Re: Rado's Sigma and the Halting Problem for Programs

Source: http://coding.derkeiler.com/Archive/General/comp.theory/2004−08/2169.html

---

From: peter_douglass (*baisly_at_gis.net*)
Date: 08/13/04

```
Date: Thu, 12 Aug 2004 23:22:59 GMT
```

"r.e.s." <r.s@ZZmindspring.com> wrote in message
news:%HASc.18697$cK.14152@newsread2.news.pas.earthlink.net...

> "peter_douglass" <baisly@gis.net> wrote ...
> −snip−
> > I believe the Halting Theorem may be proven without recourse
> > to reductio ad absurdum. Here is my stab at it.
> −snip−

> > Assume that from a Turing Machine M we can construct
*> ^^^^^^*
> > a Turing Machine M' such that the following holds:
> > Eval(M',x) == begin
> > if Eval(M, <G'(M'),x>) == "no"
> > then return "yes"
> > else loop_forever() ;
> > end
> −snip−

> Can you prove, without reductio ad absurdum, that such
> an M' can always be constructed?

Hmm. I'm not sure. I think you have pointed out a bug
in the proof. Better would be

```
Eval(M',x) == begin
                 if Eval(M,<x,x>) == "no"
                   then return "yes"
                   else loop_forever() ;
              end
```

This change does not affect the form of the proof
because when evaluating Eval(M',G'(M'))
the body is still instantiated as

begin
 if Eval(M,<G'(M'),G'(M')> == "no"
 etc.

> *The self−referring definition of M' would seem to make*
> *that less tractable than the usual type of definition of*
> *a "derived TM" for reductio ad absurdum, e.g. (using the*
> *usual shorthand suppressing the encoding of TMs & inputs):*
> *M'(P) converges iff M(P,P) converges to "no".*
> *Defining M' in terms of M alone is tractable, but your*
> *version may not be. (?)*

Good observation. Hope there is nothing so wrong with
the "corrected" version.

−−PeterD