

Formal Methods in Software Engineering

First-Order Logic — Spring 2025

Konstantin Chukharev

§1 Introduction to FOL

Motivation

Propositional logic (PL) is not powerful enough for many applications.

- PL cannot reason about *natural numbers* directly.
- PL cannot reason about *infinite* domains.
- PL cannot express *abstract* properties.
- PL cannot represent the *internal structure* of propositions.
- PL lacks *quantifiers* for generalization.

First-order logic (FOL) *extends* propositional logic by adding *variables*, *predicates*, *functions*, and *quantifiers*, providing a structured way to reason about objects, their properties, and relationships.

Unlike propositional logic, which is limited to fixed truth values for statements, FOL allows complex expressions like “All humans are mortal” ($\forall x. \text{Human}(x) \rightarrow \text{Mortal}(x)$) or “There exists a solution to the problem” ($\exists x. \text{Solution}(x)$).

What is First-Order Logic?

Similar to PL, first-order logic is a formal system with a *syntax* and *semantics*.

First-order logic is an umbrella term for different *first-order languages*.

Syntax of a logic consists of *symbols* and *rules* for combining them into *well-formed formulas* (WFFs).

Symbols of a first-order language are divided into *logical symbols* and *non-logical parameters*.

Logical symbols:

- Parantheses: $(,)$
- Logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
- Variables: x, y, z, \dots
- Quantifiers: \forall and \exists

Parameters:

- Equality: $=$
- Constants: e.g. $\perp, 0, \emptyset, \dots$
- Predicates: e.g. $P(x), Q(x, y), x > y, A \subset B, a \prec ab$
- Functions: e.g. $f(x), g(x, y), x + y, x \oplus y$

Note: For connectives, just \wedge and \neg are enough. The others can be expressed in terms of them.

Note: For quantifiers, just \forall is enough, since $\exists x. \varphi$ can be expressed as $\neg \forall x. \neg \varphi$.

Predicates and Functions

Predicates are used to express properties or relations among objects.

Functions are similar to predicates but return a value, not necessarily a truth value.

Each predicate and function symbol has a fixed *arity* (number of arguments).

- Equality is a special predicate with arity 2.
- Constants can be seen as functions with arity 0.

First-Order Language

First-order language is specified by its *parameters*.

Propositional logic:

- Equality: *no*
- Constants: *none*
- Predicates: A_1, A_2, \dots
- Functions: *none*

Set theory:

- Equality: *yes*
- Constants: \emptyset
- Predicates: \in
- Functions: *none*

Number theory:

- Equality: *yes*
- Constants: 0
- Predicates: $<$
- Functions: S (successor), $+$, \times , \exp

§2 Many-Sorted FOL

Syntax

The *syntax* of a logic consists of *symbols* and *rules* for combining them.

The *symbols* of a first-order language include:

1. Logical symbols: $(,), \neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$
2. Infinite set of variables: x, y, z, \dots
3. Signature $\Sigma = \langle \Sigma^S, \Sigma^F \rangle$, where:
 - Σ^S is a set of *sorts* (also called *types*), e.g. Bool, Int, Real, Set.
 - Σ^F is a set of *function symbols*, e.g. $=, +, <, \dots$

Signatures

Definition 1: Signature $\Sigma = \langle \Sigma^S, \Sigma^F \rangle$ consists of:

- Σ^S is a set of *sorts* (also called *types*), e.g. `Bool`, `Int`, `Real`, `Set`
- Σ^F is a set of *function symbols*, e.g. $=$, $+$, $<$

Definition 2: Each *function symbol* $f \in \Sigma^F$ is associated with an *arity* n (number of arguments) and a *rank*, $(n + 1)$ -tuple of sorts: $\text{rank}(f) = \langle \sigma_1, \sigma_2, \dots, \sigma_{n+1} \rangle$. Intuitively, f denotes a function that takes n values of sorts $\sigma_1, \dots, \sigma_n$ and returns an output of sort σ_{n+1} .

- Functions of arity 0 are called *constants*, which are said to have sort σ if $\text{rank}(f) = \langle \sigma \rangle$.
- Functions that *return* sort `Bool` are called *predicates*.

For every signature $\Sigma = \langle \Sigma^S, \Sigma^F \rangle$, we assume that:

- Σ^S includes a distinguished sort `Bool`.
- Σ^F contains distinguished constants \top and \perp of sort `Bool`, and distinguished predicate symbol \doteq with $\text{rank}(\doteq) = \langle \sigma, \sigma, \text{Bool} \rangle$ for every sort $\sigma \in \Sigma^S$.

Equality

TODO: axioms of equality

- reflexivity
- substitution for functions
- substitution for formulas

First-Order Languages

A first-order language is defined w.r.t. a signature $\Sigma = \langle \Sigma^S, \Sigma^F \rangle$.

Number Theory:

- $\Sigma^S = \{\text{Nat}\} \cup \{\text{Bool}\}$
- $\Sigma^F = \{0, S, <, +, \times\} \cup \{\top, \perp, \dot{=}_{\text{Bool}}, \dot{=}_{\text{Nat}}\}$
 - $\text{rank}(0) = \langle \text{Nat} \rangle$
 - $\text{rank}(S) = \langle \text{Nat}, \text{Nat} \rangle$
 - $\text{rank}(<) = \langle \text{Nat}, \text{Nat}, \text{Bool} \rangle$
 - $\text{rank}(+) = \text{rank}(\times) = \langle \text{Nat}, \text{Nat}, \text{Nat} \rangle$

Set Theory:

- $\Sigma^S = \{\text{Set}\} \cup \{\text{Bool}\}$
- $\Sigma^F = \{\emptyset, \in, \cup, \cap\} \cup \{\top, \perp, \dot{=}_{\text{Bool}}, \dot{=}_{\text{Set}}\}$
 - $\text{rank}(\emptyset) = \langle \text{Set} \rangle$
 - $\text{rank}(\in) = \langle \text{Set}, \text{Set}, \text{Bool} \rangle$
 - $\text{rank}(\cup) = \text{rank}(\cap) = \langle \text{Set}, \text{Set}, \text{Set} \rangle$

