

Re: [PO] Re: Can a regular Turing Machine provide Protected Memory?

Source: <http://coding.derkeiler.com/Archive/General/comp.theory/2004-08/3678.html>

From: Peter Olcott (olcott_at_worldnet.att.net)

Date: 08/30/04

Date: Mon, 30 Aug 2004 12:13:01 GMT

"Owen Jacobson" <angstrom@lionsanctuary.net> wrote in message
news:pan.2004.08.30.07.41.46.286184@lionsanctuary.net...

> *On Sun, 29 Aug 2004 13:34:44 +0000, Peter Olcott wrote:*

>

> > *"Simon G Best" <s.g.best@btopenworld.com> wrote:*

> >

> > > *Peter Olcott wrote:*

> > >

> > > > *"Simon G Best" <s.g.best@btopenworld.com> wrote:*

> > > >

> > > > *I still don't see how this is other than one condition that causes*

> > > > *failure, such that the absence of this condition still permits*

> > > > *success.*

> > >

> > > *It doesn't /cause/ failure. Halts behaves /exactly/ the same under*

> > > */both/ conditions (one condition being where it's being used within*

> > > *another TM, and the other condition being where it's used on its own).*

> >

> > *I have already addressed this issue to you again in the preceding post. I*

> > *have already shown that this is not true. Whenever Halts detects that it*

> > *is run inside of another TM, it simply halts. If the UTM that is running*

> > *Halt reports that there is a state transition out of the final state of*

> > *Halt, then it knows that another TM has invoked it.*

>

> *The ability of the machine to examine the properties of that machine is*

> *not universal to all Turing machines.*

>

> *Therefore, if your "augmented" UTM truly is Turing-equivalent, we must be*

> *able to simulate your UTM on another UTM that does not provide the*

> *capability to examine states and must necessarily run identically when*

> *this happens.*

That would not be a problem. These states are merely stored as data on the UTM's tape. The UTM is then asked to look at its own tape, and do some processing on the data that is there. It is then requested

comp.theory: Re: [PO] Re: Can a regular Turing Machine provide Protected Memory?

to place this result on a specific location of its own tape, so that the other process can access this result. The simplest way to do this might be to allocate some of the infinite tape as a stack. If we do this then the ordinary calling conventions of higher level languages can be directly implemented.

- > *From that statement, we can construct a Turing machine that*
- > *contradicts whatever Halts reports about that machine (already proven and*
- > *belaboured elsewhere).*

This has not yet been shown.

- > *The only other conclusion we can derive is that your "augmented" "UTM"*
- > **cannot* be correctly run on another UTM. (There is no third*
- > *option -- either it can be run correctly on a UTM, or it can't.)*
- > *Therefore, if we were to accept that your augmented model could*
- > *correctly analyze Turing machines, we would of necessity also have to*
- > *accept that your model of computing is *not*, at least for this problem,*
- > *Turing-equivalent: it's more powerful. We already knew that*
- > *non-Turing-machines might hypothetically be able to determine if Turing*
- > *machines halt.*
- >
- > *The last option is that your augmented UTM running Halts can't be*
- > *correctly run on another UTM for the simple reason that it can't be*
- > *correctly run anywhere; others are debating this point well enough.*
- > *Either way, you have said nothing about whether Turing machines can*
- > *correctly determine if Turing machines (with their inputs) halt, which is*
- > *the point of the Halting problem.*
- >
- > --
- > *Some say the Wired doesn't have political borders like the real world,*
- > *but there are far too many nonsense-spouting anarchists or idiots who*
- > *think that pranks are a revolution.*
- >