

# Formal Methods in Software Engineering

**First-Order Logic** — Spring 2025

Konstantin Chukharev

# §1 Introduction to FOL

## Motivation

First-order logic (FOL) extends propositional logic by introducing quantifiers and predicates, enabling reasoning about objects, their properties, and relationships. Unlike propositional logic, which is limited to fixed truth values for statements, FOL allows expressions like “All humans are mortal” ( $\forall x. \text{Human}(x) \rightarrow \text{Mortal}(x)$ ) or “There exists a solution to this problem” ( $\exists x. \text{Solution}(x)$ ).

Applications of FOL are diverse, spanning verifying algorithms, formalizing mathematical theories, and modeling knowledge in AI and databases. For instance, “Every student has an ID” translates to  $\forall x. (\text{Student}(x) \rightarrow \text{HasID}(x))$ , ensuring structured reasoning about such properties.

## What is First-Order Logic?

First-order logic provides a structured way to reason about entities and their relationships. It builds upon propositional logic by adding variables, predicates, functions, and quantifiers, allowing complex statements to be expressed.

In comparison to propositional logic, which handles static true/false statements (e.g., “A implies B” as  $A \rightarrow B$ ), FOL enables richer representations such as “If a person is a student, they have an ID” ( $\forall x. (\text{Student}(x) \rightarrow \text{HasID}(x))$ ), with  $x$  representing individuals within a domain.

# Syntax of First-Order Logic

First-order logic defines a formal language consisting of several components.

## □ **Alphabet**

Logical symbols (**not**, **and**, **or**, **imply**, **iff**), quantifiers (**forall**, **exists**), variables ( $x, y, z, \dots$ ), constants ( $a, b, c, \dots$ ), functions ( $f(x), g(x, y), \dots$ ), and predicates ( $P(x), Q(x, y), \dots$ ).

## □ **Terms**

Terms denote objects in the domain and can be constants, variables, or function applications. Examples include  $x$  (variable),  $a$  (constant), and  $f(x, y)$  (function).

## □ **Atomic Formulas**

Atomic formulas are created by applying predicates to terms, such as  $P(x)$  (“ $x$  satisfies property  $P$ ”) or  $Q(f(x), y)$  (“ $f(x)$  and  $y$  are related by  $Q$ ”).

## Well-Formed Formulas (WFFs)

Formulas in FOL are defined inductively.

**Base Case:** Atomic formulas, such as  $P(x)$ , are WFFs.

**Inductive Step:** If  $\alpha$  and  $\beta$  are WFFs, then the following are also WFFs:

- $\neg\alpha$
- $(\alpha \wedge \beta)$
- $(\alpha \vee \beta)$
- $(\alpha \rightarrow \beta)$
- $(\alpha \leftrightarrow \beta)$
- $(\forall x.\alpha)$
- $(\exists x.\alpha)$

**Examples:**  $\forall x.(P(x) \rightarrow Q(x)), \exists y.(R(y) \wedge P(f(a, y)))$ .

## Semantics of First-Order Logic

The semantics of FOL specify how formulas are interpreted.

A **domain** ( $D$ ) represents the set of all objects under consideration. **Variable assignments** map variables to elements in  $D$ , while **interpretations** assign meanings to constants, functions, and predicates.

### **Truth Conditions:**

1.  $P(t_1, \dots, t_n)$  is true if  $(t_1, \dots, t_n)$  is in the interpretation of  $P$ .
2.  $\forall x.\varphi$  is true if  $\varphi$  is true for all assignments  $x \in D$ .
3.  $\exists x.\varphi$  is true if  $\varphi$  is true for some assignments  $x \in D$ .

**Example:** Let  $D = \{1, 2, 3\}$ , and  $P(x)$  mean “ $x$  is even.”

- $\forall x.P(x)$  is false (counter-example:  $x = 3$  is not even).
- $\exists x.P(x)$  is true (example:  $x = 2$  is even).

## Examples

Suppose that  $P$  is a unary predicate, and  $Q$  is a binary predicate. Which of the following are true?

