

# Re: Lambda Calculus and Turing Equivalence

*Source:* <http://sci.tech-archive.net/Archive/sci.math/2004-12/3988.html>

---

*gds\_at\_best.cut.here.com*

*Date:* 12/02/04

Date: Thu, 02 Dec 2004 06:16:01 GMT

At 1 Dec 2004 12:04:41 -0800, examachine@gmail.com (Eray Ozkural exa) wrote:

>torbenm@diku.dk (Torben Ægidius Mogensen) wrote in message  
>news:<7zsm6qvyqk.fsf@pc-032.diku.dk>...  
>> A TM does not have an *\_infinite\_* tape, it has an *\_unbounded tape\_*.  
>> This means that the tape is guaranteed to be as large as you need it  
>> or, equivalently, that the tape will grow as you need it but always  
>> stay finite.  
>>  
>> It isn't hard to write a TM simulator in pure lambda calculus. The  
>> converse is harder, but possible.  
>  
>Precisely my point.  
>  
>I am saying to those who say that PCs are not TMs, like Chapman and  
>Harris, that they do not know the difference between unbounded  
>(potentially infinite) and actually infinite. (Although they claim  
>that they do) The latter is a set theoretical notion that is not  
>required to depict TMs, I doubt this distinction was so clear back  
>then.

Part of the problem is that there is a fair amount of literature that describes TMs as having infinite memory, e.g. the second paragraph of [http://en.wikipedia.org/wiki/Turing\\_machine](http://en.wikipedia.org/wiki/Turing_machine). (Arguably, this is an informal description.)

I haven't read all of the articles in this and related threads, but so far none that I have seen have commented on the difficulty TMs have in modeling "ongoing computation," such as what is found in operating systems. (The Wikipedia article mentioned above has a short discussion of this.)

—gregbo  
gds at best dot com