15-252: More Great Ideas in Theoretical Computer Science

Fall 2017

Gödel's Incompleteness Th

Proving the famous

"Gödel Incompleteness Theorems"

is **easy** if you use computer science.

It's a Great Application of Theoretical Computer Science to mathematics.

Formalization of proofs

Euclid's *Elements* (ca. 300 BCE), on **plane geometry**.

Canonized the idea of giving a rigorous, axiomatic deduction for all theorems.

Formalization of proofs

Euclid's 5 axioms of plane geometry:

1. To draw a straight line from any point to any point.
2. To produce a finite straight line continuously in a straight line.
3. To describe a circle with any center and radius.
4. That all right angles are equal to one another.
5. If a straight line falling on two straight lines make the interior angles on the same side less than two right angles, the two straight lines, if produced indefinitely, meet on that side on which are the angles less than the two right angles.

WTF?

His proofs were not 100% formal, either.

At least he was trying!

Formalization of proofs

19^{th} century: True rigor developed.

Culminated in the understanding that
all math proofs can be completely formalized
using the language of First Order Logic
and an associated Deductive Calculus.

First Order Logic

A formal language for logical modeling.

English: "Alex has the coolest father."

FOL: $\forall x\ (\neg(x=\mathbf{a}) \rightarrow \mathrm{IsCooler}(\mathrm{Father}(\mathbf{a}),\mathrm{Father}(x)))$

- Includes basic Boolean connectives $\wedge$, $\vee$, $\neg$, $\rightarrow$
- Variables like x stand for *objects*, not true/false
- Also has $\forall$ (for all), $\exists$ (there exists), = (equals)
- You get to invent your own **vocabulary**, meaning *function names* (like Father), *relation names* (like IsCooler), and *constant names* (like **a**).
- You always have in mind a real-world / math-world **interpretation** of the vocabulary.

First Order Logic

+ Deductive Calculus

Deductive Calculus:

A textbook set of fixed rules that lets you deduce new FOL statements from older ones.

- If you have S and S→T, you can deduce T
- If you have IsCool(**a**), can deduce ∃x IsCool(x)
- If you have S, and S does not contain the variable name x, you can deduce ∀x S
- Plus 9 more rules like this (or more or fewer, depending on whose textbook you look in)

First Order Logic
+ Deductive Calculus

Important Note:

Deductive Calculus is
100% syntactic string manipulation.
You can write a 50-line computer program
that checks if a sequence of deductions is valid.

Using FOL to formalize parts of math

1. Take some area of math you want to reason about.
2. Invent an appropriate vocabulary
   (function, relation, and constant names).
3. Specify some axioms which are true under the interpretation you have in mind.
4. Go to town, deducing theorems from the axioms using Deductive Calculus.

Example 1: Arithmetic for 6-year-olds

Mojżesz Presburger

1929

Example 1: Arithmetic for 6-year-olds

More precisely: A theory of $\mathbb{N}$ and +.

Constant names: **0** and **1**

Function name: Plus(·,·)

Axioms:

#1: $\forall x\ \neg(\mathbf{0} = \text{Plus}(x,\mathbf{1}))$

#2: $\forall x\ \forall y\ (\text{Plus}(x,\mathbf{1}) = \text{Plus}(y,\mathbf{1})) \rightarrow (x=y)$

#3: $\forall x\ \text{Plus}(x,\mathbf{0}) = x$

#4: $\forall x\ \forall y\ \text{Plus}(x,\text{Plus}(y,\mathbf{1})) = \text{Plus}(\text{Plus}(x,y),\mathbf{1})$

#5: for any sentence S with free variable x,

$$(\ S(\mathbf{0}) \wedge (\forall x\ S(x) \rightarrow S(\text{Plus}(x,\mathbf{1})))\ ) \rightarrow \forall y\ S(y)$$

Example 1: Arithmetic for 6-year-olds

More precisely: A theory of ℕ and +.

Constant names: **0**

Function name: P

This is actually an infinite "axiom schema". That's OK!

Axioms:

#1: ∀x ¬(**0** = Plus(x,**1**))

#2: ∀x ∀y (Plus(x,**1**) = Plus(y,**1**)) → (x=y)

#3: ∀x Plus(x,**0**) = x

#4: ∀x ∀y Plus(x,Plus(y,**1**)) = Plus(Plus(x,y),**1**)

#5: for any sentence S with free variable x,

(S(**0**)∧(∀x S(x)→S(Plus(x,**1**)))) → ∀y S(y)

Example 1: Arithmetic for 6-year-olds

Fact: Starting from these 5 axioms (/schema), and using only the *purely syntactic* rules of Deductive Calculus, you can…

- Prove addition is associative!
  $\forall x\ \forall y\ \forall z\ \text{Plus}(\text{Plus}(x,y),z) = \text{Plus}(x,\text{Plus}(y,z))$
- Prove addition is commutative!
  $\forall x\ \forall y\ \text{Plus}(x,y) = \text{Plus}(y,x)$
- Prove every number is even or odd!
  $\forall x\ (\exists y\ \text{Plus}(y,y) = x \vee \text{Plus}(\text{Plus}(y,y),\mathbf{1}) = x)$

Example 1: Arithmetic for 6-year-olds

You can also build up new concepts that are not part of the formal vocabulary:

