Hyperoperations and Nopt Structures

Alister Wilson

Abstract (Beta version)

The concept of formal power towers by analogy to formal power series is introduced. Bracketing patterns for combining hyperoperations are pictured. Nopt structures are introduced by reference to Nept structures. Briefly speaking, Nept structures are a notation that help picturing the seed(m)-Ackermann number sequence by reference to exponential function and multitudinous nestings thereof.
A systematic structure is observed and described.

Keywords: Large numbers, formal power towers, Nopt structures.

Contents

i Acknowledgements

I’d like to express my gratitude to Wikipedia for supplying an enormous range of high quality mathematics articles. Also Wolfram Demonstrations and WolframAlpha for providing a superlative math teaching environment and pedagogy. I’d also like to thank tetration forum, xkcd forum and fom archives for the range of discussions in areas of foundation of maths.

ii List of Figures and Tables

I Introduction

In this paper, we think about some very large numbers, and reveal some of the complexity that can sometimes be hidden. A central aspect of this paper is to explore, enumerate and survey some patterns connected with the fast-growing hierarchy and the Ackermann function. The second part of the paper is a sometimes opaque philosophy of a perspective viewpoint about the natural numbers. The third part reminds our readers that for every incremental increase in the hyperoperations, usually symbolised by an additional Knuth arrow, there are lots of numbers that fall between the two natural and workable methods of top-down and bottom-up bracketing. While addition and multiplication have the desirable associative property, exponentiation and above are non-associative and actually the number of different bracketings given an n-fold exponential expression is the same as the Catalan numbers. Having said that, most of mathematics doesn't need to concern itself with this kind of complexity, and indeed it is usually too complex to deal with anway.

The Nept and Nopt structures introduced in the fourth part of this paper are mostly a tool to help clarify and guide thinking about fast-growing structures, such as the hyperoperation hierarchy, Ackermann numbers, functional powers, slow and fast growing hierarchies, hereditary base representations, ordinal numbers and notations, Cantor Normal Form, the Veblen hierarchy, and in general, indicate some patterns in common and differences between the finite and infinite realm. Nopt structures emphasise the nestedness of recursion structures. Usually in maths, the nestedness information is somewhat hidden, by the use of new, derived symbols and definitions from other previously derived definitions and symbols in order to create various notations. Nept structures retain structural information that is not accounted for by Knuth arrow notation or other notations related to hyperoperators. And an interesting result is that although the Ackermann function increases faster than any of the hyperoperators in the hyperoperator hierarchy, by representing Ackermann numbers, or base(m) Ackermann numbers into Nept structures, with the technique of a consistent minimal symbolic notation, the growth rate of the notation is exponential. The technique of finding a minimal symbolic notation to capture essential computational information and generalise the notation helps to reveal the computational pathway connecting the constituent components. A useful observation, is that Nept structures can resolve the hyperoperator hierarchy to multi-layered nestings of exponential power towers.

The fifth part considers properties of the Conway arrow numbers by connecting them with the Knuth arrow numbers that follow the hyperoperation hierarchy, and the tower structures formed from Knuth arrow numbers. The interesting result here, is that there is a sequence from the Conway numbers that has a similar role to the Ackermann numbers. By the way Ackermann numbers are defined, it is clear they form a diagonal sequence through the hyperoperator hierarchy. By suitably defining a Nopt structure based on Knuth arrows (that I call Naropt structures, or nested arrow power towers) we find that there is a sequence of Conway numbers that form a diagonal sequence through the hierarchy of Naropt structures.

Apart from a tool to guide thinking about recursion structures it is not easy to think of useful applications. But by reinterpreting or modifying parameters and patterns in Nopt structures they may prove to be useful in other areas of maths and possibly in computer engineering as well.

II Philosophical Considerations

The following is the standard definition of N=Natural Numbers.
"Peano's successor function S(n) = n+1 uniquely covers all numbers 1.. starting from n=0 by iteration of S, and thereby defines the set of natural numbers."
To understand N better and more accurately consider
N = {N[T, SPN]} U {N[UDC]}
T = Tally; SPN = Standard Positional Notation
UDC = Unbounded Descriptive Capability
This mysterious formula acknowledges the three viewpoints about N
N[T] → As a tally without bound
(Self-referential concept about the action of counting and size representation)
N[T, SPN] → As a tally without bound OR As an SPN digit sequence without bound
(This is what people think of about N, in normal, practical situations.)
N[UDC] → Counting numbers with Unbounded Descriptive Capability
(For example, Scientific Notation, power towers etc are an extended conception of N.)
Counting numbers, in a basic and fundamental way, serve the purpose of indexing and sequencing items, entities etc and one possible way to provide lexicographic ordering.
The tally system allows the fundamental process of pointing to an item, incrementing a counter (recording the presence of the item) changing status from unread to read or removing same item from a collection set.
In this way items in the collection can be counted.
The Thoroughness Property is evident, we believe that incrementing is the unambiguous, systematic method that counts things one-by-one forever.
The reality is that "forever" should be relativised to mean towards a "horizon", that is not well-defined but accurately represents an intuition about self-reference regarding "quantity" (the transitions betweeen initial tallying, SPN, and digits-in-sequence tally), and intuition concerning "large enough".
N[T, SPN] gives the unestimable advantage of allowing a sensible method of Information Condensation while retaining Thoroughness Property.
With SPN, "large enough" can be made small by use of "base" and in so doing frees-up "large enough" to be controlled by other aspects of the description.
And so "large enough" is now the consideration of number of digits in the SPN sequential presentation.
With N[UDC] we have an Emerging Trade-Off between Thoroughness Property and Unbounded Descriptive Capability.
When considering the various big numbers more information is directed towards magnitude and less towards fine details. This is a trade-off between Descriptive Capability and Thoroughness Property. It is an emerging trade-off because there are phase transitions in the trade-off.
For example: A googol is both SPN-describable (a digit sequence of 1 followed by 100 zeros) and UDC-possible (10^100). A googolplex is not SPN-describable but is UDC-possible (10^(10^100)). For numbers between googol and googolplex it is hard to maintain Thoroughness Property. Introducing treelike structures such as HBN (Hereditary Base N) is an attempt to recapture Thoroughness Property but at the expense of greater structural pattern complexity.
Similar phenomena can be observed in the discussions concerning the infinite ordinals.

Considering large numbers and fast growing functions gives a dual reality:
A) A tangible magnitude-into-pattern transformation
B) Traditional perspective of ever-increasing patterns of magnitude

FIFF paradigm
Fuzzy Infinite Fuzzy Finite paradigm
Finite, infinite dichotomy, that is: {1,…,n} versus {1,…}
The appearance seems clear and unambiguous
But this viewpoint is biased by the dominant SPN perspective
And the evidence from considering Nopt structures shows it is a false dichotomy.

Some of the transitions:
SPN shows exponentiation as an incremental add-one-digit way. Knuth arrow notation shows Ackermann function as an incremental add-one-to-tally way.
Nopt structures show that Ackermann numbers increase exponentially with respect to Minimal Symbolic Notational requirements on a level playing field (the benchmark or yardstick of using multi-level nested layers with a fixed operation, and power towers)
Nopt structures use a sensible minimal symbolic representation.
Next stage is "zooming in on" HEFTY Nopt structure with a microscope!
Can then introduce another level of chunkiness by storing High Resolution HEFTY Nopts into little boxes… And start the process again…
And so on into the ethereal realms of incomprehensible vastness…

The Inevitable chunkiness of large numbers
In the consideration of fast growing functions there is an inevitable chunkiness that comes about due to natural limit of descriptive capabilities.
You can start out slow with 1, 2, 3 and successor function or fast with Graham's number and g-subscript power towers but the contemplation of pushing out further into the endless unboundedness of infinity calls upon chunkiness.
NOPT structures are dimensionless until an operation is specified, but even though they are dimensionless, structure can be identified and codified.
Reaching out further and more information hiding is natural and unavoidable.
What about the huge wealth of numbers between g1 and g2 from the gi-sequence leading to Graham's number? We could traverse the intermediary space by applying the standard math integer functions to the Knuth arrows. And to do this all the structure leading up to "3 hexated to 3' could be replicated but this time applying to number of Knuth arrows, padding out the hyperoperator hierarchy to dizzying realms.
The beauty of SPN (standard positional notation) numbers is they preserve the initial successor function, increment natural and successive orders of magnitude while retaining condensation property for as long as a string of digits can go.
A number of visualisations from Wolfram demonstrations show cellular automata applied to binary numbers or other base numbers; we see the condensation property that is systematic reuse and exhaustion of previous orders of magnitude.
An understanding of hyperoperators coded into NOPT structures also has systematic reuse of orders of magnitude but the condensation or thoroughness property from initial successor function is necessarily relaxed. By using exponentiation and the power towers thereof inside a NOPT structure we have the NEPT structures, and now the notion of "successor" is transformed or transmuted into "adjacent power tower".

The normal successor function we are so familiar with, that is counting distinct symbols is retained and distinctly present in the NEPT structure but we are counting Power towers symbolically adjacent to one another.

(From some perspectives, in some ways, the traditional finite, infinite separation in maths is flawed, we should think in terms of required chunkiness and layers of nestation.)

A heptation NOPT structure also contains NOPT structures of all previous orders, that is to say, hexation, pentation, tetration, exponentiation are also present and part and parcel of heptation structure. A number such as 53,672 is a 10^4 order number, and also contains information about previous orders of magnitude. A nonation number requires octation, heptation, hexation, pentation, tetration, exponentiation.

Once we get to the gi-sequence it is like a hyperdrive of magnitude that shows the transition between magnitude and pattern.

III Bracketing patterns and hyperoperations

Section 3.1 Some Examples

Take your favorite number such as 256.
256=200+50+6=128*2=64*4=32*8=16*16.
These re-expressions of 256 involve addition and multiplication.
Concerning exponentiation we have:

$$2^8 = (2^4)^2 = 16^2 = 256 = 2^{2^3} = 2^{(2^3)}$$

Concerning iterated exponentiation we have:

$$256 = 16^2 = (2^4)^2 = (2^{2^2})^2 = (2^{2.2})^2 = ((2^2)^2)^2 = (4^2)^2 = 4^{(2.2)} = 4^4$$

Concerning tetration we have:

$$256 = 16^2 = ({}^3 2)({}^3 2) = ({}^3 2)^2$$

Now let’s look at some tetration and exponentiation with bracketing pattern examples:

Table 3.1.1 Tetration examples

Bracketing pattern	Tetration examples	Value
(a b) c	${}^2({}^2 2) = {}^2 4 = 4^4$	256
a (b c)	${}^{({}^2 2)}2 = {}^4 2 = 2^{2^{2^2}}$	$65536 = 2^{16}$

Table 3.1.2 Exponentiation examples

Bracketing pattern	Exponentiation examples	Value
((a b) c) d	$((3^3)^3)^3 = 3^{(3^3)}$	3^{27}
(a b) (c d)	$(3^3)^{(3^3)} = 27^{27}$	27^{27}
(a (b c)) d	$(3^{(3^3)})^3 = (3^{27})^3 = 3^{81}$	3^{81}
a ((b c) d)	$3^{((3^3)^3)} = 3^{(27^3)}$	$3^{(27^3)}$
a (b (c d))	$3^{3^{3^3}} = 3^{3^{27}}$	$3^{3^{27}}$

Table 3.1.3 More tetration examples

Bracketing pattern	Tetration examples	Value
((a b) c) d	${}^{2}({}^{2}({}^{2}2)) = (4^{4})^{(4^{4})}$	256^{256}
(a b) (c d)	${}^{({}^{2}2)}({}^{2}2)={}^{4}4 = 4^{4^{4^{4}}}$	$4^{4^{256}}$
(a (b c)) d	${}^{2}({}^{({}^{2}2)}2))={}^{2}(65536)={}^{2}(2^{16})$	$65536^{65536} = (2^{16})^{2^{16}}$
a ((b c) d)	${}^{2({}^{2}2))}2={}^{256}2 = 2^{\cdot^{\cdot^{2}}}\}256$	$2^{\cdot^{\cdot^{2}}}\}256$
a (b (c d))	${}^{{}^{{}^{2}2}2}2={}^{{}^{4}2}2={}^{(2^{16})}2 = 2^{\cdot^{\cdot^{2}}}\}2^{16}$	$2^{\cdot^{\cdot^{2}}}\}2^{16}$

