Lecture 10: First Order Predicate Logic

MATH230

Te Kura Pāngarau | School of Mathematics and Statistics
Te Whare Wānanga o Waitaha | University of Canterbury

Outline

1. Motivation
2. First Order Logic
3. Examples

Example

Propositional logic, alone, is not enough to express the logical structure of propositions and arguments that mathematicians want to make.

Example: For all natural numbers x,y,z it is the case that if x divides the product yz, then either x divides y or x divides z.

This statement does have propositional structure, in particular the "if..., then ..." structure. However it also has structure that can't be captured with propositional logic.

Alphabet of FoL

We use the following standard language of *first-order predicate logic* to build terms and well-formed formula.

- Variables $x, y, z, \ldots$ or $x_1, x_2, x_3, \ldots$
- Constants $a, b, c, \ldots$ or $c_1, c_2, c_3, \ldots$
- Predicate (relation) symbols $R_0, R_1, R_2, \ldots$
- Function symbols $f_0, f_1, f_2, \ldots$
- Logical connectives from propostional logic $\neg, \wedge, \vee, \rightarrow, \perp$
- Quantifiers $\forall$ and $\exists$

Note: For clarity we also use parentheses and commas when writing expressions. These can be included in the symbols of our language as well.

Universal and Existential Quantifiers

The symbol $\forall$ will be interpreted as "for all..."
The symbol $\exists$ will be interpreted as "there exists..."

Example: For all real numbers x there exists another real number between x and zero.

To help with intuition we can read them as such. However, strictly speaking, they're just symbols in the alphabet of first order logic, for now.

Grammar of FoL

Terms are defined inductively as follows:

- Variables and constants are terms
- If f is an n-ary function and $t_1, ..., t_n$ are terms, then $f(t_1, ..., t_n)$ is a term.

Well-formed formulae are defined inductively as follows:

- $\perp$ is a wff
- An n-ary predicate applied to n-terms $R(t_1, ..., t_n)$ is a wff.
- If α, β are wff, then $\neg\alpha$, $(\alpha \vee \beta)$, $(\alpha \wedge \beta)$, and $(\alpha \rightarrow \beta)$ are wff.
- If α is a wff and x is a variable, then $\forall x\alpha$ and $\exists x\alpha$ are wff.

Any string of symbols that is neither a term nor a wff by virtue of the above definitions is not a part of first order logic e.g. $\perp \vee \wedge fR$

Comments

The last few slides have said how we can build strings that are part of the language of first order logic. So far, these are just strings of symbols that fit together into terms and wff with the grammar defined.

They have no meaning, *yet*. Just as in propositional logic, we have to introduce some notion of semantics on top of the language to grant these symbols any meaning.

The words *function* and *relation* maybe be familiar to you. However, in so far as the language of first order logic is concerned they are just devices for building terms and wff. The way we usually think of functions and relations is an *interpretation* of these symbols in some *model* - We will talk more about this soon.

Examples

Let R be a two-place (binary) predicate, P a one-place (unary) predicates, f a binary function, and a, b be constants.

Determine which of the following are wff and which are not.

1. Pa
2. a
3. Pab
4. $\exists x Rxb \to Px$
5. $\exists x(Px \vee Rbx)$
6. $\forall x(\exists y(R(x, y)))$
7. $\exists x(\forall y(R(x, y)P(y)P(x)))$
8. $f(y, b)$

Binding Conventions

If you want to drop parentheses, then your expressions will be interpreted using the following binding conventions:

- $\forall, \exists, \neg$ bind most tightly;
- $\wedge$ and $\vee$ bind more tightly than $\rightarrow$;
- $\rightarrow$ binds more tightly than $\leftrightarrow$.

With this convention one can unambiguously interpret the string:

$$\neg \exists x Rx \rightarrow \neg Q \wedge \forall x \neg Px$$

as the following wff:

First Order Languages

In practice we decide on a few constants, predicates, and functions depending on the part of mathematics we want to study. Thus specifying a *first order language*.

Example

Let us say we have the constant 0, unary function s, binary functions $+$ and $\times$, and $=$ as the only relation.

We collect these together into a first order language, which we can denote $\mathcal{L}$, with signature $\{0, S, +, \times, =\}$

WFF in a First Order Language

Given a first order language $\mathcal{L}$ with signature $\{s, +, \times, =, 0, 1\}$ we may write the following wff:

- $\forall x \forall y (= (x, y) \vee \neg(= (x, y)))$
- $\exists x (\forall y \neg(= (x, s(y))))$
- $= (+(0, 1), 0)$

L∃∀N Example

In the textbook L∃∀N they pick the following first order language for the entire chapter on first order logic:

$$\mathcal{L} : \{1, 2, 3, add, mul, square, even, odd, prime, lt, =\}$$

- Every natural number is even or odd, but not both.
- $Even(n) \leftrightarrow \exists x(n = mul(2, x))$.
- If x is even, then x^2 is even.
- There exists a prime number that is even.

L∃∀N Example

$\mathcal{L} : \{1, 2, 3, \mathit{add}, \mathit{mul}, \mathit{square}, \mathit{even}, \mathit{odd}, \mathit{prime}, \mathit{lt}, =\}$

Further Reading

Below is a selection of resources that I used to prepare the lecture. You might like to read over them yourself to help get a more complete picture of the topics discussed.

- L∃∀N, *Logic and Proof*. Section: 7.
- Logic Matters, *An Introduction to Formal Logic*: page 228 - 235.