Mathematical Logic 2016

Lecture 11: First order logic - Syntax and Semantics

Instructor: Ashutosh Gupta

TIFR, India

Compile date: 2016-09-10

Where are we and where are we going?

We have seen

- Propositional logic

We will see

- First order logic (FOL) syntax
- semantics

Topic 11.1

First order logic (FOL) - syntax

First order logic(FOL)

First order logic(FOL)
$=$
propositional logic + quantifiers over individuals + functions/predicates

"First" comes from this property

Example 11.1

Consider the following argument:
Humans are mortal. Socrates is a human. Therefore, Socrates is mortal.

In symbolic form,
$\forall x.(H(x) \Rightarrow M(x)) \land H(s) \Rightarrow M(s)$

- $H(x) = x$ *is a human*
- $M(x) = x$ *is mortal*
- $s =$ *Socrates*

Connectives and variables

An FOL consists of three disjoint kinds of symbols

- variables
- logical connectives
- non-logical symbols : function and predicate symbols

Variables

We assume that there is a set **Vars** of countably many variables.

- Since **Vars** is countable, we assume that variables are indexed.

$$\textbf{Vars} = \{x_1, x_2, \dots, \}$$

- The variables are just names/symbols without any inherent meaning
- We may also sometimes use x, y, z to denote the variables

Now forget all the definitions of the propositional logic. We will redefine everything and the new definitions will subsume the PL definitions.

Logical connectives

The following are a finite set of symbols that are called logical connectives.

formal name	symbol	read as	
true	$\top$	top	0-ary
false	$\bot$	bot	
negation	$\neg$	not	unary
conjunction	$\wedge$	and	binary
disjunction	$\vee$	or	
implication	$\Rightarrow$	implies	
exclusive or	$\oplus$	xor	
equivalence	$\Leftrightarrow$	iff	
equality	$\approx$	equals	binary predicate
existential quantifier	$\exists$	there is	quantifiers
universal quantifier	$\forall$	for each	
open parenthesis	(		punctuation
close parenthesis	)		
comma	,		

Non-logical symbols

FOL is a parameterized logic

The parameter is a signature $\mathbf{S} = (\mathbf{F}, \mathbf{R})$, where

- **F** is a set of function symbols and
- **R** is a set of predicate symbols.

Each symbol has arity ≥ 0

F and **R** may either be finite or infinite.

Each **S** defines an FOL.
We say, consider an FOL with signature $\mathbf{S} = (\mathbf{F}, \mathbf{R})$ ...

We write $f/n \in \mathbf{F}$ and $P/k \in \mathbf{R}$ to explicitly state the arity

With $n = 0$, f is called constant
With $k = 0$, P is called propositional variable

Syntax : terms

Definition 11.1

For signature $\mathbf{S} = (\mathbf{F}, \mathbf{R})$, $\mathbf{S}$*-terms* $T_{\mathbf{S}}$ *are given by the following grammar:*

$$t \triangleq x \mid f(\underbrace{t, \dots, t}_{n}),$$

where $x \in \mathbf{Vars}$ *and* $f/n \in \mathbf{F}$.

Example 11.2

Consider $\mathbf{F} = \{c/0, f/1, g/2\}$.
The following are terms

- $f(x_1)$
- $g(f(c), g(x_2, x_1))$
- c
- x_1

You may be noticing some similarities between variables and constants

Some notation:

- Let $\vec{t} \triangleq t_1, .., t_n$

Syntax: atoms and formulas

Definition 11.2

S-*atoms* $A_{\mathbf{S}}$ *are given by the following grammar:*

$$a \triangleq P(\underbrace{t,\ldots,t}_{n}) \mid t \approx t \mid \bot \mid \top,$$

where $P/n \in \mathbf{R}$.

Definition 11.3

S-*formulas* $\mathbf{P_S}$ *are given by the following grammar:*

$$F \triangleq a \mid \neg F \mid (F \wedge F) \mid (F \vee F) \mid (F \Rightarrow F) \mid (F \Leftrightarrow F) \mid (F \oplus F) \mid \forall x.(F) \mid \exists x.(F)$$

where $x \in \mathbf{Vars}$.