"x is even"… $\exists y \, Plus(y,y) = x$

"x < y"… $\exists z \, (\neg(z=0) \wedge Plus(x,z) = y)$

Example 2: Plane geometry done right

Alfred Tarski
1959

Example 2: Plane geometry done right

Relation names: IsBetween(x,y,z)
$\text{IsSameLength}(x_1,x_2,y_1,y_2)$

Axioms:

#1: $\forall x_1\ \forall x_2\ \text{IsSameLength}(x_1,x_2,x_2,x_1)$

#2: $\forall x\ \forall y\ \forall z\ \text{IsSameLength}(x,y,z,z)\to(x=y)$

#3: $\forall x\ \forall y\ \text{IsBetween}(x,y,x)\to(y=x)$

#4: ("Segment Extension")

$\forall x_1,x_2,y_1,y_2\ \exists z\ \text{IsBetween}(x_1,x_2,z)\wedge\text{IsSameLength}(x_2,z,y_1,y_2)$

#5−21: I won't bother to write them.

Example 2: Plane geometry done right

"m is the midpoint of $\overline{ab}$"…

IsBetween(a,m,b) ∧ IsSameLength(a,m,m,b)

"$\overline{ab}$ is parallel to $\overline{cd}$"…

(¬∃z IsBetween(a,b,z) ∧ IsBetween(c,d,z))

∧ (¬∃z IsBetween(z,a,b) ∧ IsBetween(z,c,d))

"x is on the circle that has center o

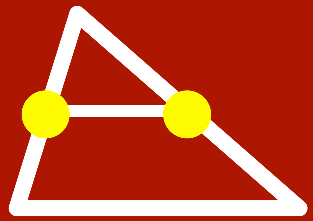
and radius the same length as $\overline{ab}$"…

IsSameLength(x,o,a,b)

Example 2: Plane geometry done right

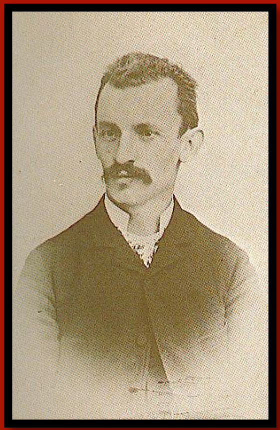
Fact: Starting from Tarski's 21 axioms, using only the purely syntactic rules of Deductive Calculus, you can prove many many things.

E.g.: "In any triangle abc, the line joining the midpoint of ab and the midpoint of bc is parallel to bc."



In fact: **Every** theorem about plane geometry in Euclid's book *Elements* can be so deduced!

More examples



Giuseppe Peano
1889

Gave a very successful list of
7 axioms/schema for
arithmetic of $\mathbb{N}$,
including multiplication.

Ernst Zermelo++
~1910's

Gave a very successful list of
9 axioms/schema for set theory.
Came to be known as "**ZFC**".

Say you are trying to axiomatize your favorite branch of math.

Some goals you should shoot for:

1. Computable axioms
2. Consistency
3. Soundness
4. Completeness

Computable axioms

It's nice if you have a finite number of axioms.

But often you need infinite families of axioms, like the Induction axiom schema in arithmetic:

for any sentence S with free variable x, have axiom

$$(S(\mathbf{0}) \wedge (\forall x\ S(x) \rightarrow S(Plus(x,\mathbf{1})))) \rightarrow \forall y\ S(y)$$

"Computable axioms" means:

L = { strings A : A is an axiom} is decidable.

An axiom system without this property is ridiculous!

Consistency

Let $A_1, \ldots, A_m$ be some axioms.

Suppose that using Deductive Calculus,
we can deduce from them some sentence S
and we can also deduce the sentence $\neg S$.

Then the axiom system is called inconsistent.

And you really screwed up!

Consistency

In fact, if your axiom system is inconsistent, then **every statement is provable.**

Theorem: Blahblahblah.

Proof: AFSOC ¬Blahblahblah.

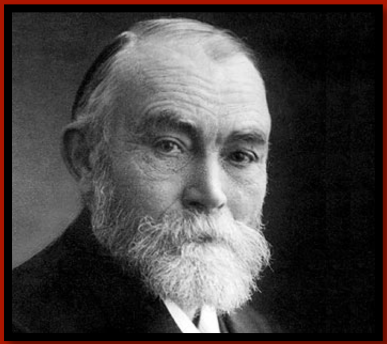
[Derive S from the axioms.]

[Derive ¬S from the axioms.]

Thus we have a contradiction.

Therefore Blahblahblah holds. □

Consistency



Frege, 1893:

Proposes axioms for set theory.

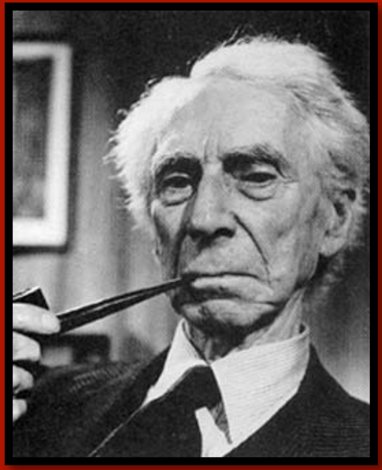
Spends 10 years writing two thick books about the system.

Russell, 1903: "Your axioms allow me to define $D = \{x : x \notin x\}$.

