4. First Order Logic

David Pereira José Proença

RAMDE 2021/2022

Requirements and Model-driven Engineering

CISTER – ISEP
Porto, Portugal

https://cister-labs.github.io/ramde2122

First Order Logic and its Syntax

The need for a richer kind of formal logic...

The limitations of Propositional Logic

So far, we have been looking into Propositional Logic for reasoning about statements, in a way that can be valuable for the process of Requirement's Engineering. Although usefull, in most cases we need a richer language (and underlying formal system) that allows us to be more precise about the concepts we need to express.

During this and next two classes...

You will be presented with the concept of First Order Logic, learn about how can we express things using its language, learn how formulas can be evaluated with respect to models (yes, we are going to talk about models), and of course we will dive into performing Natural Deduction using First Order Logic constructions.

Warning: Things are going to get a little bit more complicated, considering what has been introduced in terms of Propositional Logic. Once again, bare with me and you will get comfortable with First Order Logic in a glimpse ;-)

First Order Logic - Syntax

Lets look into this simple example

Hypothesis 1: All dogs like running.

Hypothesis 2: Zen is a dog.

Conclusion: Zen likes to run.

What can we say about the above reasoning?

Well, the argument is clearly valid! However, translating it into propositional logic would result in a unique sentence $\varphi \wedge \psi \rightarrow \theta$ which is definitely not a valid formula!

Using truth tables and considering $\varphi =$ "All dogs like running", $\psi =$ "Zen is a dog", and $\theta =$ "Zen likes to run", if $f(\varphi) =$ **true**, $f(\psi) =$ **true**, and $f(\theta) =$ **false** we would get that $f(\varphi \wedge \psi \rightarrow \theta) =$ **false**.

First Order Logic - Syntax

Representability of concepts in First Order Logic

In First Order Logic (FOL) we will be able to represent/reason about

- Objects
- Properties and relations about objects
- Properties and relations about sets of objects

Getting back to Zen's example

- $\forall x, \textbf{Dog}(x) \rightarrow \textbf{LikesToRun}(x)$
- $\textbf{Dog}(\text{zen})$
- $\textbf{LikesToRun}(\text{zen})$

We will see further ahead in this and the following classes that this kind of reasoning is valid in FOL.

First Order Logic - Syntax

FOL language

A language of FOL considers the following sets of symbols:

logical symbols of one of the following forms:

- a set of variables $S = \{x, y, \ldots, x_0, y_0, \ldots\}$
- logical connectives $\wedge, \vee, \neg$, and $\rightarrow$
- quantifiers $\forall$ (for all) and $\exists$ (exists)
- parenthesis (and)
- possibly, the equality symbol $=$

FOL language

A language of FOL considers the following sets of symbols:

Non-logical symbols of one of the following forms:

- a (possibly empty) set of functional symbols for each n-arity, represented as $\mathcal{F}_n$ (when referring to constants, we are actually talking about functional symbols with arity 0). Typically, f, g, h, ...
- a (possibly empty) set of relation symbols for each n-arity, represented as $\mathcal{R}_n$. Typically, P, Q, R, ...

First Order Logic - Syntax

FOL Terms

Let $\mathcal{L}$ be a FOL language. A term is inductively/recursively defined as follows:

- a variable $x \in \mathcal{V}$ is a term;
- a constant i.e., a symbol $c \in \mathcal{F}_0$ is also a term;
- if $t_0, \dots, t_n$ are terms and $f \in \mathcal{F}_n$ is a functional symbol, then $f(t_0, \dots, t_n)$ is a term.

Closed terms

A FOL term is said to be **closed** if no variables occur in such term.

Some quick examples

Assuming that $\mathcal{F}_0 = \{a, d\}$, that $\mathcal{F}_1 = \{f\}$, that $\mathcal{F}_2 = \{h\}$, and that $\mathcal{F}_3 = \{g\}$. Which of the following are terms and which are not?

- $f(a, g(x, g(a), a))$
- $h(d, h(f(a), x))$
- $x(d, g(y))$
- $h(h(x, x), h(y, y))$
- $f(a(x))$

Some quick examples

Assuming that $\mathcal{F}_0 = \{a, d\}$, that $\mathcal{F}_1 = \{f\}$, that $\mathcal{F}_2 = \{h\}$, and that $\mathcal{F}_3 = \{g\}$. Which of the following are terms and which are not?

