation to tetration. A comparison with hereditary base may be made for this realm. The other natural realm covers or transitions through the hyperoperations. This realm is quite a lot more esoteric and intractable but there may be a few sequences worth mentioning. The classical Ackermann numbers or base(m) Ackermann numbers fit into this natural realm. The third realm is hard to comprehend and is where Bowers array operator numbers, Graham's number and naropt structures reside. Beyond naropt structures or the corresponding Conway arrow number sequence there seems to be a "natural speed limit" because, firstly, new notation apart from Knuth arrows are needed and secondly, it's not clear if anything significantly new is gained by such a new symbol. Perhaps Conway numbers are even more dramatic early on than people expect by recursing through symbol notations at the level of length 6 Conway numbers. Furthermore, I suspect that the rather surprising Conway arrow number recurrence relation is the fastest at fast growing while preserving linkages between previous stages. Anyway, the focus of this paper is trying to survey the variety of transitional sequences through the natural numbers. It is best read sequentially. I hope you can enjoy!

Part 1 Variations of Nopt structures

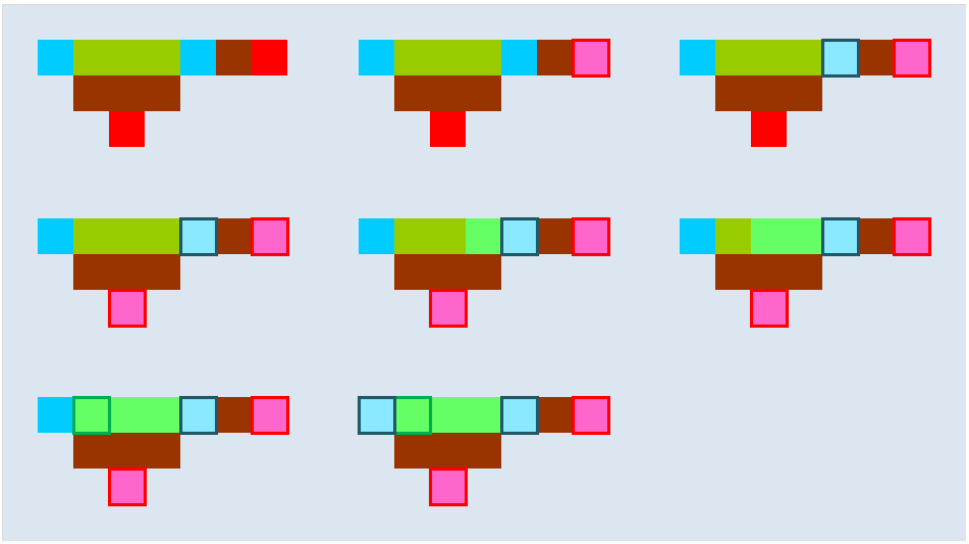
Section 1.1 Nopt structures and computational pathways

Consider the picture below for ordertype=4 nopt structures:



The computational pathway is illustrated, first the seedvalue is activated and then the formal power tower is evaluated with height supplied by the seedvalue.

The next picture shows the computational pathway for ordertype=5 nopt structures:

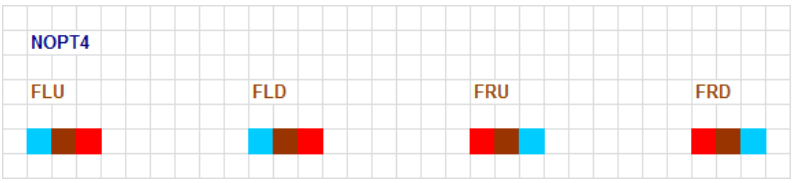


The computational pathway is illustrated: first the seedvalue is activated, then the formal power tower to the left is evaluated, then the ellipsis-controlling seedvalue is activated, the ellipsis stages are gradually passed through until the final formal power tower is reached with height supplied by the evaluated penultimate formal power tower.

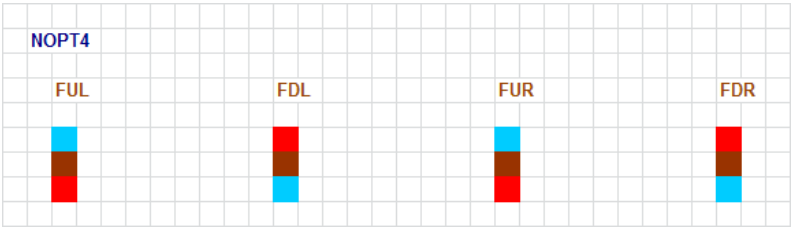
In a similar way, we can visualise the computational pathways for higher order nopt structures, noting that for ordertype 6 and higher, the ellipsis in a final component is controlled by a penultimate component, but often to reach the activation stage (the start of the ellipsis), the prior component would need to have been evaluated, which itself could be quite involved, depending on the ordertype of the prior component. In words it is difficult to explain, but this can be seen more clearly in an animation I have made of the computational pathways in nopt structures.

Section 1.2 The 8 possible folding patterns

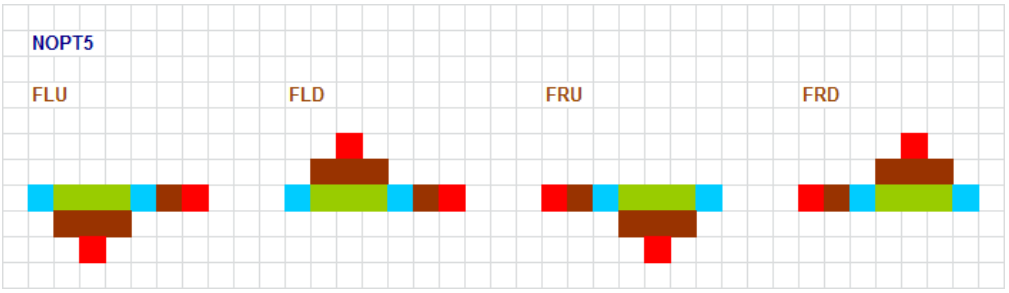
Nopt structures with the default-folding-method are used in the above examples, and this method is, I believe, the most natural way to visualise recursion, corresponding with thinking about how the base(n) Ackermann function works and typical conventions about mathematical notation. The computational pathway starts in the bottom-right position and zig-zags to the top-left position where the "answer" is obtained.
However, nopt structures can equally well be viewed as pseudo-tiling-patterns, and from this geometric point of view there are other ways to fold nopt structures.
They are shown below, in the case of ordertype=4 and ordertype=5 nopt structures.
For higher order nopt structures, please see Part 4.

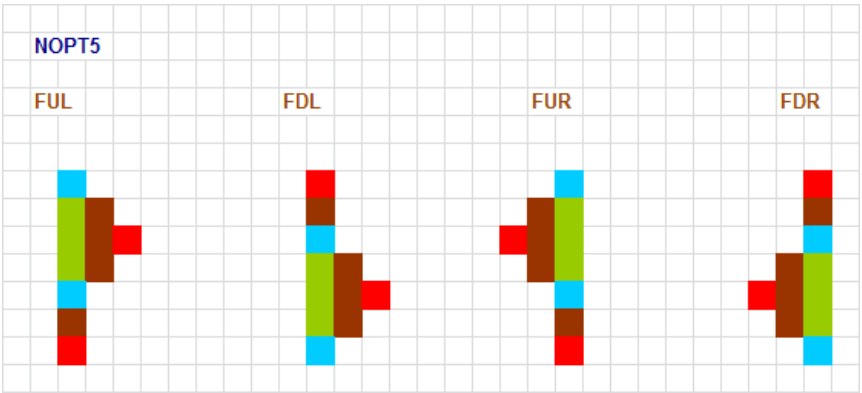


F=Fold , L=Left, U=Up, D=Down, R=Right. Eg. FLU="fold left then up"



Eg. FUL="fold up then left"





Section 1.3 Long repdigit numbers

The following numbers are all in Base10 and are all repdigit numbers:
333, 55555, 2222222, 888, 11111, 99999.
An example of a "partial arithmetic" is where we are only interested in how to do "+1" with the elements of interest. We notice that the "odd one out" from these examples is 99999.
All the others when you "+1" the number of digits is not changed.
333+1=334 (still 3 digits long), 55555+1=55556 (still 4 digits long), etc
99999+1=100000 (has changed from 5 digits to 6 digits long).
So unsurprisingly, repdigit(9) numbers are interesting for being the only numbers that when you add one also increases the number of digits by one.
Most of us are taught to do math addition on paper from right to left. If you want to do math in your head there are many good reasons why it is better to work from left to right. We usually read numbers from left to right and pronounce numbers from left to right, and so it's more natural to think about (and calculate) numbers from left to right.
However, with "+1 partial arithmetic" we may prefer to start by looking at the least significant digit. If it is a 0-8, the answer is easy, if it is a "9" we need to "carry" numbers. With repdigit(9) numbers, this carrying continues until the last "9" digit on the left: 99+1=100, 999+1=1000 and so on. Anyway, it is curious and interesting to observe that the "carry number" process continues into nopt structures:

$$\underbrace{999999999}_{9}+1=\underbrace{1000000000}_{10}=(a)$$

$$\underbrace{999...999}_{99}+1=\underbrace{1000...000}_{100}=(b)$$

$$\underbrace{999...999}_{\underbrace{999...999}_{99}}+1=\underbrace{1000...000}_{\underbrace{1000...000}_{100}}=(c)$$

and so on, into higher order nopt structures.
These numbers are "long repdigit numbers" and are interesting because they are actually describable for different nopt ordertypes. As they are repdigit or almost-repdigit numbers, they have a lower information content that allows describability. Between (a) and (b) are numbers we seldom refer to, and between (b) and (c) are heaps of numbers that are impossible to describe, apart from the redundant but true observation that any such number lies between (b) and (c) and hence has been partially described. The sequence 3^^3, 3^^^3, 3^^^^3, 3^^^^^3, turns hyperoperators into low-information-content unary representation. But the downside is, that the richness of numbers between exponentiation and tetration was neglected, and math writers haven't until now, attempted to show in what way 3^^^^^3 [3^(5)3] is much larger than 3^^^^3 [3^(4)3]. My papers help to resolve these issues, revealing some of the pattern numbers that are associated with the rich variety of "inbetween numbers".

Section 1.4 Complexity-based definitions of finite and infinite

The truly amazing computer, information, internet and technology society is all around us these days. We now have the "New Kind of Science", Mathsworld, Wolfram Alpha, Demonstrations Project and the incredible achievement of Wikipedia that has really good quality maths articles and does a fantastic job presenting a comprehensive variety of ideas. Anyway, here is another definition of finite and infinite from the philosophy of "contextual finitism" that may be useful, and is inspired by complexity thinking.

New definitions of finite, infinite from the paradigm known as "contextual finitism".

Finite
When the issue of a computation halting is not too complex, so that a termination point can be specified and understood, and is expressible by hyperarithmetic functions. Also, the algebraic formulation of the termination value is seen as relevant to the context of the problem.

Infinite
The issue of a computation halting is too complex, so that a termination point cannot be specified without using a limit argument. Also, the algebraic formulation of the termination value is seen as less relevant to the context of the problem.

