# A Theorem for Knuth-Arrows

*By Sbiis Saibian*

email: sbiissaibian@aol.com

*2014.05.21*

## ABSTRACT

Donald Knuth introduced the world to “Up-arrows” back in 1976. Since then they have become the most popular way to express the so called “hyper-operators” , first described by Wilhelm Ackermann with his 1928 *phi-function*. Up-arrow notation allows us to express certain very large numbers, far larger than those expressible with scientific notation, with extreme ease. Performing calculations on expressions involving “Knuth-arrows” however can range from difficult to impossible. Since the numbers involved typically contain more digits than particles in the known universe, and require a similarly large amount of steps to evaluate, it may seem that various expressions are impossible to compare. This is not true. It is possible to devise ways to compare numbers, even without knowing anything about their decimal form.

In this paper I prove a simple theorem for Knuth-Arrows, which can be used to compare some very large numbers (those above ones we can directly compute, up to about $f_{\omega+1}$ in the Fast Growing Hierarchy ). Even better I will show how this can be established axiomatically with 9 simple Axioms.

## INTRODUCTION

Donald Knuth invented the so called “Up-arrows” as a simple way to express very large numbers, and more specifically as a simple and intuitive way to express the ***hyper-operator hierarchy***.

We can begin the development of the *hyper-operator hierarchy* with addition, the most fundamental operator of elementary arithmetic. I will take it for granted that addition is fully understood. From this we can build a hierarchy of higher operations built on addition, and beginning with multiplication. Define multiplication as follows:

$$a * b = \begin{cases} a & if\ b = 1 \\ a + \left(a * (b - 1)\right) & if\ b > 1 \end{cases}$$

Multiplication is defined here as a recursive function using the piecewise-notation. Multiplication builds on addition, in the sense that it uses both itself and addition in its definition.

Exponentiation is defined similarly. Knuth used the "caret", "^", to represent "a" raised to the *bth* power. This was because subscripts and superscripts were difficult to write on older type-writers and computer systems. In any case we now define Knuth's caret operator (exponentiation) as follows:

$$a\hat{}b = \begin{cases} a & if\ b = 1 \\ a * (a\hat{}(b - 1)) & if\ b > 1 \end{cases}$$

Again exponentiation builds on multiplication since it uses itself and multiplication in its definition. Notice the similarity in definition to multiplication. If you take the definition for multiplication and simply swap out addition for multiplication and swap out multiplication for exponentiation, then you get the definition for exponentiation. This suggests a way to continue indefinitely beyond elementary arithmetic. First we can define a series of operator symbols. Knuth chose to include an extra *caret,* or "up-arrow" between the arguments to represent going up one level along the hierarchy of operators. The first hyper-operator is "a^^b" , popularly called *tetration* (from *tetra + exponentiation*). "a^^b" is pronounced "a *tetrated to the bth". .* Next we have "a^^^b", called *pentation* (from *penta + exponentiation*). "a^^^b" is pronounced "a *pentated to the bth*". We can continue with "a^^^^b" (*hexation*), "a^^^^^b" (*heptation*), "a^^^^^^b" (*octation*), etc. Each new operator notation is formed by simply adding an additional up-arrow. Let "@" represent a string of up-arrows "^^^...^^^" where there is at least 1 up-arrow. In this case we can define the hyper-operator hierarchy itself in a recursive fashion by defining each new operator as a recursion on the previous one. We define the *next operator*, @^, as follows:

$$a@\wedge b = \begin{Bmatrix} a & if\ b = 1 \\ a@(a@\wedge(b-1)) & if\ b > 1 \end{Bmatrix}$$

In this simple way we obtain an endless sequence of new operators building on the previous set of operators. There is no highest hyper-operator in this sequence, just as there is no largest number. However, each new hyper-operator is much more powerful than the previous one and allows us to express compactly much larger numbers.

It is also possible to define the whole sequence as a single ternary function. We will use the notation $\langle b, p, k \rangle$, devised by Jonathan Bowers, to represent this ternary function. Here "b" will be called the ***base***, "p" will be called the ***polyponent*** (my generalized term for exponent based on *poly + exponent*), and "k" will be called the ***Knuth-degree***, corresponding to the number of Knuth-arrows. This is not that different than how Ackermann originally defined the hyper-operators ...

$$\langle b, p, k \rangle = \begin{Bmatrix} b \wedge p & if\ k = 1 \\ b & if\ p = 1\ and\ k > 1 \\ \langle b, \langle b, p-1, k \rangle, k-1 \rangle & if\ p > 1\ and\ k > 1 \end{Bmatrix}$$

In this notation we have that...

$\langle a, b, k \rangle =$ a^^^...^^^b w/k ^s

The theorem that I will prove in this paper is the following:

***Theorem I*** [TI]

$$Let\ b, m, n, k \in Z^+$$

$$\langle \langle b, m, k \rangle, n, k \rangle < \langle b, m+n, k \rangle : b, k > 1$$

In order to prove Theorem I it will be necessary to establish some basic properties of the hyper-operator sequence via a series of lemmas.

Before we jump into the proof, I will provide a brief introduction to the power of this notation for readers unfamiliar with Knuth-arrows. For readers already versed in arrow notation, feel free to skip ahead to the next section, "*Establishing our Axioms*."

