4CCS1ELA: Elementary Logic with Applications
(Formal) Syntax & Semantics of First-Order Logic

Enrico Malizia

(parts of these slides are based on material by Dr. Odinaldo Rodrigues)

Department of Informatics
King's College London, UK

Term 1, 2020/21

Different semantics for FOL

- The language of first-order logic (or predicate logic) is characterized by the terms, the functions, the predicates, the quantifiers, and so on
- However, this is just a set of symbols
- Symbols do not carry with them their own meaning. . .
  - What does the sequence of characters "c o n f e t t i" mean?

English "confetti"

Italian "confetti" (if you wonder, these are sugar-coated almonds)

Different semantics for FOL

- Different meanings can be associated with a sequence of symbols
- Hence, different semantics can be associated with a first-order language; Two of them are:
  - Classical semantics
  - Herbrand semantics
- These two semantics are different but very similar, and under some circumstances equivalent
- The classical semantics is usually adopted in mathematical logics
- The Herbrand semantics is usually adopted in relational database theory & applications
- In these slides we will give intuitions on the classical semantics; if interested, more formal (and intricate) definitions of the classical semantics can be found in additional material on KEATS

Syntax of FOL: Vocabulary

- We start by looking at the symbols of a first-order language

Definition

A *vocabulary* (or, *signature*) σ is a collection of:

- a *non-empty* set $\{c_1, \ldots, c_n\}$ of **constants** (symbols);
- a possibly empty set $\{f_1, \ldots, f_n\}$ of **functions** (symbols), each of them associated with an integer $arity(f_i) = a_i \geq 1$, i.e. the *arity* of f_i;
- a possibly empty set $\{P_1, \ldots, P_n\}$ of **predicates** (or, **relations**) (symbols), each of them associated with an integer $arity(P_i) = a_i \geq 0$, i.e. the *arity* of P_i.

Syntax of FOL: Alphabet

- Besides the vocabulary, we need logical connectives & other symbols

Definition

Given a vocabulary σ, the *alphabet* $\mathcal{A}_\sigma$ *of a first-order language over* σ is the set of symbols comprising:

- a countably infinite set of *variables* $\{x, y, z, \ldots\}$ (with subscripts and superscripts);
- '$\top$', '$\bot$' (i.e. Boolean values `true`, `false`);
- '$\neg$', '$\wedge$', '$\vee$', '$\rightarrow$', '$\leftrightarrow$' (i.e. the logical connectives);
- '$\forall$', '$\exists$' (i.e. the quantifiers' symbols);
- '$=$' (i.e. the equality symbol; '$t_1 \neq t_2$' will replace '$\neg(t_1 = t_2)$');
- ')', '(' (i.e. the parentheses);
- ',' (i.e. the comma);
- all the constant, predicate, and function, symbols (with their associated arity) from σ.

Syntax of FOL: Expressions

- An *expression* of a first-order language is any finite sequence of symbols of the alphabet of the language
- ${\mathcal{A}_\sigma}^*$ is the set of all expressions that can be build by the symbols in the alphabet $\mathcal{A}_\sigma$
- Since no constraint is imposed over how symbols can be arranged in expressions, surely many expressions are nonsensical; e.g.,

$$)\forall \leftrightarrow \exists(\wedge x((Friend(\forall, \vee)$$

- Therefore, we need to select some expressions that make sense for us

Syntax of FOL: Terms

- Defining the statements making sense requires the concept of term

Definition

The σ*-terms of a first-order language over* σ are the expressions from ${\mathcal{A}_\sigma}^*$ defined via the following rules:

- every variable is a σ-term;
- every constant in σ is a σ-term;
- if the expressions $t_1, \ldots, t_n$ are σ-terms and f is an n-ary function in σ, then the expression $f(t_1, \ldots, t_n)$ is also a σ-term.

Syntax of FOL: Formulas

- We are now ready to define first-order formulas

Definition

Given a vocabulary σ, the *σ-formulas of a first-order language over σ* are the expressions from ${\mathcal{A}_\sigma}^*$ defined via the following rules:

- $\top$ and $\bot$ are σ-formulas;
- if t_1 and t_2 are σ-terms, then $t_1 = t_2$ is also a σ-formula (an **atomic formula**);
- if $t_1, \ldots, t_n$ are σ-terms and P is an n-ary predicate in σ, then the **atom** $P(t_1, \ldots, t_n)$ is also a σ-formula (another **atomic formula**);
- if ϕ is a σ-formula, then $\neg(\phi)$ is also a σ-formula;
- if ϕ and ψ are σ-formulas, then $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, and $(\phi \leftrightarrow \psi)$ are also σ-formulas;
- if x is a variable and ϕ is a σ-formula $(\exists x)(\phi)$ and $(\forall x)(\phi)$ are also σ-formulas

Syntax of FOL: Language

- The *first-order (or predicate logic) language over the vocabulary* σ, denoted by $\mathcal{L}[\sigma]$, is the set of all expressions over ${\mathcal{A}_\sigma}^*$ that are moreover σ-formulas
- To slimline the formulas of our first order languages, parentheses can be omitted by resorting to the precedence rules of the logical connectives