Part 2 Transitional sequences

Section 2.1 From exponentiation to tetration

The consecutive whole numbers can be written in unary or in base(n) with n>=2. In daily life, we use the base(10) natural numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, …
Important examples are base(2), base(3), etc, as well as the hereditary base versions.
Base(2) uses binary {0,1} strings and begins: 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, …
Base(3) uses ternary {0,1,2} strings and begins: 0, 1, 2, 10, 11, 12, 20, 21, 22, 100, 101, 102, 110, …
The hereditary base versions are:

hb(2): $0,\ 1,\ 2,\ 2+1,\ 2^2,\ 2^2+1,\ 2^2+2,\ 2^2+2+1,\ 2^{2+1},\ 2^{2+1}+1,\ ...$

hb(3): $0,\ 1,\ 2,\ 3,\ 3+1,\ 3+2,\ 3*2,\ 3*2+1,\ 3*2+2,\ 3^2,\ 3^2+1,\ 3^2+2,\ 3^2+3,\ ...$

When we do maths we need to know the context of the numbers. Base2 and Base3 numbers are familiar mathematical entities, but given a string of digits, can we say by looking at them, what base is being assumed? If I present the string of digits "110" it could be in Base10 (one hundred and ten) or in Base2 (where it equals 4+2=6 in Base10). We make forays into other bases but we have the best experience, feeling and understanding for Base10 where we often compare numbers in our minds. Anyway, "110" could also be a number in Base3 (and then it is 9+3=12).
As soon as we want to describe a subsequence of the natural numbers, we are dealing with something quite different. Sequences are functions from natural numbers to natural numbers, so as functions, their domain is the set of natural numbers. The reason that sequences have dom(f)=N is that we like to talk about the first, second, third, members of the sequence, so we use the natural numbers to index (or point to) values in the sequence. As you may have guessed, it is sometimes convenient to start a sequence from 0 instead of 1, and refer to the zeroth, first, second, third members of a sequence.
Anyway, I hope to encourage some wonder at the possible ways to transition from natural numbers up towards tetration and beyond.

Section 2.2 The slow road to tetration: hereditary base(n)

The normal base(10) counting numbers 1,2,3,… can be re-expressed in terms of hereditary base(n) relative to a finite base(n). The "hereditary" adjective means that the exponents at each level are relative to base(n).
For example: the counting numbers represented as hereditary base(2) numbers can be grouped according to maximum height of the component exponentiation power towers.

Maxheight=1: $1,\ 2,\ 2+1,$

Maxheight=2: $2^2, 2^2+1, 2^2+2, 2^2+2+1, 2^{2+1}, 2^{2+1}+1, 2^{2+1}+2, 2^{2+1}+2+1, 2^{2+1}+2^2,$
$2^{2+1}+2^2+1, 2^{2+1}+2^2+2, 2^{2+1}+2^2+2+1,$

Maxheight=3: $2^{2^2}, 2^{2^2}+1, 2^{2^2}+2, 2^{2^2}+2+1, 2^{2^2}+2^2, 2^{2^2}+2^2+1, 2^{2^2}+2^2+2,$
$2^{2^2}+2^2+2+1, 2^{2^2}+2^{2+1}, 2^{2^2}+2^{2+1}+1, 2^{2^2}+2^{2+1}+2, 2^{2^2}+2^{2+1}+2+1,$
$2^{2^2}+2^{2+1}+2^2,$
$2^{2^2}+2^{2+1}+2^2+1, 2^{2^2}+2^{2+1}+2^2+2, 2^{2^2}+2^{2+1}+2^2+2+1, 2^{2^{2+1}}, 2^{2^{2+1}}+1, 2^{2^{2+1}}+2,...$

The last term with Maxheight=3 is :

$$2^{2^{2+1}}+2^{2^2+2+1}+2^{2^2+2}+2^{2^2+1}+2^{2^2}+2^{2+1}+2^2+2+1=65535.$$

The first term with Maxheight=4 is at $2^{2^{2^2}}=2^{16}=65536$.

So, this sequence gradually extends towards tetration. The 4th term of the sequence reaches a height of 2, the 16th term of the sequence reaches a height of 3, and the 65536th term of the sequence reaches a height of 4. Much further on in the sequence, the first term to reach a maxheight of 5 is the $2^{65536}th$ term.

Section 2.3 Understanding hereditary base

The usual and common way to think about the counting numbers is with Base10 and using Place Value Notation. With PVN, we observe that addition is represented by concatenation of digits, multiplier by numeral digit value (less than the base) and position of digit indicating power of the base, starting from 0, 1, 2, as the exponents of the base.

So consider a base10 number such as 274039541.

The operations of addition and multiplication are quite visible, but the consecutive powers of the base (base to power of exponent) are indirectly present. We need to do some "meta-counting" of the digits in order to understand it better, such as the order of magnitude.

274039541 is easier to read as 274,039,541 and we can see clearly there are 9 digits and the number is somewhere between 200 million and 300 million.

We can write 274,039,541 in expanded form as:

$$2\cdot10^8+7\cdot10^7+4\cdot10^6+3\cdot10^4+9\cdot10^3+5\cdot10^2+4\cdot10^1+1\cdot10^0$$

As usual, we use "." to mean multiplication and "^" to mean "to the power of".

So far the issue of hereditary base has not arisen because the highest exponent (8) is less than the base (10).

But with the number 274,039,541,735 there are 12 digits.

The expanded form is:

$$2.10^{11}+7.10^{10}+4.10^9+\cdots+5.10^0$$

But what about the exponent "11"?

The hereditary base10 form is:

$$2.10^{10+1}+7.10^{10}+4.10^9+\cdots+5.10^0$$

So that all the exponents are written in expanded form as well.

Numbers in Base10 can be written in Base2 or Base3.

For example 21 (base10) = 16+4+1 = 10101 (base2)

The expanded form of 10101 is:

$$1.2^4+1.2^2+1.2^0=2^{2^2}+2^2+2^0$$

Another example:

$$2205_{10}=3^7+2.3^2=10000200_3=3^{2.3+1}+2.3^2$$

In the next section, we consider hereditary base examples with long and sparse bitstrings.

Section 2.4 Hereditary base examples with long and sparse bitstrings

Consider the following number:

$2^{2^6}+2^6=2^{2^{2^2+2}}+2^{2^2+2}$ How to write this as a long and sparse bitstring?

Firstly, recognise that in Base2, $2^n=\underbrace{100..00}_{n+1 \text{ digits}}$. So, $2^{2^6}=\underbrace{100..00}_{2^6+1 \text{ digits}}$ and $2^6=\underbrace{100..00}_{6+1 \text{ digits}}$

Therefore $2^{2^6}+2^6=\underbrace{100...001000000}_{2^6+1 \text{ digits}}$

Here are some more examples:

Example A	**Example B**
$3^{3+1}+3^3+3^2+1$ 11101 (Base3) 81 + 27 + 9 + 1 = 118	$3^{3.2+2}+3^2.2+3.2+1$ 100000221 (Base3) 6561 + 18 + 6 + 1 = 6,586
Example C	**Example D**
$3^{3^2+1}+3^{3^2}+3^{3.2}+3$ 11001000010 (Base3) 59049 + 19683 + 729 + 3 = 79,464	$3^{3^2.2+1}.2+3^{3^2.2}.2$ 22000000000000000000 (Base3) 2324522934 + 774840978 = 3,099,363,912

Example E

$3^{3^2.2+3.2+2}.2+3^2+1$

=2000000000000000000000000101

=2541865828329*2 + 9 + 1 = 5083731656658 + 10

=5,083,731,656,668 (about 10^12).

Example F

$3^{3^3+3.2}.2+3^{3^2+1}.2+3^2$

2000000000000000000000020000000100

=5559060566555523*2 + 59049*2 + 9

=11,118,121,133,229,153 (about 10^16).

Example G

$3^{3^3.2}+3^{3^3+1}.2+1$

1000000000000000000000000020000000000000000000000000001

= 581497370030400596903901 69 + 22876792454961*2 + 1

= 58,149,737,003,085,813,275,300,092 (about 10^25).

Example H

What is $3^{3^3}-1$ in Base3 or hereditary base 3?

$3^{3^3}=\underbrace{1000...000}_{3^3+1 \text{ digits}}{}_{(3)}$ (*) So, $3^{3^3}-1=\underbrace{222...222}_{3^3 \text{ digits}}{}_{(3)}$ (**)

This is the answer in Base3 but in hereditary base 3, we put the number in expanded form and each of the exponents are in hereditary base 3. So we have that:

$$3^{3^3}-1=3^{3^2.2+3.2+2}.2+3^{3^2.2+3.2+1}.2+3^{3^2.2+3.2}.2+3^{3^2.2+3+2}.2$$
$$+3^{3^2.2+3+1}.2+3^{3^2.2+3}.2+3^{3^2.2+2}.2+3^{3^2.2+1}.2+3^{3^2.2}.2$$
$$+3^{3^2+3.2+2}.2+3^{3^2+3.2+1}.2+3^{3^2+3.2}.2+3^{3^2+3+2}.2+3^{3^2+3+1}.2$$
$$+3^{3^2+3}.2+3^{3^2+2}.2+3^{3^2+1}.2+3^{3^2}.2+3^{3.2+2}.2+3^{3.2+1}.2$$
$$+3^{3.2}.2+3^{3+2}.2+3^{3+1}.2+3^3.2+3^2.2+3.2+2$$

And sure, the equation is unwieldy, but it illustrates hereditary base quite well, and notice that if you count them, there are $3^3=27$ terms in the sum, just as there are 27 digits in equation (**). This last example is a long sum of tree-like expressions and certainly *not* a "long and sparse bitstring", but, of course, if you add one to it you do get a long and sparse bitstring as in equation (*).

Section 2.5 The Binob array

Consider the normal binary sequence, usually associated with base(2), but instead, interpreted as base(3) strings that have no occurrence of the digit "2". Also, the binary sequence could be interpreted as base(n) strings for n=3,4,5,…
This gives rise to a family of sequences, where binary is interpreted using a base higher than base(2). Let's call this family the Binob family, where "Binob" stands for "BINary in Other Base".