Section 3.2 Top-down versus bottom-up

TD = Top Down bracketing
BU = Bottom Up bracketing

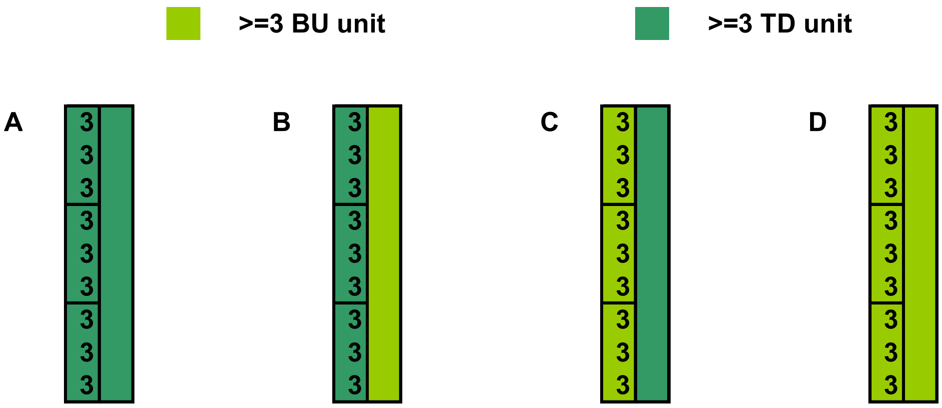
Figure 3.2.1 Mixed Top-down and Bottom-up examples

TD THEN TD
A (3^(3^3)) ^ ((3^(3^3)) ^ (3^(3^3)))

TD THEN BU
B ((3^(3^3)) ^ (3^(3^3))) ^ (3^(3^3))

BU THEN TD
C ((3^3)^3) ^ (((3^3)^3) ^ ((3^3)^3))

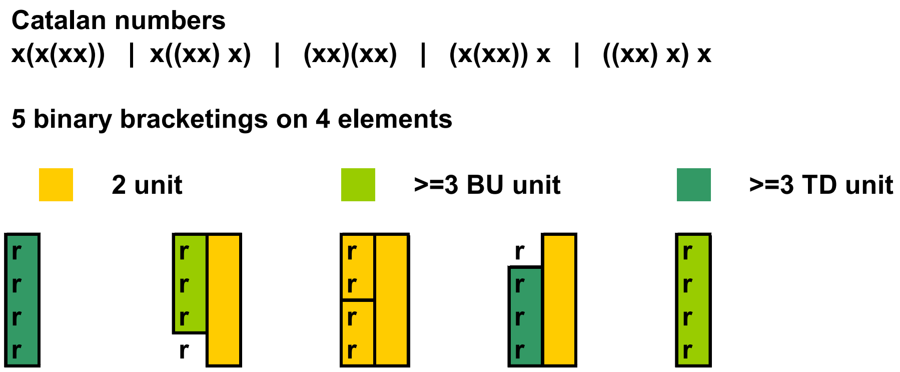
BU THEN BU
D (((3^3)^3) ^ ((3^3)^3)) ^ ((3^3)^3)



Section 3.3 Bracketing patterns and binary operations

We can represent a bracketing pattern in the following BUTDJ way:
BU and TD components (>= 3) together with Joins of 2 components.
The innermost components on the left and outermost components on the right.
The “r ” represents an arbitrary symbol or number.

Figure 3.3.1 The 5 binary bracketings on 4 elements.



The Catalan numbers have the following recurrence relation:

$$C_0 = 1 \quad and \quad C_{n+1} = \sum_{i=0}^{n} C_i C_{n-i} \quad for \quad n \geq 0$$

Starting from n=0, the first few Catalan numbers are: 1, 1, 2, 5, 14, 42, 132, …

Table 3.3.1 Exponential representation of the 14 binary bracketings on 5 elements

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

Figure 3.3.2 Corresponding butdj-representation of these 14 binary bracketings

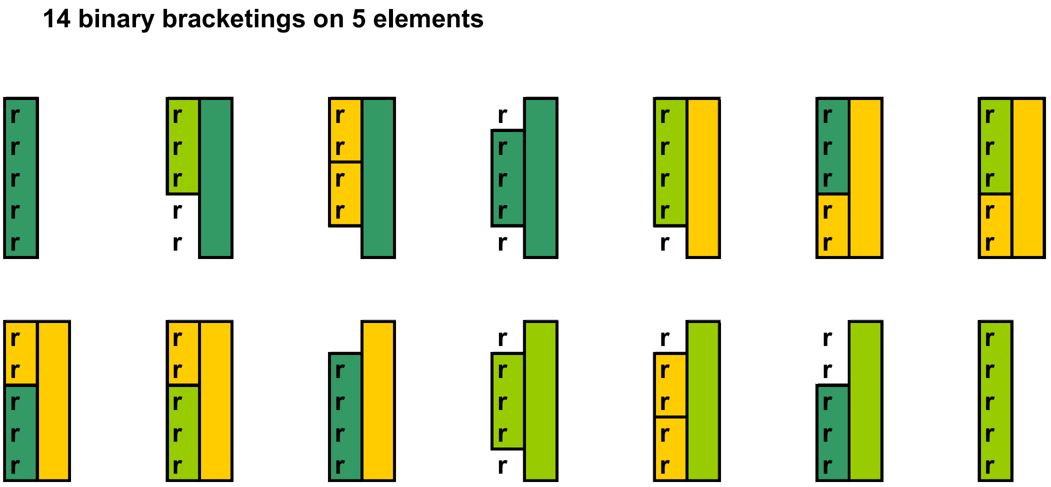
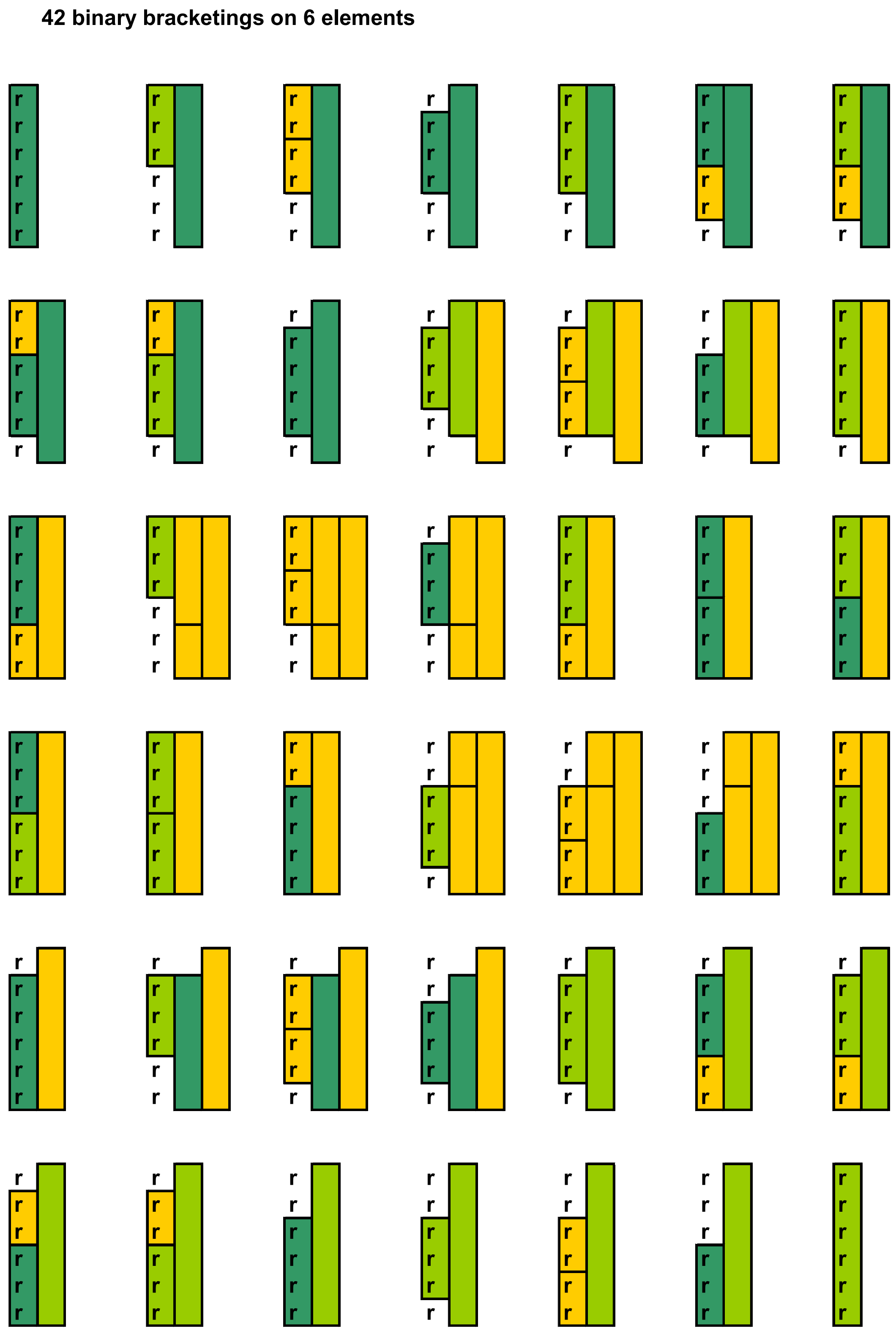


Figure 3.3.3 butdj-representations of the 42 binary bracketings on 6 elements



Section 3.4 Bracketing patterns with exponentiation and tetration

Table 3.4.1 4 element, 3 operations, mixed exponentiation-tetration examples

I use base_number=2 in all the examples, other small numbers, eg bn=3 are suitable.

Catalan numbers

xxxx | (xx) xx | x(xx) x | xx(xx) | (xxx) x | x(xxx)
((xx) x) x | (x(xx)) x | (xx)(xx) | x((xx) x) | x(x(xx))
the last five of which are binary.
5 binary bracketings on 4 elements

<3,3,3>					
	$((2^2)^2)^2$	$(2^{(2^2)})^2$	$(2^2)^{(2^2)}$	$2^{((2^2)^2)}$	$2^{(2^{(2^2)})}$
<3,3,4>					
	${}^{2}((2^2)^2)$	${}^{2}(2^{(2^2)})$	$(2^2)^{({}^{2}2)}$	$2^{({}^{2}(2^2))}$	$2^{(2^{({}^{2}2)})}$
<3,4,3>					
	$({}^{2}(2^2))^2$	$(2^{({}^{2}2)})^2$	${}^{(2^2)}(2^2)$	$2^{(({}^{2}2)^2)}$	$2^{({}^{(2^2)}2)}$
<3,4,4>					
	${}^{2}({}^{2}(2^2))$	${}^{2}(2^{({}^{2}2)})$	${}^{({}^{2}2)}(2^2)$	$2^{({}^{2}({}^{2}2))}$	$2^{({}^{({}^{2}2)}2)}$
<4,3,3>					
	$(({}^{2}2)^2)^2$	$({}^{(2^2)}2)^2$	$({}^{2}2)^{(2^2)}$	${}^{((2^2)^2)}2$	${}^{(2^{(2^2)})}2$
<4,3,4>					
	${}^{2}(({}^{2}2)^2)$	${}^{2}({}^{(2^2)}2)$	$({}^{2}2)^{({}^{2}2)}$	${}^{({}^{2}(2^2))}2$	${}^{(2^{({}^{2}2)})}2$
<4,4,3>					
	$({}^{2}({}^{2}2))^2$	$({}^{({}^{2}2)}2)^2$	${}^{(2^2)}({}^{2}2)$	${}^{(({}^{2}2)^2)}2$	${}^{({}^{(2^2)}2)}2$
<4,4,4>					
	${}^{2}({}^{2}({}^{2}2))$	${}^{2}({}^{({}^{2}2)}2)$	${}^{({}^{2}2)}({}^{2}2)$	${}^{({}^{2}({}^{2}2))}2$	${}^{({}^{({}^{2}2)}2)}2$

Table 3.4.2 4 element, 3 operations, mixed (hyp=3,4) and with comparisons

<3,3,3> [Note: 65,536 = 2^16]

$((2^2)^2)^2$	$(2^{(2^2)})^2$	$(2^2)^{(2^2)}$	$2^{((2^2)^2)}$	$2^{(2^{(2^2)})}$
=256	=256	=256	=65,536	=65,536

<3,3,4>

${}^{2}((2^2)^2)$	${}^{2}(2^{(2^2)})$	$(2^2)^{({}^{2}2)}$	$2^{({}^{2}(2^2))}$	$2^{(2^{({}^{2}2)})}$
=16^16	=16^16	=256	=2^256	=2^16

<3,4,3> [Note: 4^^4 = 4^(4^256) and 2^^5 = 2^(2^16)]

$({}^{2}(2^2))^2$	$(2^{({}^{2}2)})^2$	${}^{(2^2)}(2^2)$	$2^{(({}^{2}2)^2)}$	$2^{({}^{(2^2)}2)}$
=2^16	=256	=4^(4^256)	=2^16	=2^^5

<3,4,4>

${}^{2}({}^{2}(2^2))$	${}^{2}(2^{({}^{2}2)})$	${}^{({}^{2}2)}(2^2)$	$2^{({}^{2}({}^{2}2))}$	$2^{({}^{({}^{2}2)}2)}$
=256^256	=16^16	=4^(4^256)	=2^256	=2^^5

<4,3,3> [Note: (2^16)^2 = 2^32 and 2^^16 = 2./.2)16]