Example 11.3

Consider $\mathbf{F} = \{s/0\}$ *and* $\mathbf{R} = \{H/1, M/1\}$
The following is a $(\mathbf{F}, \mathbf{R})$*-formula:* $\forall x.(H(x) \Rightarrow M(x)) \wedge H(s) \Rightarrow M(s)$

Unique parsing

For FOL we will ignore the issue of unique parsing,

and assume

all the necessary precedence and associativity orders are defined

for ensuring human readability and unique parsing.

Some terminology

We may write some functions and predicates in infix notation.
For example, we may write $+(a, b)$ as $a + b$ and similarly $<(a, b)$ as $a < b$.

We may not mention **S** if from the context the signature is clear.

Since we know arity of each symbol, we need not write "," "(", and ")" to write a term unambiguously.
For example, $f(g(a, b), h(x), c)$ can be written as *fgabhxc*.

Definition 11.4
subterms and subformulas are naturally defined.

Closed terms and quantifier free

Definition 11.5

A closed term is a term without variable. Let $\hat{T}_{\mathbf{S}}$ be the set of closed $\mathbf{S}$-terms. Sometimes closed terms are also referred as ground terms.

Definition 11.6

A formula F is quantifier free if there is no quantifier in F.

Free variables

Definition 11.7

A variable $x \in \textbf{Vars}$ is free in formula F if

- $F \in A_{\mathbf{S}}$: *x occurs in F,*
- $F = \neg G$: *x is free in G,*
- $F = G \circ H$: *x is free in G or H, for some binary operator $\circ$, and*
- $F = \exists y.G$ *or* $F = \forall y.G$: *x is free in G and* $x \neq y$.

Let $FV(F)$ denote the set of free variables in F.

Exercise 11.1

Is x free?

- $H(x)$ ✓
- $H(y)$ ✗
- $\forall x.H(x)$ ✗
- $x \approx y \Rightarrow \exists x.G(x)$ ✓

Sentence

Definition 11.8

*A variable $x \in$ **Vars** is bounded in formula F if x occurs in F and x is not free.*
In $\forall x.G$ ($\exists x.G$), we say the quantifier $\forall x$ ($\exists x$)has scope G and bounds x.

Definition 11.9

A formula F is a sentence if it has no free variable.

Exercise 11.2

Is the following formula a sentence?

- $H(x)$ ✗
- $\forall x.H(x)$ ✓
- $x = y \Rightarrow \exists x.G(x)$ ✗
- $\forall x.\exists y.\ x = y \Rightarrow \exists x.G(x)$ ✓

Topic 11.2

FOL - semantics

Semantics : models

Definition 11.10

For signature $\mathbf{S} = (\mathbf{F}, \mathbf{R})$, *a* **S***-model* m *is a*

$$(D_m; \{f_m : D_m^n \rightarrow D_m | f/n \in \mathbf{F}\}, \{P_m \subseteq D_m^n | P/n \in \mathbf{R}\}),$$

where D_m *is a nonempty set. Let* **S***-Mods denotes the set of all* **S***-models.*

Some terminology

- D_m is called domain of m.
- f_m assigns meaning to f under model m.
- Similarly, P_m assigns meaning to P under model m.

Example 11.4 (Running example)

Consider $\mathbf{S} = (\{\cup/2\}, \{\in /2\})$.
$m = (\mathbb{N}; \cup_m = max, \in_m = \{(i,j) | i < j\})$ *is a* **S***-model.*

Commentary: Models are also known as interpretation/structure.