THE BINOB ARRAY

b10	bin	d4	d3	d2	d1	d0	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1
2	10	0	0	0	1	0	2	3	4	5	6	7	8	9	10
3	11	0	0	0	1	1	3	4	5	6	7	8	9	10	11
4	100	0	0	1	0	0	4	9	16	25	36	49	64	81	100
5	101	0	0	1	0	1	5	10	17	26	37	50	65	82	101
6	110	0	0	1	1	0	6	12	20	30	42	56	72	90	110
7	111	0	0	1	1	1	7	13	21	31	43	57	73	91	111
8	1000	0	1	0	0	0	8	27	64	125	216	343	512	729	1000
9	1001	0	1	0	0	1	9	28	65	126	217	344	513	730	1001
10	1010	0	1	0	1	0	10	30	68	130	222	350	520	738	1010
11	1011	0	1	0	1	1	11	31	69	131	223	351	521	739	1011
12	1100	0	1	1	0	0	12	36	80	150	252	392	576	810	1100
13	1101	0	1	1	0	1	13	37	81	151	253	393	577	811	1101
14	1110	0	1	1	1	0	14	39	84	155	258	399	584	819	1110
15	1111	0	1	1	1	1	15	40	85	156	259	400	585	820	1111
16	10000	1	0	0	0	0	16	81	256	625	1296	2401	4096	6561	10000

Section 2.6 The stem-2-base-3 sequence

In number theory, there are the even and odd numbers, squares and cubes, of natural numbers.
I would like to introduce a curious sequence sharing features of base2 and base3 that I have decided to refer to as the "stem-2-base-3" sequence. The sequence is supposed to be interpreted in base3, in other words, the allowable digits are {0,1,2} in the usual order. Here is the sequence:
0, 1, 2, 10, 11, 12, 20, 100, 101, 110, 111, 120, 200, 1000, 1001, 1010, 1011,1100, 1101, 1110, 1111, 1200, 2000, 10000, 10001, 10010, 10011, 10100, …
Clearly, it is permissible to see these numbers as base3 numbers.
The "stem-2-base-3" sequence contains the complete binary sequence (in blue colour), but with regular insertions of two numbers of the form **(120…0, 200…0)** that appear just before numbers of the form **(100…0).** Indeed, the binary strings within the sequence should be interpreted as all of the ternary strings that have no occurrence of the digit "2".
The "stem-2-base-3" sequence has the feature of accentuating information immediately before a change in magnitude. At the stage when we transition from (n)-digit numbers to (n+1)-digit numbers we bring in the extra numbers **(120…0, 200…0)** to "pad out" the **(111…1)** to **(100…0)** transition. Later on in the paper we present an interesting use for these sequences, and for the reason just stated, about padding out information at the stage just before a magnitude transition.

Anyway, let's have a look at a table to ponder over the nature of the "stem-2-base-3" sequence.

b10	b3	b2	s2b3	b10s23	b10	b3	b2	s2b3	b10s23
1	1	1	1	1	17	122	10001	1100	36
2	2	10	2	2	18	200	10010	1101	37
3	10	11	10	3	19	201	10011	1110	39
4	11	100	11	4	20	202	10100	1111	40
5	12	101	12	5	21	210	10101	1200	45
6	20	110	20	6	22	211	10110	2000	54
7	21	111	100	9	23	212	10111	10000	81
8	22	1000	101	10	24	220	11000	10001	82
9	100	1001	110	12	25	221	11001	10010	84
10	101	1010	111	13	26	222	11010	10011	85
11	102	1011	120	15	27	1000	11011	10100	90
12	110	1100	200	18	28	1001	11100	10101	91
13	111	1101	1000	27	29	1002	11101	10110	93
14	112	1110	1001	28	30	1010	11110	10111	94
15	120	1111	1010	30	31	1011	11111	11000	108
16	121	10000	1011	31	32	1012	100000	11001	109

Section 2.7 The stalk-0-bud-3 sequence

We can form a "stalk-0-bud-3" sequence by starting with the "stem-2-base-3" sequence and "pruning the stem" so that we're only left with the "bud". So we start with:
0, 1, 2, 10, 11, 12, 20, 100, 101, 110, 111, 120, 200, 1000, 1001, 1010, 1011,1100, 1101, 1110, 1111, 1200, 2000, 10000, 10001, 10010, 10011, 10100, …
and from this sequence, only keep numbers of the form { 120…0, 200…0, 100…0 }.
Once the other numbers from the sequence are removed, we have:
2, 10, 12, 20, 100, 120, 200, 1000, 1200, 2000, 10000, …
These base(3) numbers can be converted into base(10) numbers:
2, 3, 5, 6, 9, 15, 18, 27, 45, 54, 81, …
Or we could also retain the repdigit(1) numbers of the form (111…1) to give "slightly fatter buds":
1, 2, 10, 11, 12, 20, 100, 111, 120, 200, 1000, 1111, 1200, 2000, 10000, …
And these base(3) numbers can be converted into base(10) numbers:
1, 2, 3, 4, 5, 6, 9, 13, 15, 18, 27, 40, 45, 54, 81, …
So these two "stalk-0-bud-3" sequences focus in on changes in magnitude relative to exponentiation that is encoded in the position of a digit in a ternary string.
We can then define the "minimal stalk-0-bud-n" family of sequences as follows:
1, 10, 100, 1000, 10000, …
where the binary strings are interpreted in base(n) where n>=2.
In base(2) these are the powers of 2, in base(3) the powers of 3 and so on.

Let's compare the standard binary sequence in hb(2) with "minimal stalk-0-bud-2" in hb(2):
1, 10, 11, 100, 101, 110, 111, 1000, 1001, …
hb2: $1,\ 2,\ 2+1,\ 2^2,\ 2^2+1,\ 2^2+2,\ 2^2+2+1,\ 2^{2+1},\ 2^{2+1}+1,\ \ldots$
1,10,100,1000,10000,100000,1000000,10000000,100000000, … (base2)
hb2: $1,\ 2,\ 2^2,\ 2^{2+1},\ 2^{2^2},\ 2^{2^2+1},\ 2^{2^2+2},\ 2^{2^2+2+1},\ 2^{2^{2+1}},\ \ldots$

Now let's compare the ternary sequence in hb(3) with "minimal stalk-0-bud-3" in hb(3):
1, 2, 10, 11, 12, 20, 21, 22, 100, …
hb3: $1,\ 2,\ 3,\ 3+1,\ 3+2,\ 3*2,\ 3*2+1,\ 3*2+2,\ 3^2,\ ...$
1,10,100,1000,10000,100000,1000000,10000000,100000000, … (base3)
hb3: $1,\ 3,\ 3^2,\ 3^3,\ 3^{3+1},\ 3^{3+2},\ 3^{3*2},\ 3^{3*2+1},\ 3^{3*2+2},\ ...$

Section 2.9 Lexicographic binary-indexed trees

Consider the lexicographic binary sequence. There are 2 lexbin numbers with length 1, namely 0 and 1.
There are 2^2 lexbin numbers with length 2, namely 00, 01, 10, 11.
There are 2^3 lexbin numbers with length 3, namely 000, 001, 010, 011, 100, 101, 110, 111.
In general, there are 2^n lexbin numbers with length n, and they are:
00…00, 00…01, 00…10, 00…11, up to 11…10, 11…11.
This is because there are n positions, and each digit is either 0 or 1, giving 2^n such numbers.
They can be ordered first by string length (string length=1,2,3,…) and then by the standard lexicographic ordering. This gives the sequence:
0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, 0010, 0011, …
It is clear that the number of lexbin numbers with string length <= n is given by:

$$|\{lexbin : length(lexbin) <= n\}| = \sum_{i=1}^{n} 2^i.$$

We can use the lexbin numbers with finite length to index sequences that transition from exponentiation to tetration. Select a number for base, eg b=3, and consider the backbone sequence that is 3, 3^^2, 3^^3, 3^^4, 3^^5 etc. Now this sequence is already a tetration sequence but we can form a transitional sequence in the following way:
For 3^^n, which is an exponential power tower of height n, associate the 2^n lexbin numbers. The lexbin number 00…00 (n 0's) corresponds to 3^^n.
For each digit "1" in the lexbin index number we either add a constant number, c,
where (1<=c<b), or multiply by a constant, k, where (1<k<b), and this happens at the level corresponding to the position of the "1" in the lexbin number. For each digit "0", nothing is done at the corresponding level. The resulting numbers are well-defined, due to the exponent laws of top-down exponentiation in exponential power towers.
Now, we can form a notation to describe these numbers:
Tree(b, lexbin index number, level operation) where the function "tree" takes the base number b, forms a base(b) power tower of height=length(lexbin) and applies the level operation as defined above. From this description, and the lexicographic ordering of lexbin numbers, we can form an ordered sequence of "trees" that form a transitional sequence between b^^n and b^^(n+1).

Example (1) $< tree(3, a_1a_2...a_n, +1)\ such\ that\ n \geq 1, a_1a_2...a_n\ is\ lexbin. >$
Example (2) $< tree(3, a_1a_2...a_n, \times 2)\ such\ that\ n \geq 1, a_1a_2...a_n\ is\ lexbin. >$
Let's look at the first few values of these sequences

E.g. (1) $< tree(3, a_1a_2...a_n, +1)\ such\ that\ n \geq 1, a_1a_2...a_n\ is\ lexbin. >$

$3,$	$3+1,$		
$3^3,$	$3^3+1,$	$3^{3+1},$	$3^{3+1}+1,$
$3^{3^3},$	$3^{3^3}+1,$	$3^{3^3+1},$	$3^{3^3+1}+1,$
$3^{3^{3+1}},$	$3^{3^{3+1}}+1,$	$3^{3^{3+1}+1},$	$3^{3^{3+1}+1}+1,$ etc.

E.g. (2) $< tree(3, a_1a_2...a_n, \times 2)\ such\ that\ n \geq 1, a_1a_2...a_n\ is\ lexbin. >$

$3,$ $3.2,$

$3^3,$ $3^3.2,$ $3^{3.2},$ $3^{3.2}.2,$

$3^{3^3},$ $3^{3^3}.2,$ $3^{3^{3.2}},$ $3^{3^{3.2}}.2,$

$3^{3^{3.2}},$ $3^{3^{3.2}}.2,$ $3^{3^{3.2}.2},$ $3^{3^{3.2}.2}.2,$ etc.

So, this is another way to consider transitions between exponentiation and tetration.

Section 2.9 The Etindao numbers

As we move towards tetration, we need to leave behind the Place Value Notation method of notation due to the number of digits in PVN being too large to easily represent.
The key idea is to find systematic ways to describe certain "natural" classes of hereditary base numbers. For describing large numbers, hereditary base is the way to go as it immediately puts the focus on the exponent of the number.
For example In terms of information content:
100,000,000 (Base3) uses 9 digits of information under the PVN assumption.
Written in hb(3) as 3^{3*2+2} uses 4 digits and 3 operations, or 7 units of information.
If we were to write a googol (Base10) in PVN notation it would require 101 digits.
$\text{googol}=10^{100}=10^{10^2}$ is written in hereditary base10 and only uses 5 numerical symbols and exponential operations with the usual top-down exponent evaluation assumption.
An interesting idea, is to use a binary, ternary, quaternary, etc, lexicode sequence to point to locations on the hereditary base tree where we can insert operations and digits at particular levels.
As these numbers are Exponential Towers Indexed with Appended Operations I shall refer to these numbers as "Etindao numbers".
So here is our well-used binary sequence:
0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, …
but the lexicode version looks a little different:
0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, 0001, 0010, 0011,…
and etindao[+1] refers to appended operation of "+1" wherever there is a "1" in the lexicode.
For example, we can interpret a "1" at position "n" to indicate $"+1"$ at level "n" of the hb(2) tree:

$$2,\ 2+1,\ 2^2,\ 2^2+1,\ 2^{2+1},\ 2^{2+1}+1,\ 2^{2^2},\ 2^{2^2}+1,$$

$$2^{2^2+1},\ 2^{2^2+1}+1,\ 2^{2^{2+1}},\ 2^{2^{2+1}}+1,\ 2^{2^{2+1}+1},\ 2^{2^{2+1}+1}+1,\ \ldots$$

We could do the same $"+1"$ thing with an hb(3) etindao tree. Also, with the hb(3) etindao tree, we could interpret a "1" at position "n" in the lexicode to indicate $"*2"$ at level "n" of the hb(3) etindao tree, since multiplication by 2 is part of the hb(3) method. Clearly, a ternary lexicode could be used, if we wanted to refer to appending $"+1"$ OR $"*2"$ at level "n" of an hb(3) etindao tree.
Or a quaternary lexicode could be used if we wanted to refer to appending $"+1"$ OR $"*2"$ OR $"*2+1"$ at level "n" of an hb(3) etindao tree. The family of sequences of etindao trees show the various ways of moving in "binary-not-apply-OR-apply-the-operations fashion" up to tetration numbers and it is clear that the lexicode orders a sequence of etindao trees from smaller to bigger.

