Logic, Computability and Incompleteness

Intro & Overview

Wolfgang Schwarz

19 September 2025

Course info

Course info

- This course introduces central results about the nature, scope, and limitations of formal logic, mathematical proof, and algorithmic computability.
- You must have taken Logic 1 or something equivalent.
- Logic 1 was easy. This course is harder.

Course info

Website, Readings, Exercises

- The website for this course is `www.wolfgangschwarz.net/logic3`.
- Each week I will post extensive lecture notes with exercises.
- Please read the notes, do the exercises, and prepare at least one question for the seminar.
- Resist the temptation to ask an AI whenever you feel stuck or confused.

Course info

Assessment

- 20% Take-home test, released 13 October
- 80% Final exam, sometime in December

What I assume you know

What I assume you know

We'll be interested in formal theories in the language of first-order predicate logic.

I assume that you are familiar with the symbols

- $\neg$ (not)
- $\wedge$ (and)
- $\vee$ (or)
- $\rightarrow$ (if ...then)
- $\leftrightarrow$ (if and only if)
- $\forall$ (for all)
- $\exists$ (there is)

and how they can be used to form sentences:

$$P \wedge Q \qquad \forall x(Fx \rightarrow Gx) \qquad \exists x(Fx \wedge Gx)$$

What I assume you know

We will spend very little time trying to prove statements in first-order logic.

Instead, we'll spend a lot of time proving statements about first-order logic, and about proofs in first-order logic.

These meta-level proofs will be in English, with some additional mathematical notation.

What I assume you know

Theorem 3.1: Cantor's Theorem

The set of all sets of natural numbers is not countable.

Proof. Assume for reductio that the set of all sets of natural numbers is countable. This means that we can list them as $S_0, S_1, S_2, \ldots$. Now consider the set $D = \{n \in \mathbb{N} : n \notin S_n\}$. This is a set of natural numbers, so it must be somewhere in the list. That is, there is some S_k such that $D = S_k$. Now the number k is either in D or not. If k is in D then by definition of D, k is not in S_k. This is impossible, as $D = S_k$. If k is not in D, then by definition of D, k is in S_k. Again, this is impossible. □

What I assume you know

Common meta-level argument forms

- Reductio/Indirect Proof: Assume $\neg P$. Derive a contradiction. Conclude P.
- Conditional Proof: Assume P. Derive Q. Conclude that if P then Q.
- Contraposition: Assume $\neg Q$. Derive $\neg P$. Conclude that if P then Q.
- Generalization: Show that arbitrary object is F. Conclude that all objects are F.

What I assume you know

Set notation

- $\{1, 2, 3\}$ is the set whose members are 1, 2, and 3.
- $\{x : A(x)\}$ is the set of all x such that $A(x)$.
- $\emptyset$ is the empty set (the set with no members).
- $x \in s$ means that x is a member of the set s.
- $x \subseteq y$ means that every member of x is also a member of y.

Historical overview

Historical overview

1. Propositional logic
2. First-order predicate logic
3. Completeness of first-order logic
4. Formal theories
5. Computability
6. Turing machines
7. Recursive functions
8. Arithmetic representability
9. Gödel's first incompleteness theorem
10. Gödel's second incompleteness theorem

Historical overview

Mathematics in the 19th century

By the 1800s, mathematics had become sophisticated, but its foundations were recognized to be shaky.

Cauchy, Weierstrass, Dedekind, Cantor, and others tried to remedy this situation by offering precise definitions and rigorous proofs.

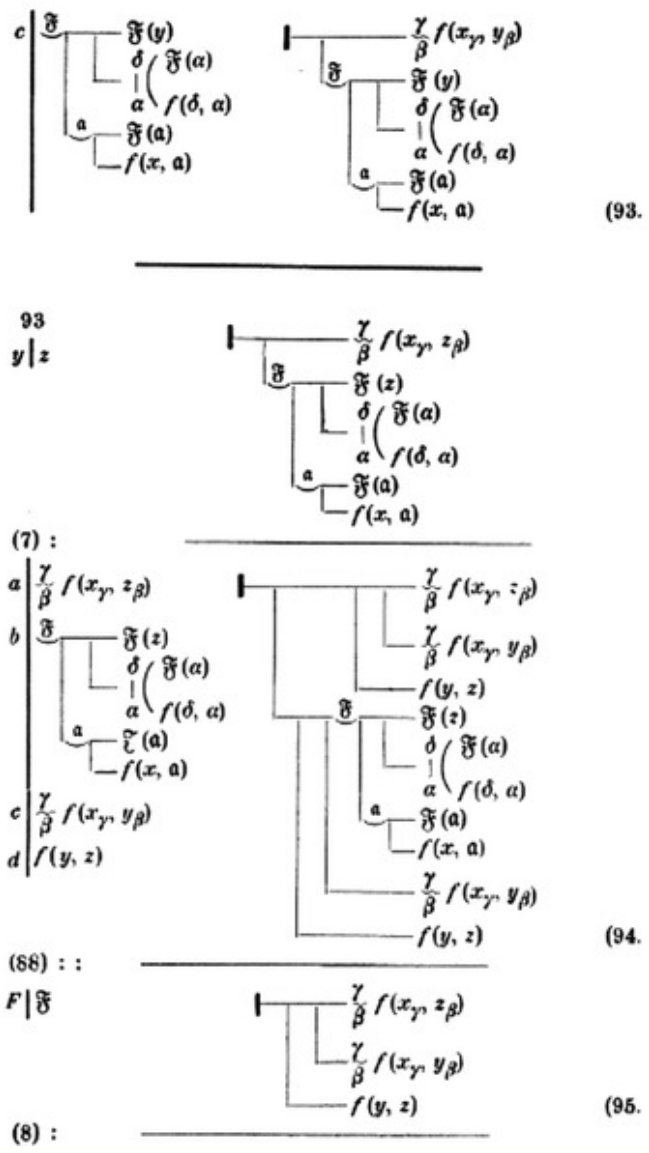
Vision: All mathematical truths should be logically derivable from clearly stated principles (axioms).

