

Introduction to Mathematical Logic

Introduction to Mathematical Logic

Key concepts, definitions, and examples for a first-time learner

1. What Is Mathematical Logic?

- **Purpose** – Provides a formal language to express mathematical statements and a systematic way to reason about them.
- **Two main branches**
 1. **Proof Theory** – Studies the structure of mathematical proofs.
 2. **Model Theory** – Studies the interpretation of logical languages in mathematical structures.

2. Formal Languages

Component	Description	Example
Alphabet	Finite set of symbols (variables, logical connectives, quantifiers, parentheses, etc.)	$p, q, r, \neg, \forall, \exists, (,)$
Terms	Built from variables and function symbols; denote objects of the domain.	$x, f(x, y)$
Atomic formulas	Predicates applied to terms; the simplest statements.	$P(x), R(f(a), b)$
Formulas	Constructed from atomic formulas using logical connectives and quantifiers.	$\neg P(x) \vee \exists y R(x, y)$
Sentences	Formulas with no free variables (all variables bound).	$\forall x \exists y R(x, y)$

3. Logical Connectives (Propositional Logic)

Symbol	Name	Truth-functional definition
$\neg$	Negation	True iff ϕ is false
$\phi \wedge \psi$	Conjunction	True iff both ϕ and ψ are true

Introduction to Mathematical Logic

Symbol	Name	Truth-functional definition
$\phi \vee \psi$	Disjunction	True iff at least one of ϕ , ψ is true
$\phi \rightarrow \psi$	Implication	False only when ϕ is true and ψ is false
$\phi \leftrightarrow \psi$	Biconditional	True iff ϕ and ψ have the same truth value

Note: Truth tables give the precise semantics for each connective.

4. Quantifiers (Predicate Logic)

Symbol	Name	Informal reading
$\forall x \phi(x)$	Universal quantifier	"For every x , $\phi(x)$ holds."
$\exists x \phi(x)$	Existential quantifier	"There exists an x such that $\phi(x)$ holds."

Scope – The part of a formula to which a quantifier applies.

Bound vs. free variables – A variable is *bound* if it lies inside the scope of a quantifier; otherwise it is *free*.

5. Semantics: Structures & Interpretations

A **structure** (or *model*) $\mathcal{M}$ for a language L consists of:

1. A non-empty **domain** (D) (the set of objects we talk about).
2. An **interpretation** of each constant symbol as an element of (D).
3. An interpretation of each n -ary function symbol as a function ($D^n \rightarrow D$).
4. An interpretation of each n -ary predicate symbol as a subset of (D^n) (i.e., a relation).

Evaluation – Given a structure $\mathcal{M}$ and a variable assignment (s), we define the truth value of any formula recursively:

- Atomic: ($\mathcal{M} \models P(t_1, \dots, t_n)[s]$) iff ($\llbracket t_1 \rrbracket_s, \dots, \llbracket t_n \rrbracket_s \in P$)