Section 2.10 Laddered exponents and hereditary base

This section is about a more involved way to think about the hb(3) trees based on all possible binary operation bracketings, Catalan numbers, and nested applications of the exponent laws.
It relates to an idea that Conway & Guy refer to in their "Book of Numbers" known as "laddered exponents". In "Hyperoperations and Nopt structures" I showed the Catalan number ways to bracket n-fold towers of exponents (known as laddered exponents). It is also possible to expand the brackets to obtain hb(3) tree representations.
There isn't really a unique way to enumerate (and order) the Catalan-number possible binary bracketings on n symbols, but we can use a natural method (derived from the recursion definition of Catalan numbers) that I used in the pictures from my other papers, and the Catalan-number-trees are colored using my 3-color butdj-coloring-method (see my other papers "Hyperoperations and Nopt structures" and "Hierarchies and Nopt structures"). There is also the butdj-block-design method that makes it easier to visualise the levels of nestings compared to only looking at the Catalan number trees (see "Hyperoperations and Nopt structures" paper). Together, they help to give an accurate feel for a correct enumeration of the possibilities and a proper expanded form into hereditary base(2) or hereditary base(3).

The Catalan numbers (the internet has copious information about them): …

$C_0 = 1$ and $C_1 = 1$ and

$C_2 = C_0C_1 + C_1C_0 = 1+1 = 2$

$C_3 = C_0C_2 + C_1C_1 + C_2C_0 = 2+1+2 = 5$

$C_4 = C_0C_3 + C_1C_2 + C_2C_1 + C_3C_0 = 5+2+2+5 = 14$

$C_5 = C_0C_4 + C_1C_3 + C_2C_2 + C_3C_1 + C_4C_0 = 14+5+4+5+14 = 42$

The Catalan numbers apply to enumerating binary bracketings, in particular, n-fold exponential towers, or laddered exponents. We can then expand the laddered exponents into hereditary base form in order to count the number of distinct values.

$2^{(2^2)} = 2^4 = 16$ and $(2^2)^2 = 4^2 = 16$

$3^{(3^3)} = 3^{27} = 7625597484987$ and $(3^3)^3 = 27^3 = 19683$

We can summarise the information into a table:

Height=3 patterns	a(aa)	(aa)a	Num distinct values
Base=2	16	16	1 / 2
Base=3	7,625,597,484,987	19,683	2 / 2

The 5 possible 4-fold exponential base=2 towers expanded into hb2:

$2^{(2^{(2^2)})} = 2^{2^{2^2}}$ $2^{(2^2)^2} = 2^{2^{2^2}}$ $(2^2)^{(2^2)} = 2^{2^{2+1}}$ $(2^{(2^2)})^2 = 2^{2^{2+1}}$ $((2^2)^2)^2 = 2^{2^{2+1}}$

Counting the number of distinct values (where "0" represents value previously occurred)
(1 + 0 + 1 + 0 + 0) = 2, (i.e. 2 / 5)
The 5 possible 4-fold exponential base=3 towers expanded into hb3:

$3^{(3^{(3^3)})} = 3^{3^{3^3}}$ $3^{(3^3)^3} = 3^{3^{3^2}}$ $(3^3)^{(3^3)} = 3^{3^{3+1}}$ $(3^{(3^3)})^3 = 3^{3^{3+1}}$ $((3^3)^3)^3 = 3^{3^3}$

Counting the number of distinct values (where "0" represents value previously occurred)
(1 + 1 + 1 + 0 + 1) = 4, (i.e. 4 / 5)

Height=4	a(a(aa))	a((aa)a)	(aa)(aa)	(a(aa))a	((aa)a)a	Num dval
Base=2	2^16=65536	2^16=65536	2^8=256	2^8=256	2^8=256	2 / 5
Base=3	3^(3^27)	3^19683	3^81	3^81	3^27	4 / 5

What about the 14 possible 5-fold exponential base=3 towers?

$3^{3^{3^{3}}}$	$3^{3^{(3^{3})^{3}}}$	$3^{(3^{3})^{(3^{3})}}$	$3^{(3^{3^{3}})^{3}}$	$3^{((3^{3})^{3})^{3}}$	$(3^{3})^{(3^{3^{3}})}$	$(3^{3})^{((3^{3})^{3})}$
$(3^{3^{3}})^{(3^{3})}$	$((3^{3})^{3})^{(3^{3})}$	$(3^{3^{3}})^{3}$	$(3^{(3^{3})^{3}})^{3}$	$((3^{3})^{(3^{3})})^{3}$	$((3^{3^{3}})^{3})^{3}$	$(((3^{3})^{3})^{3})^{3}$

Expanding these laddered exponents into hereditary base 3 gives:

$$3^{3^{3^{3}}} = 3^{3^{3^{3}}} \quad 3^{3^{(3^{3})^{3}}} = 3^{3^{3^{2}}} \quad 3^{(3^{3})^{(3^{3})}} = 3^{3^{3^{3}+1}} \quad 3^{(3^{3^{3}})^{3}} = 3^{3^{3^{3}+1}}$$

$$3^{((3^{3})^{3})^{3}} = 3^{3^{3^{3}}} \quad (3^{3})^{(3^{3^{3}})} = 3^{3^{3^{3}}+1} \quad (3^{3})^{((3^{3})^{3})} = 3^{3^{3^{2}}+1}$$

$$(3^{3^{3}})^{(3^{3})} = 3^{3^{3.2}} \quad ((3^{3})^{3})^{(3^{3})} = 3^{3^{3+2}} \quad (3^{3^{3}})^{3} = 3^{3^{3^{3}}+1} \quad (3^{(3^{3})^{3}})^{3} = 3^{3^{3^{2}}+1}$$

$$((3^{3})^{(3^{3})})^{3} = 3^{3^{3+2}} \quad ((3^{3^{3}})^{3})^{3} = 3^{3^{3+2}} \quad (((3^{3})^{3})^{3})^{3} = 3^{3^{3+1}}$$

Counting the number of distinct values (where "0" represents value previously occurred)
(1 + 1 + 1 + 0) + (1 + 1 + 1) + (1 + 1 + 0 + 0) + (0 + 0 + 1) = 9, (i.e. 9 / 14)

Or expanding the 14 base=2 laddered exponents into hereditary base 2 gives:

$$2^{2^{2^{2}}} = 2^{2^{2^{2}}} \quad 2^{2^{(2^{2})^{2}}} = 2^{2^{2^{2}}} \quad 2^{(2^{2})^{(2^{2})}} = 2^{2^{2^{2}+1}} \quad 2^{(2^{2^{2}})^{2}} = 2^{2^{2^{2}+1}}$$

$$2^{((2^{2})^{2})^{2}} = 2^{2^{2^{2}+1}} \quad (2^{2})^{(2^{2^{2}})} = 2^{2^{2^{2}+1}} \quad (2^{2})^{((2^{2})^{2})} = 2^{2^{2^{2}+1}}$$

$$(2^{2^{2}})^{(2^{2})} = 2^{2^{2^{2}}} \quad ((2^{2})^{2})^{(2^{2})} = 2^{2^{2^{2}}} \quad (2^{2^{2^{2}}})^{2} = 2^{2^{2^{2}+1}}$$

$$(2^{(2^{2})^{2}})^{2} = 2^{2^{2^{2}+1}}$$

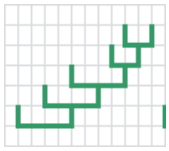
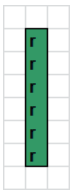
$$((2^{2})^{(2^{2})})^{2} = 2^{2^{2^{2}}} \quad ((2^{2^{2}})^{2})^{2} = 2^{2^{2^{2}}} \quad (((2^{2})^{2})^{2})^{2} = 2^{2^{2^{2}}}$$

Counting the number of distinct values (where "0" represents value previously occurred)
(1 + 0 + 1 + 0) + (0 + 1 + 0) + (1 + 0 + 0 + 0) + (0 + 0 + 0) = 4, (i.e. 4 / 14)

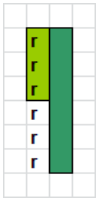
Next up, I show how to do this for 6-fold exponential power towers, where it can be a bit confusing to keep track of the 42 laddered exponents, so I've taken a lot of care by including the *Catalan number trees and block-designs with butdj colorings* to act as a visual check of the accuracy of (1) the enumerations, (2) the bracketing patterns and (3) expansions into hereditary base.

- **Comments:**
- The butdj coloring method is not unique for a particular tree.
- Having said that, I came up with a style that makes sense to me, and seems to correspond with the way the bracketing patterns are evaluated into hereditary base by nested applications of the exponent laws.
- Of course, these numbers are pattern numbers and can't be evaluated numerically as they're huge numbers in terms of magnitude, but maybe as pattern numbers they could be evaluated (into hereditary base) symbolically by a computer program.
- The coloring method corresponds with the natural symmetry in the Catalan numbers
- The Catalan number trees are colored the same way as the block-diagram representations

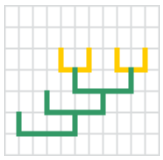
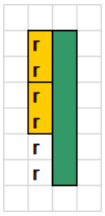
The 42 laddered exponents on 6-fold exponential power towers, with Base=3 and the parentheses expanded out to give a "pure" hereditary base form and with the usual top-down exponent evaluation assumption. This is necessary in order to determine what trees give the same or different values from one another, in other words, to count the number of distinct values. (There are 7 pages with 6 laddered exponents per page.)