Semantic of FOL

- We have seen how FOL formulas are written, i.e., the syntax of FOL. We now see how to evaluate whether a FOL formula is `true` or `false`, i.e., the semantic of FOL
- Similarly to propositional logic, the truth-value of a FOL formula depends on the truth-values assigned to its "components"
- In propositional logic, an interpretation is a truth-value assignment to the propositions
- An interpretation $\mathcal{I}$ for a first-order logic language consists of a nonempty domain D of objects, over which the variables may range, together with an assignment of a meaning to the predicate, constant and function, symbols
- Also in this case, an interpretation enables a truth-value to be assigned to a first-order formula

Semantic of FOL: Example

- Consider the following formula:

$$\exists x(x + x = 3)$$

- Consider an interpretation where the symbol "$+$" is interpreted as the usual sum symbol operation; and the symbol "3" is interpreted as the object "number 3"
- If the domain in the interpretation is the set $\mathbb{N}$ of natural numbers, then the formula is `false` in the interpretation
- On the other hand, if the domain in the interpretation is the set $\mathbb{Q}$ of rational numbers, then the formula is `true` in the interpretation
- Hence, the domain in the interpretation can make a difference in the truth value of a formula

Semantic of FOL: Example

- Consider the following formula:

$$\forall x \exists y A(x, y)$$

- Consider an interpretation with domain $\mathbb{N}$ (set of natural numbers)
- If the binary predicate A is interpreted as "*successor*", i.e. $A(x, y)$ is true whenever y is the successor of x in $\mathbb{N}$ (e.g., we have $A(2, 3)$ and $A(5, 6)$ true, whereas $A(3, 6)$ and $A(4, 1)$ are false), then the formula is `true` (because every number in $\mathbb{N}$ has a successor in $\mathbb{N}$)
- On the other hand, if the binary predicate A is interpreted as "*predecessor*", i.e. $A(x, y)$ is true whenever y is the predecessor of x in $\mathbb{N}$ (e.g., we have $A(3, 2)$ and $A(6, 5)$ true, whereas $A(3, 6)$ and $A(4, 1)$ are false), then the formula is `false` (because the number 0 does not have any predecessor in $\mathbb{N}$)
- Hence, different interpretations of the predicates can make a difference in the truth value of a formula

Semantic of FOL: Satisfaction of a formula

Definition

Let ϕ be a σ-formula of a first-order language $\mathcal{L}[\sigma]$, and let $\mathcal{I}$ be an interpretation for $\mathcal{L}[\sigma]$. The interpretation $\mathcal{I}$ **satisfies** the formula ϕ, denoted by $\mathcal{I} \models \phi$, based on the following inductive rules:

- $\mathcal{I} \models \top$ always; and $\mathcal{I} \models \bot$ never;
- for terms t_1 and t_2, $\mathcal{I} \models t_1 = t_2$ iff t_1 and t_2 are interpreted in $\mathcal{I}$ as the same object in the domain;
- for an n-ary predicate P and terms $t_1, \ldots, t_n$, $\mathcal{I} \models P(t_1, \ldots, t_n)$ iff $P(t_1, \ldots, t_n)$ is `true` in $\mathcal{I}$;
- for a formula ϕ, $\mathcal{I} \models \neg(\phi)$ iff $\mathcal{I} \models \phi$ does not hold;
- for formulas ϕ and ψ, $\mathcal{I} \models (\phi \wedge \psi)$ iff $\mathcal{I} \models \phi$ *and* $\mathcal{I} \models \psi$;
- for formulas ϕ and ψ, $\mathcal{I} \models (\phi \vee \psi)$ iff $\mathcal{I} \models \phi$ *or* $\mathcal{I} \models \psi$;
- for formulas ϕ and ψ, $\mathcal{I} \models (\phi \rightarrow \psi)$ iff $\mathcal{I} \models (\neg\phi \vee \psi)$;
- for formulas ϕ and ψ, $\mathcal{I} \models (\phi \leftrightarrow \psi)$ iff $\mathcal{I} \models ((\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi))$;

  [...]

Semantic of FOL: Satisfaction of a formula

Definition

- for a variable x and a formula ϕ, $\mathcal{I} \models (\exists x)(\phi)$ iff there exists an object $d \in D$ such that $\mathcal{I} \models \phi(x/d)$
  - For a D of finite size, $\mathcal{I} \models (\exists x)(\phi)$ equals $\mathcal{I} \models \bigvee_{d \in D} \phi(x/d)$
- for a variable x and a formula ϕ, $\mathcal{I} \models (\forall x)(\phi)$ iff, for all objects $d \in D$, $\mathcal{I} \models \phi(x/d)$
  - For a D of finite size, $\mathcal{I} \models (\forall x)(\phi)$ equals $\mathcal{I} \models \bigwedge_{d \in D} \phi(x/d)$

Notation: in the formulas above, '$\phi(x/d)$' denotes the formula obtained from ϕ in which all the *free* occurrences of x are substituted for the object d
Example: let ϕ be $P(x) \wedge \exists x \exists y P(x, y)$ and $D = \mathbb{N}$; then $\phi(x/0)$ is $P(0) \wedge \exists x \exists y P(x, y)$