Propositional Logic:

- $\Sigma^S = \{\text{Bool}\}$
- $\Sigma^F = \{\neg, \wedge, \vee, \dots, p_1, p_2, \dots\} \cup \{\top, \perp, \dot{=}_{\text{Bool}}\}$
 - $\text{rank}(p_i) = \langle \text{Bool} \rangle$
 - $\text{rank}(\neg) = \langle \text{Bool}, \text{Bool} \rangle$
 - $\text{rank}(\wedge) = \text{rank}(\vee) = \langle \text{Bool}, \text{Bool}, \text{Bool} \rangle$

Arrays Theory:

- $\Sigma^S = \{\text{Array}_{\langle X, Y \rangle}\} \cup \{\text{Bool}\}$
 - X is a sort of *indices*.
 - Y is a sort of *values*.
- $\Sigma^F = \{\text{read}, \text{write}\} \cup \{\top, \perp, \dot{=}_{\text{Bool}}, \dot{=}_{\text{Array}}\}$
 - $\text{rank}(\text{read}) = \langle \text{Array}_{\langle X, Y \rangle}, X, Y \rangle$
 - $\text{rank}(\text{write}) = \langle \text{Array}_{\langle X, Y \rangle}, X, Y, \text{Array}_{\langle X, Y \rangle} \rangle$

Expressions

Definition 3: An *expression* is a finite sequence of symbols.

Examples:

- $\forall x_1 ((< 0 \ x_1) \rightarrow \neg \forall x_2 (< x_1 \ x_2))$
- $x_1 < \forall x_2))$
- $x_1 < x_2 \rightarrow \forall x : \text{Nat}. x > 0$

Note: Most expressions are **not** *well-formed*.

Terms (simple)

Definition 4 (Variables): A set X of Σ -variables, or simply *variables*, is a countable set of variable names, each associated with a sort from Σ^S .

Based on variables, we can build *terms*. Intuitively, terms are expressions that evaluate to values.

Definition 5 (Terms): The Σ -terms over X , or simply *terms*, are defined inductively:

- Each variable x in X is a term of sort σ .
- If $c \in \Sigma^F$ is a constant symbol of sort σ , then c is a term of sort σ .
- If t_1, \dots, t_n are terms of sorts $\sigma_1, \dots, \sigma_n$, and f is a function symbol with $\text{rank}(f) = \langle \sigma_1, \dots, \sigma_n, \sigma \rangle$, then $(f\ t_1 \ \dots\ t_n)$ is a term of sort σ .
- (*Nothing else is a term.*)

Formulas (simple)

Based on terms, we can build *atoms*. Intuitively, atoms are expressions that evaluate Boolean values.

Definition 6 (Atoms): The Σ -atoms over X , or simply *atoms*, are terms of the form $(p\ t_1\ \dots\ t_n)$, where t_1, \dots, t_n are terms of sorts $\sigma_1, \dots, \sigma_n$, and p is a predicate symbol with $\text{rank}(p) = \langle \sigma_1, \dots, \sigma_n, \text{Bool} \rangle$.

In addition to sorted Σ -variables X , also consider *propositional variables* \mathcal{B} .

Definition 7 (Formulas): The Σ -formulas over X and \mathcal{B} , or simply *formulas*, are defined inductively:

- Each propositional variable p in \mathcal{B} is a formula.
- Each Σ -atom over X is a formula.
- If α and β are formulas, then so are $\neg\alpha$, $(\alpha \vee \beta)$, $(\alpha \wedge \beta)$, $(\alpha \rightarrow \beta)$, and $(\alpha \leftrightarrow \beta)$.
- For each variable $x \in X$ and sort $\sigma \in \Sigma^S$, if α is a formula, then so are $\forall x : \sigma. \alpha$ and $\exists x : \sigma. \alpha$.

Terms

A *term* is a *well-formed* S-expression built from function symbols, variables, and parentheses.

Definition 8 (Term): Let \mathcal{B} be the set of all variables and all constant symbols in some signature Σ .

For each function symbol f in Σ^F of arity n , define *term-building operation* \mathcal{T}_f :

$$\mathcal{T}_f(\varepsilon_1, \dots, \varepsilon_n) := (f \ \varepsilon_1 \ \dots \ \varepsilon_n)$$

Well-formed terms are expressions generated from \mathcal{B} by $\mathcal{T} = \{\mathcal{T}_f \mid f \in \Sigma^F\}$.

Examples:

- | | | | |
|---------------------------------|-------------------------|----------------------------|--|
| • $(+ \ x_2 \ (S \ 0))$ ✓ | • $(x_2 + 0)$ ✗ | • $(+ \ x_2 \ \perp)$ ✓ | • $(\text{select } a)$ ✗ |
| • $(S \ (S \ (S \ (S \ 0))))$ ✓ | • $(S \ 0 \ 0)$ ✗ | • $(S \ \perp)$ ✓ | • $(\text{select } a \ i)$ ✓ |
| • $(S \ (0 \ 0))$ ✗ | • $(S \ (< \ 0 \ 0))$ ✓ | • $(\doteq \ 0 \ \perp)$ ✓ | • $(\text{select } (\text{store } a \ i \ x) \ j)$ ✓ |

Well-sortedness

Note: Not all well-formed terms are meaningful. For this, we need to take into account *sorts*.

Definition 9: *Sort system* is a proof system over sequents of the form $\Gamma \vdash t : \sigma$.