$(({}^{2}2)^2)^2$	$({}^{(2^2)}2)^2$	$({}^{2}2)^{(2^2)}$	${}^{((2^2)^2)}2$	${}^{(2^{(2^2)})}2$
=256	=2^32	=256	=2./.2)16	= 2./.2)16

<4,3,4>

${}^{2}(({}^{2}2)^2)$	${}^{2}({}^{(2^2)}2)$	$({}^{2}2)^{({}^{2}2)}$	${}^{({}^{2}(2^2))}2$	${}^{(2^{({}^{2}2)})}2$
=16^16	=(2^16)^^2	=256	=2./.2)256	= 2./.2)16

<4,4,3>

$({}^{2}({}^{2}2))^2$	$({}^{({}^{2}2)}2)^2$	${}^{(2^2)}({}^{2}2)$	${}^{(({}^{2}2)^2)}2$	${}^{({}^{(2^2)}2)}2$
=2^16	=2^32	=4^(4^256)	= 2./.2)16	= 2./.2)16

<4,4,4>

${}^{2}({}^{2}({}^{2}2))$	${}^{2}({}^{({}^{2}2)}2)$	${}^{({}^{2}2)}({}^{2}2)$	${}^{({}^{2}({}^{2}2))}2$	${}^{({}^{({}^{2}2)}2)}2$
=256^256	=(2^16)^^2	=4^(4^256)	=2./.2)256	= 2./.2)2^16

Ordering the inequalities:
256 < **2^16** < 2^32 < 16^16 < 2^256 < **256^256**
< 2^(2^16) < (2^16)^^2 < 4^(4^256) < 2./.2)16 < 2./.2)256 < **2./.2)2^16**
Notice the values in bold:
256 (bu itera exp) 2^16 (td itera exp)
256^256 (bu itera tetra) 2./.2)2^16 (td itera tetra)

Figure 3.4.3 Diagrams of the 4 element, 3 operations, mixed (hyp=3,4) examples

((xx) x) x | (x(xx)) x | (xx)(xx) | x((xx) x) | x(x(xx))
5 binary bracketings on 4 elements

number **exp** **tet**
bu **td** **join**

Section 3.5 Bracketing and 4 consecutive hyperoperations

Table 3.5.1 3 elements and 2 binary operations by 4 hyperoperations

I use base_number=3 in the examples, other small numbers, eg bn=2 are suitable.

<3,td>

$3^{(3^3)}$ $3^{({}^3 3)}$ $3^{({}_3 3)}$ $3^{(3_3)}$

<3,bu>

$(3^3)^3$ ${}^3(3^3)$ ${}_3(3^3)$ $(3^3)_3$

<4,td>

${}^{(3^3)}3$ ${}^{({}^3 3)}3$ ${}^{({}_3 3)}3$ ${}^{(3_3)}3$

<4,bu>

$({}^3 3)^3$ ${}^3({}^3 3)$ ${}_3({}^3 3)$ $({}^3 3)_3$

<5,td>

${}_{(3^3)}3$ ${}_{({}^3 3)}3$ ${}_{({}_3 3)}3$ ${}_{(3_3)}3$

<5,bu>

$({}_3 3)^3$ ${}^3({}_3 3)$ ${}_3({}_3 3)$ $({}_3 3)_3$

<6,td>

$3_{(3^3)}$ $3_{({}^3 3)}$ $3_{({}_3 3)}$ $3_{(3_3)}$

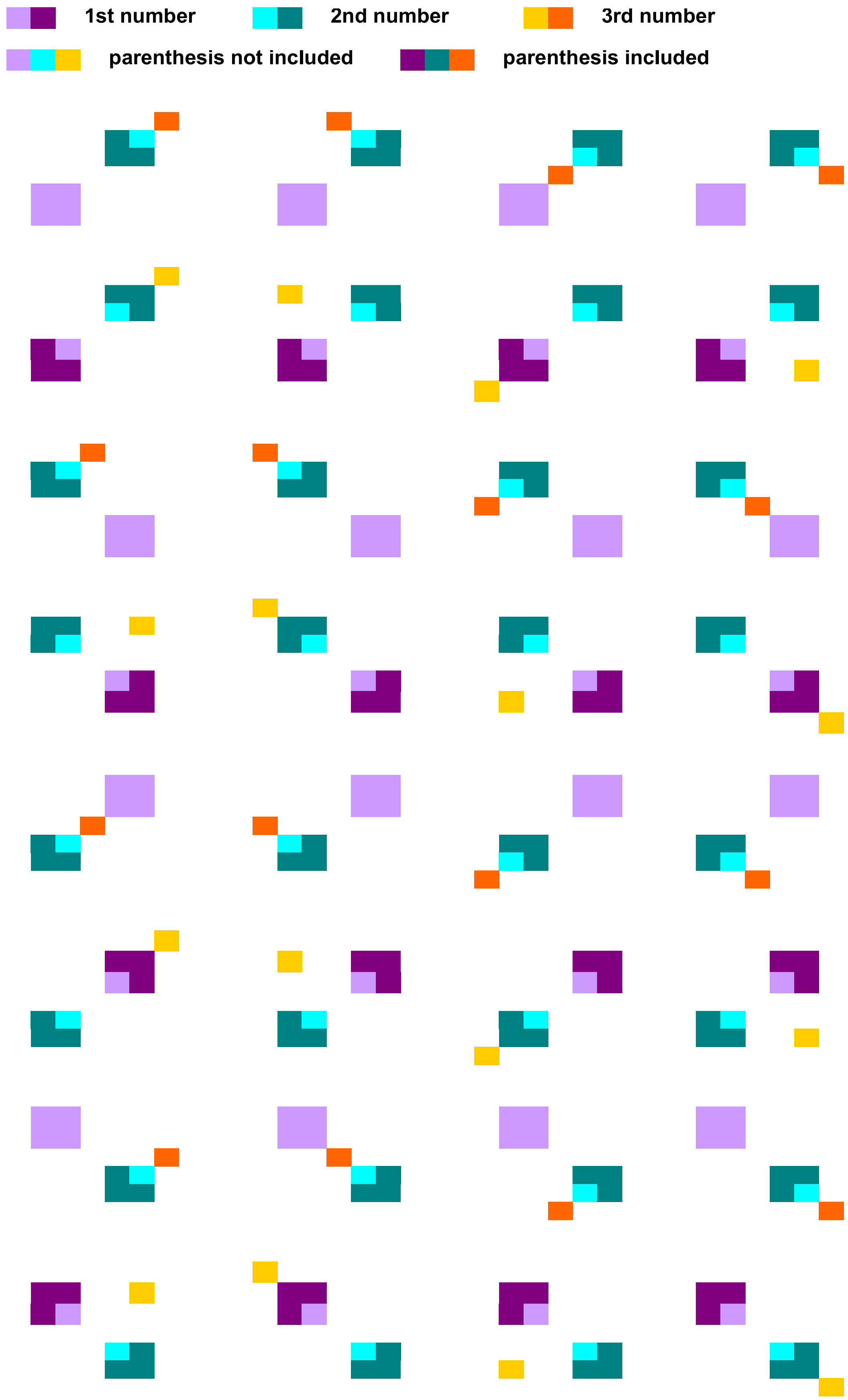
<6,bu>

$(3_3)^3$ ${}^3(3_3)$ ${}_3(3_3)$ $(3_3)_3$

Using a standard notation for hyperoperations (3,4,5,6), and moving in a direction that is counter-clockwise from the top-right corner, the operations being used are: exponentiation, tetration, pentation and hexation. Of course, pentation and hexation produce very big numbers indeed.
These numbers are way, way too big for computers to resolve into a string of digits.
We can contemplate the patterns produced and admire the complexity.
How to read these expressions – some selected examples:

$3^{({}_3 3)}$	" 3 exponentiated to {3 pentated to the 3} "
$({}_3 3)^3$	" {3 pentated to the 3} exponentiated to the 3 "
${}^3(3_3)$	" {3 hexated to the 3} tetrated to the 3 "

Figure 3.5.2 A colored square diagram of Table 3.5.1:



Section 3.6 A quick look at the start of the Grzegorczyk hierarchy:

The functions at finite levels $(\alpha < \omega)$ of any fast-growing hierarchy coincide with those of the Grzegorczyk hierarchy:

$$f_0(n) = n + 1$$
$$f_1(n) = f_0^n(n) = n + n = 2n$$
$$f_2(n) = f_1^n(n) = 2^n n$$

Moving from $f_2(n)$ to $f_3(n) = f_2^n(n)$ is quite a lot more complicated:

$$f_2^2(n) = (2^{2^n n})(2^n n)$$

$$f_2^3(n) = 2^{(2^{2^n n})(2^n n)}(2^{2^n n})(2^n n)$$

$$f_2^4(n) = 2^{2^{(2^{2^n n})(2^n n)}(2^{2^n n})(2^n n)} 2^{(2^{2^n n})(2^n n)}(2^{2^n n})(2^n n)$$

If we let:

$$a_1 = 2^n n \ , \ a_2 = 2^{2^n n} = 2^{a_1} \ , \ a_3 = 2^{(2^{2^n n})(2^n n)} = 2^{a_2 a_1} \ , \ldots$$

We can find an expression for $f_2^n(n)$:

$$f_3(n) = f_2^n(n) = 2^{a_{n-1} \cdots a_3 a_2 a_1} . \cdots . 2^{a_3 a_2 a_1} 2^{a_2 a_1} 2^{a_1} a_1$$

Obviously, the complexity increases quite dramatically so usually inequalities are used when discussing the Grzegorczyk hierarchy.

Section 3.7 Reconsidering top-down and bottom-up

As the example from Grzegorczyk hierarchy shows, the complexity is apparent when creating fast-growing hierarchies from functional powers.
Even starting from a simple formula f_0 = n + 1, and by the time you get to f_3, the unwieldy nature becomes apparent.
Hyperoperations use top-down as the natural method and top-down is the default bracketing for hyperoperations. For example, repeated tetration is pentation.
We prefer to use the term "iterated tetration" to refer to tetration when using the bottom-up bracketing method. Bottom-up method, in particular, bottom-up pentation (or iterated tetration) is the method that is used to generate the epsilon numbers (at least, up to e_w).

So let's consider an example of bu-pentation on a small finite number:

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

Now this is tetration or repeated exponentiation so the natural bracketing is used:

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

Now let's iterate taking the current number and tetrating it to 3.

$$^{3}(^{3}3) = (3^{3^{3}})^{(3^{3^{3}})^{(3^{3^{3}})}}$$ = y, say.

$$^{3}(^{3}(^{3}3)) = [y]^{[y]^{[y]}}$$ = z, say.

$$^{3}(^{3}(^{3}(^{3}3))) = \{z\}^{\{z\}^{\{z\}}}$$ and let's stop here.

The point being that the picture developing is fractaline in nature.
Actually, this is the same kind of picture that forms when considering the epsilon numbers:

$$\omega \uparrow\uparrow \omega = \omega^{\omega^{\omega^{\cdot^{\cdot^{\cdot}}}}} = \varepsilon_0$$

$$(\omega \uparrow\uparrow \omega) \uparrow\uparrow \omega = \varepsilon_0^{\varepsilon_0^{\varepsilon_0^{\cdot^{\cdot^{\cdot}}}}} = \varepsilon_1$$

$$((\omega \uparrow\uparrow \omega) \uparrow\uparrow \omega) \uparrow\uparrow \omega = \varepsilon_1^{\varepsilon_1^{\varepsilon_1^{\cdot^{\cdot^{\cdot}}}}} = \varepsilon_2$$

$$(((\omega \uparrow\uparrow \omega) \uparrow\uparrow \omega) \uparrow\uparrow \omega) \uparrow\uparrow \omega = \varepsilon_2^{\varepsilon_2^{\varepsilon_2^{\cdot^{\cdot^{\cdot}}}}} = \varepsilon_3$$

A thought about various approaches to large numbers.

Addition and multiplication are commutative and associative.
But exponentiation, tetration, and so on, are not. Some examples:
3^4=3*3*3*3=81 whereas 4^3=4*4*4=64
2^(3^4)=2^81= 2,417,851,639,229,258,349,412,352 whereas (2^3)^4 = 8^4 = 4,096
3^^2 = 3^3 = 27 whereas 2^^3 = 2^(2^2) = 2^4 = 16
2^^(2^^2) = 2^^4 = 2^(2^(2^2)) = 2^16 = 65,536 whereas (2^^2)^^2=4^^2=4^4=256
For exponentation, tetration and so on, different bracketing produces different results.

