MultiType Dimensional Place Value Notation By Alister “Mike Smith” Wilson

Contents

Introduction to Multi-Dimensional Place Value Notation
Unary to PVN transform
PVN to Unary transform
Brace notation and the PVN to Unary transform
Examples of repdigit or repeated pattern numbers
Long repdigit numbers
Extending Place Value Notation into pure noptiles
Recap of the way Place Value Notation keeps growing into noptiles
The principle of unary spillover into new PVN counting variable and new order type
If you follow the same kind of logic In Base(3) what happens?
If you follow the same kind of logic In Base(2) what happens?

Introduction to Multi-Dimensional Place Value Notation.

If we are counting in the system of Place Value Notation it is well-known that we can only write down or store in a physical computer the small natural numbers that is to say where the information required is manageable. For example I can write down the following base 10 number:
234789378793457827945823465628748568263475672345762873458726863478567
And although the number is precisely described it is also very difficult to pronounce!
If I continued this logic to a number so big it required a book to notate all the numbers we would have a very big number and almost impossible to pronounce.
However some big numbers are easy to pronounce such as googol and googolplex.
A googol can be typed up using a word processor as it is 1 followed by 100 zeroes.
However it is impossible to use one dimensional PVN to describe a googolplex. There are simply way to many zeroes that are needed. Using exponents it is possible to refer to a googolplex. So in a sense googolplex is imaginary number. Does it make sense to say that numbers can be incremented one by one up to a googolplex? Not if you hope to write all the numbers down. But, if you omit that requirement and extend the binary logic of base 10 (the use and reuse of orders of magnitude) it is surprising how many numbers can be theoretically passed in an incremental fashion and using PVN. Some people would say there is a horizon to this method, the horizon occuring depending on how far you can imagine a one dimensional string of digits continuing. This is usually what people think about when imagining the extent of PVN. The one dimensional restriction is unlikely to take us further than a googol. After that, it becomes more and more hard to imagine as one dimensional. Using other dimensions, we are still limited in terms of information content but the imagination is free to see further than a one dimensional horizon. This is sometimes taken advantage of by set theorists, but you don’t need set theory you can still get multi dimensional using place value notation. For some reason mathematicians have been uncomfortable about this idea prefering to use exponents and power towers to describe big numbers. But you can also do it using PVN if you allow the use of bracketing structures and stick to numbers with low information content such as repdigit numbers. But using repdigit numbers in bracketing structures it is possible to imagine incrementing the natural numbers on a much grander scale by extending into multi-dimensional

bracketing structures. These numbers occur in the mind, the ideosphere, as concepts with logical features and otherwise they are not practical to consider.
Let's start with one googol.
Usually represented as 10^100, that is to say 1 followed by 100 zeroes.

With bracketing you can represent a googol as: $\underbrace{10...00}_{101}$

This is a 101 digit number with a 1 followed by a hundred zeroes in other words a googol.

A googolplex is 1 followed by a googol zeroes. So using bracketing: $\underbrace{10...00}_{\underbrace{10...00}_{101}+1}$

But hey, I've never seen a googolplex look this way, you probably say. But wait a moment and think what it says. A googolplex is 1 followed by a googol zeroes. That is, it has (googol + 1) digits. Now the expression can be seen as an accurate representation of a googolplex. We will explore this logic into quite a lot of detail for the remainder of the paper: about PVN into higher dimensions.

Unary to PVN transform

we use this all the time in everyday life
let's say the unary symbol is @
then some examples
@@@@@ -> 5
@@@@@@@@ -> 8
@@@@@@@@@@@@@@ -> 14
this is going from unary to PVN

PVN to Unary transform

6 -> @@@@@@
15 -> @@@@@@@@@@@@@@@
22 -> @@@@@@@@@@@@@@@@@@@@@@
we use this transform when checking the amount of something, for example, our generous boss gives us $12 in $1 notes and we check the number of notes counting as we go:
1 1 1 1 1 1 1 1 1 1 1 1
1 2 3 4 5 6 7 8 9 10 11 12 … aha yes $12

Brace notation and the PVN to Unary transform

$22 = \underbrace{@@...@@}_{22}$ yay! Count 22 occurences of some symbol and say there are 22 of these symbols then the number represented is 22.

$$\underbrace{11...11}_{22} = ?$$

Now replace the @'s with 1's and interpret the number as PVN number, not unary!
The number represented is now 1,111,111,111,111,111,111,111 (and much bigger than 22)

$$\underbrace{11...11}_{\underbrace{11...11}_{22}} = ?$$

Now we're saying that the number represented has 1,111,111,111,111,111,111,111 digits.
So we are imagining that the PVN number 1,111,111,111,111,111,111,111 has been transformed into unary in order to match up with the digits of a new number and all of these digits are 1's and then we interpret this string of digits as a new PVN number.
This is the intended interpretation of the brace notation.

