

How Dr. Seuss would prove the halting problem undecidable

« [Semantic Wave 2008: Industry Roadmap to Web 3.0 Social Web Technologies, UMBC, Spring 2008, MW 7:10pm](#) »

How Dr. Seuss would prove the halting problem undecidable

Tim Finin, 11:12am 19 January 2008

I just discovered (via del.icio.us/polular) an extraordinary proof of the [halting problem](#) by linguist [Geoffrey Pullum](#), now at the University of Edinburgh. What's unusual about it is that it's written as a poem in the style of Dr. Seuss.

Geoffrey K. Pullum, Scooping the loop snooper: An elementary proof of the undecidability of the halting problem. *Mathematics Magazine* 73.4 (October 2000), 319-320.

It's a marvelous proof, sure to liven up any undergraduate theory of computation class. But I noticed errors in the proof — not logical errors, but a transcriptional ones in the form of a mangled word, perhaps introduced by an OCR system. The third line of the fifth stanza reads *"that would take and program and call P (of course!)"* which has problems in syntax, semantics, rhythm and meter. I'd guess it should be *"that would take any program and call P (of course!)"*. Similarly, *"the"* in the third line in the third stanza should probably be *"they"*. Most of the online version I found had these errors, but I eventually found what I take to be a correct version on the [QED](#) blog. I've not been able to get to the original version in *Mathematical Magazine* to verify the corrected version which I include below.

Scooping the Loop Snooper

an elementary proof of the undecidability of the halting problem

Geoffrey K. Pullum, University of Edinburgh

No program can say what another will do.
 Now, I won't just assert that, I'll prove it to you:
 I will prove that although you might work til you drop,
 you can't predict whether a program will stop.

Imagine we have a procedure called P
 that will snoop in the source code of programs to see
 there aren't infinite loops that go round and around;
 and P prints the word "Fine!" if no looping is found.

You feed in your code, and the input it needs,
 and then P takes them both and it studies and reads
 and computes whether things will all end as they should
 (as opposed to going loopy the way that they could).

Well, the truth is that P cannot possibly be,

Ebiquity Blog

search ebiquity blog posts

GO

[Home](#) | [Archive](#) | [Login](#) | [Feed](#)

Follow [@ebiquity](#) on Twitter

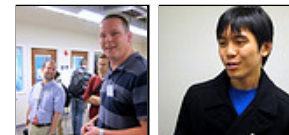
Ebiquity Recent Posts

- UMBC team 2nd in DARPA Shredder Challenge
- Estimating the Impact of Web Technology Conferences
- On Facebook it is 4.74 degrees of separation, not six
- Ebiquity Semantic Web Meetup, 6-8pm Tue 11/15
- Honda Asimo robot gains more autonomy

Ebiquity rdfs:seeAlso

- Google and Microsoft Talk Artificial Intelligence
- Microsoft drops Dryad; puts its big-data bets on Hadoop
- Android sees a 472% increase in malware since July
- TOP500 Supercomputing Sites
- Computer Programming for Children, Minus Cryptic Syntax
- Army of 'socialbots' steal gigabytes of Facebook user data • [The Register](#)
- John McCarthy, 1927–2011
- Introduction to Linguistics for NLP

Ebiquity on Flickr



Ebiquity community

- AISL
- Assured Information Sharing

because if you wrote it and gave it to me,
I could use it to set up a logical bind
that would shatter your reason and scramble your mind.

Here's the trick I would use – and it's simple to do.
I'd define a procedure – we'll name the thing Q -
that would take any program and call P (of course!)
to tell if it looped, by reading the source;

And if so, Q would simply print "Loop!" and then stop;
but if no, Q would go right back to the top,
and start off again, looping endlessly back,
til the universe dies and is frozen and black.

And this program called Q wouldn't stay on the shelf;
I would run it, and (fiendishly) feed it *itself*.
What behaviour results when I do this with Q?
When it reads its own source, just what will it do?

If P warns of loops, Q will print "Loop!" and quit;
yet P is supposed to speak truly of it.
So if Q's going to quit, then P should say, "Fine!" -
which will make Q go back to its very first line!

No matter what P would have done, Q will scoop it:
Q uses P's output to make P look stupid.
If P gets things right then it lies in its tooth;
and if it speaks falsely, it's telling the truth!

I've created a paradox, neat as can be -
and simply by using your putative P.
When you assumed P you stepped into a snare;
Your assumptions have led you right into my lair.

So, how to escape from this logical mess?
I don't have to tell you; I'm sure you can guess.
By *reductio*, there cannot possibly be
a procedure that acts like the mythical P.

You can never discover mechanical means
for predicting the acts of computing machines.
It's something that cannot be done. So we users
must find our own bugs; our computers are losers!