Patterns to large numbers:

Abbreviations: Fin=finite, inf=infinite, bu=bottom-up method, td=top-down method
a. (fin, td): <3^^3, 3^^(3^^3), 3^^(3^^(3^^3)), …>
b. (fin, bu): <3^^3, (3^^3)^^3, ((3^^3)^^3)^^3, …>
c. (inf, td): <w^^w, w^^(w^^w), w^^(w^^(w^^w)), …>
d. (inf, bu): <w^^w, (w^^w)^^w, ((w^^w)^^w)^^w, …>
So pentation (the next operation after tetration) uses **method a.** from above:
3^^^2=3^^3, 3^^^3=3^^(3^^3), 3^^^4=3^^(3^^(3^^3))
And **method d.** corresponds to the epsilon sequence: <e0, e1, e2, …>
It is curious that finite numbers via hyperoperations use the top-down method
Whereas infinite ordinals via epsilon numbers use the bottom-up method.
The other **methods b. and c.** are also possible to consider.

Table 3.7.1 A comparison of bu and td iterated tetration

The comparison below shows how compared to regular pentation, bu-pentation has a more complicated, fractaline nature, with nested layers of parentheses.

Bu-pentation	td-pentation
(2^^2)^^2 = (2^2)^(2^2)	2^^(2^^2) = 2^2^2^2
((2^^2)^^2)^^2 = ((2^2)^(2^2))^((2^2)^(2^2))	2^^(2^^(2^^2)) = 2^2^…(2^16)…^2^2

IV Nopt Structures

Section 4.1 Introduction to Nept and Nopt structures

If you’re reading this paper, and fairly new to hyperoperations, maybe you should try to “figure out” with pencil and paper some examples such as 3^3, 3^^3, 3^^^3, and 3^^^^3 in order to get some intuition about the patterns I’ll be discussing below.
The way I suggest, is to use the notations a^a, ${}^a a$, ${}_a a$, *and* a_a, for exponentiation, tetration, pentation and hexation, and see how they relate to each other, and how these hyperoperations can be reexpressed as nestings of exponential power towers.
I hope that the ideas about Nept and Nopt structures show some new ways of looking at some familiar things, and that with some patience, the material below can be understood and is reasonably clear. The Glossary of terms provides additional explanations of the terminology and definitions.

A NEPT structure, or Nested Exponentional Power Tower structure is the representation of a large number as multi-layered nested exponentional power towers.
A NOPT structure, or Nested Operational Power Tower structure is a generalisation of multi-layered nested exponential power towers.

Usually we think of power towers where the operation is exponentiation, but the concept of nested power towers could be used with other operations from the hyperoperation hierarchy as well as any others where height of power tower is well-defined and each power tower in the expression produces a natural number and the operation is a strictly increasing function on the natural numbers.

Other operations are possible for putting into nested power towers for example, such as (+, iterated addition) and (*, iterated multiplication) and (^^, tetration, iterated tetration) and so on. Considering very large numbers, other applications of NOPT structures are possible, by using powerful operations such as the g_i-sequence from Graham’s number construction, where $g_1 = 3 \uparrow\uparrow\uparrow\uparrow 3 = 3 \uparrow^4 3$, and g_{64} = Graham’s number. Even more powerful operations can be considered, such as g-subscript towers where each “subscript tower” has a well-defined height (of nested g-subscript symbols, ending with g_1) (see Section 5.1).

Section 4.2 Defining Nopts from Nepts

In this section, we see there is a natural way to define Nopts from Nepts.
By way of motivation, we need to consider the Ackermann number sequence and the Base(m)-Ackermann number sequence that is related to the Ackermann number sequence.
The notation $m\uparrow\uparrow\ldots\uparrow n$ was introduced in 1976 by Donald Knuth.
A similar function was defined by W. Ackermann in 1928.
The Ackermann numbers are the numbers
$1\uparrow 1,\ 2\uparrow\uparrow 2,\ 3\uparrow\uparrow\uparrow 3,\ 4\uparrow\uparrow\uparrow\uparrow 4,\ldots$
The first Ackermann number is 1, the second is 4, and the third is

$$3^{3^{\cdot^{\cdot^{3}}}}\Big\}\,3^{3^{3}}$$

So in words, the third Ackermann number is a tower of threes where the number of threes is:

$$3^{3^{3}} = 7{,}625{,}597{,}484{,}987.$$

Before we look at the fourth Ackermann number it's worth comparing
$3\uparrow\uparrow\uparrow 3$ with $4\uparrow\uparrow\uparrow 4$ which can be written:

$$4\uparrow\uparrow\uparrow 4=\underbrace{4^{\cdot^{\cdot^{4}}})4^{\cdot^{\cdot^{4}}})4^{\cdot^{\cdot^{4}}})4}_{4}$$

Anyway, the fourth Ackermann number can be written as nested layers of nestings of exponential power towers as follows:

$$4\uparrow\uparrow\uparrow\uparrow 4=\underbrace{4^{\cdot^{\cdot^{4}}})\cdots)4^{\cdot^{\cdot^{4}}})4}_{\underbrace{4^{\cdot^{\cdot^{4}}})\cdots)4^{\cdot^{\cdot^{4}}})4}_{\underbrace{4^{\cdot^{\cdot^{4}}})4^{\cdot^{\cdot^{4}}})4^{\cdot^{\cdot^{4}}})4}_{4}}}$$

Related to the Ackermann number sequence is the Base(m)-Ackermann number sequence: $m\uparrow m,\ m\uparrow\uparrow m, m\uparrow^{3} m,\ldots$. Consider an example with m=3.

The first number in this sequence is $3^{3}=27$ and this number is small and is easy to write using place value notation or standard positional notation.

The second number is ${}^{3}3=3^{3^{3}}=3^{27}=7625597484987$
The second base(3) Ackermann number can be reexpressed slightly differently to emphasize a structural pattern:

$$3^{3^{3}}\Big\}\,3$$

The third base(3) Ackermann number is

$$3\uparrow\uparrow\uparrow 3=\underbrace{3^{\cdot^{\cdot^{3}}})\,3^{3^{3}})\,3}_{3}$$

The canonical Nopt structure is the Nept structure where the operation being used is exponentiation. The canonical relationship between Nept and Nopt structures is that

an OrderType of a Nopt structure can be well-defined by comparison with a Nept structure. In other words, the OrderType of a Nopt structure borrows from the canonical Nept structure related to some hyperoperation:
Hyperoperation ($n \geq 4$) < - > NEPT ($n \geq 4$) < - > NOPT ($n \geq 4$)

The Nopt structures with OrderType 4 or 5 correspond to the appearance of tetration and pentation numbers when they are written into Nept form.
Nopt structures with OrderType 4 or 5 are (the only) Linear NOPT structures.
We can use a sensible notation, a minimal symbolic notation, to represent these Nopt structures:

Tetration nopt: $\theta)n$

Pentation nopt: $\underset{\breve{n}}{\theta\ldots\theta)n}$

Hexation NOPT structure has OrderType 6 and has 2-dimensional array structure:

$$\begin{array}{cc}
\theta\ldots\theta)n & \\
\vdots &)n \\
\overset{\breve{\wedge}}{\longleftrightarrow} & \\
\theta\ldots\theta)n & \\
\breve{n} &
\end{array}$$

Hepation NOPT structure has OrderType 7 and has (2,1)-dimensional array structure.

$$\begin{array}{cccc}
\theta\ldots\theta)n & & \theta\ldots\theta)n & \\
\vdots & \ldots\} & \vdots &)n \\
\leftrightarrow & & \leftrightarrow & \\
\theta\ldots\theta)n & \breve{n} & \theta\ldots\theta)n & \\
{\scriptstyle \breve{n}} & & {\scriptstyle \breve{n}} &
\end{array}$$

Octation NOPT structure has OrderType 8 and has (2,2)-dimensional array structure.

$$\begin{array}{ccccc}
\theta\ldots\theta)n & & \theta\ldots\theta)n & & \\
\vdots & \ldots\} & \vdots &)n & \\
\leftrightarrow & & \leftrightarrow & & \\
\theta\ldots\theta)n & & \theta\ldots\theta)n & & \\
{\scriptstyle \breve{n}} & & {\scriptstyle \breve{n}} & & \\
 & \vdots & & &)n \\
 & \overset{\breve{\wedge}}{\longleftrightarrow} & & & \\
\theta\ldots\theta)n & & \theta\ldots\theta)n & & \\
\vdots & \ldots\} & \vdots &)n & \\
\leftrightarrow & & \leftrightarrow & & \\
\theta\ldots\theta)n & \breve{n} & \theta\ldots\theta)n & & \\
{\scriptstyle \breve{n}} & & {\scriptstyle \breve{n}} & &
\end{array}$$

Nonation NOPT structure has OrderType 9 and has (2,2,1)-dimensional array structure:

$$
\begin{array}{cccccccc}
\theta\ldots\theta)n & & \theta\ldots\theta)n & & \theta\ldots\theta)n & & \theta\ldots\theta)n & \\
\vdots & \ldots\} & \vdots &)n & \vdots & \ldots\} & \vdots &)n \\
\underset{\breve{n}}{\theta\overset{\leftrightarrow}{\ldots}\theta})n & & \underset{\breve{n}}{\theta\overset{\leftrightarrow}{\ldots}\theta})n & & \underset{\breve{n}}{\theta\overset{\leftrightarrow}{\ldots}\theta})n & & \underset{\breve{n}}{\theta\overset{\leftrightarrow}{\ldots}\theta})n & \\
 & \vdots & & \cdots\} & & \vdots & &)n \\
 & \leftrightarrow & & & & \leftrightarrow & & \\
\theta\ldots\theta)n & & \theta\ldots\theta)n & & \theta\ldots\theta)n & & \theta\ldots\theta)n & \\
\vdots & \ldots\} & \vdots &)n & \vdots & \ldots\} & \vdots &)n \\
\underset{\breve{n}}{\theta\overset{\leftrightarrow}{\ldots}\theta})n & \breve{n} & \underset{\breve{n}}{\theta\overset{\leftrightarrow}{\ldots}\theta})n & & \underset{\breve{n}}{\theta\overset{\leftrightarrow}{\ldots}\theta})n & \breve{n} & \underset{\breve{n}}{\theta\overset{\leftrightarrow}{\ldots}\theta})n & \\
 & & & \breve{n} & & & &
\end{array}
$$

By zig-zagging from bottom right corner, in left-up-left-up-left-up-… fashion we can build up NOPT structures with higher dimensions. The computation pathway proceeds from bottom right corner in left-up-left-up-left-up-… fashion until the top left corner.

Linear NOPT structures have Linear Ellipsis Structure.
Non-linear NOPT structures (OrderType >= 6) have Multi-layered Ellipsis Structure. The Ellipsis corresponding to the final top-left corner component of the NOPT structure is where the final computation is achieved. Superficially, it looks like a Linear Nopt structure, but looking at it carefully, notice that the Ellipsis length (of linearly arranged nested power towers) is given by a small number in Tetration and Pentation NOPT structures, but in larger NOPT structures is given by a Multi-layered Ellipsis expression, that is the combined effort of the rest of the interlinked components of the NOPT structure.
The induced Multi-layered Ellipsis Structure from a NOPT structure has a very similar structure to the well-known H-Fractal.

A NOPT-6 structure (hexation) has a 2-dimensional array structure and the induced Multi-layered Ellipsis Structure corresponds with a small H-Fractal at the first level.
A NOPT-7 structure (heptation) has a (2,1)-dimensional array structure and the induced Multi-layered Ellipsis Structure corresponds with the H-Fractal at the second level of resolution.
A NOPT-8 structure (octation) has a (2,2)-dimensional array structure and the induced Multi-layered Ellipsis Structure corresponds with the H-Fractal at the third level of resolution.
A NOPT-9 structure (nonation) has a (2,2,1)-dimensional array structure and the induced Multi-layered Ellipsis Structure corresponds with the H-Fractal at the fourth level of resolution.

You can observe that in terms of written symbolic requirements, going from NOPT-(I) to NOPT-(I+1) structure requires twice as many symbol expressions, due to the symbol folding that proceeds zigzagging L-U-L-U-L-U-L-U from bottom right corner to top right corner.

Section 4.3 Seed Values: “n” and “theta) n”

NOPT structures require a fixed SeedValue, that has the roles of (1) either initiating a Linear NOPT Component OR (2) Controlling Ellipsis Length of a Component.

It can be noticed that the induced SeedValues from a NOPT Structure have a similar role to the omega (w) limit which is used as a “default way” of describing the length of an infinite list. I think that omega is a shorthand for saying “we don’t know how long an unending list of items goes on for, but we’ll call it omega”. This seems more sensible than saying “yes we do know, the list goes on for as many natural numbers there are”. This is because, while the process of generating naturals (into Standard Positional Notation) is well-defined and clear, the extent of the natural numbers depends on how creative we get by combining powerful operations with NOPT structures. In other words, the set of Natural Numbers is well-defined at the level of (+1)-iterative description, but not at the level of unbounded descriptive capabilities. As the examples shown above demonstrate, there are plenty of intermediary ellipsis stages that large numbers need to pass through in order to reach an elusive target magnitude goal.

