

EQUIPOTENCY OF E# with FGH_ω

Author: Sbiis Saibian

Email: sbiissaibian@aol.com

Date: 2015.03.27

ABSTRACT

What does it mean to say the order of a particular googological system is such and such in the fast growing hierarchy? This notion has been used to gauge the strength of various notations for the purposes of comparison. For example we can say that Conway's Chain-arrows has order-type ω^2 . This is it's "ordinal strength" in FGH.

Informally what we mean when we assign such an ordinal to a system is that it can only reasonably express numbers of roughly this size given a reasonable character limit. More formally the ordinal is the smallest ordinal "a" such that $f_a(n)$ eventually dominates any function definable within a given system. It is this latter definition which we will be using. So when we say Chain Arrows is of order- ω^2 we are actually saying that every function definable in Chain Arrows grows slower than $f_{\omega^2}(n)$.

The goal in this article is to demonstrate that E# is indeed on the order of ω . This statement can be made precise enough that it can be proven. The basic strategy will be to show that for every member of FGH_ω there is a function in E# with at least the same growth rate. This proves that the limit ordinal of E# can be no smaller than ω , since no function in FGH_ω dominates over all member functions of E#.

Technically this doesn't prove it's exactly ω , only that it's at least of this order. To do this it would be necessary to demonstrate that f_{ω} grows faster than any function definable in E#. This we will also demonstrate at the end of the article. Before we begin here are some definitions:

PRELIMINARY DEFINITIONS

We begin by establishing some important definitions...

A function (from positive-integers to positive-integers) is **STRICTLY INCREASING FUNCTION (S.IN)** if and only if:

For EVERY $a < b$ we have $f(a) < f(b)$

When $f(x) = x$ we say that "x" is a **FIXED-POINT of f**. When $f(x) < x$ we say that "x" is a **DEFICIENCY-POINT of f**. When $f(x) > x$ we say that "x" is a **ABUNDANCY-POINT of f**.

Thus an **EVERYWHERE ABUNDANT FUNCTION (E.AB)** is a function such that every point is an abundancy-point. ie:

For EVERY x we have $x < f(x)$

In googology we may adopt the following definition for growth rate.

We will say $g(n)$ grows faster than $f(n)$ if and only if for every $\Delta n, k \geq 0$, there exists an N such that:

$$f(n + \Delta n) + k < g(n) : n > N$$

In other words, if we have two sequences, and no matter how much of a head start we give f , Δn , or a handicap, k , g always "wins out" eventually (gains a permanent and irreversible lead), then g is clearly a faster growing sequence than f .

This is symbolized:

$$f(n) [<] g(n)$$

If the condition fails for at least one combination of the integer constants $(\Delta n, k)$, then $g(n)$ does not grow faster than $f(n)$ and we can write:

$$f(n) [!<] g(n)$$

If $f(n) [!<] g(n)$ and $g(n) [!<] f(n)$ then $f(n)$ and $g(n)$ are of the same growth rate and we can write:

$$f(n) [=] g(n)$$

A function in **growth class- α** , where α is an ordinal, is a function within the same **class** as f_{α} in the Fast Growing Hierarchy. In other words:

$f(n)$ is of growth class- α if and only if

$$f(n) [=] f_{\alpha}(n)$$

or

" α " is the smallest ordinal such that:

$$f(n) [<] f_{\alpha}(n)$$

$f(n)$ is above class- α if and only if:

$$f_{\alpha}(n) [<] f(n)$$

A **SYSTEM OF NOTATION** is a legal set of **CONSTRUCTIONS** only involving a predefined set of functions. For example if our system of notation only contains the function $S(n)$, then the legal constructions are: n , $S(n)$, $S(S(n))$, $S(S(S(n)))$, etc. Functional powers are not allowed. Each construction will involve a finite set of arguments (in this case only 1 argument is possible). Some of these arguments may take on a constant, while others can take on the independent variable n . Every such construction defines a function with some growth rate.

The **LIMIT ORDINAL** of a System of Notation is the least member of FGH with a growth rate greater than any function definable within that system of notation using constructions.

