

Mathematical Logic Course Summary: The Big Picture

Math 557, Fall 2025 - A Comprehensive Overview

Introduction

This course presents a journey through mathematical logic, weaving together three fundamental perspectives that give us deep insight into the nature of mathematical reasoning itself. The lectures progress from basic syntax and semantics through profound meta-mathematical results that reveal both the power and the inherent limitations of formal systems. What emerges is a unified picture connecting proof theory, model theory, computability theory, and their stunning intersection in Gödel's incompleteness theorems.

Part I: Foundations - Building the Language of Logic (Lectures 1-5)

The course begins by establishing the formal apparatus needed to reason rigorously about logic itself. We start with **syntax** - the formal rules for constructing well-formed terms and formulas in a first-order language. The readability and unique readability lemmas ensure that formulas have unambiguous structure, which is essential for everything that follows.

The semantic side introduces **structures** (models) that give meaning to syntactic objects. A structure \mathcal{M} consists of a universe M together with interpretations of constants, functions, and relations. The **satisfaction relation** $\mathcal{M} \models \varphi[\alpha]$ recursively defines when a formula φ is true in structure \mathcal{M} under assignment α .

The **substitution lemma** bridges syntax and semantics, showing that syntactic substitution corresponds semantically to changing the assignment. This seemingly technical result is crucial for connecting formal manipulations with their semantic meaning.

Validities - formulas true in all structures under all assignments - emerge as the semantic counterpart to syntactic proof. The fundamental question arises: *Can we characterize validities syntactically?* This question drives much of the first half of the course.

Part II: Proof and Completeness - Bridging Syntax and Semantics (Lectures 5-11)

The course introduces **formal proof systems** with logical axioms (propositional tautologies, equality axioms, quantifier axioms) and inference rules (Modus Ponens, Generalization). We write $T \vdash \varphi$ when φ has a formal proof from theory T , contrasting with the semantic notion $T \models \varphi$ (logical consequence).

The **Soundness Theorem** ($T \vdash \varphi \Rightarrow T \models \varphi$) is relatively straightforward, but the converse is far more subtle. The **Completeness Theorem** - stating that $T \vdash \varphi \iff T \models \varphi$ - is one of the crowning achievements of mathematical logic.

The proof strategy for completeness is constructive and beautiful: Given a consistent theory T , we build a model by:

1. Extending T to a **Henkin theory** T_H by adding witnesses for existential statements
2. Further extending to a complete theory T' using Lindenbaum's lemma
3. Constructing the **term model** $\mathcal{A}_{T'}$, where constant terms are identified under provable equality
4. Showing $\mathcal{A}_{T'} \models T'$ and hence $\mathcal{A}_{T'} \models T$

This construction reveals deep connections: every consistent theory has a model, and if the language is countable, so is the model.

The **Compactness Theorem** - a theory has a model iff every finite subtheory has a model - emerges as a key consequence. Compactness has remarkable applications, from constructing non-standard models of arithmetic to proving that every set can be linearly ordered (without choice!).

Part III: Model Theory - Understanding Mathematical Structures (Lectures 12-17)

The course shifts focus to studying structures themselves and relationships between them. Two structures are **elementarily equivalent** ($\mathcal{M} \equiv \mathcal{N}$) if they satisfy the same sentences. This is weaker than isomorphism but captures the idea that structures are "logically indistinguishable."

Elementary substructures ($\mathcal{M} \preceq \mathcal{N}$) strengthen ordinary substructures by requiring that formulas with parameters from M have the same truth value in both structures. The **Tarski-Vaught test** provides a practical criterion: $\mathcal{M} \preceq \mathcal{N}$ iff whenever \mathcal{N} satisfies $\exists x\psi(x, \vec{a})$ for $\vec{a} \in M$, there exists a witness in M .

The **Löwenheim-Skolem theorems** reveal striking cardinality phenomena: - **Downward**: Every infinite structure has elementary substructures of every smaller infinite cardinality - **Upward**: Every infinite structure has elementary extensions of every larger cardinality

These theorems show that first-order logic cannot pin down infinite cardinalities - a significant limitation. For instance, the complete ordered field \mathbb{R} has countable elementary substructures (like the algebraic numbers), so completeness isn't first-order expressible in the language of ordered rings.

