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 cannor reason about *infinite* domains.
- PL cannot express *abstract* properties.
- PL cannot respresent 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" ($\forall x.\, \text{Human}(x) \to \text{Mortal}(x)$) 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: $x, y, z, \ldots$
- Quantifiers: $\forall$ and $\exists$

Parameters:

- Equiality: $=$
- Constants: e.g. $\bot, 0, \emptyset, \ldots$
- 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:
- Equiality: *no*
- Constants: *none*
- Predicates: A_1, A_2, ...
- Functions: *none*

Set theory:
- Equiality: *yes*
- Constants: $\emptyset$
- Predicates: $\in$
- Functions: *none*

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

§2 Formalizing FOL

Syntax

Definition 1 (Signature): A *vocabulary* (also known as *signature*) of a language is a collection of symbols used to construct sentences in that language. A signature $\Sigma = \langle \mathcal{V}, \mathcal{F}, \mathcal{R} \rangle$ consists of:

- A set of *variables* $\mathcal{V}$, e.g., x, y, z, ...
- A set of *function symbols* $\mathcal{F}$, e.g., S, $+$, $\times$, ...
- A set of *relation symbols* $\mathcal{R}$, e.g., $=$, $<$, $\in$, ...

Each function and relation symbol has an associated *arity* (number of arguments).

- Functions of arity 0 are called *constants*.
- Relations of arity 1 are called *predicates*.

Statements

Definition 2 (Term): A *first-order term* over Σ is defined inductively:

- Each variable $x \in \mathcal{V}$ is a term.
- If $t_1, ..., t_n$ are terms and f is an n-ary function symbol from $\mathcal{F}$, then $f(t_1, ..., t_n)$ is a term.

Definition 3 (Atom): A *first-order atom* over Σ is defined as follows:

- If $t_1, ..., t_n$ are terms and R is a relation symbol from $\mathcal{R}$, then $R(t_1, ..., t_n)$ is an atom.

Definition 4 (Formula): A *first-order formula* over Σ is defined inductively:

- Each atom is a formula.
- If α and β are formulas, then $\neg\alpha$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \rightarrow \beta)$, and $(\alpha \leftrightarrow \beta)$ are formulas.
- If α is a formula and $x \in \mathcal{V}$ is a variable, then $\forall x.\, \alpha$ and $\exists x.\, \alpha$ are formulas.

Free and Bound Variables

- A variable x is *bound* in $\forall x.\, \alpha$ and $\exists x.\, \alpha$.
- A variable x is *free* in α if it is not bound in α.
- A formula is *closed* (also called a *sentence*) if it contains no free variables.

Grammar

$$\langle \text{Form} \rangle ::= \langle \text{Atom} \rangle \mid \neg \langle \text{Form} \rangle \mid \langle \text{Form} \rangle \wedge \langle \text{Form} \rangle \mid \ldots \mid (\text{'}\forall\text{'} \mid \text{'}\exists\text{'}) \mathcal{V}. \langle \text{Form} \rangle$$

$$\langle \text{Atom} \rangle ::= \mathcal{R} \text{ '(' } \langle \text{Term} \rangle^* \text{ ')'}$$

$$\langle \text{Term} \rangle ::= \mathcal{V} \mid \langle \text{Function} \rangle \text{ '(' } \langle \text{Term} \rangle^* \text{ ')'}$$

Semantics

Definition 5 (Model): A *possible world* (also known as *model*, or *structure*, or *interpretation*) is a mathematical object that gives meaning to the symbols of a language.

A *first-order model* $\mathcal{M} = \langle \mathcal{U}, \nu, \mathcal{I} \rangle$ for $\Sigma = \langle \mathcal{V}, \mathcal{F}, \mathcal{R} \rangle$ consists of:

- A *domain* $\mathcal{U}$ is a non-empty set of objects (*universe of disclosure*).
- A *variable valuation* $\nu : \mathcal{V} \longrightarrow \mathcal{U}$ assigning to each variable $x \in \mathcal{V}$ an element of $\mathcal{U}$.
- An *interpretation* function $\mathcal{I}$ for the function and relation symbols in Σ.
  - An *interpretation* of an n-ary function symbol $f \in \mathcal{F}$ is an n-ary total function $\mathcal{I}(f) : \mathcal{U}^n \longrightarrow \mathcal{U}$.
  - An *interpretation* of an n-ary relation symbol $R \in \mathcal{R}$ is an n-ary relation $\mathcal{I}(R) \subseteq \mathcal{U}^n$.