Now if $D \in D$ then $D \notin D$.

And if $D \notin D$ then $D \in D$.

Inconsistency, boom!"



Consistency

Frege, 1893:

Proposes axioms for set theory.

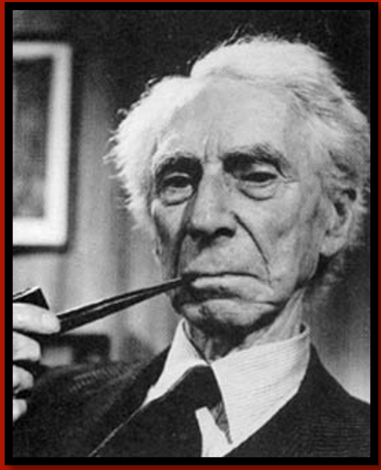
Spends 10 years writing two thick books about the system.

Russell, 1903: “Your axioms allow me to define $D = \{x : x \notin x\}$.

Now if $D \in D$ then $D \notin D$.

And if $D \notin D$ then $D \in D$.

Inconsistency, boom!”



Soundness

Let $A_1, \ldots, A_m$ be some axioms that model some branch of math you have in mind.

If every S that you can deduce is actually *true* (within the branch of math you have in mind) then the system is called sound.

Note 1: **Sound ⇒ Consistent**

Note 2: Consistency is a totally *syntactic* concept. But soundness relies on your ability to judge mathematical truth.

Presburger's arithmetic for 6-year-olds

More precisely: A theory of $\mathbb{N}$ and **+**.

Constant names: **0** and **1**

Function name: Plus(·,·)

Axioms:

#1: $\forall x\ \neg(\mathbf{0} = \mathrm{Plus}(x,\mathbf{1}))$

#2: $\forall x\ \forall y\ (\mathrm{Plus}(x,\mathbf{1}) = \mathrm{Plus}(y,\mathbf{1})) \rightarrow (x=y)$

#3: $\forall x\ \mathrm{Plus}(x,\mathbf{0}) = x$

#4: $\forall x\ \forall y\ \mathrm{Plus}(x,\mathrm{Plus}(y,\mathbf{1})) = \mathrm{Plus}(\mathrm{Plus}(x,y),\mathbf{1})$

#5: for any sentence S with free variable x,

$$(\ S(\mathbf{0}) \wedge (\forall x\ S(x) \rightarrow S(\mathrm{Plus}(x,\mathbf{1})))\ ) \rightarrow \forall y\ S(y)$$

Presburger's arithmetic for 6-year-olds

More precisely: A theory of $\mathbb{N}$ and **+**.

Constant names: **0** and **1**

Function name: Plus(·,·)

Poll

Axioms:

#1: ∃x **0** = Plus(x,**1**)

#2: ∀x ∀y (Plus(x,**1**) = Plus(y,**1**)) → (x=y)

#3: ∀x Plus(x,**0**) = x

#4: ∀x ∀y Plus(x,Plus(y,**1**)) = Plus(Plus(x,y),**1**)

#5: for any sentence S with free variable x,

(S(**0**)∧(∀x S(x)→S(Plus(x,**1**)))) → ∀y S(y)

Presburger's arithmetic for 6-year-olds

More precisely: A theory of $\mathbb{N}$ and +.

Constant names: **0**

Function name: Pl

Still consistent:
it's validly modeling
integers mod 2!

Axioms:

#1: $\exists x\ \mathbf{0} = \text{Plus}(x,\mathbf{1})$

#2: $\forall x\ \forall y\ (\text{Plus}(x,\mathbf{1}) = \text{Plus}(y,\mathbf{1})) \rightarrow (x=y)$

#3: $\forall x\ \text{Plus}(x,\mathbf{0}) = x$

#4: $\forall x\ \forall y\ \text{Plus}(x,\text{Plus}(y,\mathbf{1})) = \text{Plus}(\text{Plus}(x,y),\mathbf{1})$

#5: for any sentence S with free variable x,

$$(\ S(\mathbf{0}) \wedge (\forall x\ S(x) \rightarrow S(\text{Plus}(x,\mathbf{1})))\ ) \rightarrow \forall y\ S(y)$$

Completeness

Let $A_1, \ldots, A_m$ be some axioms.

If, for every sentence S,
either S or ¬S is deducible from the axioms,
we say the system is complete.

If you have a branch of math in mind
that you're modeling, then…
Complete ⇔ Every true statement
can be deduced from the axioms

Completeness

Completeness, like consistency, is a *completely syntactic property*.

Completeness:

For any S, at **least** one of "S" or "¬S" can be deduced.

Consistency:

For any S, at **most** one of "S" or "¬S" can be deduced.

Completeness

When you're messing around trying to axiomatize your favorite branch of math, it's quite common to suffer from "incompleteness".

It's, like, you didn't put in "enough" axioms.