The omega limit is used for limit ordinals in the theory of infinite ordinals, but it is independent of the values of the symbols in the list.
So lim<w^1, w^2, …> = w^w, and is understood as an w-limit.
And lim<w, w^w, w^w^w, …> = e0, and is understood as an w-limit.
And lim<e0, e0^e0, e0^e0^e0, …> = e1, and is understood as an w-limit.
And lim<e0, e1, e2, …> = e_w, and is understood as an w-limit.
In the last example, it doesn’t matter that all the component symbols are much larger than omega - (e.g. e0, epsilon zero) – but we always use the “small infinite value” of omega (w) to decide the list length of any of these infinite sequences whenever a limit to the sequence is desired. Omega is like a convenient yardstick that is used in evaluating limit points for horizonal phenomena.
Whenever a list structure emerges and a limit is desired then the idea that an infinite list of symbols can be indexed or sequenced by the natural numbers is called upon.
So in theory of infinite ordinals, omega is the convenient yardstick.
In NOPT structures, the induced SeedValues have the “omega role” by deciding how long a stage should be lingered upon before the transitioning to other levels becomes necessary.

All of the “formal power towers” in the NOPT structure are given the symbol “theta”
By “formal” expression, I mean an unevaluated symbolic expression that could be evaluated given more information. The word “formal” here has the same kind of meaning as with “formal power series”.
When the SeedValue is supplied, in theory, the power tower can be evaluated because the height information about the power tower is given, then each succeeding power tower in the linear Nopt structure can be evaluated with each power tower evaluation supplying the height of the next power tower to be evaluated.

We don't have to be too rigid about SeedValues in the structure.
We could have SeedValues of the form "theta) n" or SeedValues of the form "n".
Sometimes, we may want to ensure that the next linear component in the NOPT structure has nontrivial ellipsis value and so we use the SeedValue="theta) n" style of NOPT structure. However, I think that for a standard definition of NOPT structure it is better to use the second style of NOPT structure where SeedValue=n.
The reason for this, is that we can write 3^^3, 3^^^3, 3^^^^3, 3^^^^^3 and so on, in terms of exponentiation towers, *not* tetration towers etc, and the higher the hyperoperation, the more layers of nesting are required.
With 3^^3 , only a single exponential power tower is created.
Tetration is the hyperoperation with n=4, of course, and we define the first nontrivial NOPT structure to have OrderType=4 to make a suitable correspondence with tetration, where tetration is expressed in NEPT form.
You could say that the "trivial" NOPT structure is just "n".
Here, n is a small number expressed with a linear sequence of digits with standard interpretation, using SPN (Standard Positional Notation) and SPN implicitly contains the first 3 operations (+, *, exp).
The first "nontrivial" NOPT structure is just "theta) n".
"theta" by itself is a "formal power tower", not a number as there is no height information.
"theta) n" obviously is a power tower, as the height of the power tower is given by n.

By some kind of careful inspection argument, one can notice that there are many ellipsis values that have the same magnitude within a NOPT structure, for example where the ellipsis is equal to a "controlling SeedValue". The "starting SeedValues" supply the height of the adjacent power tower, theta, in the linear NOPT component. The "controlling SeedValues" give the length of an ellipsis within a component of the NOPT structure.

Also, by careful inspection argument, one can order ellipsis lengths by increasing order of magnitude. Although it is very difficult to quantify "how much" longer one ellipsis is compared to another, the magnitudes increase very quickly as is the nature of fast-growing functions.

Section 4.4 A method for generating Nopt structures

Before I show a systematic method for generating Nopt structures I should say a few more things about Nopt structures.

In a way, HEFTY NOPT Structures or H-(Ellipsis)-Fractal Type Nested Operational Power Towers are a kind of transitional number pattern-based structure between the finite and infinite realm.

NOPT structures give a standard approach to looking at the bizarre world of numbers that straddle the finite and infinite divide. The induced Multi-layered Ellipsis Structure from a NOPT structure has similar component-connection structure to the well-known H-Fractal, given a corresponding degree of resolution.

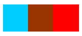
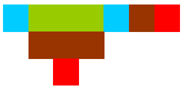
Moreover, The H-fractal structure is emergent from the NOPT structure and guides the multi-layered nestedly embedded computational pathways. The computation starts at the bottom right corner and moves towards the top left corner.

When the operation is not specified, NOPT structures are floaty and esoteric in nature, but nevertheless they still say something about the patterns arising from unlimited multi-layered nested recursion, with SeedValue being the standard stage-level phase transition marker.

Regarding these phase transitions, a comparison can be made with the phase transitions associated with hereditary base n, where the base value is used at other exponent levels apart from the level of standard positional notation, resulting in a treelike number structure. Perhaps, another comparison can be made with Cantor normal form for infinite ordinals.

In the colored-square diagram representations I will give below, the computation starts at the bottom right corner and moves towards the top left corner. The colored-square diagram approach provides a systematic way to generate Nopt structures.

Nopt stuctures with OrderTypes 4 and 5 can be shown as follows:

NOPT4  The red square represents the number n. The brown square represents a parenthesis. The blue square represents a theta symbol.	$\theta\ )\ n$
NOPT5  The brown squares represent a parenthesis. The green squares represent an ellipsis.	$\theta\ldots\theta\ )\ n$ $\breve{n}$

Nopt stucture with OrderType 6:

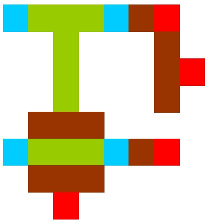
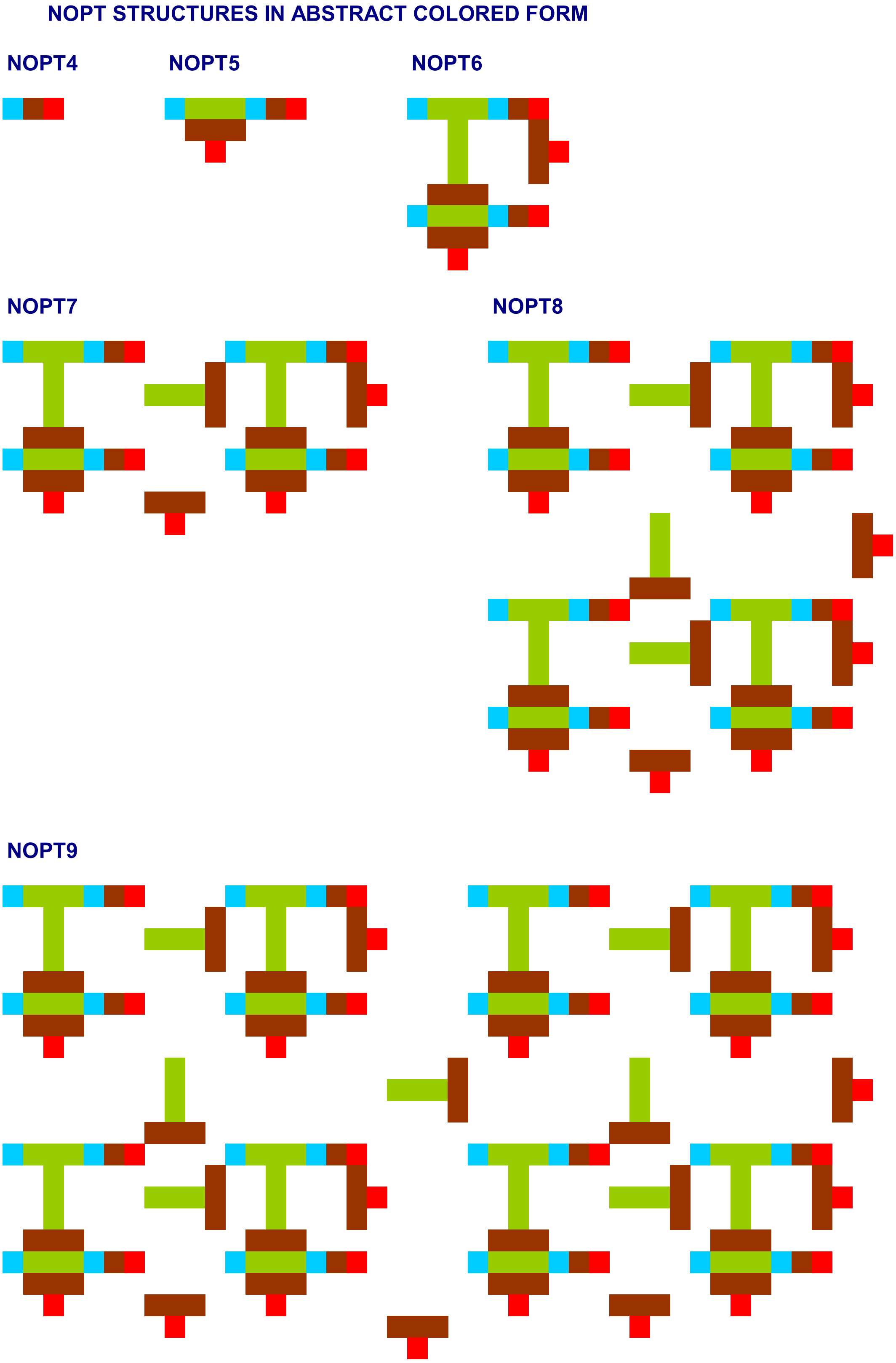
NOPT6  The red, brown, blue and green squares have same meaning as above.	$\begin{array}{ll} \theta\ldots\theta)n & \\ \vdots &)n \\ \overset{\breve{\wedge}}{\longleftrightarrow} & \\ \theta\ldots\theta)n & \\ \breve{n} & \end{array}$

Figure 4.4.1 Nopt structures with ordertypes 4 up to 9, where seed = n



With SeedValue = Theta}n, the colored-square pictures are a little different:
Nopt stuctures with OrderTypes 4 and 5 are shown as follows:

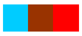
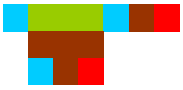
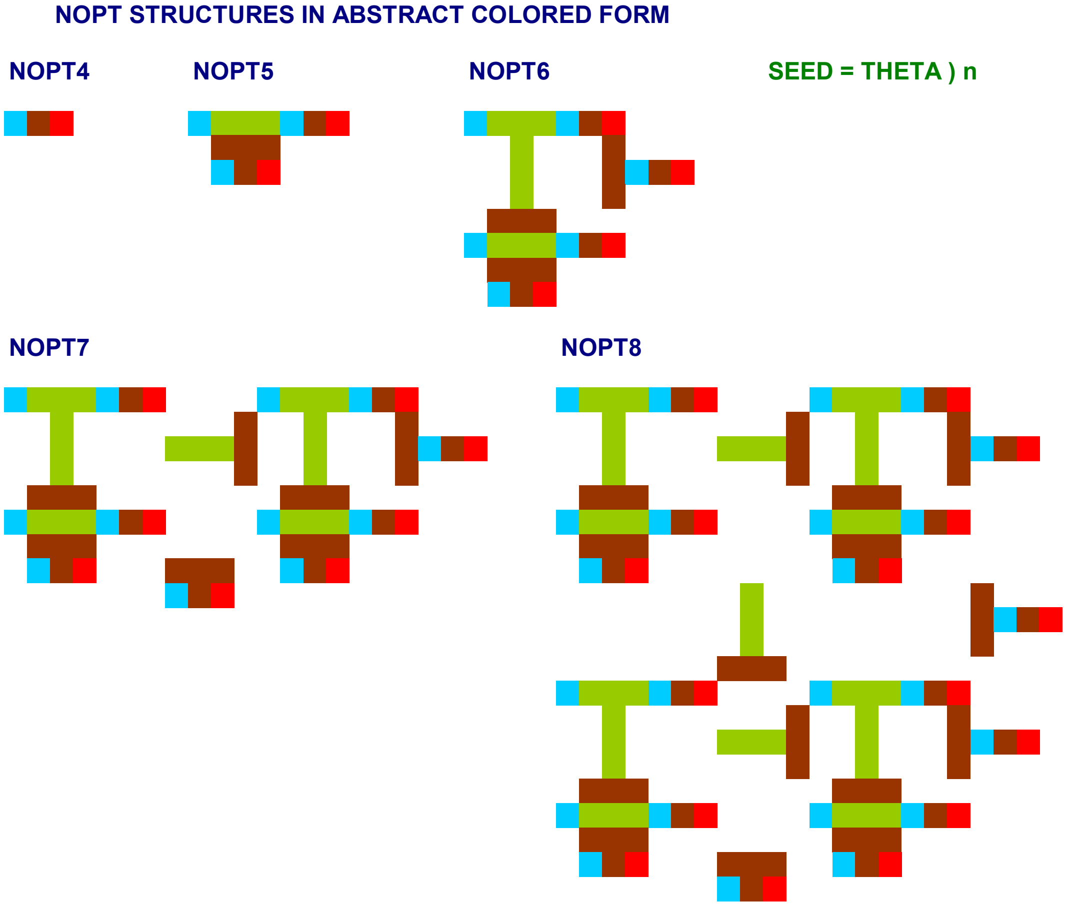
NOPT4  The red square represents the number n. The brown square represents a parenthesis. The blue square represents a theta symbol.	$\theta\ )\ n$
NOPT5  The brown squares represent a parenthesis. The green squares represent an ellipsis.	$\underbrace{\theta \ldots \theta\ )\ n}_{\theta\ )\ n}$

Figure 4.4.2 Nopt structures with ordertypes 4 up to 8, where seed = theta) n



A 3-stage process for generating Nopt(I+1) structure from Nopt(I) structure is shown below, and illustrated by the generation of Nopt10 from Nopt9.

