

Re: Alan Turing's Halting Problem is incorrectly formed

Source: <http://sci.tech-archive.net/Archive/sci.logic/2004-06/0749.html>

From: The Ghost In The Machine (*ewill_at_aurigae.athghost7038suus.net*)

Date: 06/10/04

Date: Thu, 10 Jun 2004 08:00:09 GMT

In sci.logic, |—|erc

<gotcha@beauty.com>

wrote

on Wed, 09 Jun 2004 23:40:00 GMT

<kdNxc.2919\$uB3.1201@news-server.bigpond.net.au>:

>> > *like so. Now we have a bona fide halt program that categorises the
>> > function into what class it belongs to, by giving an upper bound to the
>> > number of cycles for programs in that class.*

>> >

>> > *useful no?*

>>

>> *Not really.*

>

> *an effective halting program is not useful?*

It's not all that effective. At best, it computes whether a function can halt in N cycles in the same time one can run the function using N cycles. At worst, it requires an emulator.

Unless I'm missing something (which is admittedly possible).

>

>

>>

>> >

>> >

>> >>

>> >> *Also, TM's don't crash. :-)*

>> >

>> > *do so.*

>>

>> *A TM may perform unexpectedly because of a specification error*

>> *or unexpected input, but not crash. A TM is too simple a*

>> *beast to crash, really.*

>

>
> *it can compute anything and its simple?*

It cannot compute everything. Of course, one might make the case that that which cannot be computed may not be that interesting.

>
>
>> >
>> >>
>> >> *No, it does not disprove such a program. A fixed-cycle halter*
>> >> *is easy to write -- just run the simulated TM for that number*
>> >> *of cycles.*
>> >
>> > *not the program I was talking about, can you solve an*
>> > *exponential program in polynomial time. i.e. is there*
>> > *a haltexp that halts in polynomial time (a fixed*
>> > *cycle halter using a simple type of emulator wont work).*
>>
>> *Why not?*
>
>
> *because you can't work out any maths proofs for yourself and this one*
> *isn't in the book.*

Ah, so you're suggesting I'm part of the Godel/Turing/Cantorian Conspiracy To Suppress The Proof That All Sets Are Closed And All Reals Are Countable?

Woo.

(Hmm.... GTCCTSTPTASACAARAC. Perhaps a better name would be "The Unpronounceable Acronym Conspiracy"...)

Of course, I can't seem to work out *your* math proofs. But mine seem to work fine, although I can't claim originality. But then, this is well-trodden territory. Still, let's start with the simple stuff.

Define "equals".

I'll make it even simpler: assume two real numbers r and s have Cauchy sequences r_i and s_i . Then r and s are equal iff the sequence $t_i = r_i - s_i$ has limit 0.

If you really want I'll use Dedekind cuts instead... :-)
The two definitions are equivalent anyway.

I've already stipulated that the naive $Q \rightarrow R$ mapping will result in a gray line -- although "gray" in this context would be almost white, as the Lebesgue measure of $Q \cap [0,1]$ is 0,

whereas the Lebesgue measure of $\mathbb{R} \cap [0,1]$ is 1.
If one colors the former black and the latter white, the line
is white for all intents and purposes, even though \mathbb{Q} is dense.

But I can't say it is white, just almost white, as there
are bits of black (actually, infinite pinpricks) — $1/2$,
 $2/3$, $1/3$, and such, for instance. But there's also $\pi/4$,
which is not rational by any stretch of the imagination
(even though it's approximatable by at least one strictly
monotonic rational Cauchy sequence) and is therefore white.
If one picks a number in $[0,1]$, the probability that it
is in a set S which is a subset of $[0,1]$ turns out to be
merely the Lebesgue measure of S . Since $(\mathbb{R} - \mathbb{Q}) \cap [0,1]$
has measure 1 it is virtually certain the number
I pick is irrational — in fact, it's virtually certain
it's transcendental as well. (I'll bore you with details
of a possible mapping of \mathbb{N} to \mathbb{A} if you want; suffice it
to say for now that said mapping can be done.)

Or one can look at it this way. If numbers r and s are equal,
then their digit expansions match for any finite n . If I am
given two numbers t and u , then I can prove them unequal if I
can find a d such that $t(d)$ and $u(d)$ differ, and, as a Pedant
Point, $t(d+1)$ and $u(d+1)$ are not 0 and 9, or 9 and 0. (This
sidesteps the old $0.999... = 1.000$ problem, of course.) This
despite the fact that $\text{floor}(t * 10^n) / 10^n$ is rational for
any nonnegative integer n and any t in the interval $[0,1]$.

Therefore the diagonal construction works, if done carefully enough
to avoid the infinite 9-0 problem — even though every prefix
of the diagonal number be in the list — not unreasonable since
the list is supposed to contain all reals.

Or one can do it this way. Assume for the nonce that the sequence
 $S = \{0.3, 0.33, 0.333, \dots\}$ is also a set. This set has a lower
bound (0.3). Does it have an upper bound? The sequence is
strictly monotonically increasing; therefore, for every s_n ,
 $s_{n+1} > s_n$ and therefore there is no upper bound, although
there is a supremum, namely, $1/3$. But $1/3$ is not in the set,
since $s_n = (10^n - 1) / (3 * 10^n) \neq 1/3$ for any integer n .
(Infinity is not an integer. It's not a real number either.)

I'm not sure this argument is even all that recognizable
as a horse anymore. Between my cudgel, Barbara's shillelagh,
and other's clubs it's been liquefied.

Of course, from an effectively computable standpoint,
all numbers are finite and rational. The approximation
commonly used, for instance, has to be a multiple of a
power of 2; furthermore, the multiplier cannot exceed
 2^{52} , and the multiplicand must be between the range

sci.logic: Re: Alan Turing's Halting Problem is incorrectly formed

2^{-1076} and 2^{+972} (I might be off by a bit or two).

Also, if one adds or subtracts two numbers, the result may be rounded to be in this form under certain conditions, and some of the bit representations are sacrificed and treated as though they are either 0, +∞, −∞, or some undefined value with well-defined properties (e.g. $0 * \text{NaN} = \text{NaN}$, $\text{NaN} + 1 = \text{NaN}$).

Infinite decimal expansions are impossible to write out in full, although an interesting computational method is applicable to π , if one wants an arbitrary hex digit. See Equation 29 in

<http://mathworld.wolfram.com/PiFormulas.html>

for example.

>
> *Herc*
>

--
#191, ewill3@earthlink.net
It's still legal to go .sigless.