Barber paradox.

Barber announces:
“The barber shaves every person who does not shave themselves.”
Who shaves the barber?
Get around paradox?
The barber lies.

Russell’s Paradox.

Naive Set Theory: Any definable collection is a set.

$$\exists y \forall x (x \in y \iff P(x))$$  \hspace{1cm} (1)

$y$ is the set of elements that satisfies the proposition $P(x)$.
$P(x) = x \notin x$.
There exists a $y$ that satisfies statement (1) for $P(\cdot)$.
Take $x = y$.

$$y \in y \iff y \notin y.$$  

Oops!
What type of object is a set that contain sets?
Axioms changed.

Is it actually useful?

Write me a program checker!
Check that the compiler works!
How about.. Check that the compiler terminates on a certain input.

HALT($P, I$)

$P$ - program
$I$ - input.
Determines if $P(I)$ (P run on I) halts or loops forever.
Notice:
Need a computer
...with the notion of a stored program!!!!
(not an adding machine! not a person and an adding machine.)
Program is a text string.
Text string can be an input to a program.
Program can be an input to a program.

Implementing HALT.

$$\text{HALT}(P, I)$$
$P$ - program
$I$ - input.
Determines if $P(I)$ (P run on I) halts or loops forever.
Run $P$ on $I$ and check!
How long do you wait?
Something about infinity here, maybe?
Halt does not exist.

Halt and Turing.

Proof: Assume there is a program \( \text{HALT}(\cdot, \cdot) \).

1. If \( \text{HALT}(P, P) = \text{"halts"} \), then go into an infinite loop.
2. Otherwise, halt immediately.

Assumption: there is a program \( \text{HALT} \).

There is text that is the program Turing.

1. Can run Turing on Turing!
2. Does Turing(Turing) halt?

Turing(Turing) halts
\[ \Rightarrow \text{Turing}(\text{Turing}, \text{Turing}) = \text{halts} \]
\[ \Rightarrow \text{Turing}(\text{Turing}) \text{ loops forever.} \]

Turing(Turing) loops forever
\[ \Rightarrow \text{then HALTS(Turing}, \text{Turing}) \neq \text{halts} \]
\[ \Rightarrow \text{Turing(Turing) halts.} \]

Contradiction: Program HALT does not exist!

Proof play by play.

Assumed \( \text{HALT}(P, I) \) existed.

What is \( P \)? Text.

What is \( I \)? Text.

What does it mean to have a program \( \text{HALT}(P, I) \).

You have Text that is the program \( \text{HALT}(P, I) \).

Have Text that is the program \( \text{TURING} \).

Here it is!!

Turing(P)
1. If \( \text{HALT}(P, P) = \text{"halts"} \), then go into an infinite loop.
2. Otherwise, halt immediately.

Turing "diagonalizes" on list of program.
It is not a program!!!!

\[ \Rightarrow \text{HALT is not a program.} \]

Questions?

No computers for Turing!

In Turing's time.

No computers.

Adding machines.

\text{e.g., Babbage (from table of logarithms) 1812.}

Concept of program as data wasn't really there.
A Turing machine.
– an (infinite) tape with characters
– be in a state, and read a character
– move left, right, and/or write a character.
Universal Turing machine
– an interpreter program for a Turing machine
– where the tape could be a description of a ...

Now that's a computer!
Turing: AI, self modifying code, learning...

Church, Gödel and Turing.
Church proved an equivalent theorem. (Previously.)
Used λ calculus....which is... Lisp (Scheme)!!!
... functional part. Scheme's lambda is calculus's λ!
Programming languages! javascript, ruby, python....
Gödel: Incompleteness theorem.
Any formal system either is inconsistent or incomplete.
Inconsistent: A false sentence can be proven.
Incomplete: There is no proof for some sentence in the system.
Along the way: “built” computers out of arithmetic.
Showed that every mathematical statement corresponds to .... a natural number! ! ! ! Same cardinality as...Text.
Today:Programs can be written in ascii.

Undecidable problems.
Does a program, P, print “Hello World”?
How? What is P? Text!!!!!!
Find exit points and add statement: Print “Hello World.”
Can a set of notched tiles tile the infinite plane?
Proof: simulate a computer. Halts if finite.
Does a set of integer equations have a solution?
Example: “x^n + y^n = 1?”
Problem is undecidable.
Be careful!
Is there an integer solution to x^n + y^n = 1?
(Diophantine equation.)
The answer is yes or no. This “problem” is not undecidable.
Undecidability for Diophantine set of equations
⇒ no program can take any set of integer equations and
always correctly output whether it has an integer solution.

More about Alan Turing.
▶ Brilliant codebreaker during WWII, helped break German
Enigma Code (which probably shortened war by 1 year).
▶ Seminal paper in numerical analysis: Condition number.
Math 54 doesn’t really work.
Almost dependent matrices.
▶ Seminal paper in mathematical biology.
Person: embryo is blob. Legs, arms, head.... How?
Fly: blob. Torso becomes striped.
Developed chemical reaction-diffusion networks that
break symmetry.

Computing on top of computing...

Computer, assembly code, programming language, browser,
html, javascript..
We can't get enough of building more Turing machines.
Turing: personal.

- Tragic ending...
  - Arrested as a homosexual, (not particularly closeted)
  - given choice of prison or (quackish) injections to eliminate sex drive;
  - took injections.
  - lost security clearance...
  - suffered from depression;
  - suicided with cyanide at age 42.
  (A bite from the apple....) accident?

British Apology.

Gordon Brown. 2009. “Alan and the many thousands of other gay men who were convicted as he was convicted under homophobic laws were treated terribly. Over the years millions more lived in fear of conviction. ......... So on behalf of the British government, and all those who live freely thanks to Alan’s work I am very proud to say: we’re sorry, you deserved so much better.”

2013. Granted Royal pardon.

Back to technical..

This statement is a lie. Neither true nor false!

Every person who doesn’t shave themselves is shaved by the barber.

Who shaves the barber?

def Turing(P):
    if Halts(P,P): while(true): pass
    else:
        return
...Text of Halt...

Halt Program =⇒ Turing Program. (P =⇒ Q)

Turing(“Turing”)? Neither halts nor loops! =⇒ No Turing program.

No Turing Program =⇒ No halt program. (¬Q =⇒ ¬P)

Program is text, so we can pass it to itself, or refer to self.

Summary: decidability.

Computer Programs are an interesting thing.
Like Math.
Formal Systems.

Computer Programs cannot completely “understand” computer programs.

Computation is a lens for other action in the world.

Probability

What’s to come? Probability.
A bag contains:

What is the chance that a ball taken from the bag is blue?

For now: Counting!
Later: Probability.