Ultraproducts provide a powerful algebraic tool for constructing models. Given structures \mathcal{M}_i and an ultrafilter \mathcal{U} on index set I , the ultraproduct $\prod_i \mathcal{M}_i / \mathcal{U}$ satisfies a formula iff “most” (in the sense of \mathcal{U}) of the \mathcal{M}_i do. **Łoś’s Theorem** makes this precise, and gives a new proof of compactness.

Part IV: Homogeneous Structures and Fraïssé Theory (Lectures 18-20)

This section studies structures with maximal symmetry. A structure \mathcal{M} is **homogeneous** if every isomorphism between finitely generated substructures extends to an automorphism of \mathcal{M} . The quintessential example is $(\mathbb{Q}, <)$ - the rationals are homogeneous as a linear order.

The **age** of \mathcal{M} consists of all finitely generated structures isomorphic to substructures of \mathcal{M} . Two countable homogeneous structures with the same age are isomorphic (proved via back-and-forth).

Fraïssé’s Theorem characterizes which classes of finite structures arise as ages of homogeneous structures: a countable class \bar{K} is the age of a countable homogeneous structure iff it’s an **amalgamation class** (satisfying hereditary, joint embedding, and amalgamation properties). The proof constructs the homogeneous structure by iteratively ensuring all required properties.

The **random graph** emerges as a beautiful example: adding extension axioms to the theory of simple graphs yields a complete theory whose unique countable model is homogeneous. Remarkably, this structure captures the asymptotic behavior of finite random graphs (the **0-1 law**).

Part V: Computability - Formalizing Effective Procedures (Lectures 21-22, 24-26)

The course takes a computational turn, formalizing “effective computation” via **Turing machines** and **register machines**. These turn out to be equivalent (supporting the **Church-Turing Thesis**: the intuitive notion of computability coincides with formal computability).

Primitive recursive functions are built from initial functions (zero, successor, projections) via composition and primitive recursion. They include all standard arithmetic operations and many number-theoretic functions. The **Ackermann function** shows that computable functions can transcend primitive recursion.

Recursive functions add the μ -operator (unbounded search), yielding the full class of computable functions. Key results: - Not all computable functions are total (halting problem) - There exist non-computable functions (diagonal argument) - Various problems are **undecidable** (e.g., determining if a program halts)

The **arithmetical hierarchy** classifies formulas by quantifier complexity: - Δ_0 (bounded quantifiers): primitive recursive relations - Σ_1 (\exists followed by Δ_0): recursively enumerable sets - Π_1 (\forall followed by Δ_0): complements of r.e. sets - $\Delta_1 = \Sigma_1 \cap \Pi_1$: recursive sets

The **Definability Theorem** establishes that recursive functions are precisely those whose graphs are Σ_1 -definable in \mathbb{N} . The proof requires **Gödel’s β -function lemma**, a clever encoding using the Chinese Remainder Theorem that allows sequences to be coded without exponentiation (crucial since we work in PA^-).

Part VI: Arithmetic and Peano Arithmetic (Lectures 23-28)

The course focuses on **Peano Arithmetic (PA)** - the induction axioms plus basic axioms for successor, addition, and multiplication. While PA captures much of arithmetic, the compactness theorem guarantees **non-standard models** with “infinitely large” numbers.

PA^- (Peano arithmetic without induction) is a finitely axiomatized subtheory corresponding to discretely ordered rings. Though weaker than PA , it's strong enough to prove all true Σ_1 -sentences of arithmetic. This follows from two facts: 1. Δ_0 -formulas are **absolute** for end extensions 2. Every model of PA^- is an end extension of \mathbb{N}

The **Representation Theorem** shows that every recursive function is Σ_1 -representable in PA^- : there exists a formula $\theta(x, y)$ such that $f(n) = m$ implies both $PA^- \vdash \theta(\underline{n}, \underline{m})$ and $PA^- \vdash \exists! y \theta(\underline{n}, y)$. This bridges computation and logic: what's computable is logically representable.

Gödelization assigns natural numbers to formulas, making syntax an object of arithmetic study. The key insight: syntactic properties (being a term, being a formula, being a proof) are primitive recursive, hence Δ_0 -definable.

Part VII: Gödel's Incompleteness Theorems - The Limits of Formalization (Lectures 29-32)

The course culminates in Gödel's profound limitations results. The **Diagonal Lemma** provides the key self-referential mechanism: for any formula $\theta(v)$, there exists a sentence G with $T \vdash G \leftrightarrow \theta(\ulcorner G \urcorner)$. The sentence G "says of itself" that θ holds of its Gödel number.