Definition 6: The *term valuation* induced by a model $\mathcal{M} = \langle \mathcal{U}, \nu, \mathcal{I} \rangle$ is defined as follows:

- $\nu(t) = \nu(x)$ if t is a variable x.
- $\nu(t) = (\mathcal{I}(f))(\nu(t_1), ..., \nu(t_n))$ if t is a term $f(t_1, ..., t_n)$.

Semantics [2]

Definition 7 (Semantics of FOL): The *validation relation* $\vDash$ between a model $\mathcal{M}$ and a first-order formula φ is defined inductively:

- $\mathcal{M} \vDash R(t_1, ..., t_n)$ iff $\mathcal{I}(R)$ contains $(\nu(t_1), ..., \nu(t_n))$.
- $\mathcal{M} \nvDash \bot$.
- $\mathcal{M} \vDash \neg\varphi$ iff $\mathcal{M} \nvDash \varphi$.
- $\mathcal{M} \vDash (\alpha \wedge \beta)$ iff $\mathcal{M} \vDash \alpha$ and $\mathcal{M} \vDash \beta$. Similar for $\vee$, $\rightarrow$, and $\leftrightarrow$.
- $\mathcal{M} \vDash \exists x.\, \varphi$ iff $\mathcal{M}' \vDash \varphi$ for some $\mathcal{M}' = \langle \mathcal{U}, \mathcal{I}, \nu' \rangle$ with $\nu'(y) = \nu(y)$ for all $y \neq x$.
- $\mathcal{M} \vDash \forall x.\, \varphi$ iff $\mathcal{M}' \vDash \varphi$ for all $\mathcal{M}'$ which differ from $\mathcal{M}$ at most in the valuation of x.

§3 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, ...
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. $=$, $+$, $<$, ...

Signatures

Definition 8: 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 9: Each *function symbol* $f \in \Sigma^F$ is associated with an *arity* n (number of arguments) and a *rank*, $(n+1)$-tuple of sorts: $\text{rank}(f) = \langle \sigma_1, \sigma_2, ..., \sigma_{n+1} \rangle$. Intuitively, f denotes a function that takes n values of sorts $\sigma_1, ..., \sigma_n$ 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 $\bot$ of sort `Bool`, and distinguished predicate symbol $\doteq$ with $\text{rank}(\doteq) = \langle \sigma, \sigma, \texttt{Bool} \rangle$ 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:

- $\Sigma^S = \{\mathtt{Nat}\} \cup \{\mathtt{Bool}\}$
- $\Sigma^F = \{0, S, <, +, \times\} \cup \{\top, \bot, \doteq_{\mathtt{Bool}}, \doteq_{\mathtt{Nat}}\}$
  - $\mathrm{rank}(0) = \langle \mathtt{Nat} \rangle$
  - $\mathrm{rank}(S) = \langle \mathtt{Nat}, \mathtt{Nat} \rangle$
  - $\mathrm{rank}(<) = \langle \mathtt{Nat}, \mathtt{Nat}, \mathtt{Bool} \rangle$
  - $\mathrm{rank}(+) = \mathrm{rank}(\times) = \langle \mathtt{Nat}, \mathtt{Nat}, \mathtt{Nat} \rangle$

Set Theory:

- $\Sigma^S = \{\mathtt{Set}\} \cup \{\mathtt{Bool}\}$
- $\Sigma^F = \{\emptyset, \in, \cup, \cap\} \cup \{\top, \bot, \doteq_{\mathtt{Bool}}, \doteq_{\mathtt{Set}}\}$
  - $\mathrm{rank}(\emptyset) = \langle \mathtt{Set} \rangle$
  - $\mathrm{rank}(\in) = \langle \mathtt{Set}, \mathtt{Set}, \mathtt{Bool} \rangle$
  - $\mathrm{rank}(\cup) = \mathrm{rank}(\cap) = \langle \mathtt{Set}, \mathtt{Set}, \mathtt{Set} \rangle$

Propositional Logic:

- $\Sigma^S = \{\mathtt{Bool}\}$
- $\Sigma^F = \{\neg, \wedge, \vee, ..., p_1, p_2, ...\} \cup \{\top, \bot, \doteq_{\mathtt{Bool}}\}$
  - $\mathrm{rank}(p_i) = \langle \mathtt{Bool} \rangle$
  - $\mathrm{rank}(\neg) = \langle \mathtt{Bool}, \mathtt{Bool} \rangle$
  - $\mathrm{rank}(\wedge) = \mathrm{rank}(\vee) = \langle \mathtt{Bool}, \mathtt{Bool}, \mathtt{Bool} \rangle$