- $f(a, g(x, g(a), a))$
- $h(d, h(f(a), x))$
- $x(d, g(y))$
- $h(h(x, x), h(y, y))$
- $f(a(x))$

Some quick examples

Assuming that $\mathcal{F}_0 = \{a, b\}$, that $\mathcal{F}_1 = \{g\}$, that $\mathcal{F}_2 = \{f, h\}$, $\mathcal{R}_1 = \{R, S\}$, and $\mathcal{R}_2 = \{P, Q\}$. Which of the following are closed terms?

- $h(a, f(a, g(a), g(a)))$
- $f(h(x, g(g(a))), x)$
- $f(a, P(a, g(x)))$
- $h(g(f(a, a)), f(b, a))$
- $f(h(x, h(y, y)), g(g(b)))$
- $f(a, g(h(g(x), x(a))))$

Some quick examples

Assuming that $\mathcal{F}_0 = \{a, b\}$, that $\mathcal{F}_1 = \{g\}$, that $\mathcal{F}_2 = \{f, h\}$, $\mathcal{R}_1 = \{R, S\}$, and $\mathcal{R}_2 = \{P, Q\}$. Which of the following are closed terms?

- $h(a, f(a, g(a), g(a)))$
- $f(h(x, g(g(a))), x)$
- $f(a, P(a, g(x)))$
- $h(g(f(a, a)), f(b, a))$
- $f(h(x, h(y, y)), g(g(b)))$
- $f(a, g(h(g(x), x(a))))$

First Order Logic - Syntax

FOL Atoms

Let $\mathcal{L}$ be a FOL language. An **atom** (from the term atomic formula) is inductively/recursively defined as follows:

- if $t_0, \ldots, t_n$ are terms and $R \in \mathcal{R}_n$ is a relational symbol, then $R(t_0, \ldots, t_n)$ is an atom;
- if $\mathcal{L}$ include the equality symbol $=$ and if t_1 and t_2 are terms, then $t_1 = t_2$ is an atom.

Some examples...

- $R(b)$
- $R(x, y, z)$
- $G(f(a, b), x)$
- $R(f(a, x), g(y, b), h(c))$

First Order Logic - Syntax

FOL Formulae

Let $\mathcal{L}$ be a FOL language. The set of **formulae** is inductively/recursively defined as follows:

- an atom is a formula;
- if φ is a formula, then so is $\neg\varphi$;
- if φ and ψ are formulas, then so are $\varphi \wedge \psi$, $\varphi \vee \psi$, and $\varphi \rightarrow \psi$;
- if φ is a formula and x is a variable, then $\forall x, \varphi$ and $\exists x, \varphi$ are also formulas.

The role of parenthesis

Parenthesis can be disregarded if we assume priority conventions to the remaining constructs, that is, that quantifiers have higher precedence than other logical operator. For instance, the formula $\forall x, (P(x) \vee \neg P(x))$ does not have the same meaning that $\forall x, P(x) \vee \neg P(x)$.

Some quick examples

Assuming that $\mathcal{F}_0 = \{a, d\}$, that $\mathcal{F}_1 = \{f\}$, that $\mathcal{F}_2 = \{h\}$, and that $\mathcal{R}_2 = \{R, S\}$, which of the following expressions are formulas?

- $R(a, d)$
- $h(x, y)$
- $S(R(f(x), y), z)$
- $R(d, a) \to \exists y, S(d, y)$
- $S(\forall x, R(f(a), d), x)$
- $\forall x, R(f(a), h(a, x))$

Some quick examples

Assuming that $\mathcal{F}_0 = \{a, d\}$, that $\mathcal{F}_1 = \{f\}$, that $\mathcal{F}_2 = \{h\}$, and that $\mathcal{R}_2 = \{R, S\}$, which of the following expressions are formulas?

- $R(a, d)$
- $h(x, y)$
- $S(R(f(x), y), z)$
- $R(d, a) \rightarrow \exists y, S(d, y)$
- $S(\forall x, R(f(a), d), x)$
- $\forall x, R(f(a), h(a, x))$

A more serious example

Lets look at a FOL that focus on being able to express formulae about natural numbers, including addition and multiplication. Let $\mathcal{A}$ be a language that includes equality and such that $\mathcal{F}_0 = \{0, 1\}$, $\mathcal{F}_2 = \{+, \times\}$, and $\mathcal{R}_2 = \{<\}$.

