Standard First-Order Languages

Péter Mekis
Department of Logic, ELTE Budapest

May 17, 2016

Contents

1 Basic symbols

Logical constants $\wedge, \vee, \rightarrow, \leftrightarrow, \neg, \forall\ , \exists\ , =, (,)$

Variables x, y, z, x', x'' etc. (infinitely many)

n**-ary predicate symbols** (or *relation symbols*) P^1, Q^1, R^2, S^3, P'^4 etc. (of any finite number or infinitely many; indices can be omitted when the context makes the arity of a predicate obvious)

Individual constants a, b, c, a' etc. (of any finite number or infinitely many)

The names of the logical constants are *conjunction*, *disjunction*, *implication*, *biconditional*, *negation*, *universal quantifier*, *existential quantifier*, *identity* and brackets, respectively.

Predicate symbols and individual constants are also called the *signature* of a first-order language, or its *extralogical constants*, or *extralogical vocabulary*, or *similarity type*. A particular first-order language is specified by its signature. There are two extreme cases: The *minimal language* (or *the language of pure logic*) has no extralogical constants; the *maximal language* has an infinite number of both individual constants and predicates of each arity. (By infinite we mean countably infinite.)

Variables and individual constants are also called *individual terms* or shortly *terms*. If we don't want to specify whether a term is variable or constant, we use the parameters t, t', t'' etc for it.

In the present note we don't use function symbols (operation symbols). They will be introduced later on.

2 Syntax

We define formulas recursively. We use atomic formulas as a basis.

A1 If P is an n-ary predicate and $t_1, \ldots, t_n$ are terms, then $Pt_1 \ldots t_n$ is an atomic formula.

A2 If t and t' are terms, then $t = t'$ is an atomic formula.

F0 Atomic formulas are formulas.

F1 If φ and ψ are formulas, then

- (a) $(\varphi \wedge \psi)$ is a formula;
- (b) $(\varphi \vee \psi)$ is a formula;
- (c) $(\varphi \rightarrow \psi)$ is a formula;
- (d) $(\varphi \leftrightarrow \psi)$ is a formula;
- (e) $\neg\, \varphi$ is a formula.

F2 If x is a variable and φ is a formula, then

- (a) $\forall x\ \varphi$ is a formula;
- (b) $\exists x\ \varphi$ is a formula.

Some formulas are enclosed between brackets; these outer brackets can be omitted. There are several other conventions for omitting brackets; we don't go into the details of them. On the other hand, if the predicate-argument structure is not clear, we may emphasize it by writing $P(t_1, \ldots, t_n)$ instead of $Pt_1 \ldots t_n$.

It is handy to use infi notation for binary relations; thus, for example, we can write aRx instead of Rxa. $x \neq y$ is shorthand for $\neg\, x = y$.

3 Some derived syntactic concepts

A variable occurrence x is bound if it is within a subformula of the form $\forall x\ \varphi$ or $\exists x\ \varphi$; it is free if it is not bound. For example, in the formula

$$\exists x\ (Px \wedge xRy) \rightarrow \exists y\ xRy$$

the first three occurrences of x are bound, while the fourth one is free; and the first occurrence of y is free, while the second and the third ones are bound. A variable is free in a formula if it has at least one free occurrence; and it is bound if it has only bound occurrences. A formula is open if it has at least one free variable occurrence; it is closed if it has no free variable occurrences. For example, the formula

$$xRy \rightarrow yRx$$

is open, while the formula

$$\forall x\ \forall y\ (xRy \rightarrow yRx)$$

is closed.

4 An example: the language of arithmetic

We have four extralogical constants:

- 0 is an individual constant with the intended meaning “zero”.
- S is a binary predicate. The intended meaning of xSy is *y is the successor of x.*
- $+$ is a ternary predicate. The intended meaning of $+xyz$ is *z is the sum of x and y.*
- $\cdot$ is a ternary predicate. The intended meaning of $\cdot xyz$ is *z is the product of x and y.*

We must introduce S, $+$ and $\cdot$ as relations, because we don't have function symbols yet.

Some examples of arithmetic formulas:

1. $\neg\, \exists x\ xS0$
   Intended meaning: *Zero is not a successor of any number.*
2. $\forall x\ \exists y\ (xSy \wedge \neg\, \exists y'\ (xSy' \wedge y \neq y'))$
   Intended meaning: *Every number has exactly one successor.*
3. $\forall x\ \forall y\ \forall z\ \big(+xyz \rightarrow \big((x = z \rightarrow y = 0) \wedge (y = z \rightarrow x = 0)\big)\big)$
   Intended meaning: *If the sum of two numbers is equal to one of the numbers, then the other one is zero.*

Finally, here are the definitions of some basic arithmetic concepts:

1. $x < y \quad \Longleftrightarrow_{\text{def}} \quad \exists z\ (z \neq 0 \wedge +xzy)$
   Intended meaning: *x is less than y.*
2. $x|y \quad \Longleftrightarrow_{\text{def}} \quad \exists z\ (z \neq 0 \wedge \cdot xzy)$
   Intended meaning: *x is a divisor of y.*
3. $Ev(x) \quad \Longleftrightarrow_{\text{def}} \quad \exists y\ \exists y'\ (0Sy \wedge ySy' \wedge y'|x)$
   Intended meaning: *x is an even number.*
4. $Pr(x) \quad \Longleftrightarrow_{\text{def}} \quad \neg\, 0Sx \wedge \forall y\ (y|x \rightarrow (0Sy \vee y = x))$
   Intended meaning: *x is a prime number.*

5 Basic semantic concepts

Domain (or *universe of discourse*) A is a nonempty set; it is the set of individuals that can be referred to by the variables of the language.

Interpretation $\ldots^{\mathcal{A}}$ is a function that assigns an extension to each extralogical constant. In particular,

i the extension of an individual constant c is an individual: $c^{\mathcal{A}} \in A$;

ii the extension of an n-ary predicate symbol P is a set of n-tuples of individuals: $P^{\mathcal{A}} \subseteq A^n$.

Structure (or *first-order structure*) is an ordered triple $\mathcal{A} = \langle A, Con^{\mathcal{A}}, Pred^{\mathcal{A}} \rangle$.

Assignment (or *evaluation*) g is a function from variables to individuals. It assigns a value to each variable.

Linguists often call a first-order structure a *model for* a language. This is not to be mistaken with a *model of* a set of formulas, discussed later.

We always use the same uppercase letters for the structure and its constituents: $\mathcal{A} = \langle A, Con^{\mathcal{A}}, Pred^{\mathcal{A}} \rangle$, $\mathcal{B} = \langle B, Con^{\mathcal{B}}, Pred^{\mathcal{B}} \rangle 1$ etc.

We define the value of an arbitrary individual term as

$$|t|_{\mathcal{A},g} = \begin{cases} g(t) \text{ if } t \text{ is a variable;} \\ t^{\mathcal{A}} \text{ if } t \text{ is an individual constant.} \end{cases}$$

We will need a relation between