- $\Gamma = x_1 : \sigma_1, \dots, x_n : \sigma_n$ is a *sort context*, a set of sorted variables.
- t is a well-formed term.
- σ is a sort from Σ^S .

$$\begin{array}{c} \text{VAR} \frac{x : \sigma \in \Gamma}{\Gamma \vdash x : \sigma} \quad \text{CONST} \frac{c \in \Sigma^F \quad \text{rank}(c) = \langle \sigma \rangle}{\Gamma \vdash c : \sigma} \\[2ex] \text{FUN} \frac{f \in \Sigma^F \quad \text{rank}(f) = \langle \sigma_1, \dots, \sigma_n, \sigma \rangle \quad \Gamma \vdash t_1 : \sigma_1 \quad \dots \quad \Gamma \vdash t_n : \sigma_n}{\Gamma \vdash (f \ t_1 \ \dots \ t_n) : \sigma} \end{array}$$

Definition 10 (Σ -term): A term t is *well-sorted* w.r.t. Σ and *has sort* σ in a sort context Γ if $\Gamma \vdash t : \sigma$ is derivable in the sort system. Term t is called Σ -term.

Examples of Well-sorted Terms

Let $\Sigma^S = \{\text{Nat}\} \cup \{\text{Bool}\}$ and $\Sigma^F = \{0, S, <, +, \times, \dot{=}_{\text{Nat}}\} \cup \{\top, \perp, \dot{=}_{\text{Bool}}\}$.

- $\text{rank}(0) = \langle \text{Nat} \rangle$
- $\text{rank}(S) = \langle \text{Nat}, \text{Nat} \rangle$
- $\text{rank}(<) = \text{rank}(\dot{=}_{\text{Nat}}) = \langle \text{Nat}, \text{Nat}, \text{Bool} \rangle$
- $\text{rank}(+) = \text{rank}(\times) = \langle \text{Nat}, \text{Nat}, \text{Nat} \rangle$

Are these well-formed terms also well-sorted in the context $\Gamma = \{x_1 : \text{Bool}, x_2 : \text{Nat}, x_3 : \text{Nat}\}$?

1. $(+ 0 x_2)$ ✓
2. $(+ (+ 0 x_1) x_2)$ ✗
3. $(S (+ 0 x_5))$ ✓
4. $(< (S x_3) (+ (S 0) x_1))$ ✓
5. $(\dot{=}_{\text{Nat}} (S x_3) (+ (S 0) x_1))$ ✓

Formulas

Definition 11 (Σ -atom): Given a signature Σ , an *atomic Σ -formula*, or simply *atom*, is a Σ -term of sort `Bool` under *some* sort context Γ .

Definition 12 (Formula): *Well-formed formulas* are expressions generated from atoms by the *formula-building operations*, denoted $\mathcal{F} = \{\mathcal{F}_\vee, \mathcal{F}_\wedge, \mathcal{F}_\neg, \mathcal{F}_\rightarrow, \mathcal{F}_{\leftrightarrow}, \mathcal{E}_{x,\sigma}, \mathcal{A}_{x,\sigma}\}$.

- $\mathcal{F}_\vee(\alpha, \beta) := (\alpha \vee \beta)$
- $\mathcal{F}_\wedge(\alpha, \beta) := (\alpha \wedge \beta)$
- $\mathcal{F}_\neg(\alpha) := (\neg\alpha)$
- $\mathcal{F}_\rightarrow(\alpha, \beta) := (\alpha \rightarrow \beta)$
- $\mathcal{F}_{\leftrightarrow}(\alpha, \beta) := (\alpha \leftrightarrow \beta)$
- $\mathcal{E}_{x,\sigma}(\alpha) := (\exists x : \sigma. \alpha)$ for each variable x and sort σ in Σ^S
- $\mathcal{A}_{x,\sigma}(\alpha) := (\forall x : \sigma. \alpha)$ for each variable x and sort σ in Σ^S

Examples of Formulas

Let $\Sigma^S = \{\text{Nat}\}$, $\Sigma^F = \{0, S, <, +, \times, \dot{=}_{\text{Nat}}\}$, and let x_i be variables.

Which of the following formulas are well-formed?

1. $(\dot{=}_{\text{Nat}} 0 (S 0))$ ✓
2. $(< (S x_3) (+ (S 0) x_1))$ ✓
3. $(\dot{=}_{\text{Nat}} (+ x_1 0) x_2)$ ✓
4. $(\dot{=}_{\text{Nat}} (+ x_1 0) x_2) \rightarrow \perp$ ✓
5. $(+ 0 x_3) \wedge (< 0 (S 0))$ ✗
6. $\forall x_3 : \text{Nat}. (+ (+ 0 x_3) x_2)$ ✗
7. $\forall x_3 : \text{Bool}. (\dot{=}_{\text{Nat}} (+ 0 x_3) x_2)$ ✓ (Note: not well-sorted)
8. $\neg \exists x_0 : \text{Nat}. (< 0 x_0 (S 0))$ ✗

Well-sorted Formulas

We *extend* the sort system for terms with rules for the *logical connectives* and *quantifiers*.

$$\begin{array}{c} \text{BCONST} \frac{c \in \{\top, \perp\}}{\Gamma \vdash c : \text{Bool}} \qquad \text{NOT} \frac{\Gamma \vdash \alpha : \text{Bool}}{\Gamma \vdash (\neg \alpha) : \text{Bool}} \\[2ex] \text{CONN} \frac{\Gamma \vdash \alpha : \text{Bool} \quad \Gamma \vdash \beta : \text{Bool} \quad * \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}}{\Gamma \vdash (\alpha * \beta) : \text{Bool}} \\[2ex] \text{QUANT} \frac{\Gamma[x : \sigma] \vdash \alpha : \text{Bool} \quad \sigma \in \Sigma^S \quad Q \in \{\forall, \exists\}}{\Gamma \vdash (Q x : \sigma. \alpha) : \text{Bool}} \end{array}$$

Here, $\Gamma[x : \sigma] = \Gamma \cup \{x : \sigma\}$.

Definition 13 (Σ -formula): A formula φ is a *well-sorted* w.r.t. Σ in a sort context Γ if $\Gamma \vdash \varphi : \text{Bool}$ is derivable in the extended sort system. Formula φ is called Σ -formula.

