Re: My claim on Omega's defn

Source: http://sci.tech−archive.net/Archive/sci.logic/2005−01/2833.html

---

From: |−|erc (*H_at_r.c*)
Date: 01/31/05

```
Date: Mon, 31 Jan 2005 13:54:33 +1000
```

> >
> >>|−|erc wrote:
> >>
> >>>Omega = sum(p halts) 1 / (2 ^ size(p))
> >>>
> >>> [big snip]
> >>>
> >>>Even with infinitesimally small total probability of halting, Omega will not converge and will equal oo.
> >>
> >>
> >>You seem to be missing the point that the domain of the universal
> >>self−delimiting TM U is taken to be prefix−free −−− ie, the encoding of
> >>halting TMs is such that if x is the encoding of some halting TM, then
> >>no proper prefix of x is a encoding. Basically, any branch of the
> >>infinite−binary tree will contain at most one such encoding, and so to
> >>simply say that there are 2^n encoding with n bits is just being
> >>ignorant. If follows via Kraft's inequality that Chaitin's Omega will
> >>be bounded.
> >
> >
> > So all Omega means is there exists 1 program of that size that halts.
>
> No, what it does mean is that the set of encodings of programs is sparse
> (quite possibly meagre) is the space of all finite binary strings. It
> is a perhaps unnatural condition, but its what was done to ensure that
> the "Omega series" converges.
>
> > What a load of crap, I could design an arbirtrary UTM where Omega = 1.
> >
> > It skips 100% of programs without any reason.
>
> Then this number really doesn't mean anything. We could all come up
> with our own numerical constants, but unless there's a reason for them,
> no−one will care. It's the natural interpretation that Chaitin was able
> to get that makes it extremely worthwhile.
>

the number of programs per size must be contant for Omega to converge.
its useless.

if 1 program halts per |p|, omega = 1/2 + 1/4 + 1/8 + 1/16 ... = 1
if 2 programs halt per |p|, omega = 1/2 + 1/2 + 1/4 + 1/4 + ... = 2

Its hasn't been *modified* to fix this, this error is so huge becuase they botched it completely.

How can there only be 10 or so program that halt (for Omega = 10)
for programs of size 1,000,000,000? That's a typical 125MB file.

He is MISSING 2^1,000,000,000−10 algorithms, just for that size!

Herc