Terms of this language include:

- 0, 1, $+(1,1)$, $+(1,+(1,1))$, $+(1,+(+(1,1),1))$
- $\times(+(1,0),1)$, $\times(\times(1,1),+(0,1))$

Formulas of this language include:

- $<(\times(1,1),+(1,1))$
- $\forall x, (+(0,x) = x)$
- $\forall x, \exists y, +(x,1) = +(y,1) \to x = y$

Translating text into FOL formulae

In general, we have the following translation

Type of text	FOL formula
All P are also Q	$\forall x, (P(x) \rightarrow Q(x))$
Some P are also Q	$\exists x, (P(x) \wedge Q(x))$
No P is Q	$\forall x, (P(x) \rightarrow \neg Q(x))$
Not all P are Q	$\exists x, (P(x) \wedge \neg Q(x))$

Small exercise

Find some adequate predicates to express the following statements:

- All even numbers are prime numbers
- Not all prime numbers are even numbers
- Some prime numbers are not even
- All prime number is not even or equal to 2

Small exercise

Find some adequate predicates to express the following statements:

- All even numbers are prime numbers
  $\forall x, (\textbf{Even}(x) \rightarrow \textbf{Prime}(x))$
- Not all prime numbers are even numbers
- Some prime numbers are not even
- All prime number is not even or equal to 2

Small exercise

Find some adequate predicates to express the following statements:

- All even numbers are prime numbers
  $\forall x, (\mathbf{Even}(x) \rightarrow \mathbf{Prime}(x))$
- Not all prime numbers are even numbers
  $\neg(\forall x, (\mathbf{Prime}(x) \rightarrow \mathbf{Even}(x)))$
- Some prime numbers are not even
- All prime number is not even or equal to 2

Small exercise

Find some adequate predicates to express the following statements:

- All even numbers are prime numbers
  $\forall x, (\textbf{Even}(x) \rightarrow \textbf{Prime}(x))$
- Not all prime numbers are even numbers
  $\neg(\forall x, (\textbf{Prime}(x) \rightarrow \textbf{Even}(x)))$
- Some prime numbers are not even
  $\exists x, (\textbf{Prime}(x) \wedge \neg\textbf{Even}(x))$
- All prime number is not even or equal to 2

Small exercise

Find some adequate predicates to express the following statements:

- All even numbers are prime numbers
  $\forall x, (\textbf{Even}(x) \rightarrow \textbf{Prime}(x))$
- Not all prime numbers are even numbers
  $\neg(\forall x, (\textbf{Prime}(x) \rightarrow \textbf{Even}(x)))$
- Some prime numbers are not even
  $\exists x, (\textbf{Prime}(x) \wedge \neg\textbf{Even}(x))$
- All prime number is not even or equal to 2
  $\forall x, (\textbf{Prime}(x) \rightarrow (\neg\textbf{Even}(x) \vee x = 2))$

Bound and free variables

Bound Variable

A variable x is said to be **bound** to a formula φ if φ has a subformula ψ whose schema is $\forall x, \theta$ or $\exists x, \theta$ and x occurs in θ.

Free Variable

A variable x is said to be **free** if it is not bound.

Proposition

A formula is said to be a proposition if it does not contain free variables.

Small exercises

For each of the formulae presented bellow, identify the bound and free variables:

- $\exists x, (P(y, z) \wedge \forall y, (\neg Q(x, y) \vee P(y, z))$
- $\neg(\forall x, \exists y, P(x, y, z) \rightarrow \forall z, P(x, y, z))$
- $P(a, g(c, d))$
- $\exists x, (P(x) \rightarrow \neg Q(x))$

Small exercises

Assuming that $\mathcal{F}_0 = \{a, d\}$, that $\mathcal{F}_1 = \{f\}$, that $\mathcal{F}_2 = \{h\}$, and that $\mathcal{R}_2 = \{R, S\}$, which of the following expressions have free variables, which are propositions, and which are the atomic sub-formulas?

- $\forall x, Q(x, x) \wedge P(x, x)$
- $R(a) \wedge \exists y, (R(f(y, y)) \rightarrow P(a, y))$
- $\forall x, \forall y, x = y \rightarrow \forall x, Q(y, x)$

Substitution

Let $\mathcal{L}$ be a FOL language, φ a formula, t a term, and $x \in \mathcal{L}$ a variable. The substitution of the variable x by the term t in φ is denoted by $\varphi[t/x]$ and corresponds to replacing all the free occurrences of x in φ by the term t.