NOTE ON THE DEFINITION OF GROWTH RATES

The definition of the limit ordinal for a notation system implicitly rests on the definition of growth rates. Computer science uses Big-O notation to establish a classification system of growth rates, however I use a slightly different definition (growth class) when working in googology.

CONJECTURE

The SYSTEM OF NOTATION $E\#$ is of order- ω *

**meaning $f_\omega(n)$ is the smallest member of FGH such that it dominates every function in $E\#$*

To Prove this conjecture we will establish that:

$$f_k(n) < E_{100\#100\#\dots\#100\#}(2n) \text{ w/k-2 } 100s : k > 2$$

&

$$E_{100\#100\#\dots\#100\#}n \text{ w/k } 100s < f_\omega(n) : n > 99 \text{ \& } n \geq k+3$$

A PRELIMINARY DISCUSSION OF OUR STRATEGY

For the purposes of proving our conjecture it will not be necessary to enumerate every possible construction/function in $E\#$. It suffices to show that for every function in FGH_ω there exists a function in $E\#$ which *grows* at least as fast. This means that no order less than ω could dominate over the entire system.

We will do this by first establishing a special hierarchal sub-set of $E\#$ and show that every member of FGH_ω there is a member of this sub-set which grows at least as fast.

In order to establish that one function grows at least as fast as a member of FGH , f_k , all we need to show is that:

$$f_k(n) < f(n) : n > N$$

If this is true than f_k CAN NOT be of a higher **growth class**, since for $\Delta n=0$, there is no point at which f_k gains a permanent lead on f as required by the

definition. If the above holds it follows that $f_k(n)$ $[<=]$ $f(n)$, and that $f(n)$ $[!<]$ $f_k(n)$.

To show that $f_\omega(n)$ does indeed grow faster than $E\#$, we will show that every function in $E\#$ grows slower than a member of our hierarchal sub-set of $E\#$, and then show that every member of this hierarchy is beaten by $f_\omega(n)$ eventually.

Establishing fundamental Properties of PR Hierarchies

For the hardcore googologist's amongst you who don't mind delving into the math we can prove that $E\#$ and FGH_ω are stabilized against each other. This means that there exists a "c" such that f_k $[<]$ E_{k+c} holds for all k.

Some members of the community have expressed a high level of skepticism for using ordinals and FGH as a means to measure the strengths of systems. It's important to recognize, firstly, the necessity of **some means of comparison**. There is nothing particularly sacred about FGH, but it has the nice property that it directly relates orders of growth with ordinals.

The objections to this sort of use of ordinals and FGH basically boils down to a *lack of agreed upon formal definition*, and a *lack of formal proof*. Here I will provide both to back up my contention that such properties are not only necessary, but in fact hold for the vast majority of cases we might consider in googology. At very least I will show that my claim that $E\#$ is on par with FGH_ω is sound and can be supported formally.

Before we delve into the proof, we need to establish some important properties of googological functions. Namely we would like to show that an arbitrary number of applications of primitive recursion to a **Strictly increasing Everywhere Abundant function** also results in an **S.IN/E.AB** function.

These are ideal properties for any googological function. A function being S.IN guarantees us that if we want a larger number it suffices to have a larger argument. A function being E.AB guarantees that the number we are producing is always larger than the number we are putting in, making the function "useful" from a googological standpoint (to understand why just consider a function for which it returns the argument or produces a smaller value. In that case we'd be better off with the number we were plugging in than the result, defeating the purpose of the function in the first place).

With very few exceptions, almost all googological notations are designed implicitly with these properties in mind.

Let $f(n)$ be a S.IN/E.AB function from $\mathbb{Z}^+ \rightarrow \mathbb{Z}^+$

let g be a new function from $\mathbb{Z}^+ \rightarrow \mathbb{Z}^+$ defined as follows:

$$g(n) = \begin{cases} f(k) & : n=1 ; \\ f(g(n-1)) & : n>1 ; \end{cases}$$

where k is a positive integer.