The **First Incompleteness Theorem**: Any consistent, recursively axiomatizable theory extending PA^- is incomplete. The proof constructs a Π_1 -sentence G that "says" (via Gödelization) " G is not provable in T ." If $T \vdash G$, then G is false (since it says it's unprovable), contradicting soundness. If $T \vdash \neg G$, then G is true but unprovable. For consistent T , neither G nor $\neg G$ is provable.

The theorem has philosophical depth: no consistent formal system extending basic arithmetic can be both complete and recursively axiomatizable. Mathematics transcends any particular formalization.

The **Second Incompleteness Theorem** strengthens this: no consistent theory extending PA can prove its own consistency. The consistency statement Con_T asserts " $0 = 1$ is not provable." Using derivability conditions, one shows that if $T \vdash Con_T$, then T is inconsistent. This means: - PA cannot prove Con_{PA} (assuming PA is consistent) - Hilbert's program for proving consistency of mathematics within mathematics is impossible

Tarski's Undefinability Theorem completes the picture: truth is not definable in arithmetic. There's no formula $\theta(v)$ such that $\mathbb{N} \models \theta(\ulcorner \sigma \urcorner) \iff \mathbb{N} \models \sigma$. Syntax (provability) can be arithmetized, but semantics (truth) cannot.

Finally, **Church's Theorem** shows that the set of valid sentences is undecidable: no algorithm can determine whether an arbitrary first-order sentence is a validity. This follows from the undecidability of PA (or any consistent extension of PA^-).

The Big Picture: Three Perspectives Unite

The course reveals a stunning convergence of three perspectives on logic:

1. Syntactic (Proof-Theoretic): What can be formally proved? - Formal systems with axioms and inference rules - Deductive closure: $T^+ = \{\sigma : T \vdash \sigma\}$ - Gödel shows this is recursively enumerable but (for rich enough theories) undecidable

2. Semantic (Model-Theoretic): What is true in structures? - Structures and satisfaction - Logical consequence: $T \models \sigma$ - Compactness, Löwenheim-Skolem, elementary equivalence reveal the power and limitations of first-order logic

3. Computational: What can be effectively computed? - Turing machines, recursive functions - Representability in arithmetic - The arithmetical hierarchy classifies formulas by computational complexity

The **Completeness Theorem** unifies syntax and semantics: provability = semantic consequence. **Gödel's Theorems** use computability to show limits of syntax and semantics: undecidability enters through Gödelization, connecting the three perspectives.

The course's genius lies in showing these aren't separate subjects but facets of one coherent story. Syntax studies what we can write down and prove. Semantics studies what's true in mathematical structures. Computation studies what's effectively calculable. Logic reveals they're intimately connected, and meta-mathematics - reasoning about logic itself - uncovers profound limitations: incompleteness, undecidability, undefinability.

Key Conceptual Threads

Several themes weave throughout:

Expressiveness and Limitation: First-order logic is powerful (Completeness Theorem, expressiveness for many theories) yet limited (can't pin down infinite cardinalities, can't express completeness of ordered fields). Second-order logic is more expressive but loses completeness.

Construction and Existence: The course repeatedly constructs objects proving existence: term models for completeness, Henkin extensions for witnesses, Fraïssé limits for homogeneous structures, non-standard models via compactness and ultraproducts.

Countability Phenomena: Countable models play a special role (Löwenheim-Skolem, ω -categoricity, back-and-forth arguments). The transition from finite to infinite is crucial.

Self-Reference: The Diagonal Lemma enables sentences to "talk about themselves" through Gödelization, leading to incompleteness and undefinability. Self-reference reveals deep limits.

Hierarchy and Classification: Structures classified by elementary equivalence; formulas by quantifier complexity (arithmetical hierarchy); functions by computability (primitive recursive, recursive). Classification reveals structure in apparent chaos.

Conclusion

This course presents mathematical logic not as a collection of techniques but as a unified investigation into the nature of mathematical reasoning. We begin with basic formal languages and build to the incompleteness theorems - arguably the most philosophically significant mathematical results of the 20th century.

The journey reveals both power and humility: Logic gives us formal languages to express mathematics, proof systems to derive theorems, and model theory to understand structures. But it also shows inherent limitations: incompleteness and undecidability. These aren't defects but deep truths about the nature of formal systems.

The synthesis of proof theory, model theory, and computability theory into a coherent whole represents one of mathematics' great intellectual achievements. Understanding this synthesis - seeing how syntax, semantics, and computation interweave - is the true goal and lasting gift of this course.