## *BUILDING INTUITION FOR KNUTH-ARROWS*

Donald Knuth made the point that the distinction between the finite and infinite is not as useful as the distinction between *reasonably large* and *unreasonably large*. To say that a number is "finite" is to say almost nothing about it at all! Calling a number "finite" by no means bounds it in any practical sense of the word. A proof that a certain positive integer exists, for example, provides no actual bound on the size of such a number other than to simply say it's not infinite. $20^{th}$ Century mathematics has demonstrated on several occasions that the number may *indeed* be ***very large***, far larger than is realistic and practical by any stretch of the imagination. Perhaps some of the best examples of this are *Skewes' Number, Graham's Number,* and *TREE(3)* (the first two of these numbers are within the purview of the numbers we will discuss here, but TREE(3) goes far far beyond the scope of our present discussion.)

I concur with Knuth on this point. We will find it much more useful in this instance to distinguish between reasonably and unreasonably large. Keep in mind that however unfathomably large and never-ending the numbers discussed here may *seem to be,* they are all none the less *finite*.

For the purposes of this article we may call a positive integer ***trivially large*** if it's full decimal form can be feasibly be stored in memory. It is trivially large in the sense that, if a *large number notation,* can do no better than generate a number which we could simply *write out* in decimal than why bother having the notation in the first place when decimal notation would suffice? Call a number ***non-trivially large*** or ***remote*** if we ***can not*** feasibly express it in decimal notation.

Although in daily life we are inclined to think of *millions, billions,* and *trillions* as very large numbers, in this context they are quite benign. Any such numbers can easily be written out on an ordinary piece of paper, and so are *trivially large* by definition. Going up another level of complexity, consider numbers with *millions, billions,* or even *trillions* of digits. A human being can not hope to write most of these numbers out as the human life span measured in seconds is only about *three billion.* However on modern hard-drives we can store such numbers using Megabytes, Gigabytes, or Terabytes of data. So it is actually feasible to store such numbers. Beyond this however we hit a grey area and we can not pin the exact moment when a number becomes unfeasibly large. If we include all the data on all the computer systems in the world, then we have an estimated 500 exabytes to work with. This would get us as far as numbers with *millions of trillions* of digits. This however is merely what is feasible at the present. How much data could the world potentially hold? We could provide an upperbound on this by simply converting all the mass of the earth into raw data at the atomic level. In this case we would only have enough data to express numbers with less than $10^{52}$ digits approximately. An even harder limit is reached if we consider all of the sub-atomic particles in the observable universe, of which there is an estimated $10^{80}$. We can thus nominally say a number is *trivially large* if it is less than…

$$10^{10^{80}}$$

… and a number is *remote* if it is larger than this value. We can never know all the digits of a remote integer. Such integers can only be expressed using specialized "large number notations". A common mistake of laymen when first encountering some of the very large numbers that occur only in mathematics is to wonder why we should bother creating a special notation when we could just write out their decimal expansion. The point is that for these numbers we can't! They are non-trivially large, and this necessitates the use of highly specialized and esoteric notations in order to express them. Keep in mind that even these notations can only express a very select subset of the remote integers. Most remote integers can not be expressed in ***any*** notation, which is why I call them *remote.*

As I'll show, there are only a small minority of trivial cases for expressions involving Knuth arrows. Most expressions will result in *non-trivially large* integers. Firstly we can observe that by the rules, any time the polyponent = 1, we will get a trivially large value of the base…

b^^…^^1 = b

If the base=1 then the return value must be 1…

1^^…^^p = 1

So in order to get a non-trivially large value, both the base and polyponent must be greater than 1. The smallest such case is 2 *tetrated to the $2^{nd}$.* Here we consider tetration with a base of 2:

(Note that all operators are to be carried out from right-to-left)

2^^2 = 2^2 = 4 (trivially large)

2^^3 = 2^2^2 = 2^4 = 16 (trivially large)

2^^4 = 2^2^2^2 = 2^2^4 = 2^16 = 65,536 (trivially large)

2^^5 = 2^2^2^2^2 = 2^65,536 ~ $10^{19,726}$ (trivially large)

2^^6 = $2^{2\text{^}65,536}$ ~ $10^{10\text{^}19,726}$ (**remote!**)

So 2^^6 is the first member of this sequence that has far more digits than we could ever hope to compute, and could not be stored in decimal form in the known universe. For reasons that will become more clear as we proceed to the proof, if 2^^6 is remote, so is 2^^7,2^^8,2^^9,etc. and in fact inconceivably more so.

Let's consider the base of 3:

3^^2 = 3^3 = 27 (trivially large)

3^^3 = 3^3^3 = 3^27 = 7,625,597,484,987 (trivially large)

3^^4 = 3^3^3^3 = $3^{7,625,597,484,987}$ ~ $10^{3,638,334,640,024}$ (trivially large)

$$3\text{^^}5 = 3^{3\text{^}7,625,597,484,987} \sim 10^{10\text{^}3,638,334,640,024} \text{ (**remote!**)}$$

After 3^^5, we have further remote integers 3^^6 , 3^^7 , 3^^8, etc. Remember that we have no other way to express these number other than Knuth-arrow notation at this point, or some other equivalent and comparable notation. Knuth arrow notation allows us to express non-trivially large values well beyond what we can express in decimal notation compactly with only a handful of characters.

Let's consider base 4:

$$4\text{^^}2 = 4\text{^}4 = 256 \text{ (trivially large)}$$

$$4\text{^^}3 = 4\text{^}4\text{^}4 = 4\text{^}256 \sim 10^{154} \text{ (trivially large)}$$

$$4\text{^^}4 = 4^{4\text{^}256} \sim 10^{10\text{^}153} \text{ (**remote!**)}$$

Of the remote values so far considered, 4^^4 is the smallest. It seems tantalizingly close to being realizable. Understand that even it is ridiculously far beyond what we could hope to compute.