Arrays Theory:

- $\Sigma^S = \left\{\mathtt{Array}_{\langle \mathtt{X},\mathtt{Y} \rangle}\right\} \cup \{\mathtt{Bool}\}$
  - $\mathtt{X}$ is a sort of *indices*.
  - $\mathtt{Y}$ is a sort of *values*.
- $\Sigma^F = \{\mathrm{read}, \mathrm{write}\} \cup \{\top, \bot, \doteq_{\mathtt{Bool}}, \doteq_{\mathtt{Array}}\}$
  - $\mathrm{rank}(\mathrm{read}) = \langle \mathtt{Array}_{\langle \mathtt{X},\mathtt{Y} \rangle}, \mathtt{X}, \mathtt{Y} \rangle$
  - $\mathrm{rank}(\mathrm{write}) = \langle \mathtt{Array}_{\langle \mathtt{X},\mathtt{Y} \rangle}, \mathtt{X}, \mathtt{Y}, \mathtt{Array}_{\langle \mathtt{X},\mathtt{Y} \rangle} \rangle$

Expressions

Definition 10: An *expression* is a finite sequence of symbols.

Examples:

- $\forall x_1((< 0\ x_1) \to \neg\forall x_2(< x_1\ x_2))$
- $x_1 < \forall x_2))$
- $x_1 < x_2 \to \forall x : \texttt{Nat}.\, x > 0$

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

Terms

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

Definition 11 (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* $\mathcal{T}_f$:

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

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

Examples:

- $(+\ x_2\ (S\ 0))$ ✓
- $(S\ (S\ (S\ (S\ 0))))$ ✓
- $(S\ (0\ 0))$ ✗
- $(x_2 + 0)$ ✗
- $(S\ 0\ 0)$ ✗
- $(S\ (<\ 0\ 0))$ ✓
- $(+\ x_2\ \bot)$ ✓
- $(S\ \bot)$ ✓
- $(\doteq\ 0\ \bot)$ ✓
- $(\text{read}\ a)$ ✗
- $(\text{read}\ a\ i)$ ✓
- $(\text{read}\ (\text{write}\ a\ i\ x)\ j)$ ✓

Well-sortedness

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

Definition 12: *Sort system* is a proof system over sequents of the form $\Gamma \vdash t : \sigma$.

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

$$\textsc{Var}\ \frac{x : \sigma \in \Gamma}{\Gamma \vdash x : \sigma} \qquad \textsc{Const}\ \frac{c \in \Sigma^F \quad \mathrm{rank}(c) = \langle \sigma \rangle}{\Gamma \vdash c : \sigma}$$

$$\textsc{Fun}\ \frac{f \in \Sigma^F \quad \mathrm{rank}(f) = \langle \sigma_1, ..., \sigma_n, \sigma \rangle \quad \Gamma \vdash t_1 : \sigma_1 \quad \cdots \quad \Gamma \vdash t_n : \sigma_n}{\Gamma \vdash (f\ t_1 \cdots t_n) : \sigma}$$

Definition 13 (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 $\Sigma^S = \{\texttt{Nat}\} \cup \{\texttt{Bool}\}$ and $\Sigma^F = \{0, S, <, +, \times, \doteq_{\texttt{Nat}}\} \cup \{\top, \bot, \doteq_{\texttt{Bool}}\}$.

- $\mathrm{rank}(0) = \langle \texttt{Nat} \rangle$
- $\mathrm{rank}(S) = \langle \texttt{Nat}, \texttt{Nat} \rangle$
- $\mathrm{rank}(<) = \mathrm{rank}(\doteq_{\texttt{Nat}}) = \langle \texttt{Nat}, \texttt{Nat}, \texttt{Bool} \rangle$
- $\mathrm{rank}(+) = \mathrm{rank}(\times) = \langle \texttt{Nat}, \texttt{Nat}, \texttt{Nat} \rangle$

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

1. $(+\ 0\ x_2)$ ✓
2. $(+\ (+\ 0\ x_1)\ x_2)$ ✗
3. $(S\ (+\ 0\ x_5))$ ✓
4. $(<\ (S\ x_3)\ (+\ (S\ 0)\ x_1))$ ✓
5. $(\doteq_{\texttt{Nat}}\ (S\ x_3)\ (+\ (S\ 0)\ x_1))$ ✓

Formulas

Definition 14 (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 15 (Formula): *Well-formed formulas* are expressions generated from atoms by the *formula-building operations*, denoted $\mathcal{F} = \left\{\mathcal{F}_\vee, \mathcal{F}_\wedge, \mathcal{F}_\neg, \mathcal{F}_\to, \mathcal{F}_\leftrightarrow, \mathcal{E}_{x,\sigma}, \mathcal{A}_{x,\sigma}\right\}$.