First we prove that $g(n)$ is S.IN

To do this it is sufficient to show

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

We observe:

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

bec. f is E.AB $f(g(n)) > g(n)$

therefore:

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

which proves that...

$$g(1) < g(2) < g(3) < g(4) < \dots$$

therefore

$$g(a) < g(b) : a < b$$

so g is S.IN

Next we use this property to prove it's E.AB

$$\text{Firstly } g(1) = f(k) > k \geq 1$$

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

Now in general...

$$g(n) = f(f(f(f(\dots f(f(f(k))) \dots)))) \text{ w/n } f\text{'s}$$

Here we use the **Axiom of Minimum Successorship (AxMS)**:

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

From this we can gather that...

$$k < f(k) < f(f(k)) < f(f(f(k))) < \dots \text{ etc.}$$

$$\text{and that } f(k) \geq k+1$$

$$f(f(k)) \geq k+2$$

$$f(f(f(k))) \geq k+3$$

etc.

Therefore

$$f(f(f(\dots f(f(f(k))) \dots))) \text{ w/n } f\text{'s } \geq k+n$$

$$g(n) \geq k+n > n$$

$$g(n) > n$$

$$\therefore g(n) \text{ is E.AB}$$

So if " f " is S.IN & E.AB, and " g " iterates " f " with initial value k , then g is also S.IN & E.AB.

Note that this follows regardless of what else is true about "f". So it follows then that for any function which is S.IN/E.AB that

$PR^1 f$ is S.IN/E.AB , $PR^2 f$ is S.IN/E.AB , $PR^3 f$ is S.IN/E.AB , etc.

The E# Hierarchy

Now we define the E# Hierarchy (E#H):

$$E100\#^*(0)n = E_n$$

$$E100\#^*(1)n = E100\#n$$

$$E100\#^*(2)n = E100\#100\#n$$

$$E100\#^*(3)n = E100\#100\#100\#n$$

$$E100\#^*(4)n = E100\#100\#100\#100\#n$$

...

etc.

Under this scheme we eventually want to prove:

$$f_k(n) < E100\#^*(k-1)n$$

Establishing our Lemmas

But first we establish some basic properties of this hierarchy.

(Lemma 1) That every member is an S.IN/E.AB function

(Lemma 2) That each new function dominates all previous functions

(Lemma 3) $3@n < @(n+1)$

(Lemma 4) $@a@ < @b@ : a < b$

(Lemma 5) $@(2n) < @100\#n$

Proof of Lemma 1

To establish (1) it is only necessary to show that the base function E_n is S.IN/E.AB, and that each new function is a primitive recursion of the previous function.

$$E(n+1) = 10 * E_n > E_n$$

Therefore E_n is S.IN

$$E_1 = 10 > 1$$

and E_n is S.IN

Therefore

$$E_2 > E_1 > 1 \rightarrow E_2 > E_1 \geq 2 \rightarrow E_2 > 2$$

$$E_3 > E_2 > 2 \rightarrow E_3 > E_2 \geq 3 \rightarrow E_3 > 3$$

and in general if

$$E_k > k$$

Then

$$E(k+1) > E_k > k \rightarrow E(k+1) > E_k \geq k+1 \rightarrow E(k+1) > k+1$$

so E_n is also E.AB

Now we can also observe that if $@_n$ is S.IN/E.AB, then the next function in the hierarchy is $@_{100\#n} = @@@@...@@@_{100}$ w/n $@_s$. In other words, that it's a PR of $@_n$. The next function in the hierarchy must therefore be S.IN/E.AB. Thus every member of $E\#H$ is S.IN/E.AB.

Proof of Lemma 2

Next we need to establish:

$$n < E_n < E_{100\#n} < E_{100\#100\#n} < E_{100\#100\#100\#n} < E_{100\#100\#100\#100\#n} < \dots$$

This will be vital for later in our proof.

