CAS 701
Logic and Discrete Mathematics
Fall 2017

4 First-Order Logic

William M. Farmer

Department of Computing and Software
McMaster University

October 27, 2017

What is First-Order Logic?

- First-order logic is the study of statements about individuals constructed using functions, predicates, propositional connectives, and quantifiers.
  - 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 develop first-order logic following Chapter 5 of J. H. Gallier, *Logic for Computer Science*, Dover, 2015.

Outline

- Syntax.
- Semantics.
- Metatheorems.
- Theories.
- Hilbert-style proof systems.
- Gentzen-style sequent systems.

Syntax [1/3]

- Let $\mathbf{V} = \{x_0, x_1, \ldots\}$ be a countably infinite set of symbols called variables.
- A first-order language is a triple $\mathbf{L} = \{\mathbf{CS}, \mathbf{FS}, \mathbf{PS}\}$ where:
  1. $\mathbf{CS} = \{c_0, c_1, \ldots\}$ is a countable set of symbols called constants.
  2. $\mathbf{FS} = \{f_0, f_1, \ldots\}$ is a countable set of symbols called function symbols with a rank function r that assigns each member $f \in \mathbf{FS}$ a positive integer $r(f)$.
  3. $\mathbf{PS} = \{P_0, P_1, \ldots\}$ is a countable set of symbols called predicate symbols with a rank function r that assigns each member $P \in \mathbf{PS}$ a nonnegative integer $r(P)$.
  4. $\mathbf{V}$, $\mathbf{CS}$, $\mathbf{FS}$, and $\mathbf{PS}$ are pairwise disjoint.

Syntax [2/3]

- Let $\mathbf{L} = \{\mathbf{CS}, \mathbf{FS}, \mathbf{PS}\}$ be a first-order language.
- The set $\text{TERM}_{\mathbf{L}}$ of $\mathbf{L}$-terms is the set of strings defined inductively by:
  1. Each variable in $\mathbf{V}$ and constant in $\mathbf{CS}$ is in $\text{TERM}_{\mathbf{L}}$.
  2. If $t_1, \ldots, t_n$ are in $\text{TERM}_{\mathbf{L}}$ and f is a function symbol in $\mathbf{FS}$ with $r(f) = n > 0$, then $f\, t_1 \cdots t_n$ is in $\text{TERM}_{\mathbf{L}}$.
- The set $\text{ATOM}_{\mathbf{L}}$ of $\mathbf{L}$-atomic formulas is the set of strings defined by:
  1. $\bot$ and $\top$ are in $\text{ATOM}_{\mathbf{L}}$.
  2. Each P in $\mathbf{PS}$ with $r(P) = 0$ is in $\text{ATOM}_{\mathbf{L}}$.
  3. If $t_1, \ldots, t_n$ are in $\text{TERM}_{\mathbf{L}}$ and P is a predicate symbol in $\mathbf{PS}$ with $r(P) = n > 0$, then $P\, t_1 \cdots t_n$ is in $\text{ATOM}_{\mathbf{L}}$.
  4. If t_1 and t_2 are in $\text{TERM}_{\mathbf{L}}$, then $(t_1 \doteq t_2)$ is in $\text{ATOM}_{\mathbf{L}}$.

Syntax [3/3]

- The set $\text{FORM}_{\mathbf{L}}$ of $\mathbf{L}$-formulas is the set of strings inductively defined by:
    1. Each A in $\text{ATOM}_{\mathbf{L}}$ is in $\text{FORM}_{\mathbf{L}}$.
    2. If A and B are in $\text{FORM}_{\mathbf{L}}$, then $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \supset B)$, and $(A \equiv B)$ are in $\text{FORM}_{\mathbf{L}}$.
    3. If x in $\mathbf{V}$ and A is $\text{FORM}_{\mathbf{L}}$, then $\forall\, x\, A$ and $\exists\, x\, A$ are in $\text{FORM}_{\mathbf{L}}$.

Free and Bound Variables [1/2]