Figure 4.4.3 Stage1 of the 3-stage process generating Nopt(I+1) structure

NONATION NOPT STRUCTURES IN ABSTRACT COLORED FORM

NOPT9 to NOPT10

KEY: 🟥 **SEED** 🟫 **PARENTHESIS** 🟦 **THETA** 🟩 **ELLIPSIS**

STEP ONE MAKE ANOTHER COPY OF NOPT9

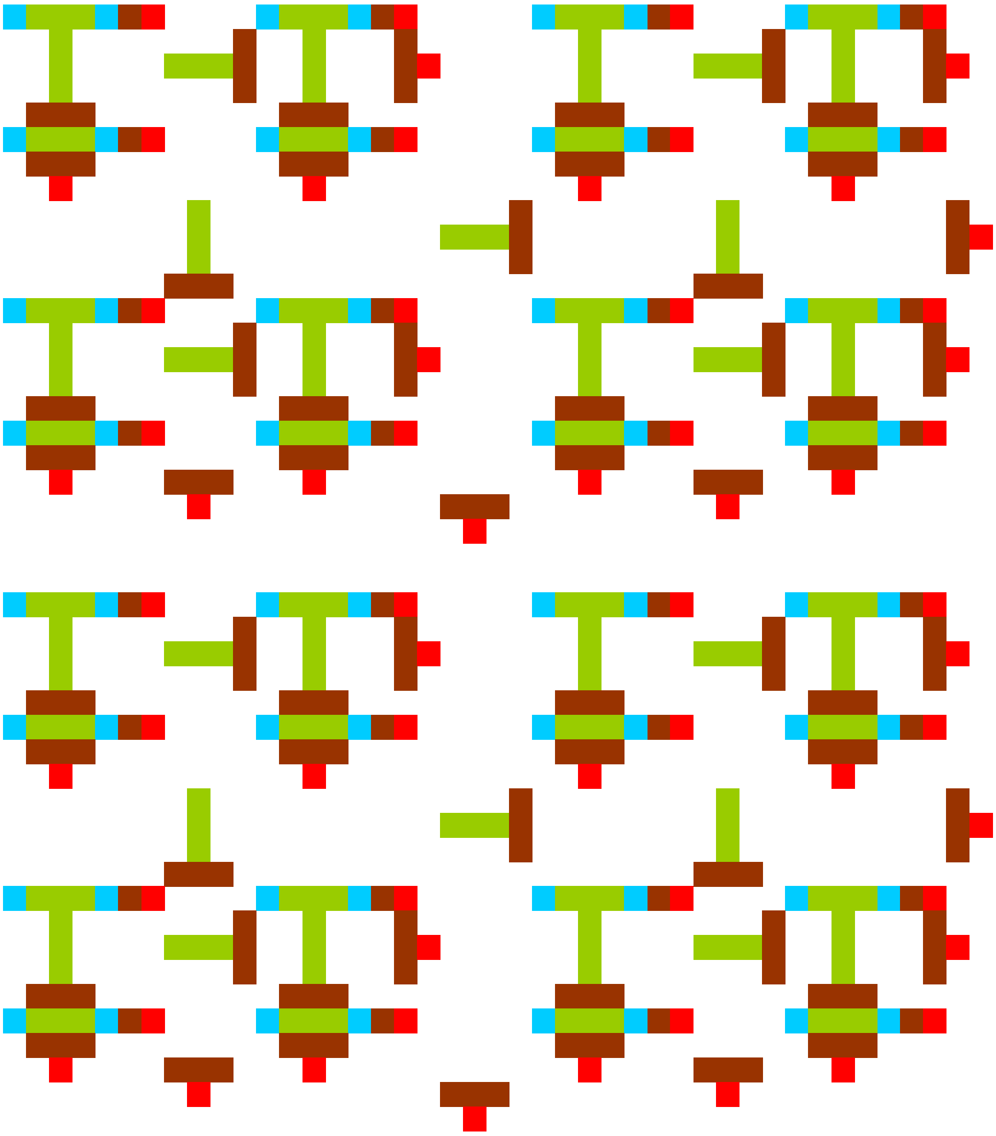


Figure 4.4.4   Stage2 of the 3-stage process generating Nopt(I+1) structure

**STEP TWO REPLACE CENTRAL CONTROLLING SEED VALUE WITH
ELLIPSIS HAVING (THE SAME) SEED LENGTH
ALLIGN ELLIPSIS/PARENTHESIS BLOCK TO MAKE IT
FLUSH ADJACENT WITH OUTER SEED VALUES**

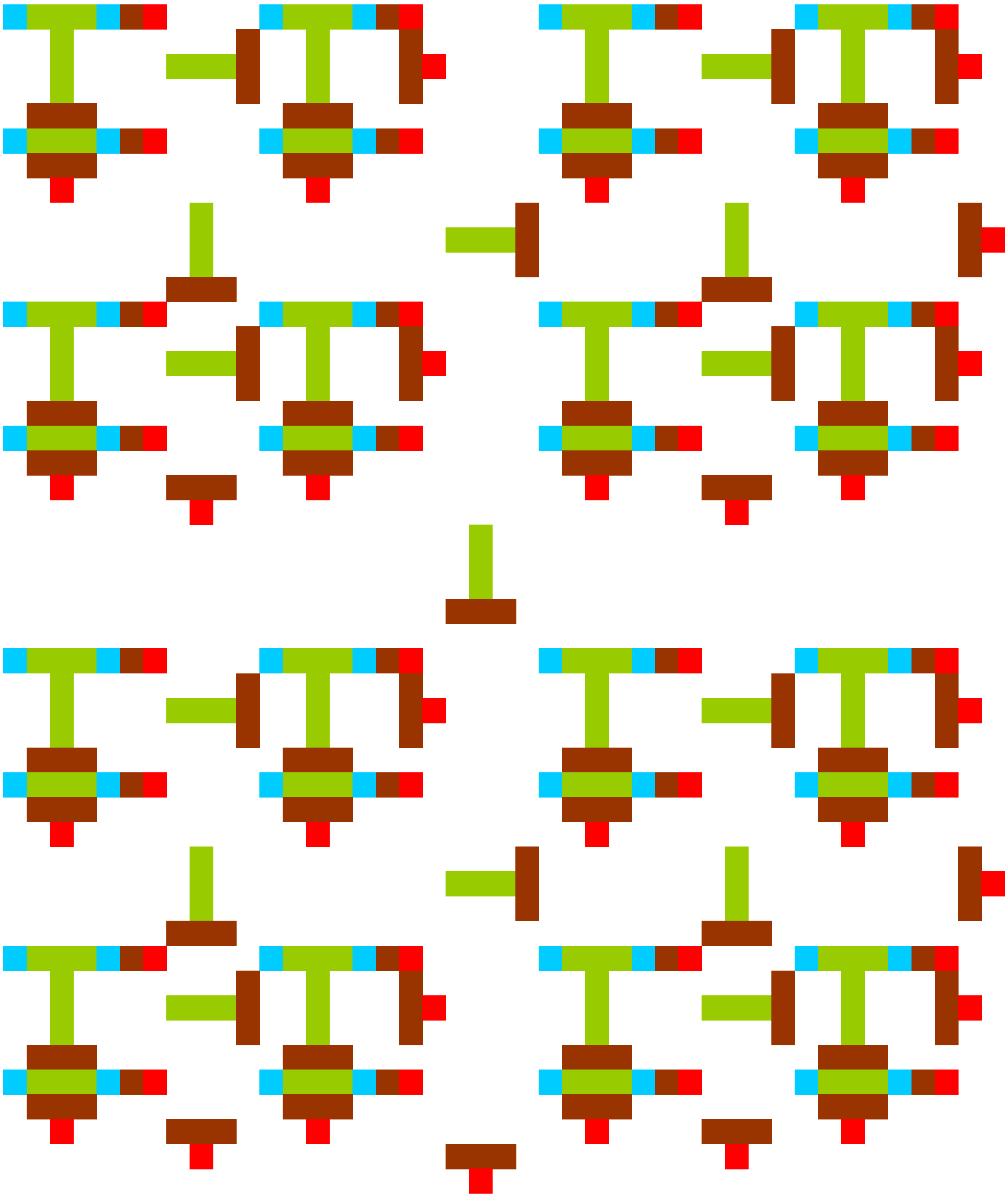
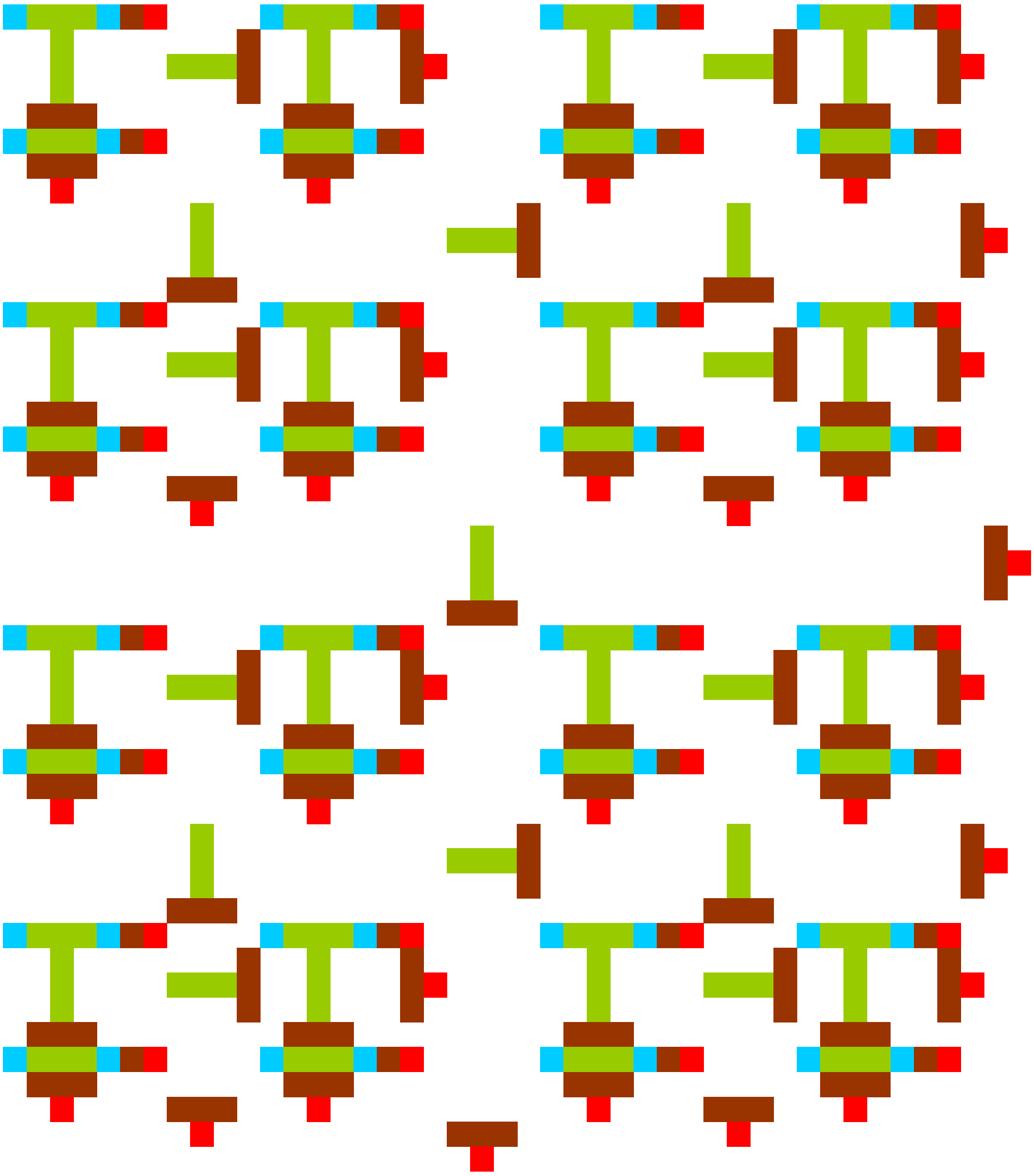