Free and Bound Variables

A variable x may occur *free* or *bound* in a Σ -formula.

Example: In $\forall x. A(x, y)$, the variable x is said to be *bound* and y is *free*.

Definition 14 (Free and bound variables): Recursive definition of *free* variables in a Σ -formula:

- x occurs free in a Σ -atom $(A \ t_1 \ \dots \ t_n)$ if some t_i contains x .
- x occurs free in $\neg A$ if x occurs free in A .
- x occurs free in $A * B$ if x occurs free in A or in B .
- x occurs free in $Q y : \sigma. A$ if x occurs free in A and x is not y .

If α contains $Q x$, then x is said to be *bound* in α .

Note: A variable can be simultaneously free and bound in a formula. For example, $x \rightarrow \forall x. P(x)$.

Definition 15 (Sentence): A formula α is *closed*, or a *sentence*, if it contains no free variables.

Free and Bound Variables [2]

Definition 16: The set \mathcal{FV} of *free variables* of a Σ -formula α is defined as follows:

$$\mathcal{FV}(\alpha) := \begin{cases} \{x \mid x \text{ is a var in } \alpha\} & \text{if } \alpha \text{ is atomic} \\ \mathcal{FV}(\beta) & \text{if } \alpha \equiv \neg\beta \\ \mathcal{FV}(\beta) \cup \mathcal{FV}(\gamma) & \text{if } \alpha \equiv (\alpha * \beta) \text{ with } * \in \{\wedge, \vee, \rightarrow, \leftrightarrow\} \\ \mathcal{FV}(\beta) \setminus \{v\} & \text{if } \alpha \equiv Qv : \sigma. \beta \text{ with } Q \in \{\forall, \exists\} \end{cases}$$

Example: Let x , y , and z be variables.

- $\mathcal{FV}(x) = \{x\}$ (if x has sort `Bool`)
- $\mathcal{FV}(x < S(0) + y) = \{x, y\}$
- $\mathcal{FV}((x < S(0) + y) \wedge (x \doteq z)) = \mathcal{FV}(x < S(0) + y) \cup \mathcal{FV}(x \doteq z) = \{x, y\} \cup \{x, z\} = \{x, z, y\}$
- $\mathcal{FV}(\forall x : \text{Nat}. x < S(0) + y) = \mathcal{FV}(x < S(0) + y) \setminus \{x\} = \{x, y\} \setminus \{x\} = \{y\}$

Scope

- TODO: scope of variables
- TODO: free/bound variable
- TODO: open/closed formula
- TODO: universal closure
- TODO: existential closure

FOL Semantics

Recall: The *syntax* of a first-order language is defined w.r.t. a signature $\Sigma = \langle \Sigma^S, \Sigma^F \rangle$, where:

- Σ^S is a set of *sorts*.
- Σ^F is a set of *function symbols*.

In PL, the truth of a formula depends on the meaning of its variables.

In FOL, the truth of a Σ -formula depends on:

1. The meaning of each sort $\sigma \in \Sigma^S$ in the formula.
2. The meaning of each function symbol $f \in \Sigma^F$ in the formula.
3. The meaning of each free variable x in the formula.

Semantics

Let α be a Σ -formula and let Γ be a sort context that includes all free variables of α .

The truth of α is determined by *interpretations* \mathcal{J} of Σ and Γ consisting of:

- An interpretation $\sigma^{\mathcal{J}}$ of each $\sigma \in \Sigma^S$, as a non-empty set, the *domain* of σ .
- An interpretation $f^{\mathcal{J}}$ of each $f \in \Sigma^F$ of rank $\langle \sigma_1, \dots, \sigma_n, \sigma_{n+1} \rangle$, as an n -ary total function from $\sigma_1^{\mathcal{J}} \times \dots \times \sigma_n^{\mathcal{J}}$ to $\sigma_{n+1}^{\mathcal{J}}$.
- An interpretation $x^{\mathcal{J}}$ of each $x : \sigma \in \Gamma$, as an element of $\sigma^{\mathcal{J}}$.

Note: We consider only interpretations \mathcal{J} such that

- $\text{Bool}^{\mathcal{J}} = \{\text{true}, \text{false}\}$, $\perp^{\mathcal{J}} = \text{false}$, $\top^{\mathcal{J}} = \text{true}$,
- for all $\sigma \in \Sigma^S$, $\doteq_{\sigma}^{\mathcal{J}}$ maps its two arguments to true iff they are identical.

Semantics: Example

Consider a signature $\Sigma = \langle \Sigma^S, \Sigma^F \rangle$ for a fragment of a set theory with non-set elements (ur-elements):

- $\Sigma^S = \{\text{Elem}, \text{Set}\}$,
- $\Sigma^F = \{\emptyset, \varepsilon\}$ with $\text{rank}(\emptyset) = \langle \text{Set} \rangle$, $\text{rank}(\varepsilon) = \langle \text{Elem}, \text{Set}, \text{Bool} \rangle$,
- $\Gamma = \{e_i : \text{Elem} \mid i \geq 0\} \cup \{s_i : \text{Set} \mid i \geq 0\}$

A possible *interpretation* \mathcal{I} of Σ and Γ :

- $\text{Elem}^{\mathcal{I}} = \mathbb{N}$, the natural numbers.
- $\text{Set}^{\mathcal{I}} = 2^{\mathbb{N}}$, all sets of natural numbers.
- $\emptyset^{\mathcal{I}} = \emptyset$, the empty set.
- For all $n \in \mathbb{N}$ and $S \subseteq \mathbb{N}$, $\varepsilon^{\mathcal{I}}(n, S) \leftrightarrow n \in S$.
- For $i \geq 0$, $e_i^{\mathcal{I}} = i$ and $s_i^{\mathcal{I}} = [0; i] = \{0, 1, \dots, i\}$.

