CS228 Logic for Computer Science 2020

Lecture 14: First-order logic - Syntax

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2020-07-28

Topic 14.1

First-order logic (FOL)

First-order logic(FOL)

First-order logic(FOL)

=

propositional logic + quantifiers over individuals + functions/predicates

"First" comes from this property

Example 14.1

Consider 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*

A note on FOL syntax

The FOL syntax may appear non-intuitive and cumbersome.

FOL requires getting used to it like many other concepts such as complex numbers.

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.

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

- 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	$=$	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

- $\mathbf{F}$ is a set of function symbols and
- $\mathbf{R}$ is a set of predicate symbols (aka relational symbols).

Each symbol is associated with an arity ≥ 0.

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

Example 14.2

We may have $\mathbf{F} = \{c/0, f/1, g/2\}$ *and* $\mathbf{R} = \{P/0, H/2, M/1\}$.

Example 14.3

We may have $\mathbf{F} = \{+/2, -/2\}$ *and* $\mathbf{R} = \{< /2\}$.

Commentary: We are familiar with predicates, which are the things that are either true or false. However, the functions are the truly novel concept.

Non-logical symbols (contd.)

$\mathbf{F}$ and $\mathbf{R}$ may either be finite or infinite.

Example 14.4

In the propositional logic, $\mathbf{F} = \emptyset$ *and*

$$\mathbf{R} = \{p_1/0, p_2/0, ...\}.$$

Each $\mathbf{S}$ defines an FOL.

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

We may not mention $\mathbf{S}$ if from the context the signature is clear.

Constants and Propositional variable

There are special cases when the arity is zero.

$f/0 \in \mathbf{F}$ is called a constant.

$P/0 \in \mathbf{R}$ is called a propositional variable.

Building FOL formulas

Let us use the ingredients to build the FOL formulas.

It will take a few steps to get there.

- terms
- atoms
- formulas

Syntax : terms

Definition 14.1

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

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

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

Example 14.5

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

Infix notation

We may write some functions and predicates in infix notation.

Example 14.6

we may write $+(a, b)$ *as* $a + b$ *and similarly* $< (a, b)$ *as* $a < b$.

Syntax: atoms

Definition 14.2

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

$$a ::= P(\underbrace{t, \dots, t}_{n}) \mid t = t \mid \bot \mid \top,$$

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

Exercise 14.1

Consider $\mathbf{F} = \{s/0\}$ *and* $\mathbf{R} = \{H/1, M/1\}$
Is the following an atom?

- $H(x)$
- s
- $M(s)$
- $H(M(s))$

Commentary: Remember: you can nest terms but not atoms.

Equaltiy within logic vs. equality outside logic

We have an equality $=$ within logic and the other we use to talk about logic.

Both are distinct objects.

Some notations use same symbols for both and the others do not to avoid confusion.

Whatever is the case, we must be vary clear about this.

Syntax: formulas

Definition 14.3

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

$$F ::= a \mid \neg F \mid (F \land F) \mid (F \lor F) \mid (F \Rightarrow F) \mid (F \Leftrightarrow F) \mid (F \oplus F) \mid \forall x.(F) \mid \exists x.(F)$$

where $x \in$ **Vars**.

Example 14.7

Consider $\mathbf{F} = \{s/0\}$ *and* $\mathbf{R} = \{H/1, M/1\}$

The following is a $(\mathbf{F}, \mathbf{R})$-*formula:*

$$\forall x.(H(x) \Rightarrow M(x)) \land H(s) \Rightarrow M(s)$$

Commentary: Notice we have dropped some parenthesis. We will not discuss the minimal parenthesis issue at length.

Unique parsing

For FOL we will ignore the issue of unique parsing,

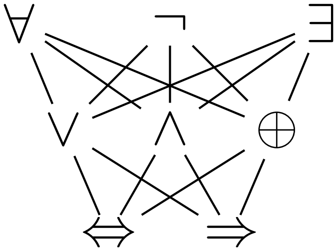
and assume

all the necessary precedence and associativity orders are defined

for ensuring human readability and unique parsing.

Precedence order

We will use the following precedence order in writing the FOL formulas



Example 14.8

The following are the interpretation of the formulas after dropping parenthesis

- $\forall x.H(x) \Rightarrow M(x) = \forall x.(H(x)) \Rightarrow M(x)$
- $\exists z \forall x. \exists y. G(x,y,z) = \exists z.(\forall x.(\exists y.G(x,y,z)))$

Clubbing simlar quantifiers

If we have a chain of same quantifier then we write the quantifier once followed by the list of variables.

Example 14.9

- $\forall z, x.\exists y.G(x, y, z) = \forall z.(\forall x.(\exists y.G(x, y, z)))$
- $\exists z, x, y.G(x, y, z) = \exists z.(\exists x.(\exists y.G(x, y, z)))$

Subterm and subformulas

Definition 14.4

A term t is subterm *of term t', if t is a substring of t'.*

Exercise 14.2

- *Is $f(x)$ a subterm of $g(f(x), y)$?*
- *Is c a subterm of c?*
- *x is a subterm of $f(x)$*

Definition 14.5

A formula F is subformula *of formula F', if F is a substring of F'.*

Example 14.10

- *$G(x, y, z)$ is a subformula of $\forall z, x.\exists y.G(x, y, z)$*
- *$P(c)$ is a subformula of $P(c)$*
- *$\exists y.G(x, y, z)$ is a subformula of $\forall z, x.\exists y.G(x, y, z)$*

Closed terms and quantifier free

Definition 14.6

A closed term is a term without any variable.
Let $\hat{T}_{\mathbf{S}}$ be the set of closed $\mathbf{S}$-terms.