Example: Tarski's plane geometry

Relation names: IsBetween(x,y,z)

IsSameLength(x_1,x_2,y_1,y_2)

Axioms:

#1: $\forall x_1 \forall x_2$ IsSameLength(x_1,x_2,x_2,x_1)

#2: $\forall x \forall y \forall z$ IsSameLength(x,y,z,z)$\to$(x=y)

#3: $\forall x \forall y$ IsBetween(x,y,x)$\to$(y=x)

#4: ("Segment Extension")

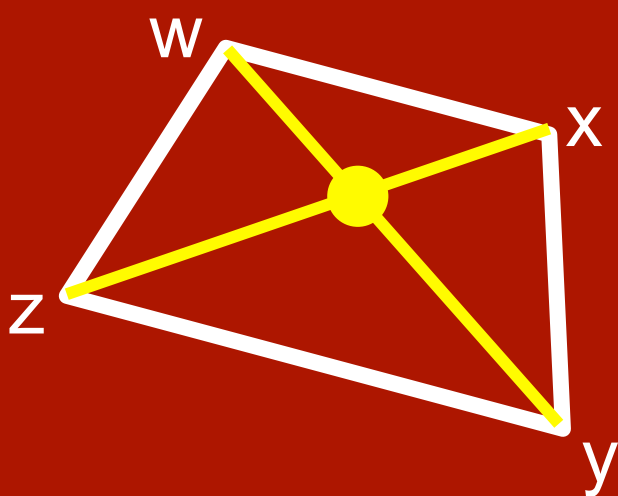
$\forall x_1,x_2,y_1,y_2\ \exists z$ IsBetween(x_1,x_2,z)$\wedge$IsSameLength(x_2,z,y_1,y_2)

#5−21: I won't bother to write them.

Example: Tarski's plane geometry

One of the 21 axioms says,

"If wxyz is a quadrilateral, then the diagonals wy and xz must intersect."



Historically, people tried hard to prove this statement using only the other axioms.

But, in fact, you can't!
(We can prove that!)

So fine, you add it as an axiom.

Say you are trying to axiomatize your favorite branch of math.

Some goals you should shoot for:

1. Computable axioms
2. Consistency
3. Soundness
4. Completeness

Presburger's arithmetic for 6-year-olds

It has computable axioms.
It's consistent.
Indeed, it's sound.

And…
Presburger proved it's complete.

Hooray! We have perfectly axiomatized arithmetic for 6-year-olds!

#5: for any sentence S with free variable x,

$$(S(\mathbf{0}) \wedge (\forall x\ S(x) \rightarrow S(\mathrm{Plus}(x,\mathbf{1})))) \rightarrow \forall y\ S(y)$$

Example: Tarski's plane geometry

Re

It has computable axioms.

It's consistent.

Indeed, it's sound.

A

And…

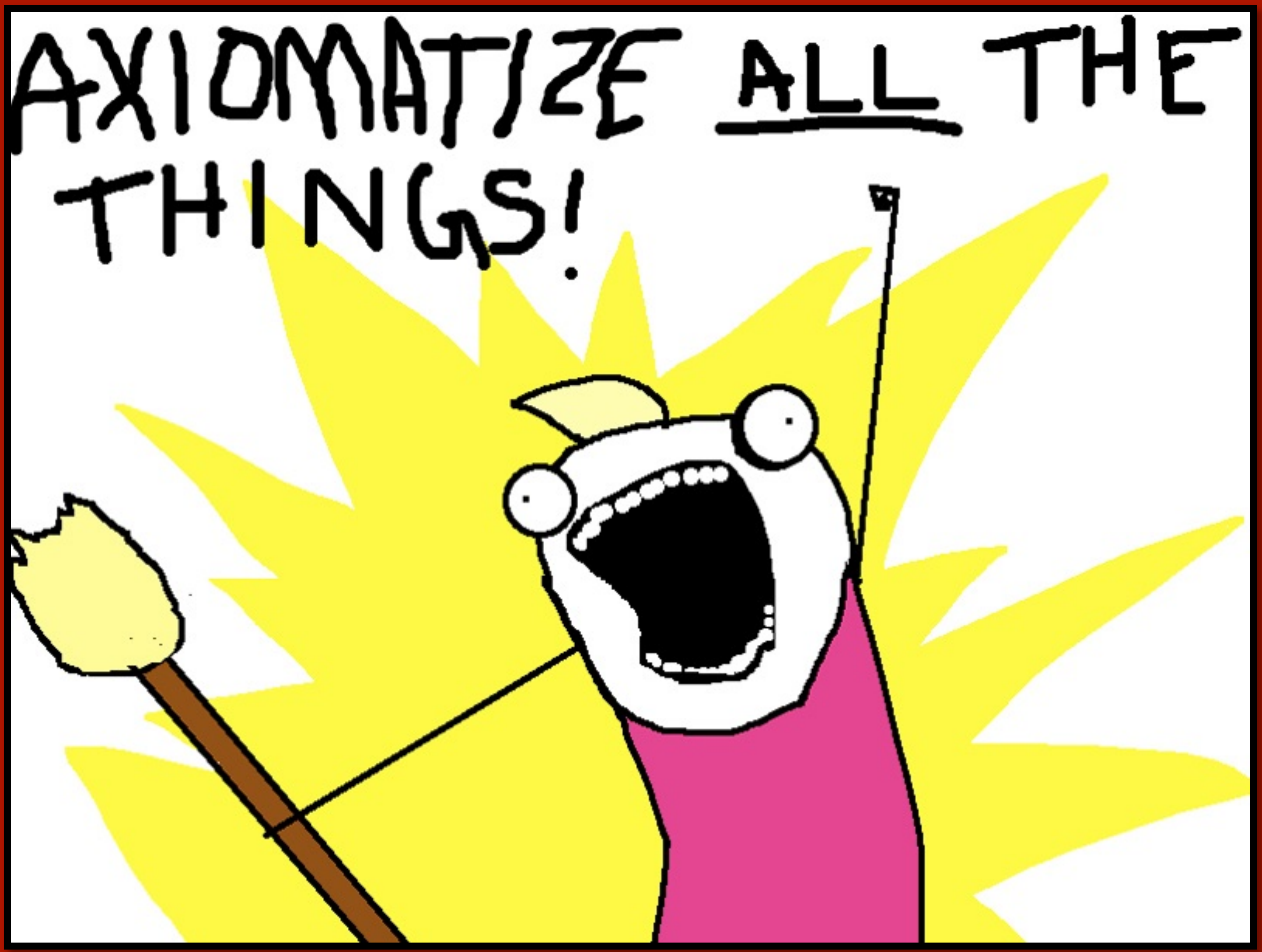
Tarski proved it's complete.