Here are some further trivially large bases:

$$5\text{^^}2 = 5\text{^}5 = 3125 \text{ (trivially large)}$$

$$5\text{^^}3 = 5\text{^}5\text{^}5 = 5\text{^}3125 \sim 10^{2184} \text{ (trivially large)}$$

$$5\text{^^}4 = 5^{5\text{^}3125} \sim 10^{10\text{^}2184} \text{ (**remote!**)}$$

$$6\text{^^}2 = 6\text{^}6 = 46{,}656 \text{ (trivially large)}$$

$$6\text{^^}3 = 6\text{^}6\text{^}6 = 6\text{^}46{,}656 \sim 10^{36,305} \text{ (trivially large)}$$

$$6\text{^^}4 = 6^{6\text{^}46,656} \sim 10^{10\text{^}36,305} \text{ (**remote!**)}$$

$$7\text{^^}2 = 7\text{^}7 = 823{,}543 \text{ (trivially large)}$$

$$7\text{^^}3 = 7\text{^}7\text{^}7 = 7\text{^}823{,}543 = 10^{695,974} \text{ (trivially large)}$$

$7\wedge\wedge 4 = 7^{7\wedge 823,543} \sim 10^{10\wedge 695,974}$ (**remote!**)

$8\wedge\wedge 2 = 8\wedge 8 = 16,777,216$ (trivially large)

$8\wedge\wedge 3 = 8\wedge 8\wedge 8 = 8\wedge 16,777,216 \sim 10^{15,151,335}$ (trivially large)

$8\wedge\wedge 4 = 8^{8\wedge 16,777,216} \sim 10^{10\wedge 15,151,335}$ (**remote!**)

$9\wedge\wedge 2 = 9\wedge 9 = 387,420,489$ (trivially large)

$9\wedge\wedge 3 = 9\wedge 9\wedge 9 = 9\wedge 387,420,489 \sim 10^{369,693,099}$ (trivially large)

$9\wedge\wedge 4 = 9^{9\wedge 387,420,489} \sim 10^{10\wedge 369,693,099}$ (**remote!**)

$10\wedge\wedge 2 = 10\wedge 10 = 10,000,000,000$ (trivially large)

$10\wedge\wedge 3 = 10\wedge 10\wedge 10 = 10^{10,000,000,000}$ (trivially large)

$10\wedge\wedge 4 = 10^{10\wedge 10,000,000,000}$ (**remote!**)

As you can gather, if the *tetraponent* ( the *tetrational-exponent*) is 3 or less than we get a trivially large value, but if it is 4 or more we get non-trivially large values. The only exception to this is for the very small bases of 2 and 3. It is also true that if the base gets large enough eventually tetrating it to the 3rd will result in a non-trivially large value but this will take a little while. The base has to be >47 in order a^^3 to be non-trivially large. Even a^^2 may be non-trivially large with a large enough base, but the base needs to be roughly $10^{78}$ for this to occur. Eventually the base itself becomes non-trivially large in which case even a^^1 is non-trivially large.

Beyond *tetration,* we find that trivial cases are even more scarce. The triple-arrow, has these cases:

2^^^2 = 2^^2 = 2^2 = 4 (trivially large)

2^^^3 = 2^^2^^2 = 2^^4 = 65,536 (trivially large)

2^^^4 = 2^^2^^2^^2 = 2^^2^^4 = 2^^65,536 (**EXTREMELY REMOTE!)**

Notice how much faster this occurred. Now consider base =3 :

3^^^2 = 3^^3 = 7,625,597,484,987 (trivially large)

3^^^3 = 3^^3^^3 = 3^^7,625,597,484,987 (**EXTREMELY REMOTE!)**

Things begin to blow up really quickly after this…

4^^^2 = 4^^4 = $4^{4\wedge 256} \sim 10^{10\wedge 153}$ (**remote!**)

5^^^2 = 5^^5 (**remote!**)

6^^^2 = 6^^6 (**remote!**)

7^^^2 = 7^^7 (**remote!**)

8^^^2 = 8^^8 (**remote!**)

9^^^2 = 9^^9 (**remote!**)

10^^^2 = 10^^10 (**remote!**)

Almost immediately *2* is enough to make the number explode! Since a^^^1 = a , for all "a", we will still have to wait until we reach non-trivially large bases before a^^^1 is non-trivially large.

After triple-arrows, every other base and operator will explode as long as the polyponent is greater than 1, except for the case where the base and polyponent is 2. 2^^^…^^^2 is *known* as the *degenerate case.* Consider the quadruple-arrow operator:

2^^^^2 = 2^^^2 = 2^^2 = 2^2 = 4 (trivially large)

2^^^^3 = 2^^^4 (**EXTREMELY REMOTE!)**

3^^^^2 = 3^^^3 (**EXTREMELY REMOTE!**)

4^^^^2 = 4^^^4 = 4^^4^^4^^4 (**EXTREMELY REMOTE!**)

etc.

2^^^^^2 = 2^^^^2 = 2^^^2 = 2^^2 = 2^2 = 4 (trivially large)

2^^^^^3 = 2^^^^4 (**EXTREMELY REMOTE!**)

3^^^^^2 = 3^^^^3 (**EXTREMELY REMOTE!**)

4^^^^^2 = 4^^^^4 (**EXTREMELY REMOTE!**)

etc.

As you can see Knuth-arrows are very powerful and quickly become so large that they can not be understood on any terms other than their own.

Now that I have demonstrated the necessity of Knuth-arrow notation to express numbers of this size, we can proceed to the proof. We will begin by presenting 2 simple axioms on which the entire proof is built.