Another *interpretation* \mathcal{J} of Σ and Γ :

- $\text{Elem}^{\mathcal{J}} = \text{Set}^{\mathcal{J}} = \mathbb{N}$.
- $\emptyset^{\mathcal{J}} = 0$.
- For all $m, n \in \mathbb{N}$, $\varepsilon^{\mathcal{J}}(m, n) \leftrightarrow m \mid n$.
- For $i \geq 0$, $e_i^{\mathcal{J}} = i$ and $s_i^{\mathcal{J}} = 2$.

There is a *infinity* of interpretations of Σ, Γ .

Term and Formula Semantics

First, extend \mathcal{I} to an interpretation $\bar{\mathcal{I}}$ for *well-sorted* Σ -terms by structural induction:

$$t^{\bar{\mathcal{I}}} = \begin{cases} t^{\mathcal{I}} & \text{if } t \text{ is a constant or a variable} \\ f^{\mathcal{I}}(t_1^{\bar{\mathcal{I}}}, \dots, t_n^{\bar{\mathcal{I}}}) & \text{if } t \text{ is a term } (f \ t_1 \ \dots \ t_n) \end{cases}$$

Example: Let $\Sigma^S = \{\text{Person}\}$, $\Sigma^F = \{\text{pa}, \text{ma}, \text{sp}\}$, $\Gamma = \{x : \text{Person}, y : \text{Person}\}$, $\text{rank}(\text{pa}) = \text{rank}(\text{ma}) = \langle \text{Person}, \text{Person} \rangle$, $\text{rank}(\text{sp}) = \langle \text{Person}, \text{Person}, \text{Bool} \rangle$.

Let \mathcal{I} be an interpretation of Σ and Γ such that:

- $\text{ma}^{\mathcal{I}} = \{\text{Jim} \mapsto \text{Jill}, \text{Joe} \mapsto \text{Jen}, \dots\}$
- $\text{pa}^{\mathcal{I}} = \{\text{Jim} \mapsto \text{Joe}, \text{Jill} \mapsto \text{Jay}, \dots\}$
- $\text{sp}^{\mathcal{I}} = \{(\text{Jill}, \text{Joe}) \mapsto \text{true}, (\text{Joe}, \text{Jill}) \mapsto \text{true}, (\text{Jill}, \text{Jill}) \mapsto \text{false}, \dots\}$
- $x^{\mathcal{I}} = \text{Jim}$ and $y^{\mathcal{I}} = \text{Joe}$

Then:

Term and Formula Semantics [2]

- $(\text{pa } (\text{ma } x))^{\bar{\mathcal{J}}} = \text{pa}^{\mathcal{J}}((\text{ma } x)^{\bar{\mathcal{J}}}) = \text{pa}^{\mathcal{J}}(\text{ma}^{\mathcal{J}}(x^{\bar{\mathcal{J}}})) = \text{pa}^{\mathcal{J}}(x^{\mathcal{J}})$
 $= \text{pa}^{\mathcal{J}}(\text{ma}^{\mathcal{J}}(\text{Jim})) = \text{pa}^{\mathcal{J}}(\text{Jill}) = \text{Jay}$
- $(\text{sp } (\text{ma } x) \ y)^{\bar{\mathcal{J}}} = \text{sp}^{\mathcal{J}}((\text{ma } x)^{\bar{\mathcal{J}}}, y^{\bar{\mathcal{J}}}) = \text{sp}^{\mathcal{J}}(\text{ma}^{\mathcal{J}}(x^{\bar{\mathcal{J}}}), y^{\mathcal{J}}) = \text{sp}^{\mathcal{J}}(\text{ma}^{\mathcal{J}}(x^{\mathcal{J}}), y^{\mathcal{J}})$
 $= \text{sp}^{\mathcal{J}}(\text{ma}^{\mathcal{J}}(\text{Jim}), \text{Joe}) = \text{sp}^{\mathcal{J}}(\text{Jill}, \text{Joe}) = \text{true}$

Further extend $\bar{\mathcal{J}}$ to *well-sorted non-atomic Σ -formulas* by structural induction:

- $(\neg\alpha)^{\bar{\mathcal{J}}} = \text{true}$ iff $\alpha^{\bar{\mathcal{J}}} = \text{false}$
- $(\alpha \wedge \beta)^{\bar{\mathcal{J}}} = \text{true}$ iff $\alpha^{\bar{\mathcal{J}}} = \text{true}$ and $\beta^{\bar{\mathcal{J}}} = \text{true}$
- $(\alpha \vee \beta)^{\bar{\mathcal{J}}} = \text{true}$ iff $\alpha^{\bar{\mathcal{J}}} = \text{true}$ or $\beta^{\bar{\mathcal{J}}} = \text{true}$
- $(\alpha \rightarrow \beta)^{\bar{\mathcal{J}}} = \text{true}$ iff $\alpha^{\bar{\mathcal{J}}} = \text{false}$ or $\beta^{\bar{\mathcal{J}}} = \text{true}$
- $(\alpha \leftrightarrow \beta)^{\bar{\mathcal{J}}} = \text{true}$ iff $\alpha^{\bar{\mathcal{J}}} = \beta^{\bar{\mathcal{J}}}$
- $(\exists x : \sigma. \alpha)^{\bar{\mathcal{J}}} = \text{true}$ iff $\alpha^{\bar{\mathcal{J}}[x \mapsto c]} = \text{true}$ for some $c \in \sigma^{\mathcal{J}}$
- $(\forall x : \sigma. \alpha)^{\bar{\mathcal{J}}} = \text{true}$ iff $\alpha^{\bar{\mathcal{J}}[x \mapsto c]} = \text{true}$ for all $c \in \sigma^{\mathcal{J}}$

Here, $\bar{\mathcal{J}}[x \mapsto c]$ denotes the interpretation that maps x to c and is otherwise identical to $\bar{\mathcal{J}}$.