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

Examples of Formulas

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

Which of the following formulas are well-formed?

1. $(\doteq_{\texttt{Nat}}\ 0\ (S\ 0))$ ✓
2. $(<\ (S\ x_3)\ (+\ (S\ 0)\ x_1))$ ✓
3. $(\doteq_{\texttt{Nat}}\ (+\ x_1\ 0)\ x_2)$ ✓
4. $(\doteq_{\texttt{Nat}}\ (+\ x_1\ 0)\ x_2) \to \bot$ ✓
5. $(+\ 0\ x_3) \wedge (<\ 0\ (S\ 0))$ ✗
6. $\forall x_3 : \texttt{Nat}.\ (+\ (+\ 0\ x_3)\ x_2)$ ✗
7. $\forall x_3 : \texttt{Bool}.\ (\doteq_{\texttt{Nat}}\ (+\ 0\ x_3)\ x_2)$ ✓ *(Note: not well-sorted)*
8. $\neg\exists x_0 : \texttt{Nat}.\ (<\ 0\ x_0\ (S\ 0))$ ✗

Well-sorted Formulas

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

$$\textsc{BConst}\ \frac{c \in \{\top, \bot\}}{\Gamma \vdash c : \texttt{Bool}} \qquad \textsc{Not}\ \frac{\Gamma \vdash \alpha : \texttt{Bool}}{\Gamma \vdash (\neg\alpha) : \texttt{Bool}}$$

$$\textsc{Conn}\ \frac{\Gamma \vdash \alpha : \texttt{Bool} \quad \Gamma \vdash \beta : \texttt{Bool} \quad * \in \{\wedge, \vee, \to, \leftrightarrow\}}{\Gamma \vdash (\alpha * \beta) : \texttt{Bool}}$$

$$\textsc{Quant}\ \frac{\Gamma[x : \sigma] \vdash \alpha : \texttt{Bool} \quad \sigma \in \Sigma^S \quad Q \in \{\forall, \exists\}}{\Gamma \vdash (Q\, x : \sigma.\, \alpha) : \texttt{Bool}}$$

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

Definition 16 (Σ-formula): A formula φ is a *well-sorted* w.r.t. Σ in a sort context Γ if $\Gamma \vdash \varphi : \texttt{Bool}$ 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 17 (Free and bound variables): Recursive definition of *free* variables in a Σ-formula:

- x occurs free in a Σ-atom $(A\ t_1 \cdots t_n)$ 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, $x \to \forall x.\, P(x)$.

Definition 18 (Sentence): A formula α is *closed*, or a *sentence*, if it contains no free variables.

Free and Bound Variables [2]

Definition 19: The set $\mathcal{FV}$ of *free variables* of a Σ-formula α is defined as follows:

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

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

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

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* $\mathcal{I}$ of Σ and Γ consisting of:

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

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

- $\texttt{Bool}^{\mathcal{I}} = \{\texttt{true}, \texttt{false}\}, \quad \bot^{\mathcal{I}} = \texttt{false}, \quad \top^{\mathcal{I}} = \texttt{true}$,
- for all $\sigma \in \Sigma^S$, $\doteq_\sigma^{\mathcal{I}}$ 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):

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

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

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

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

- $\texttt{Elem}^{\mathcal{J}} = \texttt{Set}^{\mathcal{J}} = \mathbb{N}$.
- $\emptyset^{\mathcal{J}} = 0$.
- 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 $\overline{\mathcal{I}}$ for *well-sorted Σ-terms* by structural induction:

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

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

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

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

Then:

- $(\text{pa}\ (\text{ma}\ x))^{\overline{\mathcal{I}}} = \text{pa}^{\mathcal{I}}((\text{ma}\ x)^{\overline{\mathcal{I}}}) = \text{pa}^{\mathcal{I}}(\text{ma}^{\mathcal{I}}(x^{\overline{\mathcal{I}}})) = \text{pa}^{\mathcal{I}}(x^{\mathcal{I}})$
  $= \text{pa}^{\mathcal{I}}(\text{ma}^{\mathcal{I}}(\text{Jim})) = \text{pa}^{\mathcal{I}}(\text{Jill}) = \text{Jay}$

Term and Formula Semantics [2]

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

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

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

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

Satisfiability, Entailment, Validity

We write $\mathcal{I} \models \alpha$ to denote "$\mathcal{I}$ *satisfies* α" and mean $\alpha^{\mathcal{I}} = \texttt{true}$.

We write $\mathcal{I} \not\models \alpha$ to denote "$\mathcal{I}$ *falsifies* α" and mean $\alpha^{\mathcal{I}} = \texttt{false}$.

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.

Satisfiability, Entailment, Validity [2]

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

1. $\forall x.\, p(x) \vDash p(y)$ ✓
2. $p(x) \vDash \forall x.\, p(x)$ ✗
3. $\forall x.\, p(x) \vDash \exists y.\, p(y)$ ✓
4. $\exists y\, \forall x.\, q(x, y) \vDash \forall x\, \exists y.\, q(x, y)$ ✓
5. $\forall x\, \exists y.\, q(x, y) \vDash \exists y\, \forall x.\, q(x, y)$ ✗
6. $\vDash \exists x.\, (p(x) \to \forall y.\, p(y))$ ✓

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 $\Sigma^S = \{\sigma\}$, $\Sigma^F = \{Q, \doteq_\sigma\}$, $\Gamma = \{x : \sigma, y : \sigma\}$, $\mathrm{rank}(Q) = \langle \sigma, \sigma, \texttt{Bool} \rangle$.

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

1. $\forall x : \sigma . \forall y : \sigma . x \doteq y$
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.
   $\exists x : \texttt{Nat}.\, \forall y : \texttt{Nat}.\, (x \doteq y) \vee (x < y)$
2. For every natural number there is a greater one.
   $\forall x : \texttt{Nat}.\, \exists y : \texttt{Nat}.\, (x < y)$
3. Two natural numbers are equal only if their respective successors are equal.
   $\forall x : \texttt{Nat}.\, \forall y : \texttt{Nat}.\, (x \doteq y) \to (S(x) \doteq S(y))$
4. Two natural numbers are equal if their respective successors are equal.
   $\forall x : \texttt{Nat}.\, \forall y : \texttt{Nat}.\, (S(x) \doteq S(y)) \to (x \doteq y)$
5. No two distinct natural number have the same successor.
   $\forall x : \texttt{Nat}.\, \forall y : \texttt{Nat}.\, \neg(x \doteq y) \to \neg(S(x) \doteq S(y))$
6. There are at least two natural number smaller than 3.
   $\exists x : \texttt{Nat}.\, \exists y : \texttt{Nat}.\, \neg(x \doteq y) \wedge (x < S(S(S(0)))) \wedge (y < S(S(S(0))))$
7. There is no largest natural number.
   $\neg\exists x : \texttt{Nat}.\, \forall y : \texttt{Nat}.\, (y \doteq x) \vee (y < x)$

From English to FOL [2]

1. Everyone has a father and a mother.
   $\forall x : \texttt{Person}.\, \exists y : \texttt{Person}.\, \exists z : \texttt{Person}.\, (y \doteq \mathrm{pa}(x)) \wedge (z \doteq \mathrm{ma}(x))$
2. The marriage relation is symmetric.
   $\forall x : \texttt{Person}.\, \forall y : \texttt{Person}.\, \mathrm{sp}(x, y) \rightarrow \mathrm{sp}(y, x)$
3. No one can be married to themselves.
   $\forall x : \texttt{Person}.\, \neg\, \mathrm{sp}(x, x)$
4. Not all people are married.
   $\neg \forall x : \texttt{Person}.\, \exists y : \texttt{Person}.\, \mathrm{sp}(x, y)$
5. Some people have a farther and a mother who are not married to each other.
   $\exists x : \texttt{Person}.\, \neg\, \mathrm{sp}(\mathrm{ma}(x), \mathrm{pa}(x))$
6. You cannot marry more than one person.
   $\forall x : \texttt{Person}.\, \forall y : \texttt{Person}.\, \forall z : \texttt{Person}.\, (\mathrm{sp}(x, y) \wedge \mathrm{sp}(x, z)) \rightarrow (y \doteq z)$
7. Some people are not mothers.
   $\exists x : \texttt{Person}.\, \forall y : \texttt{Person}.\, \neg(x \doteq \mathrm{ma}(y))$
8. Nobody can be both a farther and a mother.
   $\forall x : \texttt{Person}.\, \neg\exists y : \texttt{Person}.\, \neg\exists z : \texttt{Person}.\, (x \doteq \mathrm{pa}(y)) \wedge (x \doteq \mathrm{ma}(z))$