## *ESTABLISHING OUR AXIOMS*

We will use the following Axioms. All variables are positive integers:

**AXIOM I**

*Axiom of Part and Sum* [A1 – part]

*“The sum of positive integers is always greater than any of its summands”*

$$a < a+b$$

**AXIOM II**

*Axiom of Symmetry* [A2 – symmetry]

*“The order one receives sums does not change the final total”*

$$a+b = b+a$$

**AXIOM III**

*Axiom of Order* [A3 – order]

*"The order one combines sums does not change the final total"*

$(a+b)+c = a+(b+c)$

**AXIOM IV**

*Axiom of the Least* [A4 – least]

*"If one positive integer is larger than another, than it is at least as large as the successor of the smaller"*

$a < b \rightarrow a+1 \leq b$

**AXIOM V**

*Axiom of Transitivity* [A5 – transitive]

*" That which is greater than that which is greater is greater still"*

$a<b \;\&\; b<c \rightarrow a<c$

**AXIOM VI**

*Axiom of Greater Sums* [A6 – greater]

"*Greater parts, greater sum*"

$a<b\;,\; c<d \rightarrow a+c < b+d$

**AXIOM VII**

*Axiom of Substitution* [A7 – substitution]

*"If two things are equivalent they are interchangeable"*

Let E(x) be a formula or statement involving occurrences of the expression "x"

$a=b \rightarrow (E(a) = E(b))$ if E( ) is a formula

&

$a=b \rightarrow (E(a) \leftrightarrow E(b))$ if E( ) is a statement

**AXIOM VIII**

*Axiom of Induction* [A8 – induction]

" *That which holds for the first statement of a sequence, and for which the truth of one statement implies the next, may be taken to imply any desired statement in the sequence.* "

Let P(x) be the xth statement. If P(1) holds and $P(k) \rightarrow P(k+1)$ then P(x) holds for all $x \in Z^+$

**AXIOM IX**

*Axiom of multiplication* [A9 – multiplication]

" *m copies of n is the same as n copies of m"*

$m*n = n*m$

Hopefully these Axioms are very simple and easy to accept. With these 9 axioms we can prove Theorem I.

Next we use our axioms to establish the basic properties of the hyper-operators.

## *IMPORTANT PROPERTIES OF THE HYPER-OPERATORS*

Before we can begin our general proof, we must establish some important properties of the hyper-operators. We begin with the following definitions…

---

**DEFINITION 1 – Strictly Increasing Function**

*An integer-to-integer* function, f , is ***Strictly Increasing*** if and only if for any two integers, “a” and “b” , such that a<b it follows that f(a) < f(b). More formally…

f(n) is *Strictly Increasing* (Str.Inc) iff

∀a,b | f(a) < f(b) : a<b

---

**DEFINITION 2 – Everywhere Abundant Function**

*An integer-to-integer function, f, is* ***Everywhere-Abundant*** if and only if for every integer, “n”, f(n) > n. More formally…

f(n) is *Everywhere-Abundant* (E.AB) iff

∀n | n < f(n)

---

We will prove that any function of the form:

$$f(n) = \langle b, n, k \rangle : b > 1$$

For constants “b” and “k”, is a *Strictly Increasing Everywhere-Abundant function.* Such functions have the handy property that the output is always greater than the input, and that if we increase the input we also increase the output. In addition to this, I’ll also demonstrate that, increasing the *base* or *knuth-degree* is guaranteed to give us a larger value under very broad conditions.

We begin by establishing that multiplication is Str.Inc and E.AB. Define…

M(n) = b*n

where $b>1$

We have…

$M(1) = b*1 = b > 1$

$M(n+1) = b*(n+1) = b+b*n = b+M(n) > M(n)$ [A1+A2 – part + symmetry ]

$\therefore M(n) < M(n+1)$

By virtue of the 2nd statement we prove that M is *Strictly Increasing…*

$M(1) < M(2) < M(3) < M(4) < \ldots$

By virtue of the 1st and 2nd statement we prove that the function M is *Everywhere-Abundant* :

$1 < M(1)$

Assume there exists some k such that

$k < M(k)$

then we have…

$k < M(k) < M(k+1)$

$k+1 \leq M(k) < M(k+1)$ [A4 - Least]

$k+1 \leq M(k) < M(k+1)$

means

$k+1 < M(k) < M(k+1)$

or

$k+1 = M(k) < M(k+1)$

In the first case $k+1 < M(k) < M(k+1) \rightarrow k+1 < M(k+1)$ [A5 – transitive]

In the second case we have…

$k+1 = M(k)$ & $M(k) < M(k+1) \rightarrow k+1 < M(k+1)$ [A7 – substitution]

Therefore…

$$k+1 < M(k+1)$$

so…

$$k<M(k) \rightarrow k+1<M(k+1)$$

$$\therefore n < M(n)\ \forall n \text{ [A8 - Induction]}$$

Thus M is also *Everywhere-Abundant.* Next we prove the same properties for the function…

$$E(n) = b^n : b>1$$

Consider…

$$E(1) = b^1 = b > 1$$

$$E(n+1) = b^{(n+1)} = b*b^n$$

Note, b*b^n is greater than b^n because multiplication is E.AB, which means the output must be larger than the input. Therefore…

$$b*b^n > b^n = E(n)$$

$$\therefore E(n) < E(n+1)$$

From $E(n) < E(n+1)$ we have a proof that $E(n)$ is a Str.Inc function since…

$$E(1) < E(2) < E(3) < E(4) < \ldots$$

From $1 < E(1)$ & $E(n) < E(n+1)$ we know that the function is also E.AB, since…

$$1 < E(1)$$

Assume there exists a "k" such that…

$$k < E(k)$$

consider…

$$k < E(k) < E(k+1)$$

$$k+1 \leq E(k) < E(k+1) \text{ [ A4 - least ]}$$

$$k+1 < E(k+1) \text{ [A8 – induction]}$$

Therefore E(n) is both Str.Inc and E.AB.