Satisfiability, Entailment, Validity

We write $\mathcal{I} \models \alpha$ to denote “ \mathcal{I} satisfies α ” and mean $\alpha^{\overline{\mathcal{I}}} = \text{true}$.

We write $\mathcal{I} \not\models \alpha$ to denote “ \mathcal{I} falsifies α ” and mean $\alpha^{\overline{\mathcal{I}}} = \text{false}$.

Let Φ be a set of Σ -formulas. We write $\mathcal{I} \models \Phi$ to mean that $\mathcal{I} \models \alpha$ for every $\alpha \in \Phi$.

If Φ is a set of Σ -formulas and α is a Σ -formula, then Φ *entails* or *logically implies* α , denoted $\Phi \models \alpha$, if $\mathcal{I} \models \alpha$ for every interpretation \mathcal{I} of Σ such that $\mathcal{I} \models \Phi$.

We write $\alpha \models \beta$ as an abbreviation for $\{\alpha\} \models \beta$.

α and β are *logically equivalent*, denoted $\alpha \equiv \beta$, iff $\alpha \models \beta$ and $\beta \models \alpha$.

A Σ -formula α is *valid*, denoted $\models \alpha$, if $\emptyset \models \alpha$ iff $\mathcal{I} \models \alpha$ for every interpretation \mathcal{I} of Σ .

Example: Let $\Sigma^S = \{A\}$, $\Sigma^F = \{p, q\}$, $\text{rank}(p) = \langle A, \text{Bool} \rangle$, $\text{rank}(q) = \langle A, A, \text{Bool} \rangle$, and all variables v_i have sort A. Do the following entailments hold?

1. $\forall v_1. p(v_1) \models p(v_2)$ ✓
2. $p(v_1) \models \forall v_1. p(v_1)$ ✗

Satisfiability, Entailment, Validity [2]

- 3. $\forall p(v_1) \models \exists v_2. p(v_2)$ ✓
- 4. $\exists v_2 \forall v_1. q(v_1, v_2) \models \forall v_1 \exists v_2. q(v_1, v_2)$ ✓
- 5. $\forall v_1 \exists v_2. q(v_1, v_2) \models \exists v_2 \forall v_1. q(v_1, v_2)$ ✗
- 6. $\models \exists v_1. (p(v_1) \rightarrow \forall v_2. p(v_2))$ ✓

Exercise

Let α be a Σ -formula and let Γ be a sort context that includes all free variables of α .

Consider the signature where $\Sigma^S = \{\sigma\}$, $\Sigma^F = \{Q, \dot{=}_\sigma\}$, $\Gamma = \{x : \sigma, y : \sigma\}$, $\text{rank}(Q) = \langle \sigma, \sigma, \text{Bool} \rangle$.

For each of the following Σ -formulas, describe an interpretation that satisfies it.

1. $\forall x : \sigma. \forall y : \sigma. x \dot{=} y$
2. $\forall x : \sigma. \forall y : \sigma. Q(x, y)$
3. $\forall x : \sigma. \exists y : \sigma. Q(x, y)$

From English to FOL

1. There is a natural number that is smaller than any *other* natural number.

$$\exists x : \text{Nat. } \forall y : \text{Nat. } (x \doteq y) \vee (x < y)$$

2. For every natural number there is a greater one.

$$\forall x : \text{Nat. } \exists y : \text{Nat. } (x < y)$$

3. Two natural numbers are equal only if their respective successors are equal.

$$\forall x : \text{Nat. } \forall y : \text{Nat. } (x \doteq y) \rightarrow (S(x) \doteq S(y))$$

4. Two natural numbers are equal if their respective successors are equal.

$$\forall x : \text{Nat. } \forall y : \text{Nat. } (S(x) \doteq S(y)) \rightarrow (x \doteq y)$$

5. No two distinct natural number have the same successor.

$$\forall x : \text{Nat. } \forall y : \text{Nat. } \neg(x \doteq y) \rightarrow \neg(S(x) \doteq S(y))$$

6. There are at least two natural number smaller than 3.

$$\exists x : \text{Nat. } \exists y : \text{Nat. } \neg(x \doteq y) \wedge (x < S(S(S(0)))) \wedge (y < S(S(S(0))))$$

7. There is no largest natural number.

$$\neg \exists x : \text{Nat. } \forall y : \text{Nat. } (y \doteq x) \vee (y < x)$$

From English to FOL [2]

1. Everyone has a father and a mother.

$$\forall x : \text{Person}. \exists y : \text{Person}. \exists z : \text{Person}. (y \doteq \text{pa}(x)) \wedge (z \doteq \text{ma}(x))$$

2. The marriage relation is symmetric.

$$\forall x : \text{Person}. \forall y : \text{Person}. \text{sp}(x, y) \rightarrow \text{sp}(y, x)$$

3. No one can be married to themselves.

$$\forall x : \text{Person}. \neg \text{sp}(x, x)$$

4. Not all people are married.

$$\neg \forall x : \text{Person}. \exists y : \text{Person}. \text{sp}(x, y)$$

5. Some people have a father and a mother who are not married to each other.

$$\exists x : \text{Person}. \neg \text{sp}(\text{ma}(x), \text{pa}(x))$$

6. You cannot marry more than one person.

$$\forall x : \text{Person}. \forall y : \text{Person}. \forall z : \text{Person}. (\text{sp}(x, y) \wedge \text{sp}(x, z)) \rightarrow (y \doteq z)$$

7. Some people are not mothers.

$$\exists x : \text{Person}. \forall y : \text{Person}. \neg (x \doteq \text{ma}(y))$$

From English to FOL [3]

8. Nobody can be both a farther and a mother.

$$\forall x : \text{Person}. \neg \exists y : \text{Person}. \neg \exists z : \text{Person}. (x \doteq \text{pa}(y)) \wedge (x \doteq \text{ma}(z))$$

9. Nobody can be their own or farther's farther.

$$\forall x : \text{Person}. \neg ((x \doteq \text{pa}(x)) \vee (x \doteq \text{pa}(\text{pa}(x))))$$

10. Some people do not have children.

$$\exists x : \text{Person}. \forall y : \text{Person}. \neg (y \doteq \text{pa}(x)) \wedge \neg (y \doteq \text{ma}(y))$$

Invariance of Term Values

Consider a signature Σ , a sort context Γ , and two interpretations \mathcal{I} and \mathcal{J} that agree on the sorts and symbols of Σ .