From English to FOL [3]

9. Nobody can be their own or farther's farther.
   $\forall x : \texttt{Person}.\, \neg((x \doteq \mathrm{pa}(x)) \vee (x \doteq \mathrm{pa}(\mathrm{pa}(x))))$
10. Some people do not have children.
   $\exists x : \texttt{Person}.\, \forall y : \texttt{Person}.\, \neg(y \doteq \mathrm{pa}(x)) \wedge \neg(y \doteq \mathrm{ma}(y))$

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 $(f\ t_1 \cdots t_n)$ 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, $t^{\overline{\mathcal{I}}} = f^{\mathcal{I}}(t_1^{\overline{\mathcal{I}}}, ..., t_n^{\overline{\mathcal{I}}}) = f^{\mathcal{J}}(t_1^{\overline{\mathcal{J}}}, ..., t_n^{\overline{\mathcal{J}}}) = t^{\overline{\mathcal{J}}}$.

□

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 $Q\, x : \sigma.\, \beta$ with $Q \in \{\forall, \exists\}$, then $\mathcal{FV}(\beta) = \mathcal{FV}(\alpha) \cup \{x\}$. For any c in $\sigma^{\mathcal{I}}$, $\mathcal{I}[x \mapsto c]$ 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 $\alpha \to \beta$ is valid.

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

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

- 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 $\mathcal{I} \models \alpha \to \beta$ for every interpretation $\mathcal{I}$. □

Proof *($\Leftarrow$): If $\alpha \to \beta$ 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 $\alpha \to \beta$, contradicting the assumption that $\alpha \to \beta$ 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 $\models \alpha \to \beta$ iff $\alpha \to \beta$ 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, $\Phi \vDash \alpha$ iff $\Phi \vDash \forall x : \sigma.\, \alpha$.

Proof *($\Rightarrow$): If $\Phi \vDash \alpha$ then $\Phi \vDash \forall x : \sigma.\, \alpha$.*

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 $\Phi \vDash \alpha$). 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, $\Phi \vDash \forall x : \sigma.\, \alpha$. □

Proof *($\Leftarrow$): If $\Phi \vDash \forall x : \sigma.\, \alpha$ then $\Phi \vDash \alpha$.*

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

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, $\alpha \vDash \beta$ iff $\exists x : \sigma.\, \alpha \vDash \beta$.

Proof *($\Rightarrow$): If $\alpha \vDash \beta$ then $\exists x : \sigma.\, \alpha \vDash \beta$.*

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 $\exists x : \sigma.\alpha \vDash \beta$. □

Proof *($\Leftarrow$): If $\exists x : \sigma.\, \alpha \vDash \beta$ then $\alpha \vDash \beta$.*

Let $\mathcal{I}$ be an interpretation that satisfies α. Then, trivially[1], $\mathcal{I}$ satisfies $\exists x : \sigma.\, \alpha$. By assumption, $\mathcal{I} \vDash \beta$. Since $\mathcal{I}$ was arbitrary, $\alpha \vDash \beta$. □

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

§4 Proofs in FOL

Semantic Arguments for FOL

(a) $\dfrac{\mathcal{I} \vDash \neg\alpha}{\mathcal{I} \nvDash \alpha}$

(b) $\dfrac{\mathcal{I} \nvDash \neg\alpha}{\mathcal{I} \vDash \alpha}$

(c) $\dfrac{\mathcal{I} \vDash \alpha \wedge \beta}{\mathcal{I} \vDash \alpha, \mathcal{I} \vDash \beta}$

(d) $\dfrac{\mathcal{I} \nvDash \alpha \wedge \beta}{\mathcal{I} \nvDash \alpha \mid \mathcal{I} \nvDash \beta}$

(e) $\dfrac{\mathcal{I} \vDash \alpha \vee \beta}{\mathcal{I} \vDash \alpha \mid \mathcal{I} \vDash \beta}$