Now assume there is a function that is both Str.Inc and E.AB called f(n). We prove that the function g(n) is also Str.Inc and E.AB. Define g(n):

$$g(1) = b$$

$$g(n+1) = f(g(n))$$

where $b>1$

To prove "g" is Strictly Increasing we simply observe:

$$g(n) < f(g(n))$$

because f is E.AB

therefore…

$$g(n) < f(g(n)) = g(n+1)$$

$$\therefore g(n) < g(n+1)$$

Thus g(n) is Str.Inc. Furthermore we have…

$$1 < g(1)$$

Assume there exists a k such that…

$$k < g(k)$$

consider…

$$k < g(k) < g(k+1)$$

$$k+1 \leq g(k) < g(k+1) \text{ [ A4 - least ]}$$

$$k+1 < g(k+1) \text{ [A8 – induction]}$$

So “g” is also E.AB. Thus since b^n is Str.Inc and E.AB so are the functions b^^n , b^^^n , b^^^^n, b^^^^^n , b^^^^^^n, etc.

Thus in this way we have demonstrated that all the hyper-operators, as well as the elementary functions b*n and b^n, are *Strictly Increasing* and *Everywhere Abundant.* This becomes our first *lemma*…

---

***LEMMA I*** [L1]

b*n is *Str.Inc & E.AB ,*

$\forall f(n) = \langle b, n, k \rangle : b > 1 \text{ , are } Str.Inc \ \& \ E.AB$

---

This lemma allows us to know, for example, that 3^^^3 is definitely less than 3^^^4 since 3^^^n is Str.Inc. We also know that 3 < 3^^^3 and 4 < 3^^^4 since 3^^^n is E.AB.

This lemma on its own however is not sufficient for the proof of Theorem I. For this we want to demonstrate that each new up-arrow does actually provide larger values. For the sake of completeness I will also demonstrate that a larger base always leads to larger values. This is not strictly necessary for the proof, but provides a strong axiomatic foundation for manipulating and comparing expressions using Knuth-arrows. At the end of the paper I’ll show how this fact can be used along with Lemma I and Theorem I to quickly obtain powerful results with great ease.

First I’ll prove that a larger base means a larger value. We first consider the multiplication functions:

f(n) = b*n , g(n) = c*n

where b<c

Firstly

f(1) = b*1 = b < c = c*1 = g(1)

∴f(1) < g(1)

Next we have…

If f(k) < g(k)

f(k+1) = b*(k+1) = b*k+b

b*k < c*k & b < c → b*k+b < c*k+c [A5 – greater]

c*k+c = c*(k+1) = g(k+1)

→ f(k+1) < g(k+1) [A5+A7 – transitivity+substitution]

∴ f(n) < g(n) : ∀n [A8 – induction]

We now use multiplication to prove the same property for exponentiation…

f(n) = b^n , g(n) = c^n : b<c

b^1 = b < c = c^1

b^1 < c^1 [A7 – substitution]

If b^k < c^k

b^(k+1) = b*b^k

We know that multiplication is Str.Inc, therefore…

b*b^k < b*c^k

But we also know that a larger base results in a larger value so…

b*c^k < c*c^k = c^(k+1)

Thus… b^n < c^n for all n.

Again we generalize this. Assume that b@n < c@n, as well as "@" being Str.Inc and E.AB.

b@^1 = b < c = c@^1

b@^1 < c@^1 [A7]

Assume b@^k < c@^k

b@^(k+1) = b@(b@^k)

b@(b@^k) < c@(b@^k) [b@n < c@n]

c@(b@^k) < c@(c@^k) [@ is Str.Inc]

→ b@^(k+1) < c@^(k+1)

∴All the hyper-operators produce larger values if the base is increased.

This is our second lemma…

---

***LEMMA II*** [L2]

$$b * p < c * p : b < c$$

$$\langle b, p, k\rangle < \langle c, p, k\rangle : b < c$$

---

Lastly we will prove that each new hyper-operator does in fact produce much larger values than all previous hyper-operators.

Firstly we must observe that…

b*1 = b^1 = b^^1 = b^^^1 = b^^^^1 = …

So all the operators produce the same value if the polyponent =1. At 2 however we have…

2*2 ≤ b*2 [ Lemma II ]

This works since we established a larger base means a larger output. Next we have…

b*2 ≤ b*b = b^2 ≤ b^b = b^^2 ≤ b^^b = b^^^2 ≤ b^^^b = b^^^^2 ≤ …

By substitution [A7] and transitivity [A5] this establishes that if the polyponent is 2, then the output is at least 4. Note that the ≤ comes into play only because "b" might be 2, in which case we have…

$$4 = 2*2 = 2\string^2 = 2\string^\string^2 = 2\string^\string^\string^2 = 2\string^\string^\string^\string^2 = \ldots$$

However if b>2 then we can replace the ≤ with < in each instance and we obtain…

$$b*2 < b\string^2 < b\string^\string^2 < b\string^\string^\string^2 < b\string^\string^\string^\string^2 < \ldots$$