To establish the above claim it suffices to show that:

$$@_n < @_{100\#n}$$

$$\phi_{100\#1} = \phi_{100} > \phi_1$$

$$\text{Assume } \phi_{100\#k} > \phi_k$$

$$\phi_{100\#(k+1)} = \phi(\phi_{100\#k}) > \phi(\phi_k)$$

This holds because ϕ_n is S.IN and $\phi_{100\#k} > \phi_k$ by assumption

also

$$\phi(\phi_k) \geq \phi(k+1)$$

because ϕ_n is E.AB, and by the AxMS $\phi_n \geq n+1$

Therefore:

$$\phi_{100\#(k+1)} > \phi(k+1)$$

QED

So...

$$n < \phi_n < \phi_{100\#n} < \phi_{100\#100\#n} < \phi_{100\#100\#100\#n} < \phi_{100\#100\#100\#100\#n} < \dots \text{ etc.}$$

Even this won't be quite sufficient for our purposes. We must also prove Lemma 3.

Proof of Lemma 3

We can begin by observing that:

$$3\phi_n < 10\phi_n = \phi(n+1)$$

now

$$\text{Assume } 3\phi_n < \phi(n+1)$$

Prove this also holds for the next function...

$$\text{Begin with } \phi_{100\#1} = \phi_{100}$$

$$\text{and } \phi_{100\#2} = \phi(\phi_{100}) \geq \phi_{101} > 3\phi_{100} = 3\phi_{100\#1}$$

Now assume:

$$3\phi_{100\#n} < \phi_{100\#(n+1)}$$

We prove for the next case:

$$\begin{aligned} 3^{n+2} &= 3(3^{n+1}) > 3(3^n) = 3(3^n + 3^n + 3^n) \\ &> 3((3^n)+1) \\ &> 3(3^n) = 3^{n+1} \end{aligned}$$

So not only is it true that $3^n < 3^{n+1}$, it's also easy to prove.

Proof of Lemma 4

A last potential stumbling block to our proof is that we need to know that in E_n if we increase any argument that we are guaranteed to get a larger output.

We can provide proof for this as well:

The proof is done through a double induction. We begin with the last argument and work our way backwards to any arbitrary argument. Firstly:

$$a < b : a < b$$

because we showed E_n is S.I.N

So now we prove it for the 2nd to last argument:

$$a < b$$

$$aa < bb : E_n \text{ is S.I.N and } a < b$$

$$aaa < bbb : E_n \text{ is S.I.N and } aa < bb$$

$$aaaa < bbbb : E_n \text{ is S.I.N and } aaa < bbb$$

...

therefore:

$$aaa\dots aaa < aaa\dots aab \text{ for arbitrary but equal numbers of } a$$

therefore:

$$a^n < b^n$$

now since this is true we can go even further...

First we need to point out that the fixed arguments needn't be "100" for the function to be S.IN and E.AB. Remember that we proved that if the base function is S.IN/E.AB then it doesn't matter what the base value is. For any particular base the result will be a S.IN/E.AB function. Therefore:

$$a^n < b^n$$

$$a^{n^2} = a^{(a^n)} < b^{(a^n)} \text{ [previously established]}$$

$$< b^{(b^n)}$$

since $a^n < b^n$ and b^n is S.IN

$$\text{thus } b^{(a^n)} < b^{(b^n)}$$

$$a^{(a^n)} < b^{(b^n)}$$

$$a^{n^2} < b^{n^2}$$

Furthermore:

$$a^{n^3} =$$

$$a^{(a^{(a^n)})} < b^{(a^{(a^n)})}$$

$$< b^{(b^{(a^n)})} < b^{(b^{(b^n)})} = b^{n^3}$$

and...

$$a^{n^4} = a^{(a^{(a^{(a^n)})})} < b^{(a^{(a^{(a^n)})})}$$

$$< b^{(b^{(a^{(a^n)})})} < b^{(b^{(b^{(a^n)})})}$$

$$< b^{(b^{(b^{(b^n)})})} = b^{n^4}$$

...

$$a^{n^k} < b^{n^k} : a < b$$

Assume $a^n < b^n$

$$\text{Then } a^{n^2} = a^{(a^n)} < b^{(a^n)} < b^{(b^n)} = b^{n^2}$$

$$a^{n^3} = a^{(a^{(a^n)})} < b^{(a^{(a^n)})} < b^{(b^{(a^n)})} = b^{n^3}$$

...

$$a_n^k < b_n^k$$

So in general increasing an arbitrary argument while leaving the others fixed will produce a larger value.