Figure 4.4.5 Stage3 of the 3-stage process generating Nopt(I+1) structure

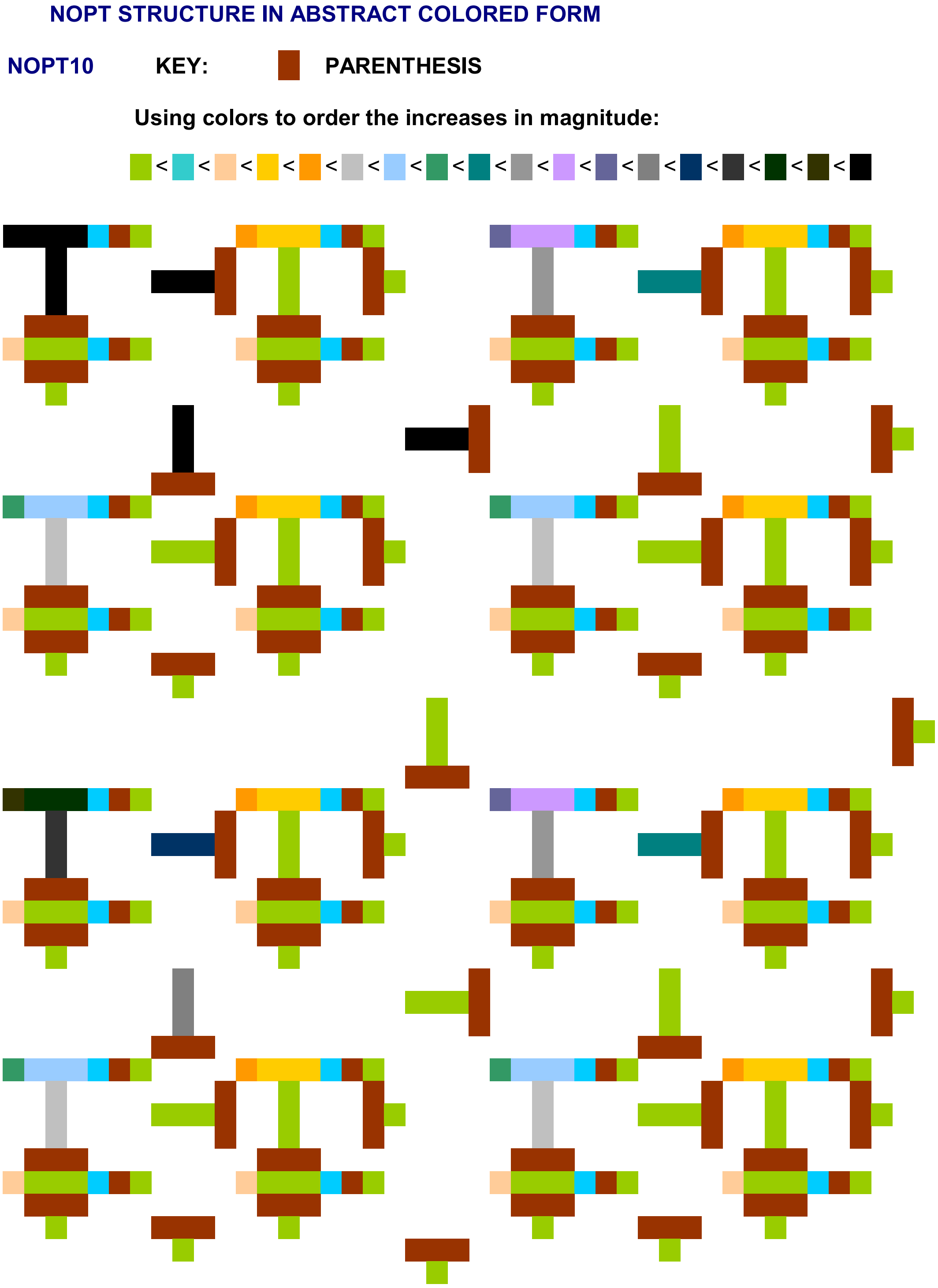
STEP THREE SUPPLY SEED VALUE TO NEW CENTRAL ELLIPSIS/PARENTHESIS BLOCK
ALLIGN STRUCTURES SO THE BROWN PARENTHESIS FROM CENTRAL ELLIPSIS BLOCK IS FLUSH ADJACENT TO OUTER THETA VALUES



Section 4.5 Magnitude inequalities inside Nopt structures

The idea of visualising the ellipsis lengths in increasing magnitude order has been color-coded in the picture below. Notice the increasing magnitudes of thetas as well.

Figure 4.5.1 Magnitude inequalities inside Nopt10 structure



V Applying Nopt Structures

Section 5.1 The gi-sequence with g-subscript towers.

The Seed(m)-Ackermann number sequence (with m>=3) is the sequence defined by:

$$f(n) = m \uparrow^n m, \quad n = 1,2,3,...$$

This starts with exponentiation, tetration and so on with a fixed base seedvalue = m.
Consider the Seed(3)-Ackermann-number sequence:
$f(n) = 3 \uparrow^n 3$ = <3^3, 3^^3, 3^^^3, …>
Now consider starting with f(4) and applying functional powers:
$f(4) = 3 \uparrow^4 3$ = 3^^^^3 = g1

$f(f(4)) = 3 \uparrow^{3 \uparrow^4 3} 3$ = 3(^3(^4)3)3 = g2 (using inline notation)

$f^3(4)$ = 3(^3(^3(^4)3)3)3 = g3 and so on.
Usually, this notation would increase in the vertical direction, with the inline notation it increases in a line in the horizontal direction. These numbers form a sequence:
g1, g2, g3, … (the gi-sequence)
3(^4)3, 3(^3(^4)3)3, 3(^3(^3(^4)3)3)3, …
Graham's number comes from the sequence defined by

$$g_0 = 4 \ , \ g_{k+1} = 3 \uparrow^{g_k} 3$$

With the functional power notation note that in particular,

$g_{64} = f^{64}(4)$ and

$$g_n = f^n(4), \quad n \geq 1$$

This is the formula linking f-superscripts with g-subscripts.
Now, consider g-subscript towers.
The "g_n" notation emphasises that subscript towers can be formed.
g_1 = f^1(4) = f(4) = g1
g_2 = f^2(4), g_3 = f^3(4), g_4 = f^4(4), … g_64 = f^64(4), …
g_(g_1) = f^(g_1)(4) = f^(f^1(4))(4)
g_(g_(g_1)) = f^(f^(f^1(4))(4))(4)

$$g_{g_{g_1}} = f^{f^{f^1(4)}(4)}(4)$$

etc…
We can avoid having a tower of functional powers by defining:

$h(n) = g_n$ for $n \geq 0$

The function h enumerates the gi-sequence.

$$h^2(n) = h(h(n)) = g_{h(n)} = g_{g_n}$$

The sequence

$$h(1), h^2(1), h^3(1), h^4(1), ...$$

Is the sequence that ends up making a g-subscript tower:

$$g_1, g_{g_1}, g_{g_{g_1}}, g_{g_{g_{g_1}}}, \cdots$$

And we can use NOPT structures to abbreviate the notation.
We can rewrite the following:
“ g_(g_...(g_(g_1))…) } where there are g_1 g’s ”
into a NOPT structure with
Theta = gst [a formal g-subscript tower (or gst), finishing with g_1]
Seed = g_1 and OrderType = 4 (there is only a seedvalue, parenthesis and theta)
The standard way to present a NOPT structure, NS is
NS = [OrderType, Theta, Seed]
NS = [OT4, gst, g_1] = “ g_(g_...(g_(g_1))…) } with g_1 g’s ”
Notice, by the way that NS = [OT3, gst, g_1] = g_1
Then, if we want to, we could contemplate the sequence <[OT(I), gst, g_1]> (I>=3)
So, hopefully, the relationship between the gi-sequence and g-subscript towers is now a little clearer.

The relationship with Conway chains.
From wikipedia, a Conway chain is defined as follows:
1. The chain p represents the number p.
2. $p \to q$ represents the exponential expression p^q.
3. $X \to 1 \to (q+1) = X$
4. $X \to (p+1) \to (q+1) = X \to (X \to p \to (q+1)) \to q$
It is clear that the sequence

$$< 3 \to 3 \to n >, \quad n \geq 1$$

is another notation for the Seed(3) Ackermann number sequence.
It is not to hard to see that the sequence

$$< 3 \to 3 \to n \to 2 >, \quad n \geq 2$$

forms a sequence that corresponds with the sequence

$f^n(1), \quad n \geq 2$ where $f(n) = 3 \uparrow^n 3$ starting with

$$f^2(1) = f(f(1)) = 3 \uparrow^{3\uparrow^1 3} 3 = 3 \uparrow^{3^3} 3 = 3 \uparrow^{27} 3$$

(Please refer to Table 5.2.2)
So the gi-sequence, g1, g2, g3, … that leads to Graham’s number has parallel growth, or keeps pace with, the sequence $< 3 \to 3 \to n \to 2 >, \quad n \geq 2$.
Also, the sequence

$$< 3 \to 3 \to n \to 3 >, \quad n \geq 2$$

has parallel growth, or keeps pace with, the sequence

$$g_1, g_{g_1}, g_{g_{g_1}}, g_{g_{g_{g_1}}}, \cdots.$$

To see this is so, please refer to the next section.

Section 5.2 Nopt structures and Conway chained arrows.

Table 5.2.1 Conway arrow number sequences

3→3→2→2 = 3→3→(3→3→1→2) →1 = 3→3→(3→3) **= 3→3→27**	**3→3→2→4** = 3→3→(3→3→1→4)→3 = 3→3→(3→3)→3 **= 3→3→27→3**
3→3→3→2 = 3→3→(3→3→2→2) →1 = 3→3→(3→3→2→2)	**3→3→3→4** = 3→3→(3→3→2→4)→3 = 3→3→(3→3→27→3)→3
3→3→4→2 = 3→3→(3→3→3→2) →1 = 3→3→(3→3→3→2)	**3→3→4→4** = 3→3→(3→3→3→4)→3 = 3→3→(3→3→(3→3→27→3)→3)→3
3→3→5→2 = 3→3→(3→3→4→2) →1 = 3→3→(3→3→4→2) etc	**3→3→5→4** = 3→3→(3→3→4→4)→3 = 3→3→(3→3→(3→3→(3→3→27→3)→3)→3)→3 etc
3→3→2→3 = 3→3→(3→3→1→3)→2 = 3→3→(3→3)→2 **= 3→3→27→2**	**3→3→2→5** = 3→3→(3→3→1→5)→4 = 3→3→(3→3)→4 **= 3→3→27→4**
3→3→3→3 = 3→3→(3→3→2→3)→2 = 3→3→(3→3→27→2)→2	**3→3→3→5** = 3→3→(3→3→2→5)→4 = 3→3→(3→3→27→4)→4
3→3→4→3 = 3→3→(3→3→3→3)→2 = 3→3→(3→3→(3→3→27→2)→2)→2	**3→3→4→5** = 3→3→(3→3→3→5)→4 = 3→3→(3→3→(3→3→27→4)→4)→4
3→3→5→3 = 3→3→(3→3→4→3)→2 = 3→3→(3→3→(3→3→(3→3→27→2)→2)→2)→2 etc	**3→3→5→5** = 3→3→(3→3→4→5)→4 = 3→3→(3→3→(3→3→(3→3→27→4)→4)→4)→4 etc

Table 5.2.2 Linking the Conway arrow number sequences together

So, using another notation… **3→3→2→2** = **3→3→27** = $3\uparrow^{27}3$ **3→3→2→3** = **3→3→27→2** = $\left.3\uparrow^{3\uparrow^{\cdot^{\cdot^{\cdot^{3\uparrow^{27}3}}}}3}3\right\}27$	**3→3→2→4** = **3→3→27→3** = $\left.3\to3\to(^{3\to3\to(^{\cdot^{\cdot^{\cdot^{3\to3\to27\to2}}}})\to2})\to2\right\}27$ **3→3→2→5** = **3→3→27→4** = $\left.3\to3\to(^{3\to3\to(^{\cdot^{\cdot^{\cdot^{3\to3\to27\to3}}}})\to3})\to3\right\}27$

These expressions above are Knuth arrow power towers and Conway arrow power towers. We need to see them again for comparison later on.

Let's see how to interpret the Conway arrow power towers by converting them to Knuth arrow power towers.

Conway chained arrow notation is usually presented with inline notation, however, it is still well-defined to consider them as power towers. The amount of inward-outward nesting is translated into height (or depth if you like) of the power tower. It can be observed that, at least for small Conway chained arrows of length = 4, we can convert them into Knuth arrow power towers but Conway chained arrows grow so fast, it is only possible to do this for Conway chained arrows of length = 4. For the first non-trivial length = 5 Conway chained arrows, the ability to convert them to multi-layered nestations of Knuth arrow power towers runs out of steam. This can be made clearer with the help of Nopt structures.

