CS 208 : Automata Theory and Logic DATE 15/04/2024

Lecture - 38

Topic: First Order Logic

Scribed by: Arohan Hazarika (22B3948)
Checked and compiled by:

A Brief Introduction

First-order logic (FOL), also known as first-order predicate logic, is an extension of propositional logic that allows for more complex statements involving quantifiers such as $\forall$ and $\exists$, as well as variables and predicates. In contrast to propositional logic, which deals with simple propositions and their logical connectives, first-order logic enables us to express relationships between objects, properties of objects, and the quantification of variables over domains of discourse.

Notation

The symbols are the basic syntactic elements of FOL and are used to write statements in shorthand notation.

- **Variables:** They are usually denoted by smalls $(p, q, r, x, y, z, ...etc)$. Unlike propositional logic, the variables can take any values depending on the underlying set and not just true or false

- **Constants:** They are specific elements of the underlying set and again unlike propositional logic, they can take any values depending on the underlying set

- **Function Symbols:** Each function symbol is associated with an **arity** (a positive integer specifying the number of arguments). For example, $f(x, y, c)$ is a function where x, y are variables and c is a constant.

- **Predicate Symbols:** Every predicate symbol, e.g. $P, Q, R...$ maps variables, constants and functions from the underlying Universal set to True or False. It is also called as "Predicate Calculus"

- **Fixed Symbols:** These are similar to symbols used in propositional logic like $\wedge, \vee, \neg, \rightarrow$ etc alongwith two new symbols $\forall$ and $\exists$

Equality in FOL

It is a very special binary predicate which is widely used in math and logic is known as **Equality** symbol. It has only two possible outcomes, either true or false.

Syntax of FOL

There are two classes of syntactic objects: *terms* and *formulas*

- **Terms:** Every variable, constant and function symbol which takes variables as arguments is a term. Example: $f(t_1, t_2, t_3, ...)$ with $t_1, t_2, ...$ as variables and f as a function symbol is a term. Note that constants are also functions.
- **Atomic Formulas:** Every predicate symbol with arguments as variables and constant is an atomic formula. Example: $P(t_1, t_2, t_3, ...)$ with $t_1, t_2, ...$ as variables and P as a predicate is an atomic formula. Predicate basically maps values from underlying set to true or false.

Rules for formuling formulas

- Every atomic formula is a formula
- If φ is a formula, so are $\neg\varphi$ and (φ)
- If φ_1 and φ_2 are formulas, so is $\varphi_1 \wedge \varphi_2$
- If φ is a formula, so is $\exists x \varphi$ for any variable x

Primitive fixed symbols

Symbols which are sufficient to express any other fixed symbol. Examples: $\wedge$,$\neg$,$\exists$ is a set of primitive fixed symbols
Note: Formulas with other fixed symbols definable in terms of formulas with primitive symbols.

- $\varphi_1 \vee \varphi_2 \triangleq \neg(\neg\varphi_1 \wedge \neg\varphi_2)$
- $\varphi_1 \rightarrow \varphi_2 \triangleq \neg\varphi_1 \vee \varphi_2$
- $\varphi_1 \leftrightarrow \varphi_2 \triangleq (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1)$
- $\forall x\ \varphi \triangleq \neg(\exists x \neg\varphi)$

FOL formulas as strings

- **Alphabets:** Set of variable names, constants, functions, predicates and fixed symbols.
- **Well-formed formula:** Strings formed according to above rules mentioned.

  Example: $\forall x_1(\forall x_2(((x_1 = a) \vee (x_1 = b)) \wedge \neg(f(x_2) = f(x_1))))$ is well formed but

  $\forall(\forall x_1(x_1 = ab)\neg()x_2)$ is not well formed

  Well-formed formulas can be represented using parse trees similar to what we did in

  Propositional Logic

Vocabulary

Set of constants, functions, predicates, e.g. $\{a, b, f, =\}$

Free Variables in a Formula

Free variables are those that are not quantified in a formula. Let free(φ) denote the set of free variables in φ.

- If φ is an atomic formula, free(φ)=$\{x \mid \text{x occurs in in } \varphi\}$
- If $\varphi = \neg\psi$ or $\varphi = (\psi)$, then free(φ) = free(ψ).
- If $\varphi = \varphi_1 \wedge \varphi_2$, then free($\varphi$) = free($\varphi_1$) $\cup$ free(φ_2).
- If $\varphi = \exists x \varphi_1$, then free($\varphi$) = free($\varphi_1$) $\setminus \{x\}$.

If φ has free variables $\{x, y\}$, we write $\varphi(x, y)$.
A formula with no free variables is a sentence, e.g., $\exists x \forall y f(x) = y$.
Example:

$$\begin{aligned}
&\text{free}((\exists x P(x,y)) \wedge (\forall y Q(x,y))) && (1)\\
&= \text{free}(\exists x P(x,y)) \cup \text{free}(\forall y Q(x,y)) && (2)\\
&= (\text{free}(P(x,y)) \setminus \{x\}) \cup (\text{free}(Q(x,y)) \setminus \{y\}) && (3)\\
&= \{x,y\} \setminus \{x\} \cup \{x,y\} \setminus \{y\} && (4)\\
&= \{x,y\} && (5)
\end{aligned}$$

Bound Variables in a Formula

Bound variables are those that are quantified in a formula. Let bnd(φ) denote the set of bound variables in φ.

- If φ is an atomic formula, bnd(φ) = $\emptyset$.
- If $\varphi = \neg\psi$ or $\varphi = (\psi)$, then bnd(φ) = bnd(ψ).
- If $\varphi = \varphi_1 \wedge \varphi_2$, then bnd($\varphi$) = bnd($\varphi_1$) $\cup$ bnd(φ_2).
- If $\varphi = \exists x \varphi_1$, then bnd($\varphi$) = bnd($\varphi_1$) $\cup \{x\}$.

Example:

$$\begin{aligned}
&\text{bnd}((\exists x P(x,y)) \wedge (\forall y Q(x,y))) && (6)\\
&= \text{bnd}(\exists x P(x,y)) \cup \text{bnd}(\forall y Q(x,y)) && (7)\\
&= \text{bnd}(P(x,y)) \cup \{x\} \cup \text{bnd}(Q(x,y)) \cup \{y\} && (8)\\
&= \emptyset \cup \{x\} \cup \emptyset \cup \{y\} && (9)\\
&= \{x\} \cup \{y\} && (10)\\
&= \{x,y\} && (11)
\end{aligned}$$

Note: free(φ) and bnd(φ) are not complements.

Substitution in FOL

Suppose $x \in \text{free}(\varphi)$ and t is any term. We can substitute x with t wherever x is free in our formula φ provided free variables in t remain free in the resulting formula. This can be better explained by the following example:

$$\begin{gathered} \varphi \triangleq (\exists y R(x,y)) \vee (\forall x R(z,x)), \text{ and } t \text{ is } f(z,x). \\ f(z,x) \text{ is free for } x \text{ in } \varphi, \text{ but } f(y,x) \text{ is not.} \end{gathered} \tag{12}$$

Note: Term t is free for x in φ if no free occurrence of x in φ is in the scope of $\forall y$ or $\exists y$ for any variable y occurring in t.

For the above example, $\varphi[f(z,x)/x] \triangleq (\exists y R(f(z,x),y)) \vee (\forall x R(z,x)),$ here t is $f(z,x)$.