Proof of Lemma 5

Now we will also want to show that:

$$f(2n) < f_{100}(n)$$

To do this we just observe that:

$$2 < 100 = f_{100}(1)$$

Note that the argument in the first case will only increase by 2 each time

$$2, 4, 6, 8, 10, 12, 14, 16, 18, \dots$$

Meanwhile we get:

$$100, 10000, 1000000, 100000000, 10000000000, 1000000000000, \dots$$

but we can prove that $f_n > 10n$

Firstly we have $10n \leq E_n$

since:

$$10(1) = 10 = E_1$$

but...

$$10(2) = 20 < 100 = E_2$$

Assume $10k < E_k : k > 1$

$$10(k+1) = 10k+10 < 10k+10k = 20k < 100k = 10 \cdot 10k < 10 \cdot E_k = E(k+1)$$

Therefore:

$$10k \leq E_k$$

Furthermore we previously established that:

$$10^n \leq E_n < E_{100\#n} < E_{100\#100\#n} < E_{100\#100\#100\#n} < \dots$$

so

$$E_{100} > E_{1000}$$
$$E_{1000} > E_{10000}$$

etc.

and in general $E_{100\#n} > E_{10^{(n+1)}}$

Now we just show that:

$$2^n < 10^{(n+1)}$$

for all n

$$2 < 10$$

assume $2^k < 10^{(k+1)}$

$$2^{(k+1)} = 2^{k+2} \leq 4^k < 20^k = 10 \cdot 2^k < 10 \cdot 10^{(k+1)} = 10^{(k+2)}$$

And this was proved only very modestly. So it follows that:

$$E_{(2n)} < E_{100\#n}$$

With all these lemmas established we can finally prove that $E_{\#}$
"keeps up", or stabilizes with FGH_{ω}

MAIN PROOF

Firstly a review:

$$f_0(n) = n+1$$
$$f_1(n) = 2n$$
$$f_2(n) = n \cdot 2^n$$
$$n < 2^n \quad \forall n$$

Proof:

$$1 < 2^1$$

$$2 < 2^2$$

Assume $k < 2^k$ & $k > 1$

$$2^{(k+1)} = 2 * 2^k > 2*k = k+k > k+1$$

Therefore:

$$n*2^n < 2^n * 2^n = 4^n < 10^n = E_n$$

Therefore $f_2(n) < E_n \forall n$

$$f_3(n) = f_2^n(n) < E_n \# n < E(E100\#n) \# n$$

[$n < E100\#n$ See Lemma 2]

$$= E100\#(2n) \text{ [Def. of } E\#]$$

Therefore:

$$f_3(n) < E100\#(2n) \forall n$$

and:

$$f_3(n) < E100\#100\#n = E100\#*(2)n$$

Moving on to the next function:

$$f_4(n) = f_3^n(n) < E100\#(2 * E100\#(2 * E100\#(2 * \dots (2 * E100\#(2n)) \dots)))$$

w/n 100s

$$< E100\#(3 * E100\#(2 * E100\# \dots$$

$$< E100\#(E100\#(1 + 2 * E100\# \dots$$

$$< E100\#(E100\#(3 * E100\# \dots$$

$$< E100\#(E100\#(E100\#(1 + 2 * E100\# \dots$$

$$< E100\#(E100\#(E100\#(3 * E100\# \dots$$

$$< E100\#(E100\#(E100\#(E100\#(1 + 2 * E100\# \dots$$

we can continue this for arbitrary nestings until we obtain ...

$$E_{100} \# (2n+1) \# n$$

$$\leq E_{100} \# 3n \# n$$

Now we can say:

$$E_{100} \# 3n \# n < E_{100} \# 10n \# n < E_{100} \# (En) \# n < E_{100} \# (E_{100} \# 100 \# n) \# n = E_{100} \# 100 \# (2n)$$