$3\to3\to2\to2 = 3\to3\to27 =$

$$3\uparrow^{27}3$$

$3\to3\to2\to3 = 3\to3\to27\to2 =$

$$3\uparrow^{3\uparrow^{\cdot^{\cdot^{\cdot^{3\uparrow^{27}3}}}}3}3\ \Big\}27$$

Now, consider the Knuth arrow power tower:

$$3\uparrow^{3\uparrow^{\cdot^{\cdot^{\cdot^{3\uparrow^{27}3}}}}3}3\ \Big\}27$$

This has Nopt structure $\theta)n$ where "theta" is the "formal" Knuth arrow power tower and n=27 is the seed value describing the number of layers, including the very top layer consisting of the number 27.

Now consider:

$3\to3\to2\to4 = 3\to3\to27\to3 =$

$$3\to3\to(^{3\to3\to(^{\cdot^{\cdot^{\cdot^{3\to3\to27\to2}}}})\to2})\to2\ \Big\}27$$

This is equal to the expression below:

$$\underbrace{3\uparrow^{3\uparrow^{\cdot^{\cdot^{\cdot^{3\uparrow^{27}3}}}}3}3\ \cdots\ \Big\}\ 3\uparrow^{3\uparrow^{\cdot^{\cdot^{\cdot^{3\uparrow^{27}3}}}}3}3\ \Big\}27}_{27}$$

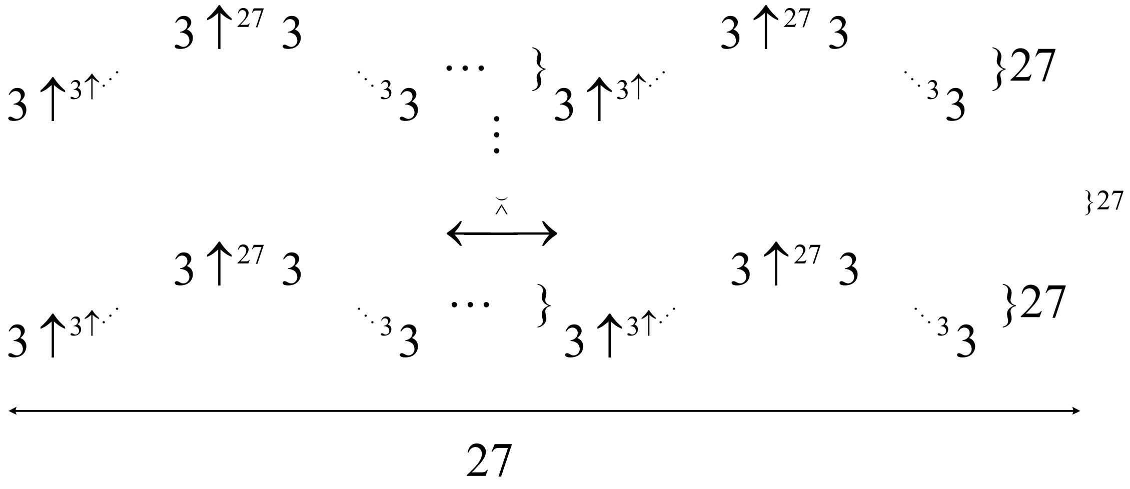
And this expression has Nopt structure $\underset{\breve{n}}{\theta ... \theta)n}$ having OrderType=5.

Now consider:

$3 \to 3 \to 2 \to 5 = 3 \to 3 \to 27 \to 4 =$

$$3 \to 3 \to \left(^{3 \to 3 \to (^{\cdot^{\cdot^{\cdot}}} \overset{3 \to 3 \to 27 \to 3}{} {}^{\cdot^{\cdot^{\cdot}}}) \to 3}\right) \to 3 \quad \}27$$

This is equal to the expression below:



And this expression has Nopt structure with OrderType=6.

$$\begin{array}{ll} \theta ... \theta)n & \\ \quad\vdots &)n \\ \overset{\breve{\wedge}}{\longleftrightarrow} & \\ \theta ... \theta)n & \\ \quad\breve{n} & \end{array}$$

Let $A = \{3 \to 3 \to 3 \to n \ : \ n \in \{1,2,3...\}\}$

Define a projection function on elements of A:

$p(3 \to 3 \to 3 \to n) = n$

Theorem 1

$$p(3 \to 3 \to 3 \to n) = OrderType\,(Naropt\,(3 \to 3 \to 3 \to n))$$

What this theorem is saying is that if you convert $3 \to 3 \to 3 \to n$ into a Nested Knuth arrow power tower (or Naropt) then the last value of the Conway chained arrow keeps pace with the OrderType of the associated Naropt structure.

In general, $3 \to 3 \to 3 \to n$ is a Naropt with OrderType = n + 2.

(The “plus 2” is because tetration (hyperoperation=4) uses 2 Knuth arrows.)

Corollary 1.1
The upshot of this is that if you want to convert Conway Chained arrows to Naropt structures this technique runs out of steam quite early on with the sequence

$$< 3 \to 3 \to 3 \to n >_{n=1}^{\infty}$$

This is due to the very fast growing nature of Conway Chained arrows.
Considering the smallest non-trivial length=5 Conway Chained arrow, that is to say:

$$3 \to 3 \to 3 \to 2 \to 2 = 3 \to 3 \to 3 \to (3 \to 3 \to 3 \to 1 \to 2) \to 1$$

$$= 3 \to 3 \to 3 \to (3 \to 3 \to 3) = 3 \to 3 \to 3 \to (3 \uparrow^3 3)$$

Is a Naropt with OrderType = $(3 \uparrow^3 3) + 2$. The OrderType is so large that the recursion structure using Knuth arrow power towers has effectively run out of gas.

Corollary 1.2
The seed(m) Ackermann number sequence does for Nept structures what the sequence $< 3 \to 3 \to 3 \to n >_{n=1}^{\infty}$ does for Naropt structures.

So in the rarefied realm of Nested Arrow power tower structures, this sequence has a similar parallel recursion structure with the recursion structure for the seed(m) Ackermann number sequence converted into nested exponential power towers.

VI Glossary

Ackermann numbers

The sequence of numbers: 1^1, 2^^2, 3^^^3, 4^^^^4, …, n^(n)^n, …

base(3) Ackermann number sequence: 3^3, 3^^3, 3^^^3, 3^^^^3, ….

The first 3 terms of this sequence (that is, 3^3, 3^^3 and 3^^^3) are fairly easy to work out from the definitions. However 3^^^^3 is more involved. These numbers are similar to the Ackermann numbers.

ellipsis

Dotdotdot (…)

formal power tower

A power tower that is unevaluated because height information is missing.

the word "formal"

Mathematicians describe "fomal power series" with a particular meaning.
In a similar way, "formal power tower" describes a process when we don't know if and when it will halt. The use of seedvalues takes the formal power towers and grounds them, initiating them, with small numbers.

gi-sequence

The sequence starting with g1=3^^^^3, g(k+1)=3^g(k)^3 and g64= Graham's number.

Graham's number construction

A source of fascination and inspiration. A desire to understand g1=3^^^^3 led me to thinking about NEPT and NOPT structures. To work out "3 hexated to 3" requires some effort. A description of "4 hexated to 4" can be found at Wikipedia article "Ackermann function"

g-subscript towers

A number of the form g_(g_(g_...(g_(g1)…) where the number of the g's in the expression is specified.
For example: g_(g_(g_...(g_(g1)…) } where there are g1 g's.

height of power tower

A natural number that says how many numeral symbols are in the power tower.
The height of a power tower may be a small number or require a nested power tower.

hyperoperation hierarchy

Continuing the operations that start with addition, multiplication, exponentiation, to tetration, pentation and so on.

in theory

This is a way of reminding the reader that the issue of a computation halting may become so lengthy that it can only be understood in stages, with assumptions about what can reasonably be ignored in an information-presentation by the writer and the reader.

in principle

A synonym for “in theory”

layered nested expression, multi-layered nested expression

The transitions between nested expressions, layered nested expressions and multi-layered nested expressions are something that NEPT and NOPT structures can describe quite nicely.

minimal symbolic notation

By considering a symbolic expression that may be folded and copied, a doubling process occurs. To keep the notation manageable, a minimal symbolic notation needs to be described. The minimal symbolic notation should be efficient, and needs to capture the essential information.

multi-layered Nested Exponentional Power Towers

When a number is huge, many layers of nestation are required

NEPT

Nested Exponential Power Tower

NOPT

Nested Operational Power Tower

NAROPT

Nested Arrow Power Tower or Nested Knuth Arrow Tower

omega and epsilon

There are the symbols ω (or w) for omega and ε (or e) for epsilon.
These numbers are the same as described in set theory, the theory of infinite ordinals.

power tower

The typical example uses exponentiation, and is essentially written in the vertical-rightwards diagonal dimension. To be more precise, it is an up-right-diagonal sequence of numerals and ellipsis to represent continuation and there is clear information about the height of the power tower.

PVN

Place Value Notation. The other name for Standard Positional Notation.

seedvalue
A small number that starts a nested power tower or says how long a starting ellipsis is.

small number
A number that can be expressed using SPN (or PVN) notation

SPN
Standard Positional Notation

controlling seed
A small number that says how long an initial ellipsis in a component is.

starting seed
A small number that initiates a linear nesting of power towers and describes the height of the first power tower in the linear nesting.

subscript power tower or subscript tower
Some classic examples are:
g_(g_(g_...(g_(g1)…) } where there are g1 g's
e_(e_(e_...(e_(e0)…) } where there are w e's

tally
Repeated Unary operation on a basic symbol to have the written effect of distinguishing and pointing at multiple items.

theta
A Greek symbol, θ, used as part of minimal symbolic notation in Nopt structures. It represents a formal power tower.

top-down and bottom-up methods
There is top-down tetration and bottom-up tetration.
Top-down is the default method as it produces a new operation.
Some authors prefer the terms higher and lower hyperoperators.

UDC
Unbounded Descriptive Capability.
We are free to use a wide range of notations with interpretations to express a counting number. "Counting number" emphasises the magnitude aspect that "in theory" could be represented by a tally. Counting numbers are whole numbers. They are not rational numbers, reals or complex numbers. Natural numbers include small numbers, as well as a wide range of numbers that require varying degrees of complexity and effort to be described in a well-defined way.

Well-defined
Expressed clearly and unambiguously, assuming the information can be communicated and understood by other mathematicians.

VII Further Reading and Weblinks (Draft version)

While the internet has abundant resources covering the diverse spectrum of mathematical ideas, there seems to be, at the time of writing this paper, little information, directly connected with the topic of this paper.
There are many researchers, amateur and professional who have come up with fruitful ideas that can interact, furnish and exemplify the ideas in this paper.
Wikipedia and Wolframscience are the big portals of math information, with an incredible selection of mathematical concepts.
There are various math forums in the foundations of maths and theory of computer science that have interesting discussions and ideas.
This is a list of website links and papers with some brief information (Draft version)

Weblinks

Large numbers
Robert Munafo
http://www.mrob.com/pub/math/largenum.html
Robert Munafo has done a great job educating the public about the subject of large numbers in a mature and thoughtful way

Big Number Central
Jonathan Bowers
http://www.polytope.net/hedrondude/bnc.htm

Tetration and higher-order operations on transfinite ordinals
Quickfur
http://math.eretrandre.org/tetrationforum

Introduction to Nept and Nopt structures
Mike Smith (aka Alister Wilson, Dolti Fantara)
http://math.eretrandre.org/tetrationforum

Making and Understanding Large Numbers
Peter William Hurford
http://www.greatplay.net/essays/table-of-contents

Proof that G>>M
Graham's number G and the Moser number M are both humungously large, but G is very much larger than M. Tim Chow's proof outline can be seen at:
http://www-users.cs.york.ac.uk/susan/cyc/b/gmproof.htm

My number is bigger! discussion at xkcd.com
http://forums.xkcd.com/viewtopic.php?f=14&t=7469&start=1160
Entertaining debate about the subject

Online Encyclopedia of Integer Sequences
www.OEIS.com
A valuable and huge collection of number sequences

Papers

Catalan Numbers
Tom Davis

Arborescent Numbers:
Higher Arithmetic Operations and Division Trees
Henryk Trappmann

Goodstein's function
Andres Eduardo Caicedo
California Institute of Technology

Exponentials reiterated
R. Arthur Knoebel
Department of Mathematics, New Mexico State University

An overview of the ordinal calculator
Paul Budnik

International Journal of Algebra
Number Theories
Patrick St-Amant

Parabola Volume 40, Issue 1 (2004)
On Iterated Exponentiation – the Hyperexponentials
Sean Stewart

Array Notations for Super Huge Numbers
Chris Bird

On the Independence of Goodstein’s theorem
Justin T Miller

An extremely sharp phase transition threshold
for the slow growing hierarchy
Andreas Weiermann

Predicativity
Solomon Feferman

Alternative Set Theories
M. Randall Holmes

Realizing Levels of the Hyperarithmetic Hierarchy
as Degree Spectra of Relations on Computable Structures
Denis R. Hirschfeldt and Walker M. White