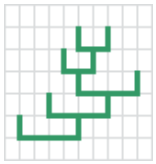
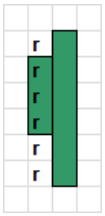
$$3^{3^{3^{3^{3}}}} = 3^{3^{3^{3^{3}}}}$$

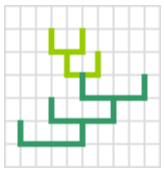
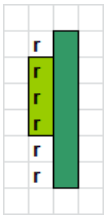
$$3^{3^{3^{(3^{3})^{3}}}} = 3^{3^{3^{3^{2}}}}$$

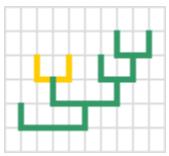
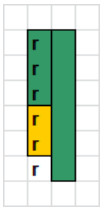
$$3^{3^{(3^{3})(3^{3})}} = 3^{3^{3^{3+1}}}$$

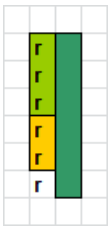
$$3^{3^{(3^{3^{3}})^{3}}} = 3^{3^{3^{3+1}}}$$

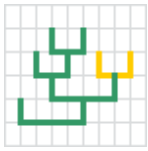
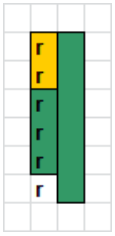
$$3^{3^{((3^{3})^{3})^{3}}} = 3^{3^{3^{3}}}$$

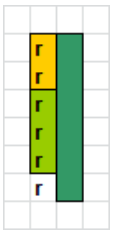
$$3^{(3^{3})^{(3^{3^{3}})}} = 3^{3^{3^{3}+1}}$$

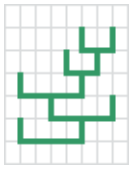
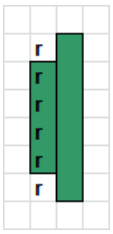
$$3^{(3^3)^{(3^3)^3}} = 3^{3^{3^{3^2+1}}}$$

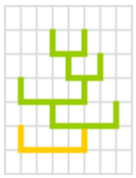
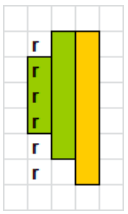
$$3^{(3^{3^3})^{(3^3)}} = 3^{3^{3^{3*2}}}$$

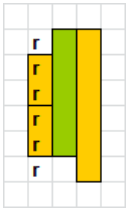
$$3^{((3^3)^3)^{(3^3)}} = 3^{3^{3^{3+2}}}$$

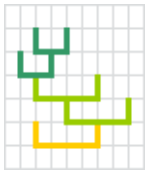
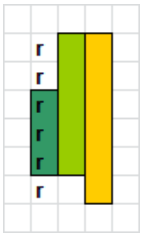
$$3^{(3^{3^{3^3}})^3} = 3^{3^{3^{3^3}*3}} = 3^{3^{3^{3^3+1}}}$$

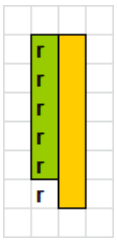
$$3^{(3^{(3^3)^3})^3} = 3^{3^{3^{3*3}*3}} = 3^{3^{3^{3^2+1}}}$$

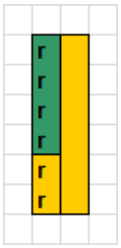
$$3^{((3^3)^{(3^3)})^3} = 3^{3^{3*(3^3)*3}} = 3^{3^{3^{3+2}}}$$

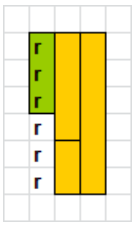
$$3^{((3^{3^3})^3)^3} = 3^{3^{3^{3+2}}}$$

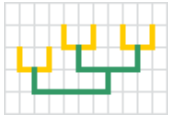
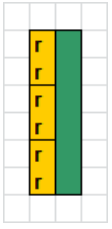
$$3^{(((3^3)^3)^3)^3} = 3^{(3^{3^3*3})} = 3^{3^{3^{3+1}}}$$

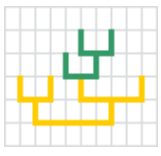
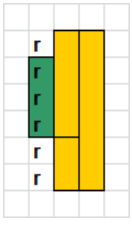
$$(3^3)^{(3^{3^{3^3}})} = 3^{3*3^{3^{3^3}}} = 3^{3^{3^{3^3}+1}}$$

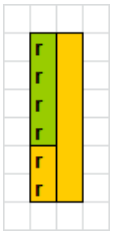
$$(3^3)^{(3^{(3^3)^3})} = 3^{3^{3^{3^2}+1}}$$

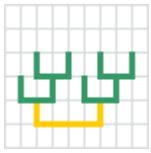
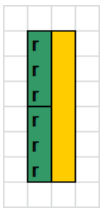
$$(3^3)^{((3^3)^{(3^3)})} = 3^{3^{3^{3+1}+1}}$$

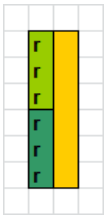
$$(3^3)^{(3^{3^3})^3} = 3^{3^{3^{3+1}+1}}$$

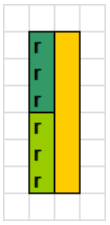
$$(3^3)^{((3^3)^3)^3} = 3^{3^{3^3+1}}$$

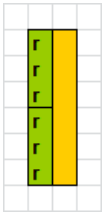
$$(3^{3^3})^{(3^{3^3})} = 3^{3^{3^3+3}}$$

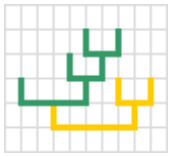
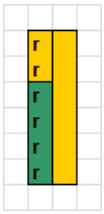
$$(3^{3^3})^{(3^3)^3} = 3^{3^{3^2+3}}$$

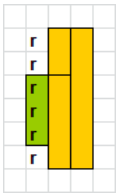
$$((3^3)^3)^{(3^{3^3})} = 3^{3^{3^3+2}}$$

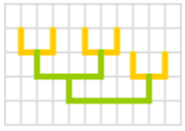
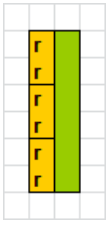
$$((3^3)^3)^{((3^3)^3)} = 3^{3^{3^2+2}}$$

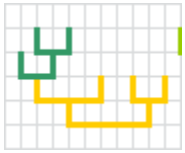
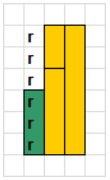
$$(3^{3^{3^3}})^{(3^3)} = 3^{3^{3^3+3}}$$

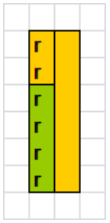
$$(3^{((3^3)^3)})^{(3^3)} = 3^{3^{3^2+3}}$$

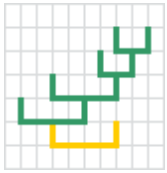
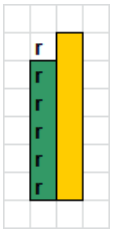
$$((3^3)^{(3^3)})^{(3^3)} = 3^{3^{3*2+1}}$$

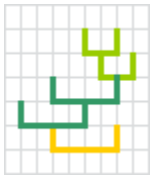
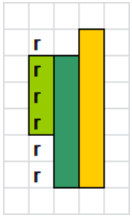
$$((3^{3^3})^3)^{(3^3)} = 3^{3^{3*2+1}}$$

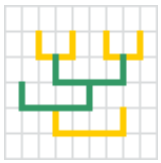
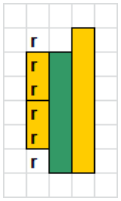
$$(((3^3)^3)^3)^{(3^3)} = 3^{3^{3*2}}$$

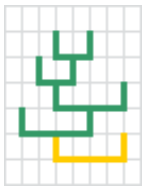
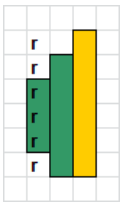
$$(3^{3^{3^3}})^3 = 3^{3^{3^3}*3} = 3^{3^{3^3}+1}$$

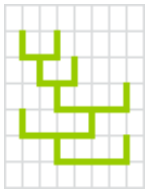
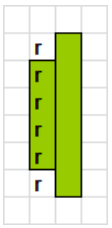
$$(3^{3^{((3^3)^3)}})^3 = 3^{3^{3^2}+1}$$

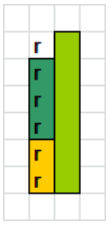
$$(3^{(3^3)^{(3^3)}})^3 = 3^{3^{3^{3+1}+1}}$$

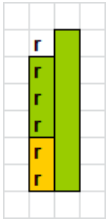
$$(3^{(3^{3^3})^3})^3 = 3^{3^{3^{3+1}+1}}$$

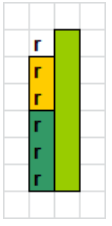
$$(3^{((3^3)^3)^3})^3 = 3^{3^{3^3+1}}$$

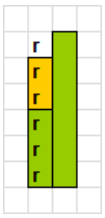
$$((3^3)^{(3^{3^3})})^3 = 3^{3^{3^3+2}}$$

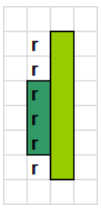
$$((3^3)^{((3^3)^3)})^3 = 3^{3^{3^2+2}}$$

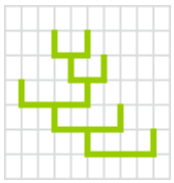
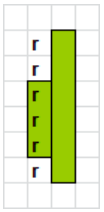
$$((3^{3^3})^{3^3})^3 = 3^{3^{3*2+1}}$$

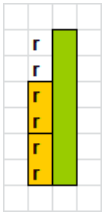
$$(((3^3)^3)^{3^3})^3 = 3^{3^{3*2}}$$

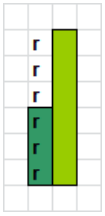
$$((3^{3^{3^3}})^3)^3 = 3^{3^{3^3+2}}$$

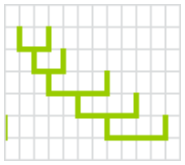
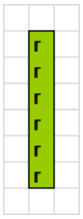
$$((3^{(3^3)^3})^3)^3 = 3^{3^{3^2+2}}$$

$$(((3^3)^{(3^3)})^3)^3 = 3^{3^{3*2}}$$

$$(((3^{3^3})^3)^3)^3 = 3^{3^{3*2}}$$

$$((((3^3)^3)^3)^3)^3 = 3^{3^{3+2}}$$

Assuming the expansions are correct (I think they are), let's see them in a list so as to count the number of distinct values. The 42 laddered exponents on 6-fold exponential power towers with base=3 (arranged in 14 rows of 3 equations):

$$3^{3^{3^{3^{3}}}} = 3^{3^{3^{3^{3}}}} \quad 3^{3^{3(3^{3})^{3}}} = 3^{3^{3^{3^{2}}}} \quad 3^{3^{(3^{3})(3^{3})}} = 3^{3^{3^{3+1}}}$$

$$3^{3^{(3^{3^{3}})^{3}}} = 3^{3^{3^{3+1}}} \quad 3^{3^{((3^{3})^{3})^{3}}} = 3^{3^{3^{3}}} \quad 3^{(3^{3})^{(3^{3^{3}})}} = 3^{3^{3^{3}+1}}$$

$$3^{(3^{3})^{(3^{3})^{3}}} = 3^{3^{3^{3^{2}}+1}} \quad 3^{(3^{3^{3}})^{(3^{3})}} = 3^{3^{3^{3*2}}} \quad 3^{((3^{3})^{3})^{(3^{3})}} = 3^{3^{3+2}}$$

$$3^{(3^{3^{3}})^{3}} = 3^{3^{3^{3}+1}} \quad 3^{(3^{(3^{3})^{3}})^{3}} = 3^{3^{3^{2}+1}} \quad 3^{((3^{3})^{(3^{3})})^{3}} = 3^{3^{3+2}}$$

$$3^{((3^{3^{3}})^{3})^{3}} = 3^{3^{3+2}} \quad 3^{(((3^{3})^{3})^{3})^{3}} = 3^{3^{3+1}} \quad (3^{3})^{(3^{3^{3}})} = 3^{3^{3^{3}}+1}$$

$$(3^{3})^{(3^{(3^{3})^{3}})} = 3^{3^{3^{2}}+1} \quad (3^{3})^{((3^{3})^{(3^{3})})} = 3^{3^{3^{3+1}+1}} \quad (3^{3})^{(3^{3^{3}})^{3}} = 3^{3^{3^{3+1}+1}}$$

$$(3^{3})^{((3^{3})^{3})^{3}} = 3^{3^{3^{3}+1}} \quad (3^{3^{3}})^{(3^{3^{3}})} = 3^{3^{3^{3}+3}} \quad (3^{3^{3}})^{(3^{3})^{3}} = 3^{3^{3^{2}+3}}$$