Related posts:

- [Parsing Perl considered undecidable](#)
- [online book: Introduction to social network methods](#)
- [Researchers prove Rubics Cube solvable in 20 moves or less](#)
- [OWL Pizzas](#)
- [Want Microsoft Source code?](#)

Categories: [GENERAL](#), [Humor](#) **Comments:** [14](#)

14 Responses to "How Dr. Seuss would prove the halting problem undecidable"

- [Koen Vervloesem](#) Says:
January 19th, 2008 at 11:25 am

If you have access to JSTOR, you can find the original Mathematics Magazine article online on <http://www.jstor.org/view/0025570x/di021216/02p00301/0>

That's the version I present on my blog QED.

- Harry Chen thinks aloud
- Journal of Web Semantics
- Search, Spam, Social Media
- UMBC CSEE
- UMBC GAIM

Ebiquity tags

Games advertising AI Policy RDF social
networking obama cyberdefense
cyberwar data Journal of Web
Semantics RDFa social network
darpa LOD Yahoo iswc IBM
Semantic Web voting
cloud computing JWS
spam Google Twitter
Microsoft Python
linked data Social
media
Facebook

■ *Tim Finin* Says:

January 19th, 2008 at 1:07 pm

Thanks. UMBC appears to have access, but I've not been able to figure out how to access it from off campus. I had hoped that using a VPN to UMBC would give me access, but that didn't work.

■ *Sam* Says:

January 24th, 2008 at 11:40 pm

This made my day! Many thanks.

■ *dr seuss the things you can do printable version* Says:

June 2nd, 2008 at 9:09 am

[...] I won't just assert that, I'll prove it to you: I will prove that although you might work til ...<http://ebiquity.umbc.edu/blogger/2008/01/19/how-dr-suess-would-prove-the-halting-problem-undecidable...> Dr. Seuss – Wikiquote If you can see things out of whack, then you can see how things ... Los Angeles [...]

■ *yamex5* Says:

May 5th, 2010 at 2:01 pm

No way to find out? Should we really say never?
How about apps that are craftily clever?
The very best minds might write such a tool
That tries to avoid becoming a fool

But no matter how wise and sneaky they'll be
When they construct their impervious P
There's always a hack with nothing to do
Who goes on to devise an updated Q!

■ *thefre* Says:

May 5th, 2010 at 2:39 pm

This is a cool poem, which present a well known result.

Unfortunately, this result is far too often misunderstood on what are its consequences.

Just because there exist no program that can take ANY program and check some properties on them, does not mean that there exist no program that can take SOME programs (with well-defined characteristics) and check some properties on them.

Which is good news, because now that I've said that, we can forget about this undecidability theorem which indeed means little in the real world, and concentrate on automatic proofs on USEFUL programs, which IMHO is a field with far too few studies, because every student is deterred to enter it when he is first taught about how this is not possible in the general case.

The general case does not matter AT ALL, because computers just don't run randomly generated programs.

So now everybody please go back to work and write some cool programs that can resonate on useful ones.

Thanks for listening :p

■ *How Dr. Seuss would prove the halting problem undecidable* Says:

May 5th, 2010 at 2:44 pm

[...] full post on Hacker News If you enjoyed this article, please consider sharing it! Tagged with: halting [...]

■ *Owen Corso* Says:

May 5th, 2010 at 5:52 pm

PUMPING LEMMA THEOREM

long live Dr. William Fleischman

- [Shawn J. Goff](#) Says:
May 5th, 2010 at 8:34 pm

To get to UMBC's journal database from off-campus: <http://aok.lib.umbc.edu/services/remoteaccess.php>

- [Sebastian](#) Says:
May 6th, 2010 at 12:50 am

Here is a link to a scanned version from the magazine: <http://www.cl.cam.ac.uk/teaching/0910/CompTheory/scooping.pdf>

- [EastZoneSoupCube - links for 2010-05-06](#) Says:
May 6th, 2010 at 10:06 am

[...] How Dr. Seuss would prove the halting problem undecidable (tags: computer humor halting problem) [...]

- [Halting Problem Poem « Eliminando La Bestia](#) Says:
May 6th, 2010 at 11:19 am

[...] Tiki « [...]

- [sdrgrderfg](#) Says:
May 17th, 2010 at 8:34 am

The version on GKP's website is slightly different:
<http://www.ling.ed.ac.uk/~gpullum/loopsnoop.html>

- [Ed The Dev .com » Blog Archive » How Dr. Seuss would teach the Halting problem](#) Says:
June 1st, 2010 at 7:05 pm

[...] In a feat of awesome nerd-ness, Geoffrey K. Pullum has written up the Computer Programming Halting Problem in Dr. Seuss style. [...]