And now we have the necessary tools to continue:

$$f_4(n) < E_{100} \# 100 \# (2n)$$

$$f_5(n) = f_4^n(n) < E_{100} \# 100 \# (2^* E_{100} \# 100 \# (2^* \dots$$

$$< E_{100} \# 100 \# (2n+1) \# n < E_{100} \# 100 \# 3n \# n < E_{100} \# 100 \# (E_{100} \# 100 \# 100 \# n) \# n <$$

$$E_{100} \# 100 \# 100 \# (2n)$$

In general let:

$$f_k(n) < E_{100} \# 100 \# \dots \# 100 \# (2n) \text{ w/k-2 } 100s$$

$$f_{k+1}(n) < E_{100} \# 100 \# \dots \# 100 \# (2n+1) \# n$$

$$< E_{100} \# 100 \# \dots \# 100 \# (E_{100} \# 100 \# \dots \# 100 \# 100 \# n) \# n$$

$$E_{100} \# 100 \# \dots \# 100 \# 100 \# (2n) \text{ w/k-1 } 100s$$

Thusly we conclude:

$$f_k(n) < E_{100} \# 100 \# \dots \# 100 \# (2n) \text{ w/k-2 } 100s$$

QED

Furthermore:

$$f_k(n) < E_{100} \# 100 \# \dots \# 100 \# 100 \# n \text{ w/k-1 } 100s \forall n, k$$

f_ω(n) eventually dominates every member of E#H

Thus FGH and E# are stabilized. Also:

There exists no k such that $f_k(n)$ grows faster than every function in E#.

Now we show that $f_\omega(n)$ is the smallest function such that it eventually dominates every function of E#H.

Firstly we need to demonstrate that every member of FGH _{ω} is a S.IN/E.AB function. The reason we can't simply apply our previous result is because that only applied strictly to stacking primitive recursions onto a S.IN/E.AB function. However as we'll see, FGH actually get's an even bigger boost in this department, so that the results that applied to applications of primitive recursion also apply here.

Firstly we have that:

$$f_0(n) = n+1$$

$$f_0(n+1) = f_0(f_0(n)) = f_0(n)+1 > f_0(n)$$

Therefore $f_0(n)$ is S.IN

also

$$f_0(n) = n+1 > n : \forall n$$

so $f_0(n)$ is E.AB

Let $f_k(n)$ be a S.IN/E.AB function.

We have already proven that:

$$f_k^n(c)$$

is also a S.IN/E.AB function of "n".

Since $f_k(n)$ is S.IN, it means that if the innermost argument increases, so does the output so that:

$$f_k(a) < f_k(b) : a < b$$

$$f_k(f_k(a)) < f_k(f_k(b)) : a < b$$

$$f_k(f_k(f_k(a))) < f_k(f_k(f_k(b))) : a < b$$

etc.

and in general:

$$f_k^n(a) < f_k^n(b) : a < b$$

Therefore:

$$f_k^a(a) < f_k^a(b) < f_k^b(b)$$

Therefore: $f_k^n(n)$ is a S.IN function.

Interestingly in the case of FGH, the functions are not strictly E.AB. This is because, for $k > 0$:

$$f_k(0) = f_{(k-1)}^0(0) = 0$$

So "0" is a fixed-point for every function in the hierarchy except $f_0(n)$. We can show however that $f_k(n)$ is E.AB for $n > 0$.

Observe:

$$f_k^1(1) = f_k(1)$$

If it holds that $f_k(1) > 1$, then it must also hold for $f_{k+1}(1)$. This is true since $f_0(1) > 1$. So it must be true for all functions in FGH_ω . In fact it can be shown that:

$$f_k(1) = f_{k-1}(1) = f_{k-2}(1) = \dots = f_2(1) = f_1(1) = f_0(1) = 2$$

Therefore for all k , $f_k(1) = 2 > 1$.

For the higher values we have:

$$f_{k+1}(n) = f_k^n(n) : n > 1$$

since all f_k are S.IN:

$$f_k^n(n) \geq n+n > n : n > 1$$

So all functions f_k are E.AB($n > 0$).

Next thing we need to show is that for each f_k ($k > 2$) there is a member of E#H which it eventually dominates. We begin with f_3 .

