David Adams
11/5/15

Alan Turing (1912 - 1954)

Computer Scientist, Mathematician, Logician, Cryptanalyst, Mathematical Biologist

- Why is he important?
    - Computer Science
    - Artificial Intelligence
- What are some cool things he did?
    - Enigma
    - Halting Problem
    - Turing Test
    - Turing Machines
    - and many more

A brief history of computer science

- Algorithms -> thousands of years old
- 1834 -> Charles Babbage -> punch cards
- 1843 -> Ada Lovelace -> first program
- 1885 -> Herman Hollerith -> more advanced punch cards -> IBM
- Just a bunch of people trying to make calculators until the 1940s
- 1953 -> Cambridge -> first CS diploma
- 1962 -> First CS diploma in USA

Enigma Machine

- Used by Germany in WWII
- In total has 159 quintillion different settings
- During WWII, Alan Turing worked for the Government Code and Cypher School
- He and his team at Bletchley Park created a machine that was able to solve the Enigma’s settings and thus decrypt messages
- Estimated to have shortened the war by two to four years

The Halting Problem

If I give you a computer program and an input to run the program on, can you ALWAYS for ANY program tell me if it will stop?

You can’t...It’s impossible...

In 1936, Alan Turing mathematically proved that there cannot be an answer to this problem and it is thus impossible.

This type of problem is called a decision problem and what Turing proved is that it is undecidable, which was one of the first problems to be proven undecidable.

Thus, a whole new way of thinking about problems was spawned (and a whole area of computer science->Computational Complexity Theory).

Why?

The answer can be shown through a proof by contradiction, which I’m not going to do now…

Essentially this means that if you assume a solution to the halting problem existed. Then, you could show that the solution shows that a program both halts and doesn’t halt at the same time, which is impossible and therefore can’t exist.

Easier way to think about it

Liar’s Paradox:

“I always lie”

“This sentence is false”

if it is True -> then it is False

if it is False -> then it is True

Nothing can be True and False at the same time, so they are undecidable!

Take these really simple two rules. Pretty easy to follow right?

1. Stop if you see a moving person.
2. Move if you see a stopped person.

Now walk in front of a mirror and see what happens?!?!

A slightly more formal explanation

https://github.com/davidtadams/HaltingProblem

Sources:

- https://en.wikipedia.org/wiki/Alan_Turing
- https://en.wikipedia.org/wiki/Computer_science
- https://en.wikipedia.org/wiki/Halting_problem
- https://www.quora.com/How-can-the-Halting-Problem-be-explained-to-a-layman

```
DEFINE DOESITHALT(PROGRAM):
{
      RETURN TRUE;
}
```

THE BIG PICTURE SOLUTION
TO THE HALTING PROBLEM