

Re: Fixpoint for LZH

Source: <http://sci.tech-archive.net/Archive/sci.math/2004-07/4382.html>

From: Ioannis (*morpheus_at_olympus.mons*)

Date: 07/17/04

Date: Sat, 17 Jul 2004 10:33:50 +0300

Dik T. Winter wrote:

[snip]

> > *I stand corrected. I was rather using "strict contraction" in the sense*
> > *of the previous article, but your and David's points still stand, even*
> > *under this definition: $\text{size}(f(\text{file})) > \text{size}(\text{file}) \Rightarrow f$ does not have*
> > *any fixed points.*
>
> *But you have not actually shown that $\text{size}(f(\text{file}))$ never equals $\text{size}(\text{file})$.*

Even if f is such that $\text{size}(f(\text{file})) = \text{size}(\text{file})$, this says nothing about the existence of a fixed point of f . Consider the file (vector) (x_1, x_2, \dots, x_n) , $x_i \in [0, 255]$, $x_i < 255$ and let f act on the file as $f(\text{file}) = f(x_1, x_2, \dots, x_n) = (x_1', x_2', \dots, x_n')$, with $x_i' = x_i + 1$. Then $\text{size}(f(\text{file})) = \text{size}(\text{file})$, but $f(\text{file}) \neq \text{file}$, so file is NOT a fixed point of f . On the other hand, if $f = I$, all files are fixed points of f .

> *With all kinds of schemes that may change filesize, and in some cases leave*
> *it unchanged, it is possible that a fixed point does exist.*

It is possible that several schemes may have a fixed point, but it appears to me as though a filesize change doesn't imply anything about the existence of said point.

> *I know of*
> *a better example back from the 80s. If you fed some file to the C compiler,*
> *captured the error messages, fed those back in again, and repeating, you*
> *might (with a particular compiler) indeed get a fixed point. Examples have*
> *been shown at that time.*

I am aware of one such example, too. It sits on my bookcase somewhere in Athens, but it concerns the Macintosh Pascal *interpreter* that was out in the 80's. The book provides for a Pascal program, which, when run, outputs itself. I am not sure this qualifies as a "fixed point" in the same sense as with a compiler, but it still made an impression :-)

> *Off-hand I see no reason why LZH (for some definition of the many variants)*
> *does not have a fixed point.*
>

> *Anyhow, as Phil stated there is a file that gunzips to itself. The
> message-id: <40c3f78f@news.orcon.net.nz>. If that is possible, it is
> very plausible that there also will be files that gzip to itself.*

Agreed. In any case, yesterday I ran a small series of tests with zip⁽ⁿ⁾ on some random files on my laptop. Starting with the file "0123456789", I got a strictly increasing sequence of sizes, as follows: 10, 120, 180, 245, 314, 387, 468, 556,...., while starting with several regular files, I got a size sequence which was strictly decreasing in the beginning, reached a minimum and then became strictly increasing again.

Opening the zipped file with Notepad, it appears that zip inserts all kinds of stuff in the file, such as the file name, etc. Perhaps there is an option with zip, which prevents it from adding extraneous junk in there, so as for it to produce non-increasing sequences. I don't know, I haven't used PC's for that long.

In that sense, zip appears to be fundamentally different from Stuffit for example, in that Stuffit for Macs if memory serves right, never increased file size. It's been a while since I was using a Mac, so my memory may be failing me on this.

--
I. N. Galidakis
<http://users.forthnet.gr/ath/jgal/>

Eventually, everything is understandable