

CAS 701 Fall 2005

01 Introduction to Mathematics and Logic

Instructor: W. M. Farmer

Revised: 9 September 2005

Popular View of What Mathematics Is

- Mathematics is a huge body of concepts and facts about such things as time, measure, pattern, space, and logical consequence.
- New concepts and facts are discovered by the definition-theorem-proof process.
- Mathematics is infallible.
 - Old concepts and facts are immutable.
- Conjectures are either proved with a proof or refuted with a counterexample.

Mathematics-as-Process View

Mathematics is a process consisting of three intertwined activities:

1. **Model creation:** Mathematical models representing mathematical aspects of the world are created.
2. **Model exploration:** The models are explored by:
 - a. Stating and proving conjectures.
 - b. Performing computations.
 - c. Creating and studying visual representations.
3. **Model connection:** Models with similar structure are connected to each other to facilitate the creation and exploration of new models.

Proof

- Mathematical proof is an essential component of the mathematics process which is unique to mathematics.
- It is a method of **discovery, communication, and certification**.
- An **informal proof** is a convincing argument that a statement about a model is true.
- A **formal proof** is a logical deduction from a set of premises to a conclusion.
 - Can be mechanically checked.
- A formal proof can be presented in two ways:
 - As a **description** of the actual deduction.
 - As a **prescription** for creating the deduction.

Lakatosian View

- Presented by **Imre Lakatos** in *Proofs and Refutations*, Cambridge University Press, 1976.
- Mathematical reasoning is dialectical.
 - Dialectic between a theory and its theorems.
 - Dialectic between a conjecture and its proof.
- New mathematics is discovered by analyzing the proofs of conjectures according to the **method of proof and refutations**.
- The definition-theorem-proof style of presentation hides the true nature of mathematics.

Mathematics for Software Engineering

Mathematics enters software engineering from three sources:

- Mathematics underlying the concepts and techniques of **computer science**.
 - This is largely **discrete mathematics**.
- Mathematics used to manage the **software development process**.
 - Many of the principal products of software engineering are essentially mathematical models (e.g., requirement specifications, design documents, software descriptions, and software code).
 - This is largely **logic**.
- Mathematics needed to understand **physical devices**.
 - This is largely **continuous mathematics**.

Software Development as Mathematics

- A rigorous software development process is a special case of the mathematics process.
 - Formulation of software requirements and design and implementation of software involves the **creation of mathematical models**.
 - Verification and analysis of software involves the **exploration of mathematical models**.
 - Reuse of software involves the **connection of mathematical models**.
- Managing software documentation is a special case of managing mathematical knowledge.

The Need for Mechanization

- The mathematical models produced by software engineering are often different than traditional mathematical models:
 - They tend to be very large with many details.
 - They are usually produced by teams of developers.
 - Most of the content is mathematically uninteresting.
 - Small mistakes can have a dire effect on such things as **mission, cost, security, and safety**.
- The traditional paper-based mathematical practice is becoming increasingly untenable for software development.
- The mathematics part of the software development process must be mechanized.

What is Logic?

- Study of the principles underlying sound reasoning.
 - Central idea: **logical consequence**.
- Branch of mathematics.
- Makes explicit several fundamental distinctions:
 - **Syntax** vs. **semantics**.
 - **Language** vs. **metalinguage**.
 - **Theory** vs. **model**.
 - **Truth** vs. **proof**.
- Principal tools: formal systems called **logics**.

Syntax vs. Semantics

- The **syntax** of language is concerned with how the expressions of the language are constructed.
 - For example, “the numeral 144 has three digits” is a statement about syntax.
- The **semantics** of a language is concerned with what the expressions of the language mean.
 - For example, “the number 144 is a perfect square” is a statement about semantics.
- This distinction is crucial in mathematics and computing.
 - Confusion between syntax and semantics is the source of many errors.
- Logic carefully disentangles the roles of syntax and semantics in reasoning.

Language vs. Metalanguage

- A **language** is for talking about a certain subject.
- A **metalanguage** for a language L is a language for talking about L itself.
- A natural language, such as English, usually serves as its own metalanguage.
 - As a result, the distinction is not explicit in English.
- A formal language, such as a logical or programming language, usually is not expressive enough to serve as its own metalanguage.
 - A metalanguage of a formal language may be a formal language, but usually it is only informal.

What is a Logic?

- Informally, a logic is a system of reasoning.
- Formally, a **logic** is a family of **formal languages** with:
 1. A common syntax.
 2. A common semantics.
 3. A notion of **logical consequence**.
- A logic may include a **proof system** for **proving** that a given formula is a logical consequence of a given set of formulas.
- Examples:
 - Propositional logic.
 - First-order logic.
 - Simple type theory (higher-order logic).