Examples of repdigit or repeated pattern numbers

An array of sequences of PVN numbers with repdigit features.

$$1, 22, 333, 4444, 55555, 666666, 7777777, 88888888, 999999999, 101010101010101010, ...$$
$$2, 33, 444, 5555, 66666, 777777, 8888888, 99999999, 1010101010101010,$$
$$3, 44, 555, 6666, 77777, 888888, 9999999, 10101010101010, 1111111111111111, ...$$
$$4, 55, 666, 7777, 88888, 999999, 101010101010, 11111111111111, ...$$
$$5, 66, 777, 8888, 99999, 1010101010, 111111111111, 12121212121212, ...$$
$$6, 77, 888, 9999, 10101010, 1111111111, 121212121212, ...$$
$$7, 88, 999, 101010, 11111111, 1212121212, 131313131313, ...$$
$$8, 99, 101010, 11111111, 1212121212, 131313131313, ...$$
$$9, 1010, 111111, 12121212, 1313131313, 141414141414, ...$$
$$10, 1111, 121212, 13131313, 1414141414, ...$$
$$11, 1212, 131313, 14141414, 1515151515, ...$$
$$12, 1313, 141414, 15151515, ...$$
$$13, 1414, 151515, 16161616, ...$$
$$14, 1515, 161616, ...$$
$$15, 1616, 171717, ...$$
$$16, 1717, ...$$
$$18, ...$$

Long repdigit numbers

Consider the following digit strings that are supposed to be Base10 repdigit numbers:
444, 22222, 5555, 888, 7777, 111, 99999.
What happens when you "+1" to the number of digits?
444 + 1 = 445 (still 3 digits long),
22222 + 1 = 22223 (still 5 digits long),
5555 + 1 = 5556 (still 4 digits long), etc
99999 + 1 = 100000 (has changed from 5 digits to 6 digits long)

Usually when you “+1” the number of digits is not changed.
However, repdigit(9) numbers are interesting for being the only numbers that when you add one also increases the number of digits by one. This is true for normal Base(10). For Base(8), the repdigit(7) numbers are the only numbers that when you add one also increases the number of digits by one.

When doing “+1 partial arithmetic” in Base(10) we usually 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.

Extending Place Value Notation into pure noptiles

First note it is clear that :

$$\underbrace{99...99}_{n}+1=\underbrace{10...00}_{n+1}$$

Now substitute $\underbrace{99...99}_{n}$ for n in the above equation and apply the result above

$$\underbrace{99...99}_{\underbrace{99...99}_{n}}+1=\underbrace{10...00}_{\underbrace{99...99}_{n}+1}=\underbrace{10...00}_{\underbrace{10...00}_{n+1}}$$

Repeat this procedure, using the result above

$$\underbrace{99...99}_{\underbrace{99...99}_{\underbrace{99...99}_{n}}}+1=\underbrace{10...00}_{\underbrace{99...99+1}_{\underbrace{99...99}_{n}}}=\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{n+1}}}$$

In particular ,

$$\underbrace{99\ldots99}_{\underbrace{99\ldots99}_{\underbrace{99\ldots99}_{9}}}+1=\underbrace{10\ldots00}_{\underbrace{99\ldots99+1}_{\underbrace{99\ldots99}_{9}}}=\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{10}}}$$

And if there are 9 layers, we note that the number of layers is the same after adding 1 and each of the layers is now either 10 or 10…00.

$$\left[\left.\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}9\right]+1=\left.\underbrace{10\ldots00}_{\substack{\vdots\\ \underbrace{10\ldots00}_{10}}}\right\}9$$

Generalise this result to get

$$\left[\left.\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}n\right]+1=\left.\underbrace{10\ldots00}_{\substack{\vdots\\ \underbrace{10\ldots00}_{10}}}\right\}n$$

Applying this result (substituting for n)

$$\left[\left.\left.\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}9\right]+1=\left.\left.\underbrace{10\ldots00}_{\substack{\vdots\\ \underbrace{10\ldots00}_{10}}}\right\}\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}9$$

And then extending to 9 nestings …

$$\left[\underbrace{\left.\left.\left.\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}\cdots\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}9}_{9}\right]+1=\underbrace{\left.\left.\left.\underbrace{10\ldots00}_{\substack{\vdots\\ \underbrace{10\ldots00}_{10}}}\right\}\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}\cdots\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}9}_{9}$$

And then generalising …

$$\left[\underbrace{\left.\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}\left.\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}\cdots\left.\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}9}_{n}\right]+1=\underbrace{\left.\underbrace{10\ldots00}_{\substack{\vdots\\ \underbrace{10\ldots00}_{10}}}\right\}\left.\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}\cdots\left.\underbrace{99\ldots99}_{\substack{\vdots\\ \underbrace{99\ldots99}_{9}}}\right\}9}_{n}$$

And from here you could substitute for n and get more complicated formulae but it is not too hard to imagine and the formulae get too messy to type up.

