

# Lecture 13: First-Order Logic

10:30 AM, Mar 10, 2009

## Contents

<b>1</b>	<b>Overview</b>	<b>1</b>
<b>2</b>	<b>Syntax</b>	<b>1</b>
<b>3</b>	<b>Semantics</b>	<b>2</b>
<b>4</b>	<b>Substitution</b>	<b>3</b>
<b>5</b>	<b>Logical Entailment</b>	<b>4</b>
5.1	Logical Inference: Natural Deduction . . . . .	4
5.2	Logical Inference: Modus Ponens, Resolution . . . . .	4

## 1 Overview

The language of predicate calculus as defined in Lecture 11 did not consider variables or quantifiers. This lecture is concerned with first-order logic, which incorporates quantification over objects using variables as placeholders. Higher-order logics allow for quantification over functions or relations.

The universal quantifier  $\forall$  can be interpreted as conjunction, with a possibly infinite number of conjuncts. In finite domains, universal quantification and conjunction are equivalent: *e.g.*, in the finite domain  $D = \{\text{SKY}, \text{MOUNTAINS}\}$ , the following formula holds:  $\forall x \text{BLUE}(x) \leftrightarrow \text{BLUE}(\text{SKY}) \wedge \text{BLUE}(\text{MOUNTAINS})$ . But in infinite domains, universal quantification extends conjunction: *e.g.*, the following formula holds in the infinite domain  $D = \mathbb{Z}$ :  $\forall x \text{ODD}(x) \vee \text{EVEN}(x)$ .

Analogously, the existential quantifier  $\exists$  can be interpreted as disjunction, with a possibly infinite number of disjuncts. As above, existential quantification and disjunction are equivalent in finite domains: *e.g.*, in domain  $D = \{\text{SKY}, \text{GRASS}\}$ , the following formula holds:  $\exists x \text{GREEN}(x) \leftrightarrow \text{GREEN}(\text{SKY}) \vee \text{GREEN}(\text{GRASS})$ . But in infinite domains, existential quantification extends disjunction: *e.g.*, the following formula holds in the infinite domain  $D = \mathbb{N}$ :  $\exists x, \forall y \neg \text{SUCC}(x, y)$ .

## 2 Syntax

First-order logic generalizes predicate calculus with variables and quantifiers. Like propositional logic and predicate calculus, a set of syntactic rules govern the construction of terms and formulas in first-order logic. The alphabet of first-order logic extends the alphabet of predicate calculus with the quantification symbols  $\{\forall, \exists\}$  and the variables  $\{x, y, z, \dots\}$ .

The following definition extends the inductive definition of the terms of predicate calculus to first-order logic:

- variables  $x, y, z, \dots$  are terms

Similarly, the following definition extends the inductive definition of the formulas of predicate calculus to first-order logic:

- if  $x$  is a variable and  $\phi$  is a formula, then
  - $\forall x \phi$  is a formula
  - $\exists x \phi$  is a formula

### 3 Semantics

The semantics of first-order logic associate meanings with formulas by extending the notion of interpretation in predicate calculus (and propositional logic) to a domain-mapping-assignment triple  $\langle D, M, \alpha \rangle$ , where  $\alpha$  maps variables to objects in the domain. In first-order logic, the mapping from terms to objects is defined (inductively) as follows:

- for all variables  $x$ ,  $\mathcal{I}[x] = \alpha(x)$
- for all constant symbols  $a$ ,  $\mathcal{I}[a] = a^M$
- for all terms  $t_1, \dots, t_n$  and  $n$ -ary function symbols  $f$ 
  - $\mathcal{I}[f(t_1, \dots, t_n)] = f^M(\mathcal{I}[t_1], \dots, \mathcal{I}[t_n])$

The definition of entailment in first-order logic extends that of the predicate calculus. For atomic formulas  $P(t_1, \dots, t_n)$ , as before,

- $\mathcal{I} \models P(t_1, \dots, t_n)$  iff  $(\mathcal{I}[t_1], \dots, \mathcal{I}[t_n]) \in P^M$

For complex first-order formulas other than those that involve quantification, the definition is as it was before: *e.g.*,  $\langle \mathcal{I}, \alpha \rangle \models \phi \vee \psi$  iff  $\langle \mathcal{I}, \alpha \rangle \models \phi$  or  $\mathcal{I} \models \psi$ . In the remaining cases, the definition is as follows:

- $\mathcal{I} \models \forall x \phi$  iff for all  $d \in D$ ,  $\mathcal{I}\{d/x\} \models \phi$
- $\mathcal{I} \models \exists x \phi$  iff there exists  $d \in D$  s.t.  $\mathcal{I}\{d/x\} \models \phi$

Here  $\mathcal{I}\{d/x\}$  is an abbreviation for  $\langle D, M, \alpha\{d/x\} \rangle$ , where the assignment  $\alpha\{d/x\}$  maps  $x$  to  $d$ , but otherwise agrees with  $\alpha$ : i.e.,

$$\alpha\{d/x\}(y) = \begin{cases} \alpha(y) & \text{if } y \neq x \\ d & \text{otherwise} \end{cases}$$

It is straightforward to extend the notions of satisfiability, validity, and unsatisfiability to first-order logic.