$$((3^{3})^{3})^{(3^{3^{3}})} = 3^{3^{3^{3}+2}} \quad ((3^{3})^{3})^{((3^{3})^{3})} = 3^{3^{3^{2}+2}} \quad (3^{3^{3^{3}}})^{(3^{3})} = 3^{3^{3^{3}+3}}$$

$$(3^{((3^{3})^{3})})^{(3^{3})} = 3^{3^{3^{2}+3}} \quad ((3^{3})^{(3^{3})})^{(3^{3})} = 3^{3^{3*2+1}} \quad ((3^{3^{3}})^{3})^{(3^{3})} = 3^{3^{3*2+1}}$$

$$(((3^{3})^{3})^{3})^{(3^{3})} = 3^{3^{3*2}} \quad (3^{3^{3^{3}}})^{3} = 3^{3^{3^{3}+1}} \quad (3^{3^{((3^{3})^{3})}})^{3} = 3^{3^{3^{3^{2}}+1}}$$

$$(3^{(3^{3})^{(3^{3})}})^{3} = 3^{3^{3^{3+1}+1}} \quad (3^{(3^{3^{3}})^{3}})^{3} = 3^{3^{3^{3+1}+1}} \quad (3^{((3^{3})^{3})^{3}})^{3} = 3^{3^{3^{3}+1}}$$

$$((3^{3})^{(3^{3^{3}})})^{3} = 3^{3^{3^{3}+2}} \quad ((3^{3})^{((3^{3})^{3})})^{3} = 3^{3^{3^{2}+2}} \quad ((3^{3^{3}})^{3^{3}})^{3} = 3^{3^{3*2+1}}$$

$$(((3^{3})^{3})^{3^{3}})^{3} = 3^{3^{3*2}} \quad ((3^{3^{3^{3}}})^{3})^{3} = 3^{3^{3^{3}+2}} \quad ((3^{(3^{3})^{3}})^{3})^{3} = 3^{3^{3^{2}+2}}$$

$$(((3^{3})^{(3^{3})})^{3})^{3} = 3^{3^{3*2}} \quad (((3^{3^{3}})^{3})^{3})^{3} = 3^{3^{3*2}} \quad ((((3^{3})^{3})^{3})^{3})^{3} = 3^{3^{3+2}}$$

Counting the number of distinct values (where "0" represents value previously occurred)
(1 + 1 + 1) + (0 + 1 + 1) + (1 + 1 + 1) + (0 + 0 + 0) + (0 + 1 + 1) + (1 + 1 + 0) + (1 + 1 + 1) + (1 + 1 + 0) + (1 + 1 + 0) + (1 + 0 + 0) + (0 + 0 + 0) + (0 + 0 + 0) + (0 + 0 + 0) + (0 + 0 + 1) = 21 (i.e. 21 / 42)

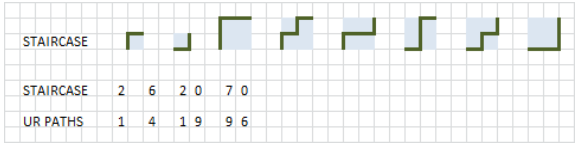
So we can make up a partial table, with the number of distinct values from Base=2 and Base=3 Laddered Exponents from 3-fold up to 6-fold power tower expressions:

Num dval	Height=3	Height=4	Height=5	Height=6
Base=2	1 / 2	2 / 5	4 / 14	[?? / 42]
Base=3	2 / 2	4 / 5	9 / 14	21 / 42

Someone with a curious mind may be interested to fill in the missing gap in this table!

Section 2.11 The standard array and finite and infinite UR-paths

The idea of a path through a grid of squares or grid of lattice points is quite well-known, such as Dyck paths and Staircase paths. Wolfram's Mathworld has information about Dyck paths, Staircase paths and others.
A similar, but different idea to Staircase paths are what I shall call UR-paths and are defined as follows. Whereas, Staircase paths are defined by edges that move right or up towards the top-right corner, finite UR-paths are defined so that they can finish at the top level at other positions apart from the corner, and they are defined via grid squares, rather than edges. The resulting combinatorics are somewhat different:
The Staircase sequence starts with: 2, 6, 20, 70, …
But the UR-path sequence starts with: 1, 4, 19, 96, …

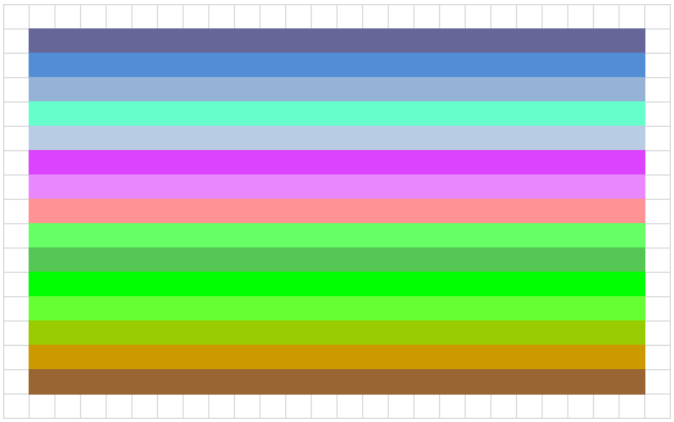


Finite UR-paths for 1x1, 2x2, 3x3 and 4x4 squares.

The “standard array” is recursively defined as follows:
The first sequence are the normal counting numbers.
Then take the 10^{th}, 100^{th}, 1000^{th}, etc number from the 1^{st} sequence to make the 2^{nd} sequence. Then, for the 3^{rd} sequence take the 10^{th}, 100^{th}, 1000^{th}, etc number from the 2^{nd} sequence, and so on.

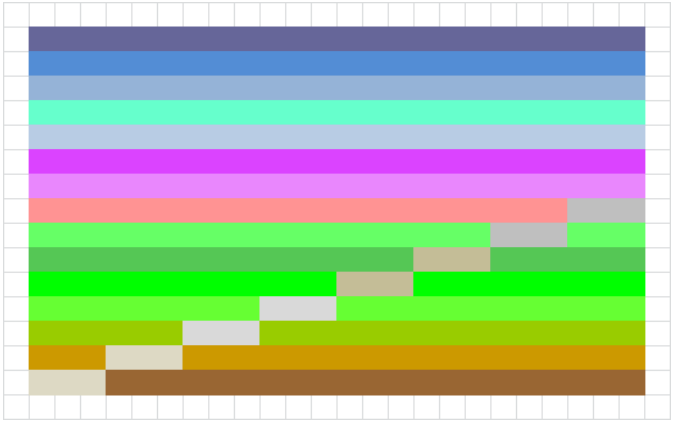
$10^{10^{10^{1}}}$	$10^{10^{10^{2}}}$	$10^{10^{10^{3}}}$	$10^{10^{10^{4}}}$	$10^{10^{10^{5}}}$	$10^{10^{10^{6}}}$	$10^{10^{10^{7}}}$	$10^{10^{10^{8}}}$	$10^{10^{10^{9}}}$	$10^{10^{10^{10}}}$	$10^{10^{10^{11}}}$	$10^{10^{10^{12}}}$	*etc*
$10^{10^{10}}$	$10^{10^{10^{2}}}$	$10^{10^{10^{3}}}$	$10^{10^{10^{4}}}$	$10^{10^{10^{5}}}$	$10^{10^{10^{6}}}$	$10^{10^{10^{7}}}$	$10^{10^{10^{8}}}$	$10^{10^{10^{9}}}$	$10^{10^{10^{10}}}$	$10^{10^{10^{11}}}$	$10^{10^{10^{12}}}$	*etc*
$10^{10^{1}}$	$10^{10^{2}}$	$10^{10^{3}}$	$10^{10^{4}}$	$10^{10^{5}}$	$10^{10^{6}}$	$10^{10^{7}}$	$10^{10^{8}}$	$10^{10^{9}}$	$10^{10^{10}}$	$10^{10^{11}}$	$10^{10^{12}}$	*etc*
10^{1}	10^{2}	10^{3}	10^{4}	10^{5}	10^{6}	10^{7}	10^{8}	10^{9}	10^{10}	10^{11}	10^{12}	*etc*
1	2	3	4	5	6	7	8	9	10	11	12	*etc*

The standard array can be pictured like this:



How to transition from exponentiation to tetration? We can define infinite sequences through the standard array. Each of the horizontal sequences in the standard array is at a particular level and doesn’t transition towards tetration. If we define a constant, maxh, to be a small finite number that represents the maximum horizontal distance before moving up to the next sequence above, then we can form a wide variety of transitional sequences from exponentiation to tetration. The other rule we need for these sequences is that to move up to the next sequence from the current sequence we can do it in either of two ways: vertical up one unit, or right-diagonal up one unit. This way we ensure there is no back-tracking. The aim here is to move towards tetration, so there is no limit for how far to go in the vertical direction. By loosening the requirement for having constant maxh, we can include more transitional sequences, but at each level the rightwards distance should be finite, before stepping up to the next sequence in the standard array. These transitional sequences are infinite UR-paths (compare with the finite UR-paths above) through the standard array such that all horizontal distances are finite. Some examples should make it clearer:

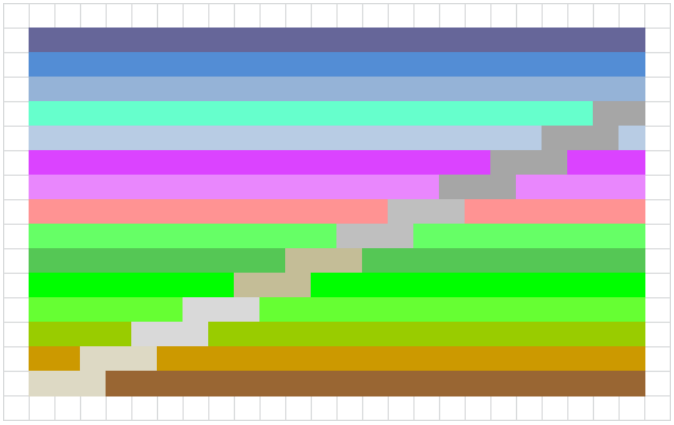
Example 1



Can you see what transitional sequence the Example 1 picture above represents? By comparing the grey squares that represent the transitional sequence with the standard array we have the sequence:

$$1,2,3,10^4,10^5,10^6,10^{10^7},10^{10^8},10^{10^9},10^{10^{10^{10}}},10^{10^{10^{11}}},10^{10^{10^{12}}},etc$$

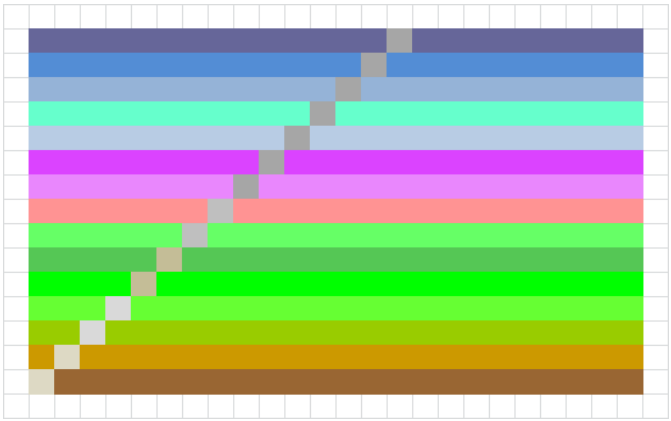
Example 2



The surrounding gridlines can be used to count the squares at each level, giving:

$$1,2,3,10^3,10^4,10^5,10^{10^5},10^{10^6},10^{10^7},10^{10^{10^7}},10^{10^{10^8}},10^{10^{10^9}},etc$$