Let's look a bit further …
What's the difference between :

(A) $\left.\underbrace{10\ldots00}_{\substack{\vdots\\ \underbrace{10\ldots00}_{10}}}\right\}n$ and (B) $\left.\underbrace{10\ldots00}_{\substack{\vdots\\ \underbrace{10\ldots00}_{10}}}\right\}(n+1)$??

Let's look at some of the steps or stages to get from (A) to (B).

$$\left[\left.\underbrace{10\ldots00}_{\substack{\vdots\\ \underbrace{10\ldots00}_{10}}}\right\}n\right]+1=\left.\underbrace{10\ldots01}_{\substack{\vdots\\ \underbrace{10\ldots00}_{10}}}\right\}n \quad \text{and} \quad \left[\left.\underbrace{99\ldots99}_{\substack{\underbrace{10\ldots00}\\ \vdots\\ \underbrace{10\ldots00}_{10}}}\right\}n\right]+1=\left.\underbrace{10\ldots00}_{\substack{\underbrace{10\ldots01}\\ \vdots\\ \underbrace{10\ldots00}_{10}}}\right\}n$$

And eventually getting to …

$$\left[\left.\underbrace{99\ldots99}_{\substack{\underbrace{99\ldots99}\\ \vdots\\ \underbrace{99\ldots99}_{10}}}\right\}n\right]+1=\left.\underbrace{10\ldots00}_{\substack{\underbrace{10\ldots00}\\ \vdots\\ \underbrace{10\ldots00}_{11}}}\right\}n \quad \text{and much later on} \quad \left[\left.\underbrace{99\ldots99}_{\substack{\underbrace{99\ldots99}\\ \vdots\\ \underbrace{99\ldots99}_{99}}}\right\}n\right]+1=\left.\underbrace{10\ldots00}_{\substack{\underbrace{10\ldots00}\\ \vdots\\ \underbrace{10\ldots00}_{100}}}\right\}n$$

And then keep going for ages until finally we get to …

$$\left[\left.\underbrace{99\ldots99}_{\underbrace{99\ldots99}_{\vdots_{\underbrace{99\ldots99}_{999,999,999}}}}\right\}n\right]+1=\left.\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{\vdots_{\underbrace{10\ldots00}_{1,000,000,000}}}}\right\}n=\left.\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{\vdots_{\underbrace{10\ldots00}_{10}}}}\right\}(n+1)$$

And we're there !! We've got from (A) to (B) !!

Consider the sets

L={9,99,999,9999,99999,999999,9999999,99999999}

F={1,10,100,1000,10000,100000,1000000,10000000}

$\underbrace{99\ldots99}_{9}+1=\underbrace{10\ldots00}_{10}$ and $\underbrace{99\ldots99}_{99}+1=\underbrace{10\ldots00}_{100}$ and …

These are ordertype=4 and notice when adding 1 the answer involves 10's and 10…00's

$\underbrace{99\ldots99}_{\underbrace{99\ldots99}_{9}}+1=\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{10}}$ and $\underbrace{99\ldots99}_{\underbrace{99\ldots99}_{\underbrace{99\ldots99}_{9}}}+1=\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{10}}}$

These are composites of ordertype=4 moving to ordertype=5

Notice when adding 1 the answer involves 10's and 10…00's

Now check the behaviour of ordertype=5

$\left[\left.\underbrace{99\ldots99}_{\vdots_{\underbrace{99\ldots99}_{9}}}\right\}9\right]+1=\left.\underbrace{10\ldots00}_{\vdots_{\underbrace{10\ldots00}_{10}}}\right\}9$ and $\left[\left.\underbrace{99\ldots99}_{\vdots_{\underbrace{99\ldots99}_{99}}}\right\}9\right]+1=\left.\underbrace{10\ldots00}_{\vdots_{\underbrace{10\ldots00}_{100}}}\right\}9$ and …

Notice when adding 1 the Answer involves 10's and 10…00's

BUT ALSO, the "9" that counts the Number of Layers IS NOT CHANGED !!

HOWEVER, when the seed value gets to 999,999,999
the “9” that counts the Number of Layers is INCREMENTED to 10 …

$$\left[\left.\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{999,999,999}}}\right\}9\right]+1=\left.\underbrace{10\ldots00}_{\substack{\vdots\\\underbrace{10\ldots00}_{1,000,000,000}}}\right\}9=\left.\underbrace{10\ldots00}_{\substack{\vdots\\\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{10}}}}\right\}10=\left.\underbrace{10\ldots00}_{\substack{\vdots\\\underbrace{10\ldots00}_{10}}}\right\}10$$

And we can also express this equation as …

$$\left[\left.\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{9}}}\right\}10\right]+1=\left.\underbrace{10\ldots00}_{\substack{\vdots\\\underbrace{10\ldots00}_{10}}}\right\}10$$