**Example:** Given the infinite alphabet  $\mathcal{A} = \{\text{ADD}, \text{MULT}, =, 0, 1, <, x_1, x_2, \dots\}$ , consider the interpretation  $\mathcal{N}$ , where the domain  $D = \mathbb{N}$ , the nonvariable symbols in  $\mathcal{A}$  are interpreted as usual, and  $\alpha(x_n) = 2n$ , for all  $n \in \mathbb{N}$ . Under this interpretation,  $\phi \equiv \text{MULT}(x_1, \text{ADD}(x_2, x_3)) = x_4$  signifies  $2(4 + 6) = 8$ . Of course,  $\mathcal{N} \not\models \phi$ . On the other hand,  $\mathcal{N} \models \text{MULT}(x_1, x_2) = x_4$ .

## 4 Substitution

A **substitution** is a function from variables to terms: *i.e.*,  $\sigma = \{t_1/x_1, \dots, t_n/x_n\}$ , where each  $x_i$  is a variable and each  $t_i$  is a term. The set of variables  $\{x_1, \dots, x_n\}$  is the **domain** of  $\sigma$ ; the set of terms  $\{t_1, \dots, t_n\}$  is the **range** of  $\sigma$ .

Before we can define the substitution operation, that is, the result of applying the substitution  $\sigma$  to the formula  $\phi$  (abbreviated  $\phi|_\sigma$ ), we need some terminology:

- The **scope** of a quantifier is that part of a formula to which the quantifier applies.
- A **bound** variable is a variable that is within the scope of some quantifier.
- A **free** variable is one that is not bound.
- A **closed** formula, or **sentence**, has no free variables.
- An **open** formula is one that is not closed.
- A **ground** term (or formula), is a term (or formula) without any variables. (Note that all formulas of propositional logic and predicate calculus are sentences, and all terms and formulas of these languages are ground.)

Now, the substitution operation is subject to the following two constraints:

- *Substitutions cannot be made for bound variables.*  
For example, consider the formula  $\phi \equiv \exists y \ y + y = 0$ , which is true in the standard interpretation of the natural numbers (let  $y = 0$ ). After applying the (illegal) substitution  $\phi|_{\{1/y\}}$ , we obtain the sentence  $\exists y \ 1 + 1 = 0$ , which is false.
  - *Substitutions cannot be made that bind variables.*  
Consider the formula  $\phi \equiv \exists y \ y + y = x$ , which states that “ $x$  is even.” After applying the (legal) substitution  $\phi|_{\{1/x\}}$ , we obtain the sentence  $\exists y \ y + y = 1$ , which is false. But after applying the (illegal) substitution  $\phi|_{\{y/x\}}$ , which yields  $\exists y \ y + y = y$ , the meaning changes—the new sentence is true (let  $y = 0$ ).
- Note that variable renaming does not change the meaning of formulas: *e.g.*,  $\exists y \ y + y = 0 \equiv \exists z \ z + z = 0$ . Hence, to preserve meaning, first rename any bound variables whose names appear in the range of a substitution: *e.g.*,  $\phi' \equiv \exists z \ z + z = x|_{\{y/x\}}$ , which reduces to  $\exists z \ z + z = y$ .

These constraints ensure soundness: i.e., no change in meaning. This property is stated formally in the following theorem, known as the Substitution Lemma.

**Theorem:** For all formulas  $\phi$ ,  $\mathcal{I} \models \phi|_{\{t_1/x_1, \dots, t_n/x_n\}}$  iff  $\mathcal{I}\{t_1/x_1, \dots, t_n/x_n\} \models \phi$ .

Given two substitutions  $\sigma$  and  $\tau$ , sequential substitution  $\sigma\tau$  is achieved via **composition** of substitutions, defined as follows:  $\phi|_{\sigma\tau} = (\phi|_{\sigma})|_{\tau}$ . For example, if  $\sigma = \{f(y)/x\}$  and  $\tau = \{b/y\}$ , then  $R(x, y)|_{\sigma\tau} = R(f(b), b)$ . But if  $\theta = \{f(y)/x, b/y\}$ , then  $R(x, y)|_{\theta} = R(f(y), b) \neq R(f(b), b)$ . Note that if  $\sigma$  binds all free variables to ground terms, then  $e|_{\sigma\sigma'} = e|_{\sigma}$ , for all substitutions  $\sigma'$  and all expressions  $e$ : e.g., if  $\sigma = \{f(0)/x, 0/y\}$ , then  $R(x, y)|_{\sigma\tau} = R(f(0), 0)$ .