Language Syntax

- A language defines a collection of **expressions** formed from:
 - **Variables**.
 - **Constants** (nonlogical constants).
 - **Constructors** (logical constants).
- Two kinds of expressions:
 - **Terms**: Denote objects or values.
 - **Formulas**: Make assertions about objects or values.
- Some languages have constructors that bind variables (e.g., \forall , \exists , λ , I , ϵ , $\{ \mid \}$).

Language Semantics

- A **model** M for a language L is a pair (D, V) where:
 1. D is a set of values called the **domain** that includes the truth values t and f .
 2. V is a function from the expressions of L to D called the **valuation function**.
- M **satisfies** a formula A of L , written $M \models A$, if $V(A) = t$.
- M **satisfies** a set Σ of formulas of L , written $M \models \Sigma$, if M satisfies each $A \in \Sigma$.
- A is a **semantic consequence** of Σ , written $\Sigma \models A$, if every model for L that satisfies Σ also satisfies A .
- A is **valid**, written $\models A$, if every model for L satisfies A .
- Σ is **satisfiable** if there exists some model for L that satisfies Σ .

Hilbert-Style Proof Systems

- A **Hilbert-style proof system \mathbf{H}** for a language L consists of:
 1. A set of formulas of L called **logical axioms**.
 2. A set of **rules of inference**.
- A **proof** of A from Σ in \mathbf{H} is a finite sequence B_1, \dots, B_n of formulas of L with $B_n = A$ such that each B_i is either a logical axiom, a member of Σ , or follows from earlier B_j by one of the rules of inference.
- A is **syntactic consequence** of Σ in \mathbf{H} , written $\Sigma \vdash_{\mathbf{H}} A$, if there is a proof of A from Σ in \mathbf{H} .
- A is a **theorem** in \mathbf{H} , written $\vdash_{\mathbf{H}} A$, if there is a proof of A from \emptyset in \mathbf{H} .
- Σ is **consistent** in \mathbf{H} if not every formula is a syntactic consequence of Σ in \mathbf{H} .

Kinds of Proof Systems

- Hilbert style.
- Symmetric sequent (Gentzen).
- Asymmetric sequent.
- Natural deduction (Gentzen, Quine, Fitch, Berry).
- Semantic tableaux (Beth, Hintikka).
- Resolution (J. Robinson).

Soundness and Completeness

- Let \mathbf{P} be a proof system for a language L .
- \mathbf{P} is **sound** if

$$\Sigma \vdash_{\mathbf{P}} A \text{ implies } \Sigma \models A.$$

- \mathbf{P} is **complete** if

$$\Sigma \models A \text{ implies } \Sigma \vdash_{\mathbf{P}} A.$$

Theories

- A **theory** is a pair $T = (L, \Gamma)$ where:
 1. L is a language (the **language** of T).
 2. Γ is a set of formulas of L (the **axioms** of T).
- M is a **model** of T , written $M \models T$, if $M \models \Gamma$.
- A theory can be viewed as a specification of its models.
- A is **valid** in T , written $T \models A$, if $\Gamma \models A$.
- A is a **theorem** of T in \mathbf{P} , written $T \vdash_{\mathbf{P}} A$, if $\Gamma \vdash_{\mathbf{P}} A$.
- T is **satisfiable** if Γ is satisfiable.
- T is **consistent** in \mathbf{P} if Γ is consistent in \mathbf{P} .

Semantic vs. Syntactic Consequence

Semantics	Syntax
semantic consequence	syntactic consequence
A is valid $\models A$	A is a theorem in \mathbf{P} $\vdash_{\mathbf{P}} A$
A is valid in T $T \models A$	A is a theorem of T in \mathbf{P} $T \vdash_{\mathbf{P}} A$
T is satisfiable	T is consistent in \mathbf{P}

- Semantic consequence and syntactic consequence are different forms of **logical consequence**.
- The semantic and syntactic notions are equivalent in the most common logics:
 - Propositional logic.
 - First-order logic (Gödel, 1930).
 - Simple type theory (Henkin, 1950).

Mathematical Problems: Fundamental Form

- Most mathematical problems can be expressed as statements of the form

$$T \models A$$

where T is a theory and A is a formula.

- There are three basic ways of deciding whether or not $T \models A$:
 1. **Model checking**: Show that $M \models A$ for each model M of T .
 2. **Proof**: Show $T \vdash_{\mathbf{P}} A$ for some sound proof system \mathbf{P} .
 3. **Counterexample**: Show $M \models \neg A$ for some model M of T .