Sometimes closed terms are also referred as ground terms.

Example 14.11

Let $\mathbf{F} = \{f/1, c/0\}$.
$f(c)$ *is a closed term, and* $f(x)$ *is not, where* x *is a variable.*

Exercise 14.3

Is the following term closed with respect to $\mathbf{F} = \{f/1, g/2, c/0\}$*?*

- $g(c, y)$
- x
- c
- $f(g(c, c))$

Quantifier-free

Definition 14.7

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

Example 14.12

$H(c)$ is quantifier-free formula and $\forall x.H(x)$ is not a quantifier-free formula.

Exercise 14.4

For signature $(\{f/1, c/0\}, \{H/1\})$, which of the following are quantifier-free?

- $\forall x.H(y)$
- $H(y) \vee \perp$
- $f(c)$
- $H(f(c))$

Free variables

Definition 14.8

*A variable $x \in$ **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 14.5

Is x free?

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

Bounded variables

Definition 14.9

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

- $F = \neg G$: *x is bounded in G,*
- $F = G \circ H$: *x is bounded in G or H, for some binary operator $\circ$, and*
- $F = \exists y.G$ *or* $F = \forall y.G$: *x is bounded in G or x is equal to y.*

Let $bnd(F)$ denote the set of bounded variables in F.

Exercise 14.6

Is x bounded?

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

Sentence

Definition 14.10

In $\forall x.(G)$, we say the quantifier $\exists x$ has scope G and bounds x.
In $\exists x.(G)$, we say the quantifier $\exists x$ has scope G and bounds x.

Definition 14.11

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

Exercise 14.7

Is the following formula a sentence?

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

Attendance quiz

For signature $(\{f/1, g/2, c/0\}, \{H/1\})$, which of the following hold?

x is a term
$f(c)$ is a ground term
$H(c)$ is an atom
$\forall x. \neg H(x)$ is a formula
$\forall x. \neg H(x)$ is a sentence
$\forall x. \neg H(y)$ is not a sentence
$f(c)$ is a closed term
$f(x)$ is not a closed term
Variable x is bounded in $\forall x. \neg H(x)$
Variable x is free in $\forall y. \neg H(x)$

x is an atom
$f(c)$ is not a ground term
$H(c)$ is a term
$\forall x. \neg H(x)$ is an atom
$\forall x. \neg H(x)$ is not a sentence
$\forall x. \neg H(y)$ is a sentence
$f(c)$ is not a closed term
$f(x)$ is a closed term
Variable x is free in $\forall x. \neg H(x)$
Variable x is bounded in $\forall y. \neg H(x)$

Semantics : structures

Definition 14.12

For signature $\mathbf{S} = (\mathbf{F}, \mathbf{R})$*, a* **S***-structure* 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***-structures.*

Some terminology

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

Topic 14.2

Problems

Exercise : compact notation for terms

Since we know arity of each symbol, we need not write "," "(", and ")" to write a term unambiguously.

Example 14.13

$f(g(a,b),h(x),c)$ can be written as $fgabhxc$.

Exercise 14.8

*Consider $\mathbf{F} = \{f/3, g/2, h/1, c/0\}$ and $x, y \in$ **Vars**.*
Insert parentheses at appropriate places in the following if they are valid term.

- $hc =$
- $gxc =$
- $fhxhyhc =$
- $fx =$

Exercise 14.9

Give an algorithm to insert the parentheses

Commentary: We will not use the compact notation in the course. It makes the formulas very difficult to read.

Exercise: DeBruijn index of quantified variables

DeBruijn index is a method for representing formulas without naming the quantified variables.

Definition 14.13

Each De Bruijn index is a natural number that represents an occurrence of a variable in a formula, and denotes the number of quantifiers that are in scope between that occurrence and its corresponding quantifier.

Example 14.14

We can write $\forall x.H(x)$ *as* $\forall.H(1)$. 1 *is indicating the occurrence of a quantified variable that is bounded by the closest quantifier. More examples.*

- $\exists y \forall x.M(x,y) = \exists\forall.M(1,2)$
- $\exists y \forall x.M(y,x) = \exists\forall.M(2,1)$
- $\forall x.(H(x) \Rightarrow \exists y.M(x,y)) = \forall.(H(1) \Rightarrow \exists.M(2,1))$

Exercise 14.10

Give an algorithm that translates FOL formulas into DeBurjin indexed formulas.

Drinker paradox

Exercise 14.11

Prove

There is someone x such that if x drinks, then everyone drinks.

Let $D(x) \triangleq x$ drinks. Formally

$$\exists x.(D(x) \Rightarrow \forall x.\ D(x))$$

https://en.wikipedia.org/wiki/Drinker_paradox

End of Lecture 14