We want to prove:

$$E_n < f_3(n) : (n > 1)$$

Firstly we observe:

$$E_1 = 10$$

$$E_2 = 100$$

$$E_3 = 1000$$

etc.

while:

$$f_3(0) = 0$$

$$f_3(1) = 2$$

$$f_3(2) = f_2(f_2(2)) = f_2(2 * 2^2) = f_2(8) = 8 * 2^8 = 2^{11} = 2048$$

so we have:

$$100 < 2048$$

$$E_2 < f_3(2)$$

Now we will show that every succeeded case f_3 wins.

Assume:

$$E_k < f_3(k)$$

$$f_3(k+1) > f_2(f_3(k)) = f_3(k) * 2^{(f_3(k))}$$

$$2^{(f_3(k))} > 10$$

provided $f_3(k) > 3$

k must be at least 2, and $f_3(2) = 2048 \gg 3$, but $f_3(n)$ is also S.IN therefore:

$$f_3(k) > 2048 \gg 3 : k > 2$$

So we have:

$$f_3(k) * 2^{(f_3(k))} > f_3(k) * 10 > E_k * 10 = E(k+1)$$

So therefore:

$$E_n < f_3(n) : n > 1$$

We can now go further:

$$f_4(n) = f_3^{(n)}(n) > E \dots E E_n \text{ w/n } E_s = E_n \# n : n > 1$$

We know that:

$$E_n \# n > E_{100} \# n : n > 100$$

yet we have:

$$f_4(100) > E_{100} \# 100$$

So in this case we have:

$$f_4(n) > E_{100} \# n : n > 99$$

We now can show inductively that:

$$f_k(n) > E_{100} \# 100 \# \dots \# 100 \# n \text{ w/k-3 } 100\text{s} : (n > 99)$$

Assume:

$$f_k(n) > E_{100} \#^{(k-3)} n : n > 99$$

$$f_{k+1}(n) > E_{100} \#^{(k-3)} n \# n \geq E_{100} \#^{(k-3)} 100 \# n = E_{100} \#^{(k-2)} n : n > 99$$

$$\text{--> } f_{k+1}(n) > E_{100} \#^{(k-2)} n : n > 99$$

Thus we have proven:

$$\mathbf{f_k(n) > E_{100} \#^{(k-3)} n : n > 99}$$

We now use this to show that $f_\omega(n)$ eventually dominates every member of $E\#H$.

Firstly we have:

$$f_\omega(n) = f_n(n) > E_{100} \#^{(n-3)} n : n > 99$$

Now imagine an arbitrary member of $E\#H$:

$$E100\#^*(k)n$$

I'll will now prove that:

$$\text{if } n > 99 \ \& \ n \geq k+3$$

$$\text{then } f_{\omega}(n) > E100\#^*(k)n : \forall k$$

Firstly if $n > 99$ then our general relation holds that:

$$f_{\omega}(n) = f_n(n) > E100\#^*(n-3)n$$

if n is also $\geq k+3$ then:

$$E100\#^*(n-3)n \geq E100\#^*(k+3-3)n = E100\#^*(k)n$$

This holds due to lemma 2.

So if $n \geq k+3$ we know that $E100\#^*(n-3)n$ contains *at least as many 100s as*

$E100\#^*(k)n$. If it's the same number then $f_{\omega}(n)$ wins automatically, but if it's more it still wins because then:

$$f_{\omega}(n) > E100\#100\#\dots\#100\#n \text{ w/} n-3 \text{ 100s}$$

$$\text{and } E100\#100\#\dots\#100\#n \text{ w/} k \text{ 100s}$$

is less provided $k < n-3$, again via lemma 2. So anyway you slice it, $f_{\omega}(n)$ dominates every function in $E\#H$ at $n > 99$ at the earliest, or $n \geq k+3$, whichever is largest.

Thus $f_{\omega}(n)$ is the first function to eventually dominate every function in $E\#H$.

So by way of phrase we can say:

$E\#$ is equipotent with FGH_{ω} , or $E\#$ is of order- ω

QED