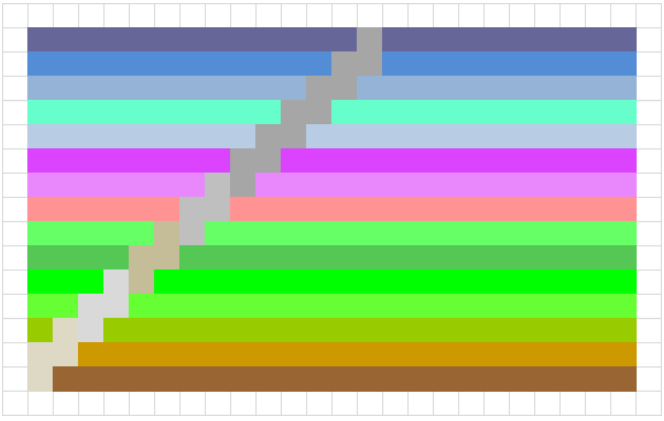
Example 3



This looks like diagonalization, and yes, it is the obvious diagonal sequence through the standard array. This sequence has the nice feature that the topmost exponent describes the height of the power tower.

$$1,10^2,10^{10^3},10^{10^{10^4}},10^{10^{10^{10^5}}},10^{10^{10^{10^{10^6}}}},10^{10^{10^{10^{10^{10^7}}}}},10^{10^{10^{10^{10^{10^{10^8}}}}}}etc$$

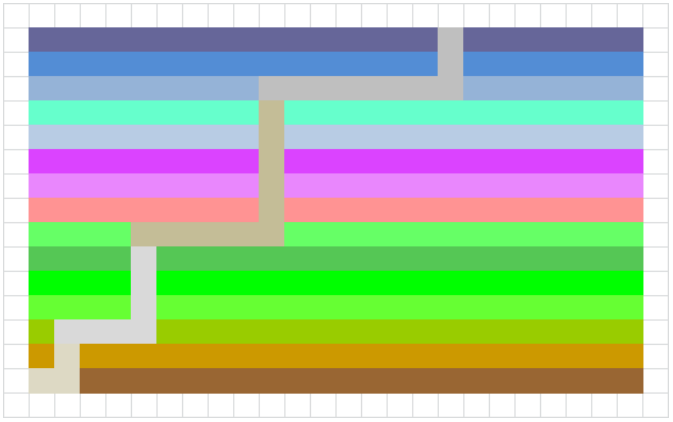
Example 4



Example 4 looks similar to Example 3, a fatter diagonal sequence through the standard array.

$1,10^{1},10^{2},10^{10^{2}},10^{10^{3}},10^{10^{10^{3}}},10^{10^{10^{4}}},10^{10^{10^{10^{4}}}},10^{10^{10^{10^{10^{5}}}}},etc$

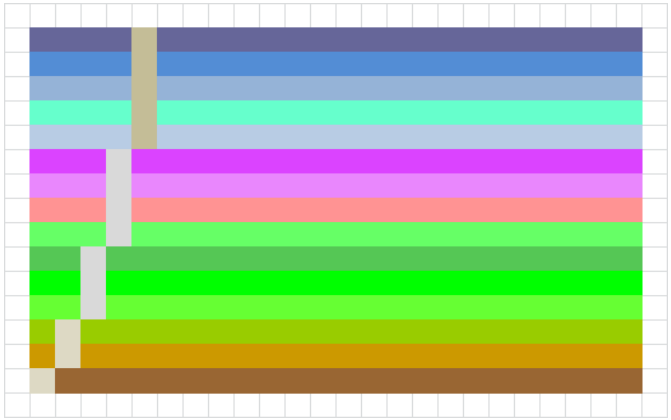
Example 5



Here we don't have a constant maxh (it increases by 2 each time: 2,4,6,8,…)
Notice that the vertical distances increase as odd numbers: 3,5,7,9,…

$1,2,10^{2},10^{10^{2}},10^{10^{3}},10^{10^{4}},10^{10^{5}},10^{10^{10^{5}}},10^{10^{10^{10^{5}}}},10^{10^{10^{10^{10^{5}}}}},10^{10^{10^{10^{10^{10^{5}}}}}},10^{10^{10^{10^{10^{10^{10^{6}}}}}}}$ *etc*

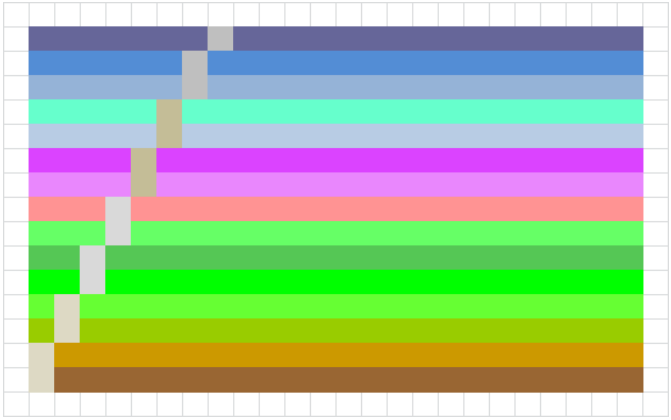
Example 6



Can you guess the transitional sequence represented?

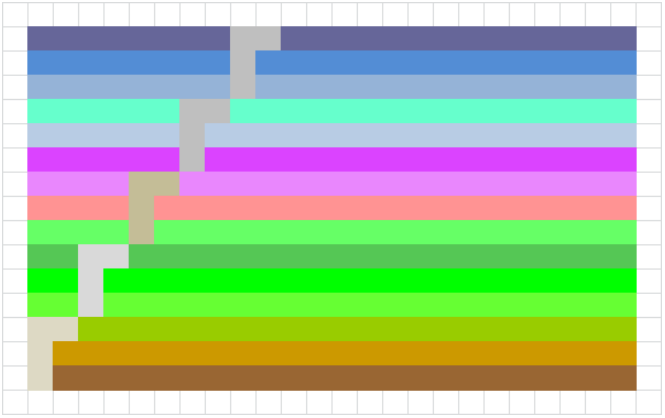
$1,10^{2},10^{10^{2}},10^{10^{10^{3}}}\ 10^{10^{10^{10^{3}}}}\ 10^{10^{10^{10^{10^{3}}}}},10^{10^{10^{10^{10^{10^{4}}}}}},10^{10^{10^{10^{10^{10^{10^{4}}}}}}},etc$

Example 7



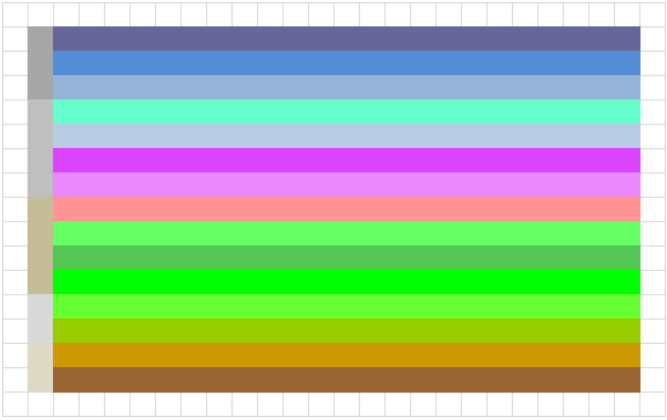
$1, 10^1, 10^{10^2}, 10^{10^{10^2}}, 10^{10^{10^{10^3}}}, 10^{10^{10^{10^{10^3}}}}, 10^{10^{10^{10^{10^{10^4}}}}}, 10^{10^{10^{10^{10^{10^{10^4}}}}}}, etc$

Example 8



$1, 10^1, 10^{10^1}, 10^{10^2}, 10^{10^{10^3}}, 10^{10^{10^{10^3}}}, 10^{10^{10^{10^{10^3}}}}, 10^{10^{10^{10^{10^{10^4}}}}}, 10^{10^{10^{10^{10^{10^{10^5}}}}}}, etc$

Example 9



$1, 10^1, 10^{10^1}, 10^{10^{10^1}}, 10^{10^{10^{10^1}}}, 10^{10^{10^{10^{10^1}}}}, etc$

Finally, this is, of course, the tetration sequence for base 10 and can be notated using Knuth arrows as follows:

$1, 10^1, 10 \uparrow\uparrow 2, 10 \uparrow\uparrow 3, 10 \uparrow\uparrow 4, 10 \uparrow\uparrow 5, etc$

In a way, this is our goal – we have considered the natural numbers and their various base and hereditary base representations, the binob array, the stem-2-base-3 sequence, the stalk-0-bud-3 sequence, lexicographic binary-indexed trees, etindao trees, laddered exponents and their expanded forms into etindao trees or almost-etindao trees (where the top exponent is less than the base but bigger than 1) and finally, infinite UR-paths through the standard array. Phew! But there is more to come! In the next section, we look at an amazing sequence that moves far away from tetration, through the esoteric and rarefied realm of the hyperoperator hierarchy. These numbers are too huge and complex to properly fathom, however, with Nopt structure visualisations to show the Nept form, at least for some iterated Knuth arrow numbers, we can begin to appreciate some of the emergent patterns and appreciate a rather wonderful sequence that pads out the hyperoperator hierarchy in quite an amenable way.

Section 2.12 A canonical transition sequence through the hyperoperators

Do you remember the "stem-2-base-3" sequence from section 2.6?
The sequence is supposed to be interpreted in Base3, in other words, the allowable digits are {0,1,2} in the usual order. Here is the sequence:
0, 1, 2, 10, 11, 12, 20, 100, 101, 110, 111, 120, 200, 1000, 1001, 1010, 1011,1100, 1101, 1110, 1111, 1200, 2000, 10000, 10001, 10010, 10011, 10100, …
This sequence can be used to index another sequence that moves through iterated Knuth arrow numbers:
Addition, multiplication and exponentiation (of natural numbers), in the context of hyperoperators, are commonly associated with the numbers 1, 2 and 3. Tetration is the 4^{th} hyperoperator and pentation is the 5^{th} hyperoperator, and so on. Knuth invented his Knuth arrow notation to start with exponentiation, tetration and so on, in order to have a simple notation to refer to the hyperoperators beyond exponentiation.
Remember that tetration is repeated exponentiation, pentation is repeated tetration and so on.
But we can also iterate a hyperoperator. Iteration is what we do before there are so many iterations that it is more convenient to count the number of operations and assign the counts as arguments to a higher order hyperoperator. Let's consider the sequence of natural numbers n=3,4,5,6,7,… but imagine them as referring to hyper(n) operators. Remember also that hyper(3) corresponds with 1 Knuth arrow, hyper(4) has 2 Knuth arrows and so on. For convenience, I'll define the Canonical Transition Sequence (CTS) relative to a fixed number (3) that acts like a "base" or "seed" number. The CTS is related to the Base(m) Ackermann sequence. (See my other papers "Hyperoperations and Nopt structures" and "Hierarchies and Nopt structures" for details about the Base(m) Ackermann number sequence and the role of seed numbers, minimal symbolic notation, Nept form, Nopt structures and Colored Square Diagrams etc.)