Historical overview

Frege's Begriffsschrift (1879) introduced a formal language in which all mathematical reasoning could be expressed.

Frege also introduced a calculus for constructing proofs in this language.

Historical overview



Historical overview

In Frege's calculus, a proof is a sequence of formulas, each of which is either an axiom or follows from earlier formulas by a rule of inference.

A1	$A \to (B \to A)$
A2	$(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
A3	$(\neg A \to \neg B) \to (B \to A)$
A4	$\forall x A \to A(x/c)$
A5	$\forall x(A \to B) \to (A \to \forall x B)$, *if x is not free in A*
MP	*From A and $A \to B$ one may infer B.*
Gen	*From A one may infer $\forall x A(c/x)$.*

Historical overview

In Frege's calculus, a proof is a sequence of formulas, each of which is either an axiom or follows from earlier formulas by a rule of inference.

More user-friendly calculi were developed in the 1920s-1950s.

- Natural deduction (ND) calculi
- Sequent calculi
- Tableau calculi

They are all equivalent in terms of what they can prove.

Historical overview

The turnstile

We write:

- $\vdash A$ for "A is provable";
- $\Gamma \vdash A$ for "A is derivable from Γ".

Examples:

- $\vdash P \to P$
- $\{Fa, \forall x(Fx \to Gx)\} \vdash Ga$

Evaluating Logical Calculi

How do we know if a proof system is any good?

- Are all its derivations valid?
- Can it derive all valid inferences?

We need a precise concept of validity that is not based on provability.

Historical overview

Semantic Consequence

An inference is valid if in every situation where the premises are true, the conclusion is also true.

A **model** specifies which statements are true in a particular situation.

A model of first-order logic consists of:

- A non-empty set of objects
- An interpretation function that assigns
  - an object to each individual constant,
  - a set of tuples of objects to each predicate symbol.

Semantic Consequence

An inference is valid if in every situationevery model where the premises are true, the conclusion is also true.

We write:

- $\models A$ for "A is true in every model";
- $\Gamma \models A$ for "A is true in every model in which all sentences in Γ are true".

Examples:

- $\models P \to P$
- $\{Fa, \forall x(Fx \to Gx)\} \models Ga$

Evaluating Logical Calculi

How do we know if a proof system is any good?

- Are all its derivations valid?
  If $\Gamma \vdash A$, do we have $\Gamma \models A$?
- Can it derive all valid inferences?
  If $\Gamma \models A$, do we have $\Gamma \vdash A$?

Soundness: If $\Gamma \vdash A$ then $\Gamma \models A$.

Completeness: If $\Gamma \models A$ then $\Gamma \vdash A$.

Historical overview

Gödel's Completeness Theorem (1929):

If $\Gamma \models A$ then $\Gamma \vdash A$.

Historical overview

Formal Theories

In the late 19th and early 20th century, rigorous axiomatizations of various mathematical domains were developed.

- Real analysis (Weierstrass, Dedekind, 1870s-1880s)
- Arithmetic (Dedekind, Peano, 1880s)
- Euclidean geometry (Hilbert, 1899)
- Set theory (Zermelo, 1908)
- Type theory (Russell, Whitehead, 1910s)

Historical overview

Peano Arithmetic (PA)

$$
\begin{array}{ll}
PA1 & \forall x \forall y\,(s(x) = s(y) \rightarrow x = y) \\
PA2 & \forall x\,(\neg(s(x) = 0)) \\
PA3 & \forall x\,(x + 0 = x) \\
PA4 & \forall x \forall y\,(x + s(y) = s(x + y)) \\
PA5 & \forall x\,(x \times 0 = 0) \\
PA6 & \forall x \forall y\,(x \times s(y) = (x \times y) + x) \\
PA7 & A(0) \wedge \forall x(A(x) \rightarrow A(s(x))) \rightarrow \forall x A(x)
\end{array}
$$

Historical overview

Formal Theories

In the late 19th and early 20th century, rigorous axiomatizations of various mathematical domains were developed.

Vision: All truths of the domain should be derivable from the axioms.

Mathematics would reduce to a game of logical derivation.