(f) $\dfrac{\mathcal{I} \nvDash \alpha \vee \beta}{\mathcal{I} \nvDash \alpha, \mathcal{I} \nvDash \beta}$

(i) $\dfrac{\mathcal{I} \vDash \alpha \qquad \mathcal{I} \nvDash \alpha}{\mathcal{I} \vDash \bot}$

(g) $\dfrac{\mathcal{I} \vDash \alpha \to \beta}{\mathcal{I} \nvDash \alpha \mid \mathcal{I} \vDash \beta}$

(h) $\dfrac{\mathcal{I} \nvDash \alpha \to \beta}{\mathcal{I} \vDash \alpha, \mathcal{I} \nvDash \beta}$

(j) $\dfrac{\mathcal{I} \vDash \alpha \leftrightarrow \beta}{\mathcal{I} \vDash \alpha, \mathcal{I} \vDash \beta \mid \mathcal{I} \nvDash \alpha, \mathcal{I} \nvDash \beta}$

(k) $\dfrac{\mathcal{I} \nvDash \alpha \leftrightarrow \beta}{\mathcal{I} \nvDash \alpha, \mathcal{I} \vDash \beta \mid \mathcal{I} \vDash \alpha, \mathcal{I} \nvDash \beta}$

§5 Other slides

Terms (simple)

Definition 20 (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*. Intuitevely, terms are expressions that evaluate to values.

Definition 21 (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 $t_1, ..., t_n$ are terms of sorts $\sigma_1, ..., \sigma_n$, and f is a function symbol with $\text{rank}(f) = \langle \sigma_1, ..., \sigma_n, \sigma \rangle$, then $(f\ t_1 \cdots t_n)$ is a term of sort σ.
- *(Nothing else is a term.)*

Formulas (simple)

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

Definition 22 (Atoms): The Σ-atoms over X, or simply *atoms*, are terms of the form $(p\ t_1 \cdots t_n)$, where $t_1, ..., t_n$ are terms of sorts $\sigma_1, ..., \sigma_n$, and p is a predicate with $\text{rank}(p) = \langle \sigma_1, ..., \sigma_n, \texttt{Bool} \rangle$.

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

Definition 23 (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 $\neg\alpha$, $(\alpha \vee \beta)$, $(\alpha \wedge \beta)$, $(\alpha \to \beta)$, 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 $\exists x : \sigma.\, \alpha$.

Syntax and Semantics of FOL

Definition 24 (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, $\bot$, $\emptyset$, ...
- A set of *function symbols* $\mathcal{F}$, e.g., S, $+$, $\times$, ...
- A set of *predicate symbols* $\mathcal{P}$, e.g., $=$, $<$, $\in$, ...

Definition 25 (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* $\mathcal{D} = \mathrm{dom}(\mathcal{M})$, 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 Example

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

Example: x is a even number.

Definition 27 (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 problem	**Formalization**	**Description**
Validity	$\vDash \varphi$	Is φ true in every world?
Satisfiability	$\exists \mathcal{M} \,.\, \mathcal{M} \vDash \varphi$	Is φ true in some world?
Model checking	$\mathcal{M} \vDash \varphi$	Is φ true in a given world?

First-Order Model Checking

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

Definition 28: 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, $\mathcal{M} \vDash \varphi$.

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

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

First-Order Model Checking [2]

Theorem 6: MCFO is NP-hard.

Proof: SAT can be reduced to MCFO.

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

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

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

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

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.

TODO: third problem is validity.

More:

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

Finite Satisfiability

Definition 29: *Finite Satisfiability Problem* (FSP) is the problem of determining whether a given first-order formula φ has a *finite* model $\mathcal{M}$ such that $\mathcal{M} \vDash \varphi$.

Example: Let φ be the first-order formula obtained as the conjunction of the following sentences, where a_i are constants:

- $R(a_0, a_1)$
- $\forall x, y.\, (R(x, y) \rightarrow \exists z.\, R(y, z))$
- $\forall x, y, z.\, (R(y, x) \wedge R(z, x) \rightarrow (y = z))$
- $\forall x.\, \neg R(x, a_0)$

Formula φ has the infinite model $R(a_0, a_1), R(a_1, a_2), \ldots$, but is not finitely satisfiable.

Theorem 7 (Trakhtenbrot): Finite satisfiability problem for first-order logic is undecidable.

TODO: Exercises

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