Hyper(8)	Hyper(7)	Hyper(6)	Hyper(5)	Hyper(4)
				1
			1	0
			1	1
		1	0	0
		1	0	1
		1	1	0
		1	1	1
	1	0	0	0
	1	0	0	1
	1	0	1	0
	1	0	1	1
	1	0	1	0

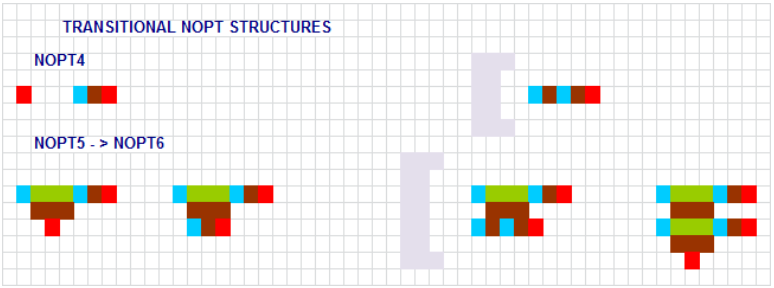
The table above can be used to define a transitional sequence using Knuth arrow notation:

$3\uparrow\uparrow 3,\ \ 3\uparrow\uparrow\uparrow 3,\ \ 3\uparrow\uparrow\uparrow (3\uparrow\uparrow 3),\ \ 3\uparrow^4 3,$

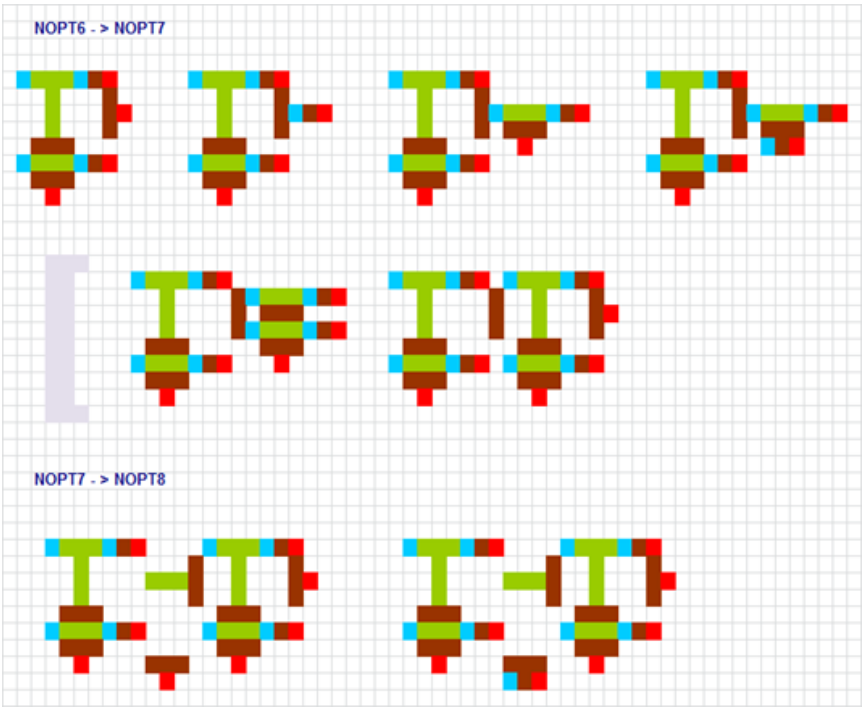
$3\uparrow^4 (3\uparrow\uparrow 3),\ \ 3\uparrow^4 (3\uparrow\uparrow\uparrow 3),\ \ 3\uparrow^4 (3\uparrow\uparrow\uparrow (3\uparrow\uparrow 3)),\ ...$

But the CTS differs a little from this binary-indexed sequence because it is based on the "stem-2-base-3" sequence. Using this sequence provides a more natural transition sequence as can be seen from the pictures below:

The CTS from tetration, pentation and approacing hyper(6)



The CTS from hyper(6), moving to hyper(7) and just beyond:



But the best way to understand this sequence is by observing an animation of it.
I have made one, and it shows how the "stem-2-base-3" sequence is important for allowing smooth transitions. Anyway, it's complicated to show in a document, so if you're feeling lost or confused, you're probably not the only one. The animation does make it clearer, maybe I'll provide a link to it later on.

Part 3 Comparing Bowers array numbers and Naropt structures

Section 3.1 Some Bowers array numbers compared with Naropt structures

Bowers array notation and naropt structures
Bower's operators
The Texan amateur mathematician Jonathan Bowers has devoted a great deal of time into finding and naming ever larger numbers. At time of writing, his largest is a colussus he has called meameamealokkapoowa oompa. Bower's basic idea is a process to far extend Knuth's arrows.
Note that we may write 4{n}3 as short-hand for $4\uparrow\uparrow\uparrow\ldots\uparrow 3$ where there are n arrows.
For still larger numbers, we need more powerful notation, such as Bowers' operators.

His first operator is {{1}} defined by:
m{{1}}2=m{m}m (2x(2-1)=2 braces and 2x2-1=3 "m"s)
m{{1}}3=m{m{m}m}m (2x(3-1)=4 braces and 2x3-1=5 "m"s)
m{{1}}4=m{m{m{m}m}m}m (2x(4-1)=6 braces and 2x4-1=7 "m"s)
In general
m{{1}}n=m{…{m}…}m (2x(n-1) braces and 2n-1 "m"s)

Let's express these as naropts to get a better feel

$$m\{\{1\}\}2 = m\uparrow^{m} m$$

$$m\{\{1\}\}3 = m\uparrow^{m\uparrow^{m} m} m$$

$$m\{\{1\}\}4 = m\uparrow^{m\uparrow^{m\uparrow^{m} m} m} m$$

In general

$$m\{\{1\}\}n = m\uparrow^{\vdots^{m\uparrow^{m} m}\vdots} m\Big\}n \text{ layers}$$

Note that:
m{{1}}m=Naropt(ordertype=4, seed=m, fpt=arrow tower)
This is enough to locate one of the largest of all mathematical constants, Graham's number.
But we can continue with a second operator {{2}} defined by:
m{{2}}2=m {{1}} m
m{{2}}3=m {{1}} (m{{2}}2)
m{{2}}4=m {{1}} (m{{2}}3)

$$m\{\{2\}\}2 = m\{\{1\}\}m = m\uparrow^{\vdots^{m\uparrow^{m} m}\vdots} m\Big\}m \text{ layers}$$

$$m\{\{2\}\}3 = m\{\{1\}\}(m\{\{2\}\}2) = m\uparrow^{\vdots^{m\uparrow^{m} m}\vdots} m\Big\}m\uparrow^{\vdots^{m\uparrow^{m} m}\vdots} m\Big\}m$$

$$m\{\{2\}\}4 = m\{\{1\}\}(m\{\{2\}\}3) = m\uparrow^{\vdots^{m\uparrow^{m} m}\vdots} m\Big\}m\uparrow^{\vdots^{m\uparrow^{m} m}\vdots} m\Big\}m\uparrow^{\vdots^{m\uparrow^{m} m}\vdots} m\Big\}m$$

$$m\{\{2\}\}m = m\{\{1\}\}(m\{\{2\}\}(m-1)) = \underbrace{m\uparrow^{\vdots^{m\uparrow^{m} m}\vdots} m\Big\}\ldots\Big\}m\uparrow^{\vdots^{m\uparrow^{m} m}\vdots} m\Big\}m}_{m}$$

So it is clear that:
m{{2}}m=Naropt(ordertype=5, seed=m, fpt=arrow tower)
Then the operators {{3}}, {{4}}, etc can all be defined analagously.

Let's check the operator {{3}}…

$$m\{\{3\}\}2 = m\{\{2\}\}m = \underbrace{m\uparrow^{\vdots\, m\uparrow^m m}\, m \}\ldots\{ m\uparrow^{\vdots\, m\uparrow^m m}\, m\}m}_{m}$$

$$m\{\{3\}\}3 = m\{\{2\}\}(m\{\{3\}\}2) = \underbrace{m\uparrow^{\vdots\, m\uparrow^m m}\, m \}\ldots\{ m\uparrow^{\vdots\, m\uparrow^m m}\, m\}m}_{\underbrace{m\uparrow^{\vdots\, m\uparrow^m m}\, m \}\ldots\{ m\uparrow^{\vdots\, m\uparrow^m m}\, m\}m}_{m}}$$

$$m\{\{3\}\}4 = m\{\{2\}\}(m\{\{3\}\}3) = \underbrace{m\uparrow^{\vdots\, m\uparrow^m m}\, m \}\ldots\{ m\uparrow^{\vdots\, m\uparrow^m m}\, m\}m}_{\underbrace{m\uparrow^{\vdots\, m\uparrow^m m}\, m \}\ldots\{ m\uparrow^{\vdots\, m\uparrow^m m}\, m\}m}_{\underbrace{m\uparrow^{\vdots\, m\uparrow^m m}\, m \}\ldots\{ m\uparrow^{\vdots\, m\uparrow^m m}\, m\}m}_{m}}}$$

$$m\{\{3\}\}m = \left.\underbrace{\begin{matrix} m\uparrow^{\vdots\, m\uparrow^m m}\, m \}\ldots\{ m\uparrow^{\vdots\, m\uparrow^m m}\, m\}m \\ \vdots \\ \underbrace{m\uparrow^{\vdots\, m\uparrow^m m}\, m \}\ldots\{ m\uparrow^{\vdots\, m\uparrow^m m}\, m\}m}_{m} \end{matrix}}_{m}\right\} m$$

m{{3}}m=Naropt(ordertype=6, seed=m, fpt=arrow tower)
In summary, we have
m{{1}}m=Naropt(ordertype=4, seed=m, fpt=arrow tower)
m{{2}}m=Naropt(ordertype=5, seed=m, fpt=arrow tower)
m{{3}}m=Naropt(ordertype=6, seed=m, fpt=arrow tower)
We begin the next level with {{{1}}}, which behaves in relation to {{-}} as {{-}} does to {-}, and so on. I think this means that:
m{{{1}}}2 = m{{m}}m=Naropt(ordertype=m, seed=m, fpt=arrow tower)
So this is where naropts run out of steam in terms of Bowers' operators.

We can press on, with a new function which counts the brackets: we write
{m,n,p,q} to mean m{{…{p}…}}n where there are q sets of brackets.
Of course, Bowers does not stop here, pushing this line of thought to ever more outrageous heights. But some numbers will always remain out of reach, such as Friedman's TREE(3).
However, after m{{{1}}}2 it is not easy to understand, we can't relate them to hyperoperations or even Knuth arrow towers in a naropt structure.

Part 4 Elaborated examples

It is possible to animate and elaborate various ideas such as:
Computational pathways for ordertypes 4, 5, 6 and 7
Folding patterns for ordertypes 4, 5, 6 and 7
Canonical transition sequence moving through ordertypes 4, 5, 6, 7, 8, 9
I have made some animations and they're kind of pretty.
I hope this paper has been interesting to read!