Semantics: assignments

Definition 11.11

An assignment is a map $\nu : \mathbf{Vars} \to D_m$

Semantics: term value

Definition 11.12

For a model m and assignment ν, we define $m^\nu : T_{\mathbf{S}} \rightarrow D_m$ as follows.

$$m^\nu(x) \triangleq \nu(x) \qquad\qquad x \in \mathbf{Vars}$$

$$m^\nu(f(t_1, \dots, t_n)) \triangleq f_m(m^\nu(t_1), \dots, m^\nu(t_n))$$

Definition 11.13

Let t be a closed term. $m(t) \triangleq m^\nu(t)$ *for any* ν*.*

Example 11.5

Consider assignment $\nu = \{x \mapsto 2, y \mapsto 3\}$ *and term* $\cup(x, y)$*.*

$m^\nu(\cup(x, y)) = max(2, 3) = 3$

Semantics: satisfaction relation

Definition 11.14

We define the satisfaction relation $\models$ among models, assignments, and formulas as follows

$$
\begin{array}{lll}
m,\nu \models \top & & \\
m,\nu \models P(t_1,\dots,t_n) & \textit{if} & (m^\nu(t_1),\dots,m^\nu(t_n)) \in P_m \\
m,\nu \models t_1 \approx t_2 & \textit{if} & m^\nu(t_1) = m^\nu(t_n) \\
m,\nu \models \neg F & \textit{if} & m,\nu \not\models F \\
m,\nu \models F_1 \lor F_2 & \textit{if} & m,\nu \models F_1 \textit{ or } m,\nu \models F_2 \\
 & & \textit{skipping other boolean connectives} \\
m,\nu \models \exists x.F & \textit{if} & \textit{there is } u \in D_m : m,\nu[x \mapsto u] \models F \\
m,\nu \models \forall x.F & \textit{if} & \textit{for each } u \in D_m : m,\nu[x \mapsto u] \models F
\end{array}
$$

Exercise 11.3

Consider sentence $F = \exists x.\forall y.\ \neg y \in x$ (what does it say to you!)
Use m and ν from previous example. Does $m,\nu \models F$?

Satisfiable, true, valid, and unsatisfiable

We say

- F is *satisfiable* if there are m and ν such that $m, \nu \models F$
- Otherwise, F is called unsatisfiable
- F is *true* in m ($m \models F$) if for all ν we have $m, \nu \models F$
- F is *valid* ($\models F$) if for all ν and m we have $m, \nu \models F$

If F is a sentence, ν has no influence in the satisfaction relation.(why?)

For sentence F, we say

- F is *true* in m if $m \models F$
- Otherwise, F is *false* in m.

Example: satisfiability

Example 11.6

Consider $\mathbf{S} = (\{s/1, +/2\}, \{\})$ *and formula* $\exists z. s(x) + y \approx s(z)$

Consider model $m = (\mathbb{N}; succ, +^{\mathbb{N}})$ *and assignment* $\nu = \{x \mapsto 3, y \mapsto 2\}$

$m^{\nu}(s(x) + y) = m^{\nu}(s(x)) +^{\mathbb{N}} m^{\nu}(y) = succ(m^{\nu}(x)) +^{\mathbb{N}} 2 = succ(3) + 2 = 6$

$m^{\nu[z \mapsto 5]}(s(x) + y) = m^{\nu}(s(x) + y) = 6$ *//Since z does not occur in the term*

$m^{\nu[z \mapsto 5]}(s(z)) = 6$

Therefore, $m, \nu[z \mapsto 5] \models s(x) + y \approx s(z)$.

$m, \nu \models \exists z. s(x) + y \approx s(z)$.

Extended satisfiability

We extend the usage of $\models$.

Definition 11.15

Let Σ be a (possibly infinite) set of formulas.
$m, \nu \models \Sigma$ *if* $m, \nu \models F$ *for each* $F \in \Sigma$.

Definition 11.16

Let M be a (possibly infinite) set of models.
$M \models F$ *if for each* $m \in M$, $m \models F$.

Implication and equisatisfiablity

Definition 11.17

Let Σ be a (possibly infinite) set of formulas.
$\Sigma \models F$ *if for each model m and assignment ν if $m, \nu \models \Sigma$ then $m, \nu \models F$.*

$\Sigma \models F$ is read Σ implies F. If $\{G\} \models F$ then we may write $G \models F$.