Hooray! We have perfectly axiomatized basic Euclidean geometry!

$\forall x_1,x_2,y_1,y_2\ \exists z\ \text{IsBetween}(x_1,x_2,z) \wedge \text{IsSameLength}(x_2,z,y_1,y_2)$

#5−21: I won't bother to write them.

A dream from the early 20th century



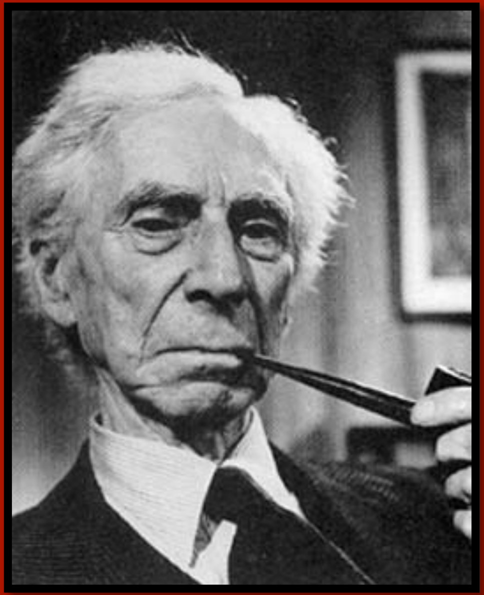
Axiomatizing all the things

After playing around, people realized you could seemingly do 100% of math using just the notions from set theory.

(Define natural numbers in terms of sets, ordered pairs in terms of sets, functions in terms of sets, sequences in terms of sets, real numbers, graphs, strings, automata, **everything** in terms of sets…)

They fixed the 9 “ZFC” axioms/schema for set theory and proceeded to go to town.

Principia Mathematica, ca. 1912



Bertrand Russell

Alfred Whitehead

Purely by combining set theory axioms with Deductive Calculus, they developed tons of number theory and some real analysis.

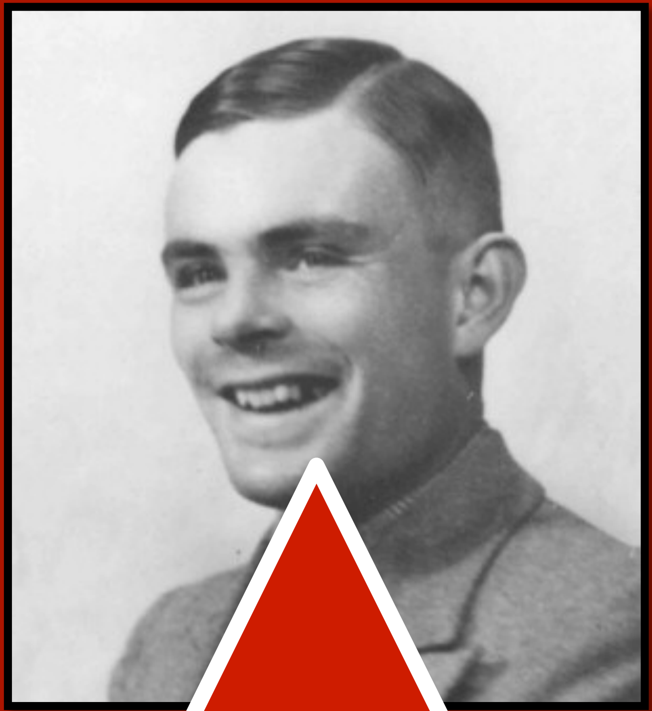
Axiomatizing all the things?

It was a huge pain (think, 500-page books…)
but it was going great.

By the end of the 1920's, mathematicians
were all pretty satisfied.

Empirical conclusion: Seemed you could formally
prove anything in math you wanted,
just from ZFC and syntactic Deductive Calculus.

By the way, all theorems in 15-251 can be so proved.



Hey, can I cut in for
a second and remind people about
my theorem?

Fine.

The Halting Problem is Undecidable

Turing's Theorem:

Let HALTS $\subseteq \{0,1\}^*$ be the language

$\{ \langle M,x \rangle$: M is a TM which halts on input x $\}$.

Then HALTS is undecidable.

It's not: "we don't know how to solve it efficiently".

It's not: "we don't know if it's a solvable problem".

We know that it is unsolvable by any algorithm.

Proof

Assume M_{HALTS} is a decider TM which decides HALTS.