At this stage we can express the general results …
Result (A)

$$\left[\underbrace{\left.\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{9}}}\right\}\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{9}}}\quad\ldots\quad\}9}_{\underbrace{\left.\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{9}}}\right\}\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{9}}}\ \ldots\ \}9}_{\vdots}}\quad\ldots\right]+1=\left[\underbrace{\left.\underbrace{10\ldots00}_{\substack{\vdots\\\underbrace{10\ldots00}_{10}}}\right\}\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{9}}}\quad\ldots\quad\}9}_{\underbrace{\left.\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{9}}}\right\}\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{9}}}\ \ldots\ \}9}_{\vdots}}\quad\ldots\right]$$

Result (B)

$$\left[\underbrace{\left.\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{999,999,999}}}\right\}\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{9}}}\quad\ldots\quad\}9}_{\underbrace{\left.\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{9}}}\right\}\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{9}}}\ \ldots\ \}9}_{\vdots}}\quad\ldots\right]+1=\left[\underbrace{\left.\underbrace{10\ldots00}_{\substack{\vdots\\\underbrace{10\ldots00}_{10}}}\right\}\left.\underbrace{10\ldots00}_{\substack{\vdots\\\underbrace{10\ldots00}_{10}}}\right\}\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{9}}}\quad\ldots\quad\}9}_{\underbrace{\left.\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{9}}}\right\}\underbrace{99\ldots99}_{\substack{\vdots\\\underbrace{99\ldots99}_{9}}}\ \ldots\ \}9}_{\vdots}}\quad\ldots\right]$$

This shows the general patterns for how these numbers change.
I hope this strange extension of PVN into pure noptiles is a little clearer.
You may like to continue the line of thinking from Results (A) and (B).

Let's look at one of these numbers more closely …

$$\left[\left.\underbrace{99\ldots99}_{\substack{\vdots \\ \underbrace{99\ldots99}_{9}}}\right\}9\right]+1=\left.\underbrace{10\ldots00}_{\substack{\vdots \\ \underbrace{10\ldots00}_{10}}}\right\}9$$

As a power tower what does it look like ??

$$\underbrace{10\ldots00}_{10}=10^{9}$$

$$\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{10}}=10^{10^{9}-1}$$

$$\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{10}}}=10^{10^{10^{9}-1}-1}$$

$$\left.\underbrace{10\ldots00}_{\substack{\vdots \\ \underbrace{10\ldots00}_{10}}}\right\}9=\left[\left.\underbrace{99\ldots99}_{\substack{\vdots \\ \underbrace{99\ldots99}_{9}}}\right\}9\right]+1=10^{10^{10^{10^{10^{10^{10^{10^{9}-1}-1}-1}-1}-1}-1}-1}<10^{10^{10^{10^{10^{10^{10^{10^{9}}}}}}}}<10\wedge\wedge 9={}^{9}10$$

One of the weirder looking but true equations of maths.
So basically ordertype=5 with PVN corresponds with ordertype=4
for exponential power towers (tetration) .

Recap of the way Place Value Notation keeps growing into noptiles

Start counting

1,2,3,…, 999,999,999

And keep going…

$$\underbrace{1000000000}_{10}\quad\underbrace{1000000001}_{10}\;\ldots\;\underbrace{9999999999}_{10}\quad\underbrace{10000000000}_{11}\;\ldots\;\underbrace{99\ldots99}_{11}\quad\underbrace{10\ldots00}_{12}\;\ldots$$

$$\underbrace{10\ldots00}_{99}\quad\underbrace{10\ldots01}_{99}\;\ldots\;\underbrace{99\ldots99}_{99}\quad\underbrace{10\ldots00}_{100}\;\ldots\;\underbrace{10\ldots00}_{999}\quad\underbrace{10\ldots01}_{999}\;\ldots\;\underbrace{99\ldots99}_{999}\quad\underbrace{10\ldots00}_{1000}\;\ldots$$

$$\underbrace{10\ldots00}_{999,999,999} \quad \underbrace{10\ldots01}_{999,999,999} \ \ldots \ \underbrace{99\ldots99}_{999,999,999} \quad \underbrace{10\ldots00}_{\underbrace{10\ldots00}_{10}} \quad \underbrace{10\ldots01}_{\underbrace{10\ldots00}_{10}} \ \ldots \ \underbrace{99\ldots99}_{\underbrace{10\ldots00}_{10}} \quad \underbrace{10\ldots00}_{\underbrace{10\ldots01}_{10}} \quad \underbrace{10\ldots01}_{\underbrace{10\ldots01}_{10}} \ \ldots$$

$$\underbrace{10\ldots00}_{\underbrace{99\ldots99}_{10}} \quad \underbrace{10\ldots01}_{\underbrace{99\ldots99}_{10}} \ \ldots \ \underbrace{99\ldots99}_{\underbrace{99\ldots99}_{10}} \quad \underbrace{10\ldots00}_{\underbrace{10\ldots00}_{11}} \quad \underbrace{10\ldots01}_{\underbrace{10\ldots00}_{11}} \ \ldots \ \underbrace{99\ldots99}_{\underbrace{10\ldots00}_{11}} \quad \underbrace{10\ldots00}_{\underbrace{10\ldots01}_{11}} \quad \underbrace{10\ldots01}_{\underbrace{10\ldots01}_{11}} \ \ldots$$