Theorem 1: If \mathcal{I} and \mathcal{J} also agree on the variables of a Σ -term t , then $t^{\overline{\mathcal{I}}} = t^{\overline{\mathcal{J}}}$.

Proof: By structural induction on t .

- If t is a variable or a constant, then $t^{\overline{\mathcal{I}}} = t^{\mathcal{I}}$ and $t^{\mathcal{J}} = t^{\overline{\mathcal{J}}}$. Since $t^{\mathcal{I}} = t^{\mathcal{J}}$ by assumption, we have $t^{\overline{\mathcal{I}}} = t^{\overline{\mathcal{J}}}$.
- If t is a term $(f\ t_1 \ \dots\ t_n)$ with $n > 1$, then $f^{\mathcal{I}} = f^{\mathcal{J}}$ by assumption and $t_i^{\overline{\mathcal{I}}} = t_i^{\overline{\mathcal{J}}}$ for $j \geq 1$ by induction hypothesis. It follows, $t^{\overline{\mathcal{I}}} = f^{\mathcal{I}}(t_1^{\overline{\mathcal{I}}}, \dots, t_n^{\overline{\mathcal{I}}}) = f^{\mathcal{J}}(t_1^{\overline{\mathcal{J}}}, \dots, t_n^{\overline{\mathcal{J}}}) = t^{\overline{\mathcal{J}}}$.

□

Invariance of Truth Values

Theorem 2: If \mathcal{I} and \mathcal{J} also agree on the *free* variables of a Σ -formula α , then $\alpha^{\overline{\mathcal{I}}} = \alpha^{\overline{\mathcal{J}}}$.

Proof: By induction on α .

- If α is an atomic formula, the result follows from the previous lemma, since α is a term and all of its variables are free in it.
- If α is $\neg\beta$ or $\alpha_1 * \alpha_2$ with $*$ $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, the result follows from the induction hypothesis.
- If α is $Qx : \sigma. \beta$ with $Q \in \{\forall, \exists\}$, then $\mathcal{FV}(\beta) = \mathcal{FV}(\alpha) \cup \{x\}$. For any c in $\sigma^{\mathcal{J}}$, $\mathcal{I}[x \mapsto c]$ and $\mathcal{J}[x \mapsto c]$ agree on x by construction and on $\mathcal{FV}(\alpha)$ by assumption. The result follows from the induction hypothesis and the semantics of \forall and \exists .

□

Corollary 2.1: The truth value of Σ -sentences is independent from how the variables are interpreted.

Deduction Theorem of FOL

Theorem 3: For all Σ -formulas α and β , we have $\alpha \models \beta$ iff $\alpha \rightarrow \beta$ is valid.

Proof (\Rightarrow): If $\alpha \models \beta$ then $\alpha \rightarrow \beta$ is valid.

Let \mathcal{I} be an interpretation and let $\gamma := \alpha \rightarrow \beta$.

- If \mathcal{I} falsifies α , then it trivially satisfies γ .
- If \mathcal{I} satisfies α , then, since $\alpha \models \beta$, it must also satisfy β . Hence, it satisfies γ .

In both cases, \mathcal{I} satisfies γ , thus $\mathcal{I} \models \alpha \rightarrow \beta$ for every interpretation \mathcal{I} . □

Proof (\Leftarrow): If $\alpha \rightarrow \beta$ is valid then $\alpha \models \beta$.

Let \mathcal{I} be an interpretation that satisfies α . If \mathcal{I} falsifies β , then it must also falsify $\alpha \rightarrow \beta$, contradicting the assumption that $\alpha \rightarrow \beta$ is valid. Thus, every interpretation \mathcal{I} that satisfies α also satisfies β . □

Corollary 3.1: For all Σ -formulas α and β , we have $\models \alpha \rightarrow \beta$ iff $\alpha \rightarrow \beta$ is valid.

Free Variables Theorem 1

Theorem 4: Consider a signature Σ and a sort context Γ . Let Φ be a set of Σ -formulas, let α be a Σ -formula with free variables from Γ , and let $x \in \mathcal{FV}(\alpha)$ be a free variable of sort σ in Γ .

Suppose x occurs free in no formulas of Φ . Then, $\Phi \models \alpha$ iff $\Phi \models \forall x : \sigma. \alpha$.

Proof (\Rightarrow): If $\Phi \models \alpha$ then $\Phi \models \forall x : \sigma. \alpha$.

Let \mathcal{I} be an interpretation that satisfies Φ . Since x does not occur free in any formulas of Φ , \mathcal{I} satisfies all formulas in Φ regardless of how it interprets x . Then, for any element $c \in \sigma^{\mathcal{I}}$, the interpretation $\mathcal{I}[x \mapsto c]$ satisfies all formulas in Φ , including α (because $\Phi \models \alpha$). Thus, it also satisfies the formula $\forall x : \sigma. \alpha$, by the semantics of \forall . Hence, every interpretation that satisfies Φ also satisfies $\forall x : \sigma. \alpha$, that is, $\Phi \models \forall x : \sigma. \alpha$. \square

Proof (\Leftarrow): If $\Phi \models \forall x : \sigma. \alpha$ then $\Phi \models \alpha$.