So each time we go up by *knuth-degree* we are indeed getting a larger value here as long as the base is greater than 2. Now let's consider the next step when the polyponent = 3.

$$b*3 < b*4 = b*(b*2) \leq b*(b\string^2) = b\string^3$$

$$\therefore b*3 < b\string^3 \text{ [A5+A7]}$$

From this we can gather…

$$b*3 < b\string^3 < b*b\string^3 = b\string^4$$

Since b*n is E.AB

$$3 < b*3$$

By the Axiom of the Least [A4] …

$$4 \leq b*3$$

Therefore…

$$4 \leq b*3 < b\string^3 < b\string^4$$

$$4 < b\string^3 < b*b\string^3 = b\string^4$$

therefore…

$$b*4 < b*b\string^3 = b\string^4$$

$$b*4 < b\string^4$$

Assume the following holds for k:

$$b*k < b\string^k$$

Now consider…

$$b*k < b^k < b*b^k = b^{(k+1)}$$

$$k+1 \leq b*k < b^k < b*b^k = b^{(k+1)}$$

$$k+1 < b^k$$

$$b*(k+1) < b*(b^k) = b^{(k+1)}$$

$$\therefore b*(k+1) < b^{(k+1)}$$

Thus we have shown that exponentiation, $b^n$, always produces larger values than $b*n$ provided $n>2$.

To prove the general case we consider b@n which is Str.Inc and E.AB. We need to show that…

$$b@n < b@^{n} : n>2$$

We have…

$$b@1 = b@^{1} = b$$

$$b@2 \leq b@b = b@^{2}$$

Next we have…

$$b@3 < b@4 \leq b@(b@2) \leq b@(b@b) = b@^{3}$$

$$b@3 < b@^{3}$$

Assume the following holds for k:

$$b@k < b@^{k}$$

Then we have…

$$k < b@k < b@^{k} < b@b@^{k} = b@^{(k+1)}$$

$$k+1 \leq b@k < b@^{k}$$

$$k+1 < b@^{k}$$

b@(k+1) < b@(b@^k) = b@^(k+1)

∴ b@(k+1) < b@^(k+1)

Thus we have…

b@n < b@^n : n>2

Thus we conclude that all the hyper-operators output larger values than the previous hyper-operator. We can state this in our third lemma…

---

***LEMMA III*** [L3]

$$b * p < b\hat{}p : p > 2$$

$$\langle b, p, k\rangle < \langle b, p, k+1\rangle : p > 2$$

---

We now use *Lemma I* and *Lemma III* to aid us in proving *Theorem I*.

## THEOREM I : TETRAPONENT LEMMA

We first prove the *Tetraponent Lemma (TL)* for positive integer values, before moving on to the more general case. A *tetraponent* ( from “tetra” + “exponent” ), is the polyponent of the “^^” operator, commonly known as *tetration.* The *Tetraponent Lemma* can be stated as…

(b^^m)^^n < b^^(m+n) : b>1

As long as the base is greater than 1, the adding of the tetraponents is a larger value. This has many interesting consequences, one of them being the ability to inductively prove it for all higher hyper-operators.

We will assume from here on in that b is a positive integer greater than 1. We can begin by getting the trivial cases out of the way. From our prior definitions it follows that…

(b^^1)^^1 < b^^2

(b^^m)^^1 < b^^(m+1)

(b^^1)^^n < b^^(1+n)

This only leaves cases in which neither "m" nor "n" is equal to 1. The smallest case of this is if m=n=2. To ease our transition to the more general case we begin by eliminating cases in which one or both of these is equal to 2. Firstly we have…

(b^^2)^^2 = (b^b)^(b^b) = b^(b*b^b) = b^b^(b+1)

b^b^(b+1) is less than b^b^b^b, because b+1 < b^b

This can be seen because b+1 < b+b = b*2 ≤ b*b = b^2 ≤ b^b

We now consider the other cases…

Assume m>2

(b^^m)^^2 = (b^^m)^(b^^m) = b^(b^^(m-1) * b^^m ) < b^(b^^m)^2

= b^b^(2*b^^(m-1)) ≤ b^b^(b*b^^(m-1)) = b^b^b^(1+b^^(m-2))

We need to show that…

1+b^^(m-2) is less than b^^(m-1)

We can observe that…

1+b^a < b^(a+1) for b>2, and "a" is an positive integer.

This is because…

1+b^a < b^a+b^a = (b^a)*2 = 2*b^a [A9 – multiplication]

≤ b*b^a [L2] = b^(a+1)

Furthermore we know from earlier that

b+1 < b^b

therefore…

1+b^b < b^(b+1) < b^b^b

1+b^b^b < b^(b^b+1) < b^b^(b+1) < b^b^b^b

1+b^b^b^b < b^(b^b^b+1) < b^b^(b^b+1) < b^b^b^(b+1) < b^b^b^b^b

etc.