$$\underbrace{10\ldots00}_{\underbrace{99\ldots99}_{99}} \quad \underbrace{10\ldots01}_{\underbrace{99\ldots99}_{99}} \ \ldots \ \underbrace{99\ldots99}_{\underbrace{99\ldots99}_{99}} \quad \underbrace{10\ldots00}_{\underbrace{10\ldots00}_{100}} \ \ldots \ \underbrace{10\ldots00}_{\underbrace{10\ldots00}_{999,999,999}} \quad \underbrace{10\ldots01}_{\underbrace{10\ldots00}_{999,999,999}} \ \ldots \ \underbrace{99\ldots99}_{\underbrace{10\ldots00}_{999,999,999}} \quad \underbrace{10\ldots00}_{\underbrace{10\ldots01}_{999,999,999}} \quad \underbrace{10\ldots01}_{\underbrace{10\ldots01}_{999,999,999}} \ \ldots$$

$$\underbrace{10\ldots00}_{\underbrace{99\ldots99}_{999,999,999}} \ \ldots \ \underbrace{99\ldots99}_{\underbrace{99\ldots99}_{999,999,999}} \quad \underbrace{10\ldots00}_{\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{10}}} \ \ldots$$

And now the pattern should be clearer, and eventually the number of layers gets to 10 layers and at this stage we use a new counter to count the number of layers :

$\left.\begin{array}{c}\underbrace{10\ldots00}\\ \vdots \\ \underbrace{10\ldots00}_{10}\end{array}\right\}10$ and we can keep going from here as before

$$\left.\begin{array}{c}\underbrace{10\ldots01}\\ \vdots \\ \underbrace{10\ldots00}_{10}\end{array}\right\}10 \ \ldots \ \left.\begin{array}{c}\underbrace{10\ldots00}\\ \vdots \\ \underbrace{10\ldots01}_{10}\end{array}\right\}10 \quad \left.\begin{array}{c}\underbrace{10\ldots01}\\ \vdots \\ \underbrace{10\ldots01}_{10}\end{array}\right\}10 \ \ldots \ \left.\begin{array}{c}\underbrace{10\ldots00}\\ \vdots \\ \underbrace{10\ldots00}_{11}\end{array}\right\}10 \ \ldots \ \left.\begin{array}{c}\underbrace{10\ldots00}\\ \vdots \\ \underbrace{10\ldots00}_{99}\end{array}\right\}10 \ \ldots$$

$$\left.\begin{array}{c}\underbrace{10\ldots00}\\ \vdots \\ \underbrace{10\ldots00}_{999,999,999}\end{array}\right\}10 \ \ldots \ \left.\begin{array}{c}\underbrace{10\ldots00}\\ \vdots \\ \underbrace{10\ldots01}_{999,999,999}\end{array}\right\}10 \ \ldots \ \left.\begin{array}{c}\underbrace{10\ldots00}\\ \vdots \\ \underbrace{99\ldots99}_{999,999,999}\end{array}\right\}10 \ \ldots \ \left.\begin{array}{c}\underbrace{99\ldots99}\\ \vdots \\ \underbrace{99\ldots99}_{999,999,999}\end{array}\right\}10 \quad \left.\begin{array}{c}\underbrace{10\ldots00}\\ \vdots \\ \underbrace{10\ldots00}_{10}\end{array}\right\}11 \ \ldots$$

The principle of unary spillover into new PVN counting variable and new order type

Relative to base 10 and PVN we can define order types as follows
ordertype[0] {0} ordertype[1] {1} ordertype[2] {2,3,4,5,6,7,8,9}
ordertype[3] {10,11,12,13,14,15,16, … , 999,999,997, 999,999,998, 999,999,999}

The reason for defining ordertype this way is it allows for arbitrary type unary spillover into new counting variable and new order type. The unary spillover is about when iterated unary of a type level is encapsulated by a new count variable using base 10 and Place Value Notation. No matter what the type level is, the unary spillover of this type level is relative to base 10.

By understanding this , you can imagine what ordertypes 4 and 5 are :

ordertype[4] PVN number range

$\underbrace{1000000000}_{10}$ or equivalently, $\underbrace{10...00}_{10}$ All the way up to …

$$\underbrace{99...99}_{\underbrace{99...99}_{\underbrace{99...99}_{\underbrace{99...99}_{\underbrace{99...99}_{\underbrace{99...99}_{\underbrace{99...99}_{\underbrace{99...99}_{999,999,999}}}}}}}}$$

This rather classic looking number has 9 layers but there is no brace on the right side that counts the number of layers. But then by adding 1 to this number and remembering **“The principle of unary spillover into new PVN counting variable and new order type”** we get to the first ordertype[5] number:

$$\left.\underbrace{10...00}_{\underbrace{\vdots}_{\underbrace{10...00}_{10}}}\right\}10$$