Definition 11.18

Let $F \equiv G$ if $G \models F$ and $F \models G$.

The above are semantic definitions.
Later, we will see the connection between logical connective $\Rightarrow$ and semantic implication $\models$.
We also need to prove that $\equiv$ are closely related to $\Leftrightarrow$.

The above definitions may appear to be abuse of notation.

Topic 11.3

Examples

Example: non-standard models

Example 11.7

Consider $\mathbf{S} = (\{\mathbf{0}/0, s/1, +/2\}, \{\})$ *and formula* $\exists z. s(x) + y \approx s(z)$

Unexpected model: *Let* $m = (\{a, b\}^*; \epsilon, append\_a, concat)$.

- *The domain of* m *is the set of all strings over alphabet* $\{a, b\}$.
- $append\_a$*: appends a in the input and*
- $concat$*: joins two strings.*

Let $\nu = \{x \mapsto ab, y \mapsto ba\}$.
Since $m, \nu[z \mapsto abab] \models s(x) + y \approx s(z)$,
$m, \nu \models \exists z. s(x) + y \approx s(z)$.

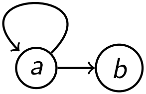
Exercise 11.4

- *Show* $m, \nu[y \mapsto bb] \not\models \exists z. s(x) + y \approx s(z)$
- *Give an assignment* ν *s.t.* $m, \nu \models x \not\approx 0 \Rightarrow \exists y.\ x \approx s(y)$.
  Show $m \not\models \forall x.\ x \not\approx 0 \Rightarrow \exists y.\ x \approx s(y)$.

Example: graph models

Example 11.8

Consider $\mathbf{S} = (\{\}, \{E/2\})$ *and* $m = (\{a, b\}; \{(a, a), (a, b)\})$.
m may be viewed as the following graph.



$$m, \{x \to a\} \models E(x, x) \wedge \exists y.(E(x, y) \wedge \neg E(y, y))$$

Exercise 11.5

Give another model and assignment that satisfies the above formula

Example : counting

Example 11.9

Consider $\mathbf{S} = (\{\}, \{E/2\})$
The following sentence is false in all the models with one element domain

$$\forall x.\neg E(x,x) \wedge \exists x \exists y.E(x,y)$$

Exercise 11.6

a. Give a sentence that is true only in a model that has more than two elements in its domains
b. Give a sentence that is true only in infinite models
c. Does the negation of the sentence in b satisfies only finite models.

Importance of $\approx$

Some of the formalisms do not include $\approx$ in their basic introduction of FOL, because $\approx$ makes the proofs hairy. They treat $\approx$ separately.

$\approx$ also makes the automation of FOL complicated, which we will soon.

Exercise 11.7

Give a sentence that is true only in models with less than or equal to two element domains

Topic 11.4

Some properties of models

Homomorphisms of models

Definition 11.19

Consider $\mathbf{S} = (\mathbf{F}, \mathbf{R})$*. Let* m *and* m' *be* $\mathbf{S}$*-models.*
A function $h : D_m \to D_{m'}$ *is a homomorphism of* m *into* m' *if the following holds.*

- *for each* $f/n \in \mathbf{F}$*, for each* $(d_1, .., d_n) \in D_m^n$

$$h(f_m(d_1, .., d_n)) = f_{m'}(h(d_1), .., h(d_n))$$

- *for each* $P/n \in \mathbf{R}$*, for each* $(d_1, .., d_n) \in D_m^n$

$$(d_1, .., d_n) \in P_m \quad \textit{iff} \quad (h(d_1), .., h(d_n)) \in P_{m'}$$

Definition 11.20

A homomorphism h *of* m *into* m' *is called isomorphism if* h *is one-to-one.*
m *and* m' *are called isomorphic if an* h *exists that is also onto.*

Example : homomorphism

Example 11.10

Consider $\mathbf{S} = (\{+/2\}, \{\})$.