Here is the description of another TM called D, which uses M_{HALTS} as a subroutine:

D:

> Given as input ⟨M⟩, the encoding of a TM M:
>
> D executes M_{HALTS}(⟨M, ⟨M⟩⟩).
>
> If this call accepts, D enters an infinite loop.
>
> If this call rejects, D halts (say, it accepts).

By definition, **D(⟨D⟩)** loops if it halts and halts if it loops.

Contradiction.

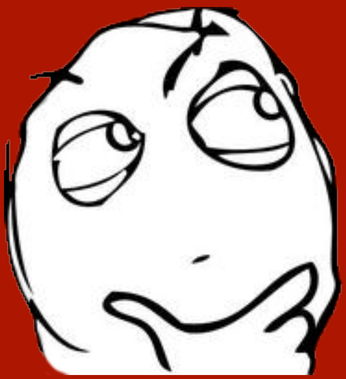
Suppose you just really cannot believe we proved that HALTS is undecidable.

How would you try to write a program H which, on input $\langle M,x \rangle$, decides if M(x) eventually halts?

Sample input:

M = “for k = 4, 6, 8, 10, 12, 14, …
check if k is the sum of 2 primes; if not, HALT”

$x = \epsilon$ (empty string)



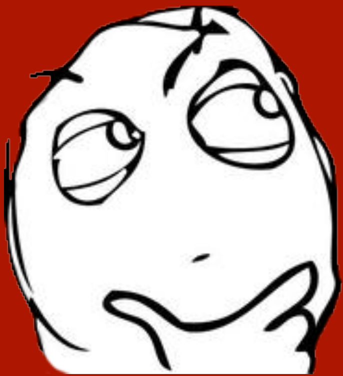
Dunno. Best idea I can think of is:
Let H simulate M(x). If M(x) halts after 1,000,000,000 steps, output "it halts". If M(x) still hasn't halted after 1,000,000,000 steps, um…

How would you try to write a program H which, on input ⟨M,x⟩, decides if M(x) eventually halts?

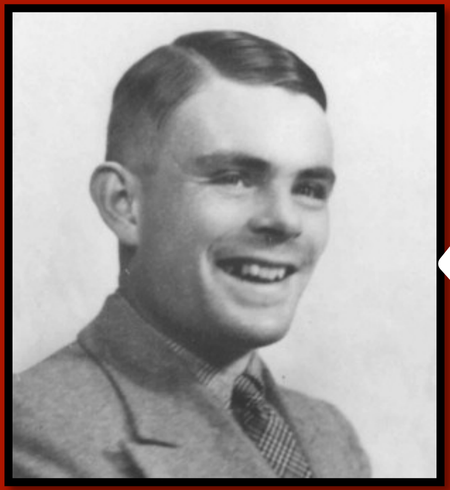
Sample input:

M = "for k = 4, 6, 8, 10, 12, 14, …
check if k is the sum of 2 primes; if not, HALT"

x = ϵ (empty string)



I have a crazy and sort of awesome idea for how to write H.



Kurt, you mathematicians always make things too complicated.

Let me explain it.

Fine.

How would you try to write a program H which, on input ⟨M,x⟩, decides if M(x) eventually halts?

Idea for H:

“ for k = 1, 2, 3, …

for all strings P of length k,

- Check if P is a valid ZFC+FOL Deductive Calculus proof of the statement ‘M(x) eventually halts’

  If so, let H halt and output “yes, M(x) halts”
- Check if P is a valid ZFC+FOL Deductive Calculus proof of the statement ‘M(x) eventually loops’

  If so, let H halt and output “no, M(x) loops” ”

By my theorem: this TM H, like all algorithms, **does not** decide the Halting Problem.

Idea for H:

" for k = 1, 2, 3, …

for all strings P of length k,

- Check if P is a valid ZFC+FOL Deductive Calculus proof of the statement 'M(x) eventually halts'

  If so, let H halt and output "yes, M(x) halts"

- Check if P is a valid ZFC+FOL Deductive Calculus proof of the statement 'M(x) eventually loops'

  If so, let H halt and output "no, M(x) loops" "

Conclusion:

There is some TM M and some string x such that ZFC+FOL Deductive Calculus **cannot prove** either of 'M(x) eventually halts' or 'M(x) eventually loops'.

But M(x) either halts or it loops!
One of these two statements is true!

∴ **There is a true mathematical statement that cannot be proved** (in ZFC+FOL Deductive Calculus).

This is basically

Gödel's First Incompleteness Theorem.

“ for k = 1, 2, 3, …

for all strings P of length k,

- Check if P is a valid ZFC+FOL Deductive Calculus proof of the statement ‘M(x) eventually halts’

  If so, let H halt and output “yes, M(x) halts”
- Check if P is a valid ZFC+FOL Deductive Calculus proof of the statement ‘M(x) eventually loops’

  If so, let H halt and output “no, M(x) halts” ”

Conclusion:

There is some TM M and some string x such that ZFC+FOL Deductive Calculus **cannot prove** either of ‘M(x) eventually halts’ or ‘M(x) eventually loops’.

Actually, this is not a 100% correct conclusion, because there's another possibility:

ZFC+FOL Deductive Calculus might have a proof that 'M(x) eventually halts' *even though it loops*, or 'M(x) eventually loops' *even though it halts*.