The ordertype[5] numbers continue all the way up to …

$$\left.\underbrace{99...99}_{\underbrace{99...99}_{\vdots_{\underbrace{99...99}_{999,999,999}}}}\right\}\left.\underbrace{99...99}_{\underbrace{99...99}_{\vdots_{\underbrace{99...99}_{999,999,999}}}}\right\}\left.\underbrace{99...99}_{\underbrace{99...99}_{\vdots_{\underbrace{99...99}_{999,999,999}}}}\right\}\left.\underbrace{99...99}_{\underbrace{99...99}_{\vdots_{\underbrace{99...99}_{999,999,999}}}}\right\}\left.\underbrace{99...99}_{\underbrace{99...99}_{\vdots_{\underbrace{99...99}_{999,999,999}}}}\right\}\left.\underbrace{99...99}_{\underbrace{99...99}_{\vdots_{\underbrace{99...99}_{999,999,999}}}}\right\}\left.\underbrace{99...99}_{\underbrace{99...99}_{\vdots_{\underbrace{99...99}_{999,999,999}}}}\right\}\left.\underbrace{99...99}_{\underbrace{99...99}_{\vdots_{\underbrace{99...99}_{999,999,999}}}}\right\}\underbrace{99...99}_{\underbrace{99...99}_{\underbrace{99...99}_{\underbrace{99...99}_{\underbrace{99...99}_{\underbrace{99...99}_{\underbrace{99...99}_{\underbrace{99...99}_{999,999,999}}}}}}}}$$

Remember, we are using “The principle of unary spillover into new PVN counting variable and new order type” so by following that principle we get to the first ordertype[6] number:

$$\underbrace{\left.\underbrace{10...00}_{\underbrace{\vdots}_{\underbrace{10...00}_{10}}}\right\}\left.\underbrace{10...00}_{\underbrace{\vdots}_{\underbrace{10...00}_{10}}}\right\}\cdots\left.\underbrace{10...00}_{\underbrace{\vdots}_{\underbrace{10...00}_{10}}}\right\}\left.\underbrace{10...00}_{\underbrace{\vdots}_{\underbrace{10...00}_{10}}}\right\}10}_{10}$$

which is the previous number plus 1.

It should now be reasonably clear how Place Value Notation works in Higher Dimensions.

If you follow the same kind of logic In Base(3) what happens?

Remember that the “key type transition number” in base10 is 999,999,999
(because we are in normal base10, and there are nine 9’s in this PVN number)
So at this type level (counting digits in a 1 dimensional PVN string of digits)
we have reached a transition from unary to PVN so we need a new type level when we get to :

$$999,999,999+1=\underbrace{1000000000}_{10}$$

Note that 10(base3)=3(base10).

So by using the method of variable type unary to base3 transform we can count in base3
That is, whenever we have 111(unary) we replace by 10(base3) no matter what the type is:

0, 1, 2, 10, 11, 12, 20, 21, 22

$22+1=\underset{10}{100}$ and now we continue to use PVN with ordertype = 4 …

$$\underset{10}{101},\ \underset{10}{102},\ \underset{10}{110},\ \underset{10}{111},\ \underset{10}{200},\ \underset{10}{201},$$

$$\underset{10}{210},\ \underset{10}{211},\ \underset{10}{220},\ \underset{10}{221},\ \underset{10}{222},\ \underset{11}{1000},$$

$$\underset{11}{1001},\ \underset{11}{1002},\ \underset{11}{1010},\ \underset{11}{1011},\ \underset{11}{1012},\ \ldots\ \underset{11}{2222},\ \underset{12}{10001},\ \ldots$$

$$\underbrace{22222}_{12},\ \underbrace{100000}_{20},\ \ldots\ \underbrace{222222}_{20},\ \underbrace{1000000}_{21},\ \ldots\ \underbrace{2222222}_{21},\ \underbrace{10000000}_{22},\ \ldots$$

We finally get to :

$\underbrace{22222222}_{22}$ and incrementing gives $\underbrace{22222222}_{22}+1=\underbrace{100000000}_{\underset{10}{100}}$

But remember the principle we are using is, no matter what the type is, whenever we have 111(unary) we replace by 10(base3) . Now the type has changed from number of digits in a 1 dimensional string of digits to the number of layers in a nested brace expression.
So in actual fact we have :

$$\left.\underbrace{22222222}_{22}+1=\underbrace{100000000}_{\underset{10}{100}}\right\}10$$

We are now using base3 to count a new type so this is ordertype = 5 with a notational folding pattern that is fold up, fold left.

