

# On computable numbers, with an application to the Entscheidungsproblem

John MacCormick

July, 2010

# In his seminal 1937 paper, “On computable numbers...”, Alan Turing:

- a) Introduces what is now known as a Turing machine
- b) Provides a concrete construction of a universal Turing machine
- c) Proves the undecidability of the Halting Problem
- d) Provides a convincing argument that any computation performed by a human can also be performed by mechanical means
- e) Solves a famous problem posed by David Hilbert
- f) All of the above

# In his seminal 1937 paper, “On computable numbers...”, Alan Turing:

- a) Introduces what is now known as a Turing machine
- b) Provides a concrete construction of a universal Turing machine
- c) Proves the undecidability of the Halting Problem
- d) Provides a convincing argument that any computation performed by a human can also be performed by mechanical means
- e) Solves a famous problem posed by David Hilbert
- f) All of the above

← the text-book answer

# In his seminal 1937 paper, “On computable numbers...”, Alan Turing:

- a) Introduces what is now known as a Turing machine *oops, Turing did not address the Halting Problem*
- b) Provides a concrete construction of a universal Turing machine
- ~~c) Proves the undecidability of the Halting Problem~~
- d) Provides a convincing argument that any computation performed by a human can also be performed by mechanical means
- e) Solves a famous problem posed by David Hilbert
- f) All of the above

*← the text-book answer*

# Turing did not address the Halting Problem

- The paper is concerned with “computable numbers”
- A computable number, by definition, has an infinite decimal expansion
- Therefore, Turing wants his machines to run forever, writing more and more digits on the tape
- So he actually proves the undecidability of the “Prints-infinitely-many-digits Problem”

If a computing machine never writes down more than a finite number of symbols of the first kind, it will be called *circular*. Otherwise it is said to be *circle-free*.

A machine will be circular if it reaches a configuration from which there is no possible move, or if it goes on moving, and possibly printing symbols of the second kind, but cannot print any more symbols of the first kind. The significance of the term “circular” will be explained in § 8.

# The vast majority of textbooks and websites give a misleading account of Turing and the halting problem

The screenshot shows a Windows Internet Explorer browser window with the title "Turing Halting problem - Bing". The address bar contains the URL "http://www.bing.com/search?q=Turing+Halting+problem&form=IE&...". The search bar displays "Turing Halting problem" with a magnifying glass icon. Below the search bar, the Bing logo is visible, followed by the word "Web".

On the left side, under "RELATED SEARCHES", there is a list of links: Turing Test, Turing Sozluk, Turing Download, Turing Program, Turing Machine, Beaufort Scale, Dean Martin, and Holtsoft.

The main content area is titled "ALL RESULTS" and shows "1-10 of 70,100 results · [Advanced](#)". The first result is "Halting problem - Wikipedia, the free encyclopedia" with a link to the article. Below the title, there is a snippet of text: "Formal statement · Importance and ... · Sketch of proof · Common pitfalls". The snippet continues: "Theorem 2.2 There exists a Turing machine whose halting problem is recursively unsolvable. A related problem is the printing problem for a simple Turing machine Z with respect to a ...". Below the snippet is a link to "en.wikipedia.org/wiki/Halting\_problem · Cached page".

The second result is "What computers can't do" with a link to the article. Below the title, there is a snippet of text: "This problem is known as the Halting Problem for Turing machines" and was first proved in the 1937 paper in which he introduced his machines. To lead up to that proof, it is ...". Below the snippet is a link to "plus.maths.org/issue5/turing · Cached page".

The third result is "Halting Problem -- from Wolfram MathWorld" with a link to the article. Below the title, there is a snippet of text: "The determination of whether a Turing machine will come to a halt given a particular input program. The halting problem is solvable for machines with less than four states." Below the snippet is a link to "mathworld.wolfram.com/HaltingProblem.html · Cached page".

At the bottom of the browser window, the status bar shows "Done", "Internet | Protected Mode: Off", and "125%".

# I'm not the first person to realize this

Petzold Book Blog - Turing and the Halting Problem - Windows Internet Explorer

http://www.charlespetzold.com/blog/2007/11/Turing-Halting-P

Turing Halting problem

Favorites Petzold Book Blog - Turing and the H...

## PETZOLD BOOK BLOG

Charles Petzold on writing books, reading books, and exercising the internal UTM

---

[Recent Entries](#)  
[Browse the Archives](#)  
[Subscribe to the RSS Feed](#)

[< Previous](#) [Next >](#)

### Turing and the Halting Problem

November 26, 2007  
New York, N.Y.

What is wrong with the following passage?

Turing found that he could phrase his version of the question in terms of the problem of deciding whether or not the  $n^{\text{th}}$  Turing machine would actually ever *stop* when acting on the number  $m$ . This problem was referred to as the *halting problem*.  
— Roger Penrose, *The Emperor's New Mind* (Oxford University Press, 1989), pg. 57

The association of Turing with the "halting problem" is an anachronism. Turing's original conception in his 1936 paper "On Computable Numbers, with an Application to the Entscheidungsproblem" was of machines that compute real numbers. Because real numbers (in general) have an infinite number of digits, these machines run forever. A machine that gets into a non-printing loop, or which enters a configuration

Done Internet | Protected Mode: Off 100%



# Am I being pedantic?

- Perhaps
- However, the modern definition of *algorithm* is a Turing machine that halts on all inputs.
- The true significance of the Halting Problem's undecidability is that there's no algorithm for identifying other algorithms
- It's interesting that this formulation took years to evolve and did not spring perfectly formed from Turing's 1937 paper.

# Turing's paper is awesome, by the way

- In it, you will find, either explicitly or implicitly:
  - Turing machines, universal Turing machines, nondeterminism, a beautiful argument for the mechanizability of human computation, subroutines, a solution to Hilbert's Entscheidungsproblem, a definition of algorithm, proofs that an immense variety of mathematical calculations can be mechanized