

sci.logic: Re: Can you find anything wrong with this solution to the Halting Problem?

Re: Can you find anything wrong with this solution to the Halting Problem?

Source: <http://sci.tech-archive.net/Archive/sci.logic/2004-07/1970.html>

From: George Greene (greeneg_at_greeneg-cs.cs.unc.edu)

Date: 07/18/04

Date: 18 Jul 2004 17:15:27 -0400

: > This is because the halting problem is only interesting
: > if the halting function has access to the same features as the programs
: > of which it has to decide whether they halt.

"Peter Olcott" <olcott@worldnet.att.net> writes:

: That is simply not true.

It IS SO TOO true, DUMBASS.

: The conclusion of the original problem

HOW THE *FUCK* would *YOU* know *THAT*??
You DON'T KONW SHIT about the original problem!
You have NEVER SEEN a treatment of "The Original Problem"
that you could get within SNIFFING distance of UNDERSTANDING!

: was that it is completely impossible to always be able to correctly
: determine

For WHAT to always correctly determine?!?!?
Suppose we had a supernatural oracle that undersood every fact about
all Turing Machines: could IT always correctly determine

: whether or not any other Turing Machine would halt,

?? WELL? COULD IT?

: regardless of any possible means to do so.

REGARDLESS OF ANY POSSIBLE MEANS??
SEZ WHO??? CAN YOU *QUOTE* "The conclusion of the original problem"
saying that??

: The determination of
: this was only framed within the context of a Turing Machine because

Re: Can you find anything wrong with this solution to the Halting Problem?

sci.logic: Re: Can you find anything wrong with this solution to the Halting Problem?

: it was considered equivalent to every other possible computer.

No, it wasn't.

Church's Thesis came AFTERWARDS.

And it's obviously NOT "equivalent to every other possible computer": it's obviously STRONGER than ALL other possible "computers" that aren't equivalent to it. Indeed, when the original problem was phrased, the whole notion of what a "computer" was was still young & fluid.

Turing published his result about the Halting problem in 1936. Church had just previously published an analogous result about his lambda calculus. Yes, it is true that you can't use one algorithm definable via the lambda-calculus to determine whether every TM halts, nor can you use a TM solve all the relevant lambda-expressions. But all this variation among equivalent formalisms does NOT excuse YOUR introduction of formalisms that are NOT equivalent and are for that matter not even formal, nor even competently described. In both Church's and Turing's original proof, it WAS necessary for the thing DOING the reasoning to be the SAME KIND of thing as the things being reasoned about.

But you didn't know that.

Which would be excusable if it weren't for the fact that even after you get informed of it, you STILL claim to know better than everybody else, when YOU'RE the one who hasn't read anything and actually NEEDS some URLs.

Start HERE:

http://en.wikipedia.org/wiki/Church-Turing_thesis

Church's Thesis is NOT some prediction that Peter Olcott will eventually fail in his bid to design some programming paradigm stronger than a TM. It is rather a proposed DEFINITION of "algorithm" and "*effectively* computable" in terms of any of a bunch of EQUIVALENT formalisms.

Not that you are smart enough to understand that, but hey, somebody else might be reading.

--

--- The history of our nation has demonstrated that separate is seldom, if ever, equal.
--- (Feb.3,2004) Supreme Judicial Court of Massachusetts (4-3), adv.Sen.#2175