Let \mathcal{I} be an interpretation that satisfies Φ . By assumption, $\mathcal{I} \models \forall x : \sigma. \alpha$. This implies that $\mathcal{I} \models \alpha$ regardless of how \mathcal{I} interprets x . Hence, $\Phi \models \alpha$. \square

Free Variables Theorem 2

Theorem 5: Consider a signature Σ and a sort context Γ . Let β be a Σ -formula, let α be a Σ -formula with free variables from Γ , and let $x \in \mathcal{FV}(\alpha)$ be a free variable of sort σ in Γ .

Suppose x does not occur free in β . Then, $\alpha \models \beta$ iff $\exists x : \sigma. \alpha \models \beta$.

Proof (\Rightarrow): If $\alpha \models \beta$ then $\exists x : \sigma. \alpha \models \beta$.

Let \mathcal{I} be an interpretation that satisfies $\exists x : \sigma. \alpha$. This means that $\mathcal{I}[x \mapsto c]$ satisfies α for some $c \in \sigma^{\mathcal{I}}$. By assumption, $\mathcal{I}[x \mapsto c]$ satisfies β as well. Since x does not occur free in β , changing the value assigned to x does not matter. It follows that \mathcal{I} satisfies β . Since \mathcal{I} was arbitrary, this shows that $\exists x : \sigma. \alpha \models \beta$. \square

Proof (\Leftarrow): If $\exists x : \sigma. \alpha \models \beta$ then $\alpha \models \beta$.

Let \mathcal{I} be an interpretation that satisfies α . Then, trivially¹, \mathcal{I} satisfies $\exists x : \sigma. \alpha$. By assumption, $\mathcal{I} \models \beta$. Since \mathcal{I} was arbitrary, $\alpha \models \beta$. \square

¹Recall that any domain $\sigma^{\mathcal{I}}$ is non-empty.

§3 Model Checking

Semantics in FOL

Definition 17 (Vocabulary): A *vocabulary* (also known as *signature*) of a language is a collection of symbols used to construct sentences in that language. A vocabulary $\mathcal{V} = \langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ consists of:

- A set of *constant symbols* \mathcal{C} , e.g., $0, \perp, \emptyset, \dots$
- A set of *function symbols* \mathcal{F} , e.g., $S, +, \times, \dots$
- A set of *predicate symbols* \mathcal{P} , e.g., $=, <, \in, \dots$

Definition 18 (Model): A *possible world* (also known as *model*, or *structure*, or *interpretation*) is a mathematical object that gives meaning to the symbols in a vocabulary. A model \mathcal{M} for \mathcal{V} consists of:

- A *domain* $\mathcal{D} = \text{dom}(\mathcal{M})$, which is a non-empty set of objects (*universe*).
- For each constant symbol c in \mathcal{C} , its interpretation $c^{\mathcal{M}}$ is an element of \mathcal{D} .
- For each k -ary function symbol f in \mathcal{F} , its interpretation $f^{\mathcal{M}}$ is a k -ary total function on \mathcal{D} .
- For each k -ary predicate symbol p in \mathcal{P} , its interpretation $p^{\mathcal{M}}$ is a k -ary relation on \mathcal{D} .

Semantics in FOL [2]

Example: Let $\mathcal{V}_{\text{field}}$ consist of 2-ary functions $+$ and \times , constants 0 and 1 , and 2-ary predicates $=$ and $<$.

One possible interpretation \mathcal{M} is:

- $\mathcal{D} = \mathbb{Z}$, the integer numbers. $0^{\mathcal{M}}$ and $1^{\mathcal{M}}$ are the usual integers zero and one.
- $+^{\mathcal{M}}, \times^{\mathcal{M}}, =^{\mathcal{M}}, <^{\mathcal{M}}$ are the usual arithmetic operators on integers.

Alternatively, $\mathcal{D} = \mathbb{R}$, $0^{\mathcal{M}} := 3.14159$, $1^{\mathcal{M}} := 42$. There are infinitely many possible interpretations!

Predicate Logic Statements

Definition 19 (Formula): A *formula* is a statement about the objects in a world.

Example: x is a even number.

Definition 20 (Sentence): A *sentence* is a statement about a world.

Example: Every even natural number greater than 2 is a sum of two prime numbers.

Some Computational Challenges

Decision problems:

1. Given a sentence, is it true in every world? (Validity)
2. Given a sentence, is it true in some world? (Satisfiability)
3. Given a sentence and a world, is it true in that world? (Model checking)

FOL and Computation

First basic computational problem in predicate logic is *Model Checking*.

Definition 21: Model checking problem for first-order logic is the problem of determining whether a given first-order formula φ is satisfied by a given structure \mathcal{M} , formally, $\mathcal{M} \models \varphi$.

$$\text{MCFO} = \{\langle \mathcal{M}, \varphi \rangle \mid \mathcal{M} \models \varphi\}$$

Note: MCFO is decidable in $|\mathcal{M}|^{|\varphi|}$ time.

FOL and Computation [2]

Theorem 6: MCFO is NP-hard.

Proof: SAT can be reduced to MCFO.

Let φ be a propositional formula. For example, $\varphi := (p_1 \vee \neg p_2 \vee p_3) \wedge \dots$

Construct a model \mathcal{M} with interpretations for true and false, and a first-order formula φ' :

$$\varphi' := \exists x_1, \dots, x_n. ((x_1 = \text{true}) \vee (\neg x_2 = \text{true}) \vee (x_3 = \text{true})) \wedge \dots$$

Then, φ is satisfiable iff $\mathcal{M} \models \varphi'$.

□

FOL and Computation

Second basic computational problem in predicate logic is *Finite Satisfiability Problem*.

Given a sentence φ , is it finitely satisfiable? That is, does there exist a *finite* model \mathcal{M} such that $\mathcal{M} \models \varphi$?

FSP is semi-decidable, undecidable.

Corollary: Finite Validity is not RE, but in co-RE.

More:

- set of validities in FOL is in RE
- FOL satisfiability is undecidable

TODO: Exercises

- Parse tree of a FOL formula
- Describe Euclidean geometry as a FOL theory