Inductively we can prove it for the general case, thus…

1+b^^(m-2) is less than b^^(m-1)

Therefore…

b^b^b^(1+b^^(m-2)) < b^b^b^b^^(m-1) = b^b^b^^m = b^b^^(m+1) = b^^(m+2)

therefore…

(b^^m)^^2 < b^^(m+2)

Now we need to show…

(b^^2)^^n < b^^(2+n)

We begin…

(b^^2)^^n = (b^b)^^n = (b^b)^(b^b)^(b^b)^…^(b^b)^(b^b) w/n copies of b^b

We begin in an inductive fashion…

(b^b)^(b^b) = b^(b*b^b) = b^b^(1+b)

(b^b)^(b^b)^(b^b) = b^(b*b^b^(1+b)) = b^b^(1+b^(1+b))

(b^b)^(b^b)^(b^b)^(b^b) = b^(b* b^b^(1+b^(1+b))) = b^b^(1+b^(1+b^(1+b)))

…

b^b^(1+b^(1+b^( … (1+b^(1+b^(1+b))) … ))) w/n+1 copies of "b"

The extra “baggage” of +1’s can easily be eliminated by observing that in each case the exponent to the right is some positive integer and we already proved 1+b^a < b^(a+1) in all such cases.  We can also show that 2+b^a ≤ b^(a+1)

2+b^a ≤ b+b^a ≤ b^a+b^a = 2*b^a ≤ b*b^a = b^(a+1)

Thus we can conclude…

b^b^(1+b^(1+b^( … (1+b^(1+b^(1+b))) … )))

< b^b^b^(2+b^( … (1+b^(1+b^(1+b))) … ))

…

≤ b^b^b^b^ … (2+b^(1+b^(1+b)))

≤ b^b^b^b^ … ^b^(2+b^(1+b))

≤ b^b^b^b^ … ^b^b^(2+b) w/n+1 “b”s

Now we merely need to show that 2+b <= b^b : b>1. This is accomplished handily:

2+b ≤ b+b = 2*b ≤ b*b = b^2 ≤ b^b

This demonstrates that  2+b ≤ b^b. Thus…

b^b^b^b^ … ^b^b^(2+b) w/n+1 “b”s

≤ b^b^b^b^ … ^b^b^b^b w/n+2 “b”s

thus…

(b^^2)^^n < b^^(2+n)

The only cases that remain now are ones in which both “m” and “n” are greater than 2. So we now consider the general case…

(b^^m)^^n

(where m,n>2)

We have…

(b^^m)^^2 = (b^^m)^(b^^m) = b^(b^^(m-1)*b^^m)

= b^b^(b^^(m-2)+b^^(m-1)) < b^b^b^(1+b^^(m-2))

(b^^m)^^3 = (b^^m)^(b^^m)^(b^^m) < (b^^m)^b^b^b^(1+b^^(m-2))

= b^(b^^(m-1)*b^b^b^(1+b^^(m-2)))

= b^b^(b^^(m-2)+b^b^(1+b^^(m-2))) < b^b^b^(1+b^(1+b^^(m-2)))

…

(b^^m)^^n

< b^b^b^(1+b^(1+b^(…(1+b^(1+b^(1+b^^(m-2))))…))) w/n+2 b's

< b^b^b^b^(2+b^(…(1+b^(1+b^(1+b^^(m-2))))…)))

…

≤ b^b^b^b^b^…^b^(2+b^(1+b^(1+b^^(m-2))))

≤ b^b^b^b^b^…^b^b^(2+b^(1+b^^(m-2)))

≤ b^b^b^b^b^…^b^b^b^(2+b^^(m-2)) = b^(2+b^^(m-2))#(n+1)

Now we focus on 2+b^^(m-2)

This is at least 2^^(3-2) = 2^^1 = 2

If so we have…

2+b ≤ b+b = 2*b ≤ b*b = b^2 ≤ b^b

2+b^a ≤ b+b^a ≤ b^a+b^a = 2*b^a ≤ b*b^a = b^(a+1)

2+b^b < b^(b+1) < b^b^b

2+b^b^b < b^(b^b+1) < b^b^(b+1) < b^b^b^b

2+b^b^b^b ≤ b^(b^b^b+1) < b^b^b^b^b

etc.

$$2+b\^\^(m-2) \leq b\^\^(m-1)$$

Thus 2+b^^(m-2) ≤ b^^(m-1)

Thus…

b^b^…^b^b^(b^^(m-1)) w/n b's

= b^^(m-1+n+1) = b^^(m+n)

Thus…

(b^^m)^^n < b^^(m+n)

Thus we have proven the TL.

## THEOREM I : GENERAL CASE

We now generalize our result. This will accomplished inductively. We will assume that the following holds for…

(b@m)@n < b@(m+n)

For some set of operations equal to and below "@". We then demonstrate that if this is so, then necessarily it must follow…

(b@^m)@^n < b@^(m+n)

To accomplish this we must use the properties we proved about the hyper-operators in the sub-heading "Important properties of the Hyper-Operators". Firstly we prove the trivial cases…

(b@^1)@^1 = b@^1

However…

b@^1 < b@^2

[L1 - @^ must be a strictly increasing function ]

∴(b@^1)@^1 < b@^(1+1)

(b@^m)@^1 = b@^m < b@^(m+1)

[ L1 - @^ is Str.Inc ]

∴ (b@^m)@^1 < b@^(m+1)

Next…

(b@^1)@^n = b@^n < b@^(n+1) ∵ [L1 - @^ is Str.Inc]

∴ (b@^1)@^n < b@^(1+n)

Now we consider the less trivial case of m=n=2.

(b@^2)@^2 = (b@b)@(b@b)

Recall that "@" holds under [TI], therefore…

(b@b)@(b@b) < b@(b+b@b)

b < b@b

[L1 - @ is E.AB]

Therefore…

b@(b+b@b) < b@(b@b+b@b) = b@(2*b@b) ≤ b@(b*b@b)

We have…

1 < b < b@b

2 < b@b

Therefore…

b*b@b < b^b@b < b^^b@b < b^^^b@b < …

via [L3] which states that the higher the *knuth-degree* the higher the value if the input is greater than 2. b@b must be at least 4, so this holds.