All mathematical questions could be answered mechanically.

Historical overview

"Wir müssen wissen, wir werden wissen!" (David Hilbert, 1930)

We must know, we will know!

Historical overview

The Decision Problem

If A is derivable from the axioms, we can mechanically find a derivation.

But what if A is not derivable?

How can we mechanically determine that there is no proof?

This is Hilbert's Entscheidungsproblem.

Historical overview

Hilbert's Entscheidungsproblem (1928)

Find a mechanical algorithm that can decide, for any given first-order statement and axioms, whether the statement is derivable from the axioms.

Church's Theorem (1936):

There is no such algorithm.

To prove this, we need a precise definition of "mechanical algorithm".

This led to the development of computability theory in the 1930s.

Historical overview

A computation is a sequence of discrete operations on some symbols.

At each step, the next operation is determined by the current state of the computation in accordance with a predefined set of rules.

Historical overview

Turing Machines

A Turing machine is an abstract computing device consisting of:

- an infinite tape for reading and writing symbols
- a read/write head that can move along the tape
- a finite set of internal states
- a transition function that determines the next action based on the current state and tape symbol

The Church-Turing Thesis (1936):

Everything that can be computed can be computed by a Turing machine.

Historical overview

Undecidability

Turing machines can run forever.

(Compare: searching through all proofs, if no proof exists.)

The Halting Problem:

Is there an algorithm for determining whether a given Turing machine halts on a given input?

Turing (1936): No Turing machine can solve the Halting Problem.

Corollary: No Turing machine can solve the Entscheidungsproblem.

Historical overview

Recursive Functions

Another approach to defining computability (Gödel, Kleene):

Start with obviously computable functions:

- Successor: $s(x) = x + 1$
- ...

Build complex functions using operations that preserve computability:

- Composition: $f(x) = g(h(x))$
- ...

Theorem: The recursive functions are exactly the Turing-computable functions.

Historical overview

Formal Theories

In the late 19th and early 20th century, rigorous axiomatizations of various mathematical domains were developed.

Vision: All truths of the domain should be derivable from the axioms.

This could still be the case.

Historical overview

Gödel's First Incompleteness Theorem (1931):

No sufficiently powerful and consistent formal theory can prove all truths of its domain.

Historical overview

Gödel's First Incompleteness Theorem

The language of Peano Arithmetic has non-logical symbols for numbers, addition, and multiplication.

Gödel showed how to construct a sentence G in this language that is true but not provable from the axioms of PA.

The sentence G is constructed in such a way that it is true iff it is not provable.

Gödel Numbering

We can code sentences and proofs as numbers.

These numbers are called Gödel numbers.

Properties of sentences and proofs turn into numerical properties of their Gödel numbers.

Example:

- $x_1 \mapsto 2$, $x_2 \mapsto 4$, $x_3 \mapsto 6$, ...
- Being a variable $\mapsto$ being an even number.

'x is even' is expressible in the language of PA by $\exists y(x = y + y)$.

Gödel Numbering

We can code sentences and proofs as numbers.

These numbers are called Gödel numbers.

Properties of sentences and proofs turn into numerical properties of their Gödel numbers.

Theorem:

For any computable property of sentences and proofs, the corresponding property of their Gödel numbers is expressible in the language of PA.

Historical overview

Theorem:

For any computable property of sentences and proofs, the corresponding property of their Gödel numbers is expressible in the language of PA.

There is a mechanical procedure for checking whether a sequence of formulas is a proof of a given sentence.

So:

There is a PA-formula $\mathrm{Prf}(x, y)$ that holds of x, y iff x is the code of a proof of the sentence with code y.

The Diagonal Lemma

For every PA-formula $A(x)$, there is a sentence G such that:

$$G \text{ is true iff } A(\#G),$$

where $\#G$ is the Gödel number of G.

Now apply the diagonal lemma to $\neg\exists y\,\mathrm{Prf}(y, x)$.

This gives us a sentence G such that:

$$G \text{ is true iff } \neg\exists y\,\mathrm{Prf}(y, \#G).$$

Historical overview

Gödel's First Incompleteness Theorem

G is true iff G is not provable in PA.

Suppose G is provable in PA. Then G is false. So PA is can prove a false statement about numbers. But all axioms of PA are true. So G isn't provable in PA.

So G is true. So there is a true statement about numbers that is not provable in PA.

Historical overview

Formal Theories

In the late 19th and early 20th century, rigorous axiomatizations of various mathematical domains were developed.

Vision: All truths of the domain should be derivable from the axioms.

Gödel showed that this vision often can't be realized, provided that the axioms are consistent.

Vision: We can prove that the axioms are consistent.

Historical overview

Gödel's Second Incompleteness Theorem (1931):

No sufficiently powerful and consistent formal theory can prove its own consistency.

Historical overview

1. Propositional logic
2. First-order predicate logic
3. Completeness of first-order logic
4. Formal theories
5. Computability
6. Turing machines
7. Recursive functions
8. Arithmetic representability
9. Gödel's first incompleteness theorem
10. Gödel's second incompleteness theorem