Consider $m = (\mathbb{N}, +^{\mathbb{N}})$ *and* $m = (\mathcal{B}, \oplus^{\mathcal{B}})$,

$h(n) = n \ mod \ 2$ *is a homomorphism of* m *into* m'.

Homomorphism theorem for terms and ≈-less QF formulas

Theorem 11.1

Let h be a homomorphism of m into m'. Let ν be an assignment.

1. *For each term t, $h(m^{\nu}(t)) = m'^{(\nu \circ h)}(t)$*
2. *If formula F is QF and has no symbol "≈"*

$$m^{\nu} \models F \quad \textit{iff} \quad m'^{(\nu \circ h)} \models F$$

Proof.

Simple structural induction. □

Exercise 11.8

For a QF formula F that may have symbol "≈", show

$$\textit{if} \quad m^{\nu} \models F \quad \textit{then} \quad m'^{(\nu \circ h)} \models F$$

Why the reverse direction does not work?

Homomorphism theorem with $\approx$

Theorem 11.2

Let h be a homomorphism of m into m'. Let ν be an assignment. If h is isomorphism then the reverse implication also holds for formulas with "$\approx$"

Proof.

Let us suppose $m'^{(\nu\circ h)} \models s \approx t$.
Therefore, $m'^{(\nu\circ h)}(s) = m'^{(\nu\circ h)}(t)$.
Therefore, $h(m^{\nu}(s)) = h(m^{\nu}(t))$.
Due to the one-to-one condition of h, $m^{\nu}(s) = m^{\nu}(t)$.
Therefore, $m^{\nu} \models s \approx t$. □

Exercise 11.9

For a formula F without symbol "$\approx$", show

$$\textit{if} \quad m'^{(\nu\circ h)} \models F \quad \textit{then} \quad m^{\nu} \models F.$$

Why the reverse direction does not work?

Commentary: Note that that implication direction his switch from the previous exercise.

Homomorphism theorem with quantifiers

Theorem 11.3

Let h be a isomorphism of m into m'. Let ν be an assignment.
If h is also onto then the reverse direction also holds for the quantified formulas

Proof.

Let us assume, $m^{\nu} \models \forall x.F$.
Choose $d' \in D_{m'}$.
Since h is onto, there is a d such that $d = h(d')$.
Therefore, $m^{\nu[x \mapsto d]} \models F$.
Therefore, $m'^{\nu[x \mapsto d']} \models F$.
Therefore, $m'^{(\nu \circ h)} \models \forall x.\ F$. □

Theorem 11.4

If m and m' are isomorphic then for all sentences F,

$$m \models F \quad iff \quad m' \models F.$$

Commentary: The reverse direction of the above theorem is not true.

Topic 11.5

Problems

FOL to PL

Exercise 11.10

Give the restrictions on FOL such that it becomes a propositional logic. Give an example of FOL model of a non-trivial propositional formula.

Valid formulas

Exercise 11.11

Prove/Disprove the following formulas are valid.

- $\forall x.P(x) \Rightarrow P(c)$
- $\forall x.(P(x) \Rightarrow P(c))$
- $\exists x.(P(x) \Rightarrow \forall x.P(x))$
- $\exists y \forall x.R(x,y) \Rightarrow \forall x \exists y.R(x,y)$
- $\forall x \exists y.R(x,y) \Rightarrow \exists y \forall x.R(x,y)$

Distributively

Exercise 11.12

Show the validity of the following formulas.

1. $\neg\forall x.\ P(x) \Leftrightarrow \exists x.\ \neg P(x)$
2. $(\forall x.\ (P(x) \land Q(x))) \Leftrightarrow \forall x.\ P(x) \land \forall x.\ Q(x)$
3. $(\exists x.\ (P(x) \lor Q(x))) \Leftrightarrow \exists x.\ P(x) \lor \exists x.\ Q(x)$

End of Lecture 11