## 5 Logical Entailment

A variant of Herbrand's theorem holds for **universal** logic, that is, first-order logic formulas with universal quantifiers only. But Herbrand's theorem does not hold for formulas of first-order logic with existential quantifiers.

**Counterexample:** Consider the pair of sentences  $\exists xP(x)$  and  $\neg P(a)$ . These sentences are satisfiable: e.g.,  $D = \{\circ, \diamond\}$ ,  $a^M = \circ$ , and  $P^M = \{\diamond\}$ . But these sentences do not have a Herbrand model. The Herbrand universe  $A = \{a\}$  and the Herbrand base  $B = \{P(a)\}$ . Thus, there are only two Herbrand models:  $P = \emptyset$  and  $P = A$ . But neither of these models satisfy the pair of sentences.

In other words, there is no known semantic technique for deciding logical entailment. This is not surprising, as logical entailment in first-order logic is not **decidable**, meaning there is no effective procedure (of any complexity), and there never will be, that can answer the question "is formula  $\phi$  entailed by a knowledge base KB?"

However, logical entailment in first-order logic is **semi-decidable**. That is, there do exist effective procedures that correctly conclude that a formula  $\phi$  is entailed by a knowledge base KB when in fact this is the case; but when  $\text{KB} \not\models \phi$ , such procedures need not terminate. The only semi-decision procedures known for first-order logic are proof-theoretic.

### 5.1 Logical Inference: Natural Deduction

Additional rules of natural deduction that extend the rules for propositional logic to quantified formulas Table 1. The following restrictions apply:

- ( $\forall I$ )  $a$  does not occur in  $\phi$ , or in any assumptions on which  $\phi|_{\{a/x\}}$  depends
- ( $\exists E$ )  $a$  does not occur in  $\phi$  or  $\psi$ , or in any of the assumptions used in the derivation of  $\psi$  from  $\phi|_{\{a/x\}}$  other than  $\phi|_{\{a/x\}}$  itself

**Gentzen's Soundness Theorem** First-order logic is sound.

**Gödel's Completeness Theorem** First-order logic is complete.

Natural deduction is a sound and complete proof theory for first-order logic.

### 5.2 Logical Inference: Modus Ponens, Resolution

GENERALIZED MODUS PONENS (GMP)

Introduction	Elimination
$\frac{\phi _{\{a/x\}}}{\forall x \phi} (\forall I)$	$\frac{\forall x \phi}{\phi _{\{t/x\}}} (\forall E)$
$\frac{\phi _{\{t/x\}}}{\exists x \phi} (\exists I)$	$\frac{\begin{array}{c} [\phi _{\{a/x\}}] \\ \vdots \\ \psi \end{array}}{\exists x \phi} (\exists E)$

Table 1: Rules of Natural Deduction for Quantified Formulas.

if  $\tau|_{\sigma} = \phi_i|_{\sigma}$ , then

$$\frac{\phi_1 \wedge \dots \wedge \phi_m \rightarrow \psi \quad \chi_1 \wedge \dots \wedge \chi_n \rightarrow \tau}{(\phi_1 \wedge \dots \wedge \phi_{i-1} \wedge \chi_1 \wedge \dots \wedge \chi_n \wedge \phi_{i+1} \wedge \dots \wedge \phi_m \rightarrow \psi)|_{\sigma}} \text{ (GMP)}$$

**Theorem:** GMP is sound.

**Theorem:** GMP is complete for Horn databases.

**Remark:** Not all knowledge bases are convertible to Horn databases.

GENERALIZED RESOLUTION (R)

if  $\tau_j|_{\sigma} = \phi_i|_{\sigma}$ , then

$$\frac{\phi_1 \wedge \dots \wedge \phi_m \rightarrow \psi_1 \vee \dots \vee \psi_n \quad \chi_1 \wedge \dots \wedge \chi_k \rightarrow \tau_1 \vee \dots \vee \tau_l}{(\phi_1 \wedge \dots \wedge \phi_{i-1} \wedge \chi_1 \wedge \dots \wedge \chi_k \wedge \phi_{i+1} \wedge \dots \wedge \phi_m \rightarrow \psi_1 \vee \dots \vee \psi_n \vee \tau_1 \vee \dots \vee \tau_{j-1} \vee \tau_{j+1} \vee \dots \vee \tau_l)|_{\sigma}} \text{ (R)}$$

**Theorem:** Resolution is sound.

**Theorem:** Resolution is refutation complete for normal form knowledge bases.

**Theorem:** All knowledge bases are convertible to normal form knowledge bases.