Let % be the operation such that %^ = @. In this case we eventually reach…

b%b@b = b@(b+1)

Thus we have…

b@b@(b+1) < b@b@(b+b) = b@b@(b*2) ≤ b@b@(b*b)

≤ b@b@b@b = b@^(4)

Therefore…

(b@^2)@^2 < b@^(2+2)

Next we have…

(b@^m)@^2 = (b@^m)@(b@^m) = b@(b@^(m-1) + b@^m)

< b@(b@^m + b@^m) = b@(2*b@^m) ≤ b@(b*b@^m) < b@b@^(m+1)

= b@^(m+2)

∴(b@^m)@^2 < b@^(m+2)

Next we have…

(b@^2)@^n

Here we consider the steps…

(b@^2)@(b@^2) = b@(b+b@b) < b@b@(1+b)

Let…

R(1) = 1+b

R(n+1) = 1+b@R(n)

b@b@(1+b) = b@b@R(1)

(b@^2)@^3 < (b@^2)@b@b@R(1) < b@(b+b@b@R(1)) < b@b@(1+b@R(1)) = b@b@R(2)

if…

(b@^2)@^k < b@b@R(k-1)

…

(b@^2)@^(k+1) = (b@^2)@b@b@R(k-1) < b@(b+b@b@R(k-1))

< b@b@(1+b@R(k-1)) = b@b@R(k)

Thus it holds that…

(b@^2)@^n < b@b@R(n-1)

From this we obtain…

b@b@(1+b@R(n-2)) < b@b@b@(1+R(n-2)) = b@b@b@(2+b@R(n-3))

< b@b@b@b@(2+b@R(n-4)) < …

< b@(2+b@R(1))#(n-1) < b@(b@(2+b))#(n-1) = b@(2+b)#n

≤ b@(b+b)#n = b@(b*2)#n ≤ b@(b*b)#n ≤ b@(b@b)#n = b@^(2+n)

∴ (b@^2)@^n < b@^(2+n)

This leaves only the general case in order to prove [TI] conclusively.

(b@^m)@^n

(b@^m)@(b@^m) < b@(b@^(m-1) + b@^m) < b@(b@^m + b@^m) = b@(2*b@^m)

≤ b@(b*b@^m) = b@(b*b@b@^(m-1)) < b@(b%b@b@^(m-1)) = b@b@(1+b@^(m-1))

Let R(1) = 1+b@^(m-1)

R(n+1) = 1+b@R(n)

So…

(b@^m)@^2 < b@b@R(1)

(b@^m)@^3 < (b@^m)@b@b@R(1) < b@(b@^(m-1) + b@b@R(1))

< b@(b*b@b@R(1)) ≤ b@(b%b@b@R(1)) = b@b@(1+b@R(1)) = b@b@R(2)

Let…

(b@^m)@^k < b@b@R(k-1)

(b@^m)@^(k+1) < (b@^m)@(b@b@R(k-1)) < b@(b@^(m-1)+b@b@R(k-1))

< b@(b*b@b@R(k-1)) ≤ b@(b%b@b@R(k-1)) = b@b@(1+b@R(k-1)) = b@b@R(k)

Thus…

(b@^m)@^n < b@b@R(n-1)

Unpacking it we have…

b@b@R(n-1) = b@b@(1+b@R(n-2)) < b@b@(b*b@R(n-2))

≤ b@b@b@(1+R(n-2))

= b@b@b@(2+b@R(n-3))

< …. < b@(2+b@R(1))#(n-1) < b@(b*b@R(1))#(n-1)

≤ b@(b%b@R(1))#(n-1)

= b@(b@(2+b@^(m-1))#(n-1) = b@(2+b@^(m-1)#n

< b@(b*b@b@^(m-2))#n

≤ b@(b%b@b@^(m-2))#n = b@(b@(1+b@^(m-2)))#n

< b@(b@^m)#n = b@^(m+n)

Thus we have proven [TI].

## CONCLUSION

This paper I believe has demonstrated, using sound mathematical reasoning, that inescapably…

(b^…^m)^…^n < b^…^(m+n)

And this result is built on the 9 Axioms presented in this paper

Once obtained it can easily be put to good use on any number of difficult comparison and bounding problems involving up-arrows very handily. For example, a common mentioned number in relation to Graham's Number, by means of size comparison is "a power tower of *googolplexes* a *googolplex terms high".* This can be written compactly using Knuth-Arrows as…

(10^10^100)^^(10^10^100)

Now we use [TI] to prove that this is way less than *Graham's Number…*

(10^10^100)^^(10^10^100) < (10^10^10^10)^^(10^10^100)

= (10^^4)^^(10^10^100) < 10^^(4+10^10^100) [TI]

< 10^^10^(1+10^100) < 10^^10^10^101 < 10^^10^10^10^10 < 10^^10^^4 < 10^^10^^10 = 10^^^3

At this point we may seem to be stuck since we require a base of 3 for direct comparison. However…

10 < 3^3 < 3^3^3 = 3^^3 < 3^^3^^3 = 3^^^3

Thus…

10^^^3 < (3^^^3)^^^3 [L2 – base change]

< 3^^^6 [TI] < 3^^^3^3 < 3^^^3^^^3 [L1+L3] = 3^^^^3 = G(1) << G(64)

Thus we prove that not only is *googolplex*^^*googolplex* much less than G(64), but it is even smaller than G(1). Thus TI is very versatile, and among one of its uses is the ability to perform base changes to smaller bases and still prove upper-bounds!

TI can now be used to substantiate other facts about large numbers. The lemmas used in its proof (L1,L2, and L3) are also useful in assisting TI, and may be considered as part of TI.