Keep on counting , and we get to these numbers:

$$\left.\underbrace{100000001}_{\substack{100\\10}}\right\}10,\ldots \quad \left.\underbrace{222222222}_{\substack{100\\10}}\right\}10,\ldots$$

At this stage , it is reasonable to use ellipsis (…) to save digit-space, and the meaning is understood:

$$\left.\underbrace{100\ldots00}_{\substack{101\\10}}\right\}10,\ldots \quad \left.\underbrace{100\ldots00}_{\substack{222\\10}}\right\}10,\ldots \quad \left.\underbrace{22\ldots22}_{\substack{222\\10}}\right\}10,\ldots$$

And adding 1 to the 3rd number above and we see the dominoes phenomenon yet again…

$$\left.\underbrace{10\ldots00}_{\substack{1000\\11}}\right\}10,\ldots \quad \left.\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{22}}\right\}10,\ldots \quad \left.\underbrace{10\ldots00}_{\underbrace{22\ldots22}_{22}}\right\}10,\ldots \quad \left.\underbrace{22\ldots22}_{\underbrace{22\ldots22}_{22}}\right\}10,\ldots$$

And adding 1 to the 4th number above we see the dominoes phenomenon and a spillover …

$$\left[\left.\underbrace{22\ldots22}_{\underbrace{22\ldots22}_{22}}\right\}10\right]+1=\left.\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{\substack{100\\10}}}\right\}11$$

Keep on incrementing and eventually we get to …

$$\left[\left.\underbrace{22\ldots22}_{\underbrace{22\ldots22}_{\underbrace{22\ldots22}_{22}}}\right\}11\right]+1=\left.\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{\substack{100\\10}}}}\right\}12$$

And then pushing on further …

$$\left.\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{100}_{10}}}}}\right\}20,\ldots \quad \left.\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{100}_{10}}}}}}\right\}21,\ldots \quad \left.\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{100}_{10}}}}}}}\right\}22,\ldots$$

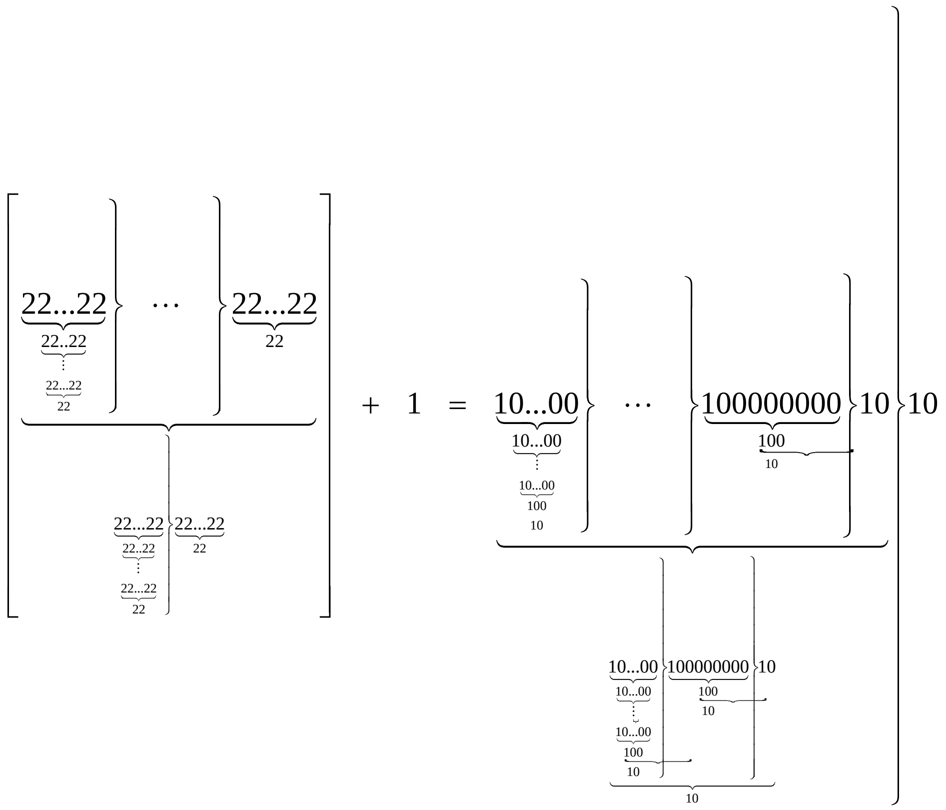
And eventually we get to …

$$\left.\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{10...00}_{\underbrace{100}_{10}}}}}}}}\right\}\underbrace{100}_{10} \quad , \quad \ldots \quad \left.\underbrace{22...22}_{\underbrace{22...22}_{\vdots_{\underbrace{22...22}_{\underbrace{22...22}_{22}}}}}\right\}\underbrace{22222222}_{22}$$

Regarding the 2nd number above , notice that when this number is incremented , there is a dominoes and cascading phenomenon (into ordertype = 6).

$$\left[\left.\underbrace{22...22}_{\underbrace{22...22}_{\vdots_{\underbrace{22...22}_{\underbrace{22...22}_{22}}}}}\right\}\underbrace{22222222}_{22}\right]+1 = \underbrace{\left.\underbrace{10...00}_{\underbrace{10...00}_{\vdots_{\underbrace{10...00}_{\underbrace{100}_{10}}}}}\right\}\left.\underbrace{100000000}_{\underbrace{100}_{10}}\right\}10}_{10}$$

Soon the formulae will be too messy to type up , …
How Type-Multidimensional Base(3) PVN changes from ordertype = 6 to ordertype = 7 …



This has been a survey of the unusual world of Type-Multidimensional Base(3) PVN.
Observe that the notational folding pattern is fold up, fold left, fold up, fold left.
In other words , the folding pattern used is FUL .
I hope this has been interesting on some level.
You can see why in every day life we stick to 1 dimensional Place Value Notation ☺

If you follow the same kind of logic In Base(2) what happens?