- Given a term t, the set $\mathrm{FV}(t)$ of free variables of t is inductively defined by:
  1. $\mathrm{FV}(x) = \{x\}$ for a variable x.
  2. $\mathrm{FV}(c) = \emptyset$ for a constant c.
  3. $\mathrm{FV}(f\ t_1\ \cdots\ t_n) = \mathrm{FV}(t_1) \cup \cdots \cup \mathrm{FV}(t_n)$ for a function symbol f with $r(f) = n > 0$.
- Given a formula A, the set $\mathrm{FV}(A)$ of free variables of A is inductively defined by:
  1. $\mathrm{FV}(\bot) = \mathrm{FV}(\top) = \emptyset$.
  2. $\mathrm{FV}(P) = \emptyset$ for a predicate symbol P with $r(P) = 0$.
  3. $\mathrm{FV}(P\, t_1\ \cdots\ t_n) = \mathrm{FV}(t_1) \cup \cdots \cup \mathrm{FV}(t_1)$ for a predicate symbol P with $r(P) = n > 0$.
  4. $\mathrm{FV}(t_1 \doteq t_2) = \mathrm{FV}(t_1) \cup \mathrm{FV}(t_2)$.
  5. $\mathrm{FV}(\neg A) = \mathrm{FV}(A)$.
  6. $\mathrm{FV}(A \,\Box\, B) = \mathrm{FV}(A) \cup \mathrm{FV}(B)$ where $\Box \in \{\wedge, \vee, \supset, \equiv\}$.
  7. $\mathrm{FV}(\Box\, x\, A) = \mathrm{FV}(A) \setminus \{x\}$ where $\Box \in \{\forall, \exists\}$.

Free and Bound Variables [2/2]

- A term or formula is open [closed] if its set of free variables is nonempty [empty].
- A sentence is a closed formula.
- Given a formula A, the set $\mathsf{BV}(A)$ of bound variables of A is inductively defined by:
  1. $\mathsf{BV}(\bot) = \mathsf{BV}(\top) = \emptyset$.
  2. $\mathsf{BV}(P) = \emptyset$ for a predicate symbol P with $r(P) = 0$.
  3. $\mathsf{BV}(P\, t_1 \cdots t_n) = \emptyset$ for a predicate symbol P with $r(P) = n > 0$.
  4. $\mathsf{BV}(t_1 \doteq t_2) = \emptyset$.
  5. $\mathsf{BV}(\neg A) = \mathsf{BV}(A)$.
  6. $\mathsf{BV}(A \,\square\, B) = \mathsf{BV}(A) \cup \mathsf{BV}(B)$ where $\square \in \{\wedge, \vee, \supset, \equiv\}$.
  7. $\mathsf{BV}(\square\, x\, A) = \mathsf{BV}(A) \cup \{x\}$ where $\square \in \{\forall, \exists\}$.

Substitutions [1/2]

The result of substituting a term t for the variable x in a term s, written $s[t/x]$, is recursively defined by:

1. $x[t/x] = t$ when s is x.
2. $y[t/x] = y$ when s is a variable $y \neq x$.
3. $c[t/x] = c$ when s is a constant c.
4. $f\ t_1 \cdots t_n[t/x] = f\ t_1[t/x] \cdots t_n[t/x]$ when s is $f\ t_1 \cdots t_n$.

Substitutions [2/2]

The result of substituting a term t for the free occurrences of a variable x in a formula A, written $A[t/x]$, is recursively defined by:

1. $\bot[t/x] = \bot$ when A is $\bot$.
2. $\top[t/x] = \top$ when A is $\top$.
3. $P[t/x] = P$ when A is P.
4. $P\, t_1 \cdots t_n[t/x] = P\, t_1[t/x] \cdots t_n[t/x]$ when A is $P\, t_1 \cdots t_n$.
5. $(t_1 \doteq t_2)[t/x] = (t_1[t/x] \doteq t_2[t/x])$ when A is $(t_1 \doteq t_2)$.
6. $(\neg B)[t/x] = \neg B[t/x]$ when A is $(\neg B)$.
7. $(B \mathbin{\square} C)[t/x] = (B[t/x] \mathbin{\square} C[t/x])$ when A is $(B \mathbin{\square} C)$ where $\square \in \{\wedge, \vee, \supset, \equiv\}$.
8. $(\square\, x\, B)[t/x] = (\square\, x\, B)$ when A is $(\square\, x\, B)$ where $\square \in \{\forall, \exists\}$.
9. $(\square\, y\, B)[t/x] = (\square\, y\, B[t/x])$ when A is $(\square\, y\, B)$ and $y \neq x$ where $\square \in \{\forall, \exists\}$.

Semantics [1/4]

- An **L**-structure is a pair $\mathbf{M} = (M, I)$ where M is a nonempty set of values (called the domain of $\mathbf{M}$) and I is a function on $\mathbf{CS} \cup \mathbf{FS} \cup \mathbf{PS}$ (called the interpretation function of $\mathbf{M}$) such that:
  1. For $c \in \mathbf{CS}$, $I(c) \in M$.
  2. For $f \in \mathbf{FS}$ with $r(f) = n > 0$, $I(f) : M^n \to M$ is an n-ary function.
  3. For $P \in \mathbf{PS}$ with $r(f) = 0$, $I(P) \in \mathrm{BOOL}$.
  4. For $P \in \mathbf{PS}$ with $r(f) = n > 0$, $I(P) : M^n \to \mathrm{BOOL}$ is an n-ary predicate.
- $I(c), I(f), I(P)$ are denoted by $c_{\mathbf{M}}, f_{\mathbf{M}}, P_{\mathbf{M}}$, respectively.
- An **L**-structure is a mathematical structure of the form

$$(M; (c_0)_{\mathbf{M}}, (c_1)_{\mathbf{M}}, \ldots; (f_0)_{\mathbf{M}}, (f_1)_{\mathbf{M}}, \ldots; (P_0)_{\mathbf{M}}, (P_1)_{\mathbf{M}}, \ldots).$$