Semantics

How to Evaluate a FOL Formula

Recalling evaluation of formulae in Propositional Logic

In PL, the evaluation of a formula φ is fully determined by the valuation given to each of its propositional variables, and the connectives involved in the formula.

But how is it done in FOL?

In FOL, to evaluate a formula we need to determine the meaning of:

- bound and free variables
- quantifiers
- functional symbols
- relational/predicate symbols

This must be done in a concrete **universe**.

Structure

The **strucuture** of an LPO language is a pair $\mathcal{S} = \langle S, \cdot^{\mathcal{S}} \rangle$ where S is a non-empty set know as the ***domain*** or ***universe*** and $\cdot^{\mathcal{S}}$ is a function such that:

- associates each constant c with a value $c^{\mathcal{S}} \in S$
- associates each n-ary symbol $f \in \mathcal{F}_n$ with a n-ary function $f^{\mathcal{S}}$ from S^n to A
- associates each n-ary predicate symbol $R \in \mathcal{R}_n$ with a relation $R^{\mathcal{S}} \subseteq S^n$

Example Structure

Let $\mathcal{L}$ be a FOL language and $\mathcal{S} = \langle \mathbb{N}, \cdot^{\mathcal{S}} \rangle$ a structure such that $\cdot^{\mathcal{S}}$ is defined as follows:

- $0^{\mathcal{S}} = 0$ and $1^{\mathcal{S}} = 1$
- $+^{\mathcal{S}}(n, m) = n + m$
- $\leq^{\mathcal{S}} (n, m) = \{(n, m) \mid n \leq m\}$

With the above structure, we can use this language to reason about partial ordering of natural numbers and sums of natural numbers. **Note the importance of the interpretation. If a different interpretation of constants, functional symbols and relations/predicates (possibly under different universe) was given, then we could be expressing completely different concepts using the same language!**

Structures and Variable Interpretations

Variable Interpretation

We associate with each language $\mathcal{L}$ and structure $\mathcal{S} = \langle A, \cdot^{\mathcal{A}} \rangle$ a variable interpretation function $s : \mathcal{V} \to A$ such that

- for each variable $x \in \mathcal{V}$, the value of $s(x)$ is defined
- if c is a constant, then $s(c) = c^{\mathcal{A}}$
- if $t_0, \ldots, t_n$ are terms and f is an n-ary function symbol, then

$$s(f(t_0, \ldots, t_n)) = f^{\mathcal{A}}(s(t_0), \ldots, s(t_n))$$

Substitution

Let $\mathcal{S} = \langle S, \cdot^{\mathcal{S}} \rangle$ and let s be a variable interpretation. We define the substitution of a variable x by a value a in s (denoted by $s[a/x]$) as $s[a/x](y) = a$ if $x = y$ and $s[a/x](y) = s(y)$ otherwise.

Satisfiability

Let $\mathcal{L}$ be a language of FOL, let $\mathcal{A} = \langle A, \cdot^{\mathcal{A}} \rangle$, and let $s : \mathcal{V} \to A$. We write $\mathcal{A} \models_s \varphi$ and say that φ is **satisfiable** under interpretation s and structure $\mathcal{A}$, and we define the relation $\models$ inductively/recursively as follows:

- $\mathcal{A} \models_s t_1 = t_2$ iff $s(t_1) = s(t_2)$
- $\mathcal{A} \models_s R(t_0, \ldots, t_n)$ iff $(s(t_0), \ldots, s(t_n)) \in R^{\mathcal{A}}$
- $\mathcal{A} \models_s \varphi \wedge \psi$ iff $\mathcal{A} \models_s \varphi$ and also $\mathcal{A} \models_s \psi$
- $\mathcal{A} \models_s \varphi \vee \psi$ iff $\mathcal{A} \models_s \varphi$ or $\mathcal{A} \models_s \psi$
- $\mathcal{A} \models_s \varphi \to \psi$ iff $\mathcal{A} \not\models_s \varphi$ or $\mathcal{A} \models_s \psi$
- $\mathcal{A} \models_s \forall x, \varphi$ iff for all $a \in A$ it is true that $\mathcal{A} \models_{s[a/x]} \varphi$
- $\mathcal{A} \models_s \exists x, \varphi$ iff exists $a \in A$ such that $\mathcal{A} \models_{s[a/x]} \varphi$

Example of Satisfiability

Let $\mathcal{L}$ be a FOL language and $\mathcal{S} = \langle \mathbb{N}, \cdot^{\mathcal{S}} \rangle$ a structure such that $\cdot^{\mathcal{S}}$ is defined as follows:

- $0^{\mathcal{S}} = 0$ and $1^{\mathcal{S}} = 1$
- $+^{\mathcal{S}}(n, m) = n + m$
- $\leq^{\mathcal{S}} (n, m) = \{(n, m) \mid n \leq m\}$

Lets try to prove that $\mathcal{S} \models_s \forall x, \leq (x, +(x, 1))$.