Semantic of FOL: Satisfaction of a formula

- If $\mathcal{I}$ satisfies ϕ, $\mathcal{I}$ is said to be a **model** of ϕ
- If $\mathcal{I}$ *does not* satisfy ϕ, it can be denoted by $\mathcal{I} \not\models \phi$
- For an interpretation $\mathcal{I}$ and a formula ϕ, we can also write $\mathcal{I}(\phi) = 1$ and $\mathcal{I}(\phi) = 0$ to say that $\mathcal{I}$ satisfies and do not satisfy ϕ, respectively
- A formula is **satisfiable** if it admits a model; e.g., this formula is satisfiable: $(\forall x)(P(x) \to Q(x))$
- A formula is **unsatisfiable** (or, a **contradiction**) if it does not admit any model; e.g., this formula is a contradiction $(\exists x)(P(x) \wedge \neg P(x))$
- A formula is **valid** (or, a **tautology**) if it is satisfied by all interpretations; e.g., this formula is a tautology $(\forall x)(P(x)) \to (\exists x)(P(x))$

Semantic of FOL: truth-value of a formula — Example

- Consider the following formula:

$$\exists x \forall y (P(y) \to x = y)$$

- Does this formula evaluate to `true` or `false` in the following interpretations? (There are no function symbols)
- $D = \{a\}$ and $P(a)$ is true ✔
- $D = \{a\}$ and $P(a)$ is false ✔
- $D = \{a, b\}$ and both $P(a)$ and $P(b)$ are true ✘
  There is no witness, as shown by the following two counter-examples for each of the possible choices of x: $(x = a; y = b)$; $(x = b; y = a)$
- $D = \{a, b\}$ and both $P(a)$ and $P(b)$ are false ✔
- $D = \{a, b\}$, and $P(a)$ is true and $P(b)$ is false ✔
  The witness is $x = a$, as for this specific choice of x there is no counter-example on y

Semantic of FOL: truth-value of a formula — Example

- Consider the interpretations where the symbols "$<$", "$+$", and "$\times$", are interpreted in the usual way as the symbols for "less than", the "sum operation", and the "multiplication operation", respectively; and "0" is the constant symbol interpreted as the "number zero"
- Tell if the following formulas are true or not depending on the domain of interpretation: $\mathbb{N}$, for the natural numbers; $\mathbb{Z}$, for the integer numbers; $\mathbb{Q}$, for the rational numbers; and $\mathbb{R}$, for the real numbers

- $(\exists z)(\forall n)(n \neq z \to z < n)$ $\mathbb{N}$: ✔, $\mathbb{Z}$: ✘, $\mathbb{Q}$: ✘, $\mathbb{R}$: ✘
- $(\forall x)(\forall y)(\exists z)(x + y = z)$ $\mathbb{N}$: ✔, $\mathbb{Z}$: ✔, $\mathbb{Q}$: ✔, $\mathbb{R}$: ✔
- $(\forall x)(\forall z)(\exists y)(x + y = z)$ $\mathbb{N}$: ✘, $\mathbb{Z}$: ✔, $\mathbb{Q}$: ✔, $\mathbb{R}$: ✔
- $(\forall x)(\forall y)(x < y \to (\exists z)(x < z \wedge z < y))$ $\mathbb{N}$: ✘, $\mathbb{Z}$: ✘, $\mathbb{Q}$: ✔, $\mathbb{R}$: ✔
- $(\forall x)(\forall y)(\exists z)(x \times y = z)$ $\mathbb{N}$: ✔, $\mathbb{Z}$: ✔, $\mathbb{Q}$: ✔, $\mathbb{R}$: ✔
- $(\forall x)(\forall z)(x \neq 0 \to (\exists y)(x \times y = z))$ $\mathbb{N}$: ✘, $\mathbb{Z}$: ✘, $\mathbb{Q}$: ✔, $\mathbb{R}$: ✔
- $(\forall y)(\neg(y < 0) \to (\exists x)(x \times x = y))$ $\mathbb{N}$: ✘, $\mathbb{Z}$: ✘, $\mathbb{Q}$: ✘, $\mathbb{R}$: ✔

Semantic of FOL: Logical consequence

- An interpretation $\mathcal{I}$ is a **model** of a set Φ of σ-formulas, denoted by $\mathcal{I} \models \Phi$, iff $\mathcal{I} \models \phi$ for all $\phi \in \Phi$
- A set Φ of σ-formulas is **satisfiable** iff Φ admits a model
- The σ-formula γ is a *(logical) consequence* of a set Φ of σ-formulas, denoted by $\Phi \models \gamma$, iff every model of Φ is also a model of γ
- $\emptyset \models \phi$, also denoted by $\models \phi$, means that ϕ is valid, because all interpretations are models of the empty set $\emptyset$
- Two σ-formulas ϕ and ψ are *logically equivalent*, denoted by $\phi \equiv \psi$, iff $\{\psi\} \models \phi$ and $\{\phi\} \models \psi$