CAS 701 Fall 2005

04 First-Order Logic

Instructor: W. M. Farmer

Revised: 12 October 2005

What is First-Order Logic?

- **First-order logic** is the study of statements about individuals using functions, predicates, and quantification.
  - First-order logic is also called **first-order predicate logic** and **first-order quantificational logic**.

- First-order logic is propositional logic plus:
  - **Terms** that denote individuals.
  - **Predicates** that are applied to terms.
  - **Quantifiers** applied to individual variables.

- First-order logic is "first-order" because quantification is over individuals but not over higher-order objects such as functions and predicates.

- There are many versions of first-order logic.

- We will define and employ a version of first-order logic named FOL.

Syntax of FOL: Languages

- Let $\mathcal{V}$ be a fixed infinite set of symbols called **variables**.

- A **language** of FOL is a triple $L = (\mathcal{C}, \mathcal{F}, \mathcal{P})$ where:
  - $\mathcal{C}$ is a set of symbols called **individual constants**.
  - $\mathcal{F}$ is a set of symbols called **function symbols**, each with an assigned arity ≥ 1.
  - $\mathcal{P}$ is a set of symbols called **predicate symbols**, each with an assigned arity ≥ 1. $\mathcal{P}$ contains the binary predicate symbol $=$.
  - $\mathcal{V}$, $\mathcal{C}$, $\mathcal{F}$, and $\mathcal{P}$ are pairwise disjoint.

Syntax of FOL: Terms and Formulas

- Let $L = (\mathcal{C}, \mathcal{F}, \mathcal{P})$ be a language of FOL.

- A **term** of L is a string of symbols inductively defined by the following formation rules:
  - Each $x \in \mathcal{V}$ and $a \in \mathcal{C}$ is a term of L.
  - If $f \in \mathcal{F}$ is n-ary and $t_1, \ldots, t_n$ are terms of L, then $f(t_1, \ldots, t_n)$ is a term of L.

- A **formula** of L is a string of symbols inductively defined by the following formation rules:
  - If $p \in \mathcal{P}$ is n-ary and $t_1, \ldots, t_n$ are terms of L, then $p(t_1, \ldots, t_n)$ is a formula of L.
  - If A and B are formulas of L and $x \in \mathcal{V}$, then $(\neg A)$ and $(A \Rightarrow B)$, and $(\forall\, x \,.\, A)$ are formulas of L.

- $=$, $\neg$, $\Rightarrow$, and $\forall$ are the **logical constants** of FOL.

Syntax of FOL: Abbreviations

$(s = t)$	denotes	$= (s, t)$.
$(s \neq t)$	denotes	$(\neg(s = t))$.
T	denotes	$(\forall\, x\, .\, (x = x))$.
F	denotes	$(\neg(\mathsf{T}))$.
$(A \vee B)$	denotes	$((\neg A) \Rightarrow B)$.
$(A \wedge B)$	denotes	$(\neg((\neg A) \vee (\neg B)))$.
$(A \Leftrightarrow B)$	denotes	$((A \Rightarrow B) \wedge (B \Rightarrow A))$.
$(\exists\, x\, .\, A)$	denotes	$(\neg(\forall\, x\, .\, (\neg A)))$.
$(\Box\, x_1, \ldots, x_n\, .\, A)$	denotes	$(\Box\, x_1\, .\, (\Box\, x_2, \ldots, x_n\, .\, A))$ where $n \geq 2$ and $\Box \in \{\forall, \exists\}$.

Free and Bound Variables

- The **scope** of a quantifier $\forall x$ or $\exists x$ in a formula $\forall x \,.\, B$ or $\exists x \,.\, B$, respectively, is the part of B that is not in a subformula of B of the form $\forall x \,.\, C$ or $\exists x \,.\, C$.

- An occurrence of a variable x in a formula A is **free** if it is not in the scope of a quantifier $\forall x$ or $\exists x$; otherwise the occurrence of x in A is **bound**.
  - An occurrence of a variable in a formula is either free or bound but never both.
  - A variable can be both bound and free in a formula.

- A formula is **closed** if it contains no free variables.

- A **sentence** is a closed formula.

Substitution

- Let x be a variable, t a term, and A a formula.
- The **substitution** of t for x in A, written

  $$A[t/x] \quad \text{or } A[x \mapsto t],$$

  is the result of replacing each free occurrence of x in A with t.
- Suppose A is $\forall y \,.\, x = y$ and t is $f(y)$. Then the substitution $A[t/x]$ is said to **capture** y.
  - Variable captures often produce unsound results.
- t is **free for** x **in** A if no free occurrence of x in A is in the scope of $\forall y$ or $\exists y$ for any variable y occurring t.
  - Hence, t is free for x in A if the substitution $A[t/x]$ does not result in any variable captures.

Semantics of FOL: Models

- A **model** for a language $L = (\mathcal{C}, \mathcal{F}, \mathcal{P})$ of FOL is a pair $M = (D, I)$ where D is a nonempty domain (set) and I is a total function on $\mathcal{C} \cup \mathcal{F} \cup \mathcal{P}$ such that:
  - If $a \in \mathcal{C}$, $I(a) \in D$.
  - If $f \in \mathcal{F}$ is n-ary, $I(f) : D^n \to D$ and $I(f)$ is total.
  - If $p \in \mathcal{P}$ is n-ary, $I(p) : D^n \to \{\mathsf{t}, \mathsf{f}\}$ and $I(p)$ is total.
  - $I(=)$ is id_D, the identity predicate on D.

- A **variable assignment** into M is a function that maps each $x \in \mathcal{V}$ to an element of D.

- Given a variable assignment φ into M, $x \in \mathcal{V}$, and $d \in D$, let $\varphi[x \mapsto d]$ be the variable assignment φ' into M such $\varphi'(x) = d$ and $\varphi'(y) = \varphi(y)$ for all $y \neq x$.

Semantics of FOL: Valuation Function

The **valuation function** for a model M for a language $L = (\mathcal{C}, \mathcal{F}, \mathcal{P})$ of FOL is the binary function V^M that satisfies the following conditions for all variable assignments φ into M and all terms t and formulas A of L:

1. Let $t \in \mathcal{V}$. Then $V^M_\varphi(t) = \varphi(t)$.
2. Let $t \in \mathcal{C}$. Then $V^M_\varphi(t) = I(t)$.
3. Let $t = f(t_1, \ldots, t_n)$. Then $V^M_\varphi(t) = I(f)(V^M_\varphi(t_1), \ldots, V^M_\varphi(t_n))$.
4. Let $A = p(t_1, \ldots, t_n)$. Then $V^M_\varphi(A) = I(p)(V^M_\varphi(t_1), \ldots, V^M_\varphi(t_n))$.
5. Let $A = (\neg A')$. If $V^M_\varphi(A') =$ f, then $V^M_\varphi(A) =$ t; otherwise $V^M_\varphi(A) =$ f.
6. Let $A = (A_1 \Rightarrow A_2)$. If $V^M_\varphi(A_1) =$ t and $V^M_\varphi(A_2) =$ f, then $V^M_\varphi(A) =$ f; otherwise $V^M_\varphi(A) =$ t.
7. Let $A = (\forall\, x \,.\, A')$. If $V^M_{\varphi[x \mapsto d]}(A') =$ t for all $d \in D$, then $V^M_\varphi(A) =$ t; otherwise $V^M_\varphi(A) =$ f.

Metatheorems of FOL

- **Completeness Theorem (Gödel 1930)**. There is a sound and complete proof system for FOL.

- **Compactness Theorem**. Let Σ be a set of formulas of a language of FOL. If Σ is finitely satisfiable, then Σ is satisfiable.

- **Undecidability Theorem (Church 1936)**. First-order logic is undecidable. That is, for some language L of FOL, the problem of whether or not a given formula of L is valid is undecidable.

A Hilbert-Style Proof System (1)

Let **H** be the following Hilbert-style proof system for a language L of FOL:

- The **logical axioms** of **H** are all formulas of L that are instances of the following schemas:
  - For propositional logic:

    A1: $A \Rightarrow (B \Rightarrow A)$.

    A2: $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$.

    A3: $(\neg A \Rightarrow \neg B) \Rightarrow (B \Rightarrow A)$.
  - For quantification:

    A4: $(\forall x \,.\, (A \Rightarrow B)) \Rightarrow (A \Rightarrow (\forall x \,.\, B))$
    provided x is not free in A.

    A5: $(\forall x \,.\, A) \Rightarrow A[x \mapsto t]$
    provided t is free for x in A.

A Hilbert-Style Proof System (2)

– For equality:

A6: $\forall x \,.\, x = x$.

A7: $\forall x, y \,.\, x = y \Rightarrow y = x$.

A8: $\forall x, y, z \,.\, (x = y \wedge y = z) \Rightarrow x = z$.

A9: $\forall x_1, \ldots, x_n, y_1, \ldots, y_n \,.\, (x_1 = y_1 \wedge \cdots \wedge x_n = y_n)$
$\Rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$
where $f \in \mathcal{F}$ is n-ary.

A10: $\forall x_1, \ldots, x_n, y_1, \ldots, y_n \,.\, (x_1 = y_1 \wedge \cdots \wedge x_n = y_n)$
$\Rightarrow (p(x_1, \ldots, x_n) \Leftrightarrow p(y_1, \ldots, y_n))$
where $p \in \mathcal{P}$ is n-ary.

- The **rules of inference** of **H** are:

  MP: From A and $(A \Rightarrow B)$, infer B.

  GEN: From A, infer $(\forall x \,.\, A)$, for any $x \in \mathcal{V}$.

More Metatheorems of FOL

- **Deduction Theorem**. $\Sigma \cup \{A\} \vdash_{\mathbf{H}} B$ implies $\Sigma \vdash_{\mathbf{H}} A \Rightarrow B$.

- **Soundness Theorem**. $\Sigma \vdash_{\mathbf{H}} A$ implies $\Sigma \models A$.

- **Completeness Theorem**. $\Sigma \models A$ implies $\Sigma \vdash_{\mathbf{H}} A$.

- **Soundness and Completeness Theorem (second form)**. Σ is consistent in **H** iff Σ is satisfiable.

Theories

- A **theory** in FOL is a pair $T = (L, \Gamma)$ where L is a language of FOL and Γ is a set of sentences of L.

- Examples:
  - Theories of partial and total orders.
  - Theories of monoids and groups.
  - Presburger arithmetic.
  - First-order Peano arithmetic.
  - Theory of real closed fields.

Algebras as Models

- If $L = (\mathcal{C}, \mathcal{F}, \mathcal{P})$ is a finite language of FOL, we may present the language as

  $$L = (c_1, \ldots, c_k, f_1, \ldots, f_m, p_1, \ldots, p_n)$$

  where $\mathcal{C} = \{c_1, \ldots, c_k\}$, $\mathcal{F} = \{f_1, \ldots, f_m\}$, and $\mathcal{P} = \{p_1, \ldots, p_n\}$.

- An algebra

  $$(D, d_1, \ldots, d_k, g_1, \ldots, g_m, q_1, \ldots . q_n)$$

  can then be considered a model for L if $M = (D, I)$ is a model for L where $I(c_i) = d_i$ for $1 \leq i \leq k$, $I(f_i) = g_i$ for $1 \leq i \leq m$, and $I(p_i) = q_i$ for $1 \leq i \leq n$.

Example: Peano Arithmetic

- Language: A language of second-order logic with an individual constant symbol 0 and unary function symbol S.
  - 0 is intended to represent the number zero.
  - S is intended to represent the successor function, i.e., $S(a)$ means $a + 1$.

- Axioms:
  - **0 has no predecessor**. $\forall x \, . \, \neg(0 = S(x))$.
  - **S is injective**. $\forall x, y \, . \, S(x) = S(y) \Rightarrow x = y$.
  - **Induction principle**.
    $\forall P \, . \, (P(0) \wedge \forall x \, . \, P(x) \Rightarrow P(S(x))) \Rightarrow \forall x \, . \, P(x)$.

- Second-order Peano arithmetic is **categorical**, i.e, it has exactly one model up to isomorphism.

Language and Theory Extensions

- Let $L_i = (\mathcal{C}_i, \mathcal{F}_i, \mathcal{P}_i)$ be a language of FOL and let $T_i = (L_i, \Gamma_i)$ be a theory of FOL for $i = 1, 2$.

- L_1 is a **sublanguage** of L_2, and L_2 is a **super language** or an **extension** of L_1, written $L_1 \leq L_2$, if $\mathcal{C}_1 \subseteq \mathcal{C}_2$, $\mathcal{F}_1 \subseteq \mathcal{F}_2$, and $\mathcal{P}_1 \subseteq \mathcal{P}_2$.

- T_1 is a **subtheory** of T_2, and T_2 is a **super theory** or an **extension** of T_1, written $T_1 \leq T_2$, if $L_1 \leq L_2$ and $\Gamma_1 \subseteq \Gamma_2$.

Conservative Theory Extension

- Let $T = (L, \Gamma)$ and $T' = (L', \Gamma')$ be theories of FOL.

- T' is a **conservative extension** of T if $T \leq T'$ and, for every formula A of L, $T' \models A$ implies $T \models A$.
  - A conservative extension of a theory adds new machinery to the theory without compromising the theory's original machinery.

- The **obligation** of a purported conservative extension is a formula that implies that the extension is conservative.

- There are two important kinds of conservative extensions that add new vocabulary to a theory:
  1. Definitions.
  2. Profiles.

Definitions

- A **definition** is a conservative extension that adds a new symbol s and a defining axiom $A(s)$ to a theory T.
  - In some logics, the defining axiom can have the form $s = D$ (where s does not occur in D).

- The obligation of the definition is

  $$\exists\, ! \, x \, . \, A(x).$$

- The symbol s can usually be eliminated from any new expression of involving s.

Profiles

- A **profile** is a conservative extension that adds a set $\{s_1, \ldots, s_n\}$ of symbols and a profiling axiom $A(s_1, \ldots, s_n)$ to a theory T.

- The obligation of the profile is

  $$\exists\, x_1, \ldots, x_n \,.\, A(x_1, \ldots, x_n).$$

- The symbols $s_1, \ldots, s_n$ cannot usually be eliminated from expressions involving $s_1, \ldots, s_n$.

- Profiles can be used for introducing:
  - Underspecified objects.
  - Recursively defined functions.
  - Algebras.