1.  $\forall x.P(x) \models P(y)$
2.  $P(x) \models \forall x.P(x)$
3.  $\forall x.P(x) \models \exists y.P(y)$
4.  $\exists x\forall y.Q(x, y) \models \forall y\exists x.Q(x, y)$
5.  $\forall x\exists y.Q(x, y) \models \exists y\forall x.Q(x, y)$
6.  $\models \exists x.(P(x) \rightarrow \forall y.P(y))$

Which models satisfy the following sentences?

Sentence	Models
$\forall x\forall y.(x = y)$	Models with a single element.
$\forall x\forall y.Q(x, y)$	Models $(A, Q)$ where $Q = A \times A$ .
$\forall x\exists y.Q(x, y)$	Models $(A, Q)$ where $\text{dom } Q = A$ .



## §2 Metatheory

## Logical Theories

A logical theory  $T$  consists of a set of axioms (assumed true formulas) and all formulas derivable from them.

### Examples:

- **Peano Arithmetic:** Defines natural numbers with axioms for addition and multiplication.
- **Group Theory:** Specifies algebraic structures with axioms for identity, inverses, and associativity.
- **Geometry:** Encodes relationships among points, lines, and planes.

A theory is **consistent** if no contradictions are derivable, and **complete** if every formula or its negation is derivable.

## Logical Entailment

Logical entailment ( $\Gamma \models \varphi$ ) means that  $\varphi$  is true in all models of  $\Gamma$ . For instance, if  $\Gamma = \{\forall x.(P(x) \rightarrow Q(x)), P(a)\}$ , then  $\Gamma \models Q(a)$ .

Entailment ensures that semantic truths are provable within a syntactic framework.

## Proof Systems for First-Order Logic

Several formal proof systems are used to reason about FOL:

**Natural Deduction:** Introduces and eliminates quantifiers and connectives. For example, from  $\forall x.P(x)$ , infer  $P(a)$  (universal elimination).

**Sequent Calculus:** Represents arguments as sequents  $(\Gamma \rightarrow \Delta)$ , where  $\Gamma$  entails  $\Delta$ . Example: From  $\Gamma \rightarrow \forall x.P(x)$ , infer  $\Gamma \rightarrow P(a)$ .

**Resolution:** Refutation-based method for proving unsatisfiability. It converts formulas into clausal form and derives contradictions to establish proofs.

## Soundness and Completeness

Soundness ensures that every provable formula is true in all models ( $\Gamma \vdash \varphi \rightarrow \Gamma \models \varphi$ ). Completeness guarantees that every formula true in all models is provable ( $\Gamma \models \varphi \rightarrow \Gamma \vdash \varphi$ ).

The **Compactness Theorem** states that if every finite subset of  $\Gamma$  is satisfiable, then  $\Gamma$  is satisfiable. This is crucial for reasoning about infinite systems.

## Decidability and Complexity

FOL is undecidable; no algorithm can determine the truth of every FOL formula. However, certain fragments, like monadic logic, are decidable. Semi-decidability allows verification of provable statements but not their refutations.

## Theorem Proving and Applications

Automated theorem proving tools like Prover9, Coq, and Lean are widely used in:

- **Verification:** Ensuring correctness of algorithms.
- **Planning:** Reasoning about sequences of actions.
- **AI:** Modeling and querying knowledge bases.

## §3 *Advanced Topics*



# Model Theory

TODO:

- Structure
- Models

# Proof Theory

TODO:

- Natural Deduction
- Sequent Calculus
- Resolution

## §4 Applications

# Modern Applications

TODO:

- Formal Verification
- Knowledge Representation
- Automated Reasoning
- Database Theory
- AI and Machine Learning

## §5 Conclusion

## Summary

First-order logic provides a foundation for reasoning about objects and their relationships. It extends propositional logic with quantifiers, predicates, and functions, enabling diverse applications in mathematics, AI, and formal verification. Understanding FOL is essential for exploring advanced topics like higher-order logic and model checking.