Proof.

By the definition of satisfiability, we know that $\mathcal{S} \models_{s[n/x]} \leq (x, +(x, 1))$. Again, by definition of satisfiability, we must show that $(s[n/x](x), s[n/x](x + 1)) \in \leq^{\mathcal{S}}$ which, by the interpretation of $\leq$ in structure $\mathcal{S}$ is the same as stating that $(s[n/x](x), s[n/x](x + 1)) \in \{(n, m) \mid n \leq m\}$. Computing the substitution, we get $(n, n + 1) \in \{(n, m) \mid n \leq m\}$, which is true! $\square$

Satisfiability, Validity, and Models

Satisfiability

Let $\mathcal{L}$ be a FLO language and φ a formula. We say that φ is satisfiable if exists a structure $\mathcal{S}$ and interpretation s such that $\mathcal{S} \models_s \varphi$.

Validity

Let $\mathcal{L}$ be a FLO language and φ a formula. We say that φ is **valid** if for all structure $\mathcal{S}$ and interpretation s, we have $\mathcal{S} \models_s \varphi$.

Propositions and Models

Let φ be a proposition (i.e., formula without free variables), and let $\mathcal{S}$ be a structure. Then, either $\mathcal{S} \models_s \varphi$ holds for all interpretations s, or $\mathcal{S} \models_s \varphi$ holds for all interpretations s. If $\mathcal{S} \models \varphi$, we say that $\mathcal{S}$ **is a model** of φ.

Lets build some models

Exercise

For each formula below, find a structure that is a model of the formula, and a structure which is not.

1. $\forall x, \forall y, x = y$
2. $\forall x, x = a$
3. $\forall x, \forall y, (P(x) \rightarrow P(y))$
4. $\exists x, f(x) = c \rightarrow \exists y, f(y) \neq c$

Lets build some models

For the formula

$$\forall x, \forall y, x = y$$

we can consider a structure

$$\mathcal{S} = \langle \{0\}, \cdot^{\mathcal{S}} \rangle$$

Note that, in this case, the only element that can substitute the variables is the only element of the domain, thus $x = y$ always holds.

To find a structure that is not a model, it is enough to consider the domain of the structure to have more than a value...

Lets build some models

For the formula

$$\forall x, x = a$$

we can consider a structure

$$\mathcal{S} = \langle \{a\}, \cdot^{\mathcal{S}} \rangle$$

Note that, in this case, the only element that can substitute the variables is the only element of the domain, thus $x = a$ always holds.

To find a structure that is not a model, it is enough to consider the domain of the structure to have more than a value...

Lets build some models

For the formula

$$\forall x, \forall y, (P(x) \rightarrow P(y))$$

we can consider a structure

$$\mathcal{S} = \langle \{0, 2, 4, 6, \ldots\}, \cdot^{\mathcal{S}} \rangle$$

where we give the following interpretation to the predicate P:

$$P^{\mathcal{S}} = \{x \mid x \text{ is even}\}$$

With this interpretation, whatever x and y we choose, they are even and so the formula holds.

To find a structure that is not a model, it is enough to consider as domain the complete set of natural numbers (it includes both even and odd numbers), while retaining the same interpretation of $P^{\mathcal{S}}$.

Lets build some models

For the formula

$$\exists x, f(x) = c \rightarrow \exists y, f(y) \neq c$$

we can consider a structure

$$\mathcal{S} = \langle \mathbb{N}, \cdot^{\mathcal{S}} \rangle$$

where we give the following interpretation:

- $c^{\mathcal{S}} = 1$
- $f^{\mathcal{S}}(x) = 1$ if $x = 1$, otherwise $f^{\mathcal{S}}(x) = 2$

To find a structure that is not a model, it is enough to consider the interpretation of f such that $f^{\mathcal{S}}(x) = 1$ for all variable x.