Remember that the “key type transition number” in Base(3) is 22 .
(because we are in Base(3), and there are two 2’s in this PVN number)
So at this type level (counting digits in a 1 dimensional PVN string of digits)
we have reached a transition from unary to PVN so we need a new type level when we get to :

$$22+1=\underset{10}{100}$$

Note that 10(Base2)=2(Base10).
So by using the method of variable type unary to Base2 transform we can count in Base2
That is, whenever we have 11(unary) we replace by 10(Base2) no matter what the type is:

But the really weird thing is that the notation for Type-Multidimensional Base(2) PVN doesn't actually settle down …

Starts off Okay: 0, 1, And then whooosh …

$$10=\underset{10}{10}=\left.\underset{10}{10}\right\}10=\underset{10}{\left.\underset{10}{10}\right\}10}=\left.\underset{10}{\left.\underset{10}{10}\right\}10}\right\}10$$

$$=\left.\underbrace{\left.\underset{10}{10}\right\}\underset{10}{10}}_{10}\right\}10=\left.\underbrace{\left.\underset{10}{10}\right\}\underset{10}{10}}_{10}\right\}10\right\}10$$

$$=\left.\underbrace{\underbrace{\left.\underset{10}{10}\right\}\underset{10}{10}}_{10}\right\}10}_{10}\right\}10=\left.\underbrace{\underbrace{\left.\underset{10}{10}\right\}\underset{10}{10}}_{10}\right\}10}_{10}\right\}10\right\}10$$

and so on , and so on , and so fourth

However , … If we relax the condition that " whenever we have 11(unary) we replace by 10(Base2) " then it is possible to have " Unary(3)-To-PVN Type-Multidimensional Base(2) PVN " :
0, 1, 10, 11, and then with the binary string "100" we have an unary(3) situation …

$$\underset{11}{100},\quad \underset{11}{101},\quad \underset{11}{110},\quad \underset{11}{111},\quad \left.\underset{\underset{11}{100}}{1000}\right\}11,\quad \ldots,\quad \left.\underset{\underset{11}{100}}{1111}\right\}11,\quad \left.\underset{\underset{11}{101}}{10000}\right\}11,\quad \ldots$$

$$\left.\underbrace{100000}_{\underset{11}{110}}\right\}11,\quad \ldots\quad \left.\underbrace{1000000}_{\underset{11}{111}}\right\}11,\quad \ldots\quad \left.\underbrace{1111111}_{\underset{11}{111}}\right\}11,$$

$$\left[\left.\underbrace{1111111}_{\underset{11}{111}}\right\}11\right]+1=\left.\underbrace{10000000}_{\underset{\underset{11}{100}}{1000}}\right\}\underset{11}{100}$$

This logic can be continued into higher ordertypes as with Base(3) above …

$$\left.\underbrace{11\ldots11}_{\underbrace{1111111}_{\underset{11}{111}}}\right\}\underset{11}{100},\quad \left.\underbrace{10\ldots00}_{\underbrace{10000000}_{\underset{\underset{11}{100}}{1000}}}\right\}\underset{11}{101},\quad \ldots\quad \left.\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{\underbrace{10000000}_{\underset{\underset{11}{100}}{1000}}}}}\right\}\underset{11}{111},$$

$$\left[\left.\underbrace{11\ldots11}_{\underbrace{11\ldots11}_{\underbrace{11\ldots11}_{\underbrace{11\ldots11}_{\underbrace{1111111}_{\underset{11}{111}}}}}}\right\}\underset{11}{111}\right]+1=\underbrace{\left.\left.\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{\underbrace{10\ldots00}_{\underbrace{10000000}_{\underset{\underset{11}{100}}{1000}}}}}}\right\}\underset{\underset{11}{100}}{1000}\right\}11}_{11}$$

At the end of the day , we always have the most ubiquitous understanding about counting , recognising distinct symbols or entities and listing them …

@

@@

@@@

@@@@ …

And let’s be happy that we have Place Value Notation in Base10 and with the uncomplicated 1 dimensional form, because this is where we do arithmetic.

1 + 2 = 3

3 + 3 = 6

6 + 4 = 10

10 + 5 = 15

15 + 6 = 21

… … …

… … … … … …

… … … … … …

… … … … … …

… … …