Conclusion:

There is some TM M and some string x such that ZFC+FOL Deductive Calculus **cannot prove** either of 'M(x) eventually halts' or 'M(x) eventually loops'.

Actually, this is not a 100% correct conclusion,
because there's another possibility:

ZFC+FOL Deductive Calculus might have a proof that 'M(x) eventually halts' *even though it loops*, or 'M(x) eventually loops' *even though it halts*.

I.e., ZFC might be **unsound**:
it might prove some false statements.

This would kind of upend all of mathematics.
Essentially everyone believes ZFC is sound.
But theoretically, it's a possibility.

What we've actually proven so far:

ZFC + FOL Deductive Calculus cannot be both

complete

and **sound**.

Complete:

for every sentence S, either S or ¬S is provable.

Sound:

for every S, if S is provable then S is true.

What we've actually proven so far:

ZFC + FOL Deductive Calculus cannot be both

complete

and **sound**.

Question:

What did this proof use about ZFC?

Answer: Not too much.

- You can define TM's and TM computation in it.
- Its axioms/axiom schemas are computable.

Gödel's First Incompleteness Theorem:

Any axiomatic system which is
"sufficiently expressive" (can define TM's)
and has **computable axioms**
cannot be both **complete** and **sound**.

Side remark:

Even Peano Arithmetic is "sufficiently expressive".
You **can** define TM's and TM computation in it,
though it is a severe pain in the neck.

A smart-aleck's attempt to circumvent Gödel's First Incompleteness Theorem:

"Let's assume ZFC is sound. Gödel's Theorem says that there's some true statement S which can't be proved in ZFC. Let's just upgrade ZFC by adding S as an axiom!"

Doesn't help:

ZFC+S is a sufficiently expressive system with computable axioms. So by Gödel's Theorem, there's still some other S' which is true but can't be proved.

A smart-aleck's attempt to circumvent Gödel's First Incompleteness Theorem:

"Maybe add in S' as another axiom?"

Still doesn't help:

Apply Gödel's Theorem to ZFC+S+S', get yet another true statement S'' which is true but cannot be proved.

*"Maybe add in **all** true statements as axioms?"*

Okay fine, but now the set of axioms is not computable. So it's a ridiculous system.

Gödel's First Incompleteness Theorem:

Any axiomatic system which is "sufficiently expressive" (can define TM's) and has **computable axioms** cannot be both **complete** and **sound**.

Sound:

for every S,
if S is provable
then S is true.

Whoahhhh, dude.

How can you say a statement S is true if you can't prove it?

Response 1

Don't get all confused. If I asked you 40 mins ago,

"Hey, is it true that 1 is the only number which appears in Pascal's Triangle more than ten times?",

you wouldn't be, like,

"Whoahhhh dude, what does true mean?"

Regular ol' mathematics doesn't suddenly become invalid just because you happen to be studying logic.

Response 2

Just so that nobody gets confused,
I'll prove an even stronger version
which doesn't mention "truth".

Gödel's 1st: full version

(with strengthening by J. Barkley Rosser)

Any axiomatic system which is
"sufficiently expressive" (can define TM's)
and has **computable axioms**
cannot be both **complete** and **consistent**.

Complete:

for every sentence S, either S or ¬S is provable.

Consistent:

for every S, you can't prove both S and ¬S.

Not only will we prove this,
there will be a bonus plot twist at the end!

For simplicity, we fix the mathematical
proof system to be ZFC.

Outline of previous proof:

1. Assume ZFC sound.
2. Reason about a certain TM.
3. Deduce that ZFC is **incomplete**.

Outline of upcoming stronger proof:

1. Assume ZFC consistent.
2. Reason about a certain TM.
3. Deduce that ZFC is **incomplete**.

We’re going to need a lemma.

Some statements are so simple that,
assuming they’re true,
they **definitely do** have a proof in ZFC.

Example: “There are 25 primes less than 100.”

This definitely has a proof:
the brute-force, brain-dead enumeration proof!

Our Brain-Dead Lemma:

If a particular TM has a particular t-step execution trace,
then there is a proof of this fact (in ZFC).

Why? Can always write (in ZFC) proofs that look like:

"Initially M in the starting state/head/tape configuration.
After 1 step, M is in state/head/tape configuration *blah*.
After 2 steps, M is in state/head/tape configuration *blah*.
After 3 steps, M is in state/head/tape configuration *blah.*
… After t steps, M is in state/head/tape configuration *blah.*
QED."

In particular, if M(x) halts, there is a proof of 'M(x) halts'.

Outline of upcoming proof of the "truth"-free stronger version of Gödel's 1st:

1. Assume ZFC consistent.
2. Reason about a certain TM.
3. Deduce that ZFC is **incomplete**.

Proof of stronger Incompleteness Theorem

Assume ZFC consistent.

Let D be the TM which on input ⟨M⟩ does:

for all strings P of length 1, 2, 3, …

- If P is a ZFC proof of 'M(⟨M⟩) halts', enter 'go right forever' state.
- If P is a ZFC proof of 'M(⟨M⟩) loops', then halt.

What can ZFC prove about D(⟨D⟩)? By consistency,
at most one of 'D(⟨D⟩) halts' or 'D(⟨D⟩) loops'.

Perhaps ZFC can prove 'D(⟨D⟩) loops'?

Then D on input ⟨D⟩ will find this proof, and thus halt.

But if D(⟨D⟩) halts **then ZFC can prove** 'D(⟨D⟩) halts'

(by Brain-Dead Lemma). This contradicts consistency.

Proof of stronger Incompleteness Theorem

Assume ZFC consistent.

Let D be the TM which on input ⟨M⟩ does:

for all strings P of length 1, 2, 3, …

- If P is a ZFC proof of 'M(⟨M⟩) halts', enter 'go right forever' state.
- If P is a ZFC proof of 'M(⟨M⟩) loops', then halt.

What can ZFC prove about D(⟨D⟩)? By consistency,
at most one of 'D(⟨D⟩) halts' or 'D(⟨D⟩) loops'.

Perhaps ZFC can prove 'D(⟨D⟩) halts'?

Then D(⟨D⟩) will run for some t steps, find this proof, and then enter the 'go right forever' state. But by Brain-Dead Lemma, **there's a proof of this fact** (the t+1 step execution trace). Thus ZFC can prove 'D(⟨D⟩) loops', contradicting consistency.

Proof of stronger Incompleteness Theorem

Assume ZFC consistent.

Let D be the TM which on input ⟨M⟩ does:

for all strings P of length 1, 2, 3, …

- If P is a ZFC proof of 'M(⟨M⟩) halts', enter 'go right forever' state.
- If P is a ZFC proof of 'M(⟨M⟩) loops', then halt.

Great! We just showed ZFC cannot prove either 'D(⟨D⟩) loops' or 'D(⟨D⟩) halts'. So ZFC is incomplete. ∎

Incidentally… does D(⟨D⟩) **actually** halt or loop?

It loops. It does not find a proof of either statement.

Proof of stronger Incompleteness Theorem

Assume ZFC consistent.

Let D be the TM which on input ⟨M⟩ does:

for all strings P of length 1, 2, 3, …

- If P is a ZFC proof of 'M(⟨M⟩) halts', enter 'go right forever' state.
- If P is a ZFC proof of 'M(⟨M⟩) loops', then halt.

Great! We just showed ZFC cannot prove either 'D(⟨D⟩) loops' or 'D(⟨D⟩) halts'. So ZFC is incomplete. □

Wait a minute.

It loops. It does not find a proof of either statement.

Proof of stronger Incompleteness Theorem

Assume ZFC consistent.

Let D be the TM which on input ⟨M⟩ does:

for all strings P of length 1, 2, 3, …

- If P is a ZFC proof of 'M(⟨M⟩) halts', enter 'go right forever' state.
- If P is a ZFC proof of 'M(⟨M⟩) loops', then halt.

Great! We just showed ZFC cannot prove either 'D(⟨D⟩) loops' or 'D(⟨D⟩) halts'. So ZFC is incomplete. ■

Wait a minute. We just showed that D(⟨D⟩) loops.

If we formalize the last 3 slides in ZFC,

we get a proof of 'D(⟨D⟩) loops'.

Did we just find a contradiction in mathematics?

Proof of stronger Incompleteness Theorem

Assume ZFC consistent.

Let D be the TM which on input ⟨M⟩ does:

for all strings P of length 1, 2, 3, …

- If P is a ZFC proof of ‘M(⟨M⟩) halts’, enter ‘go right forever’ state.
- If P is a ZFC proof of ‘M(⟨M⟩) loops’, then halt.

Great! We just showed ZFC cannot prove either
‘D(⟨D⟩) loops’ or ‘D(⟨D⟩) halts’. So ZFC is incomplete. ■

Wait a minute. We just showed that D(⟨D⟩) loops.

If we formalize the last 3 slides in ZFC,
we get a proof of ‘~~D(⟨D⟩) loops~~’.

Proof of stronger Incompleteness Theorem

Assume ZFC consistent.

Let D be the TM which on input ⟨M⟩ does:

for all strings P of length 1, 2, 3, …

- If P is a ZFC proof of 'M(⟨M⟩) halts', enter 'go right forever' state.
- If P is a ZFC proof of 'M(⟨M⟩) loops', then halt.

Great! We just showed ZFC cannot prove either 'D(⟨D⟩) loops' or 'D(⟨D⟩) halts'. So ZFC is incomplete. ∎

Wait a minute. We just showed that D(⟨D⟩) loops.

If we formalize the last 3 slides in ZFC,
we get a proof of 'ZFC consistent → D(⟨D⟩) loops'.

Proof of stronger Incompleteness Theorem

Assume ZFC consistent.

Let D be the TM which on input ⟨M⟩ does:

for all strings P of length 1, 2, 3, …

- If P is a ZFC proof of 'M(⟨M⟩) halts', enter 'go right forever' state.
- If P is a ZFC proof of 'M(⟨M⟩) loops', then halt.

Great! We just showed ZFC cannot prove either

'D(⟨D⟩) lo

The only way to avoid a contradiction:

ZFC cannot prove 'ZFC consistent'

If we formalize the last 3 slides in ZFC,

we get a proof of 'ZFC consistent → D(⟨D⟩) loops'.

Gödel's **Second** Incompleteness Theorem

(proved independently by von Neumann)

Assume ZFC (or any "sufficiently expressive" axiomatic system) is **consistent**. Then not only is it incomplete, here's **a true statement it cannot prove**: **"ZFC is consistent"**.