Semantics [2/4]

- Let $\mathbf{M} = (M, I)$ be an $\mathbf{L}$-structure.
- An assignment into $\mathbf{M}$ is any function $s : \mathbf{V} \rightarrow M$. Let $[\mathbf{V} \rightarrow M]$ be the set of all assignments into $\mathbf{M}$.
- If $s \in [\mathbf{V} \rightarrow M]$, $x \in \mathbf{V}$, and $a \in M$, $s[x := a]$ is the assignment $s' : \mathbf{V} \rightarrow M$ such that $s'(x) = a$ and $s'(y) = s(y)$ for all $y \neq x$.
- The value of a term $t \in \mathrm{TERM}_{\mathbf{L}}$ in $\mathbf{M}$ with assignment $s \in [\mathbf{V} \rightarrow M]$, written $t_{\mathbf{M}}[s]$, is a member of M recursively defined by:
  1. $x_{\mathbf{M}}[s] = s(x)$ for $x \in \mathbf{V}$.
  2. $c_{\mathbf{M}}[s] = c_{\mathbf{M}}$ for $c \in \mathbf{CS}$.
  3. $(f\ t_1 \cdots t_n)_{\mathbf{M}}[s] = f_{\mathbf{M}}((t_1)_{\mathbf{M}}[s], \ldots, (t_1)_{\mathbf{M}}[s])$ for $f\ t_1 \cdots t_n \in \mathrm{TERM}_{\mathbf{L}}$.

Semantics [3/4]

The value of a formula $A \in \mathrm{FORM}_{\mathbf{L}}$ in $\mathbf{M}$ with assignment $s \in [\mathbf{V} \rightarrow M]$, written $A_{\mathbf{M}}[s]$, is a member of BOOL recursively defined by:

1. $\bot_{\mathbf{M}}[s] = \mathbf{F}$.
2. $\top_{\mathbf{M}}[s] = \mathbf{T}$.
3. $P_{\mathbf{M}}[s] = P_{\mathbf{M}}$ for $P \in \mathrm{FORM}_{\mathbf{L}}$.
4. $(P\, t_1 \, \cdots \, t_n)_{\mathbf{M}}[s] = P_{\mathbf{M}}((t_1)_{\mathbf{M}}[s], \ldots, (t_1)_{\mathbf{M}}[s])$ for $P\, t_1 \, \cdots \, t_n \in \mathrm{FORM}_{\mathbf{L}}$.
5. $(t_1 \doteq t_2)_{\mathbf{M}}[s] = \mathbf{T}$ if $(t_1)_{\mathbf{M}}[s] = (t_2)_{\mathbf{M}}[s]$ for $(t_1 \doteq t_2) \in \mathrm{FORM}_{\mathbf{L}}$. Otherwise, $(t_1 \doteq t_2)_{\mathbf{M}}[s] = \mathbf{F}$.
6. $(\neg A)_{\mathbf{M}}[s] = H_{\neg}(A_{\mathbf{M}}[s])$ for $\neg A \in \mathrm{FORM}_{\mathbf{L}}$.
7. $(A \,\Box\, B)_{\mathbf{M}}[s] = H_{\Box}(A_{\mathbf{M}}[s], B_{\mathbf{M}}[s])$ for $(A \,\Box\, B) \in \mathrm{FORM}_{\mathbf{L}}$ where $\Box \in \{\wedge, \vee, \supset, \equiv\}$.
8. $(\forall\, x\, A)_{\mathbf{M}}[s] = \mathbf{T}$ if $A_{\mathbf{M}}[s[x := a]] = \mathbf{T}$ for all $a \in M$ for $\forall\, x\, A \in \mathrm{FORM}_{\mathbf{L}}$. Otherwise, $(\forall\, x\, A)_{\mathbf{M}}[s] = \mathbf{F}$.
9. $(\exists\, x\, A)_{\mathbf{M}}[s] = \mathbf{T}$ if $A_{\mathbf{M}}[s[x := a]] = \mathbf{T}$ for some $a \in M$ for $\exists\, x\, A \in \mathrm{FORM}_{\mathbf{L}}$. Otherwise, $(\exists\, x\, A)_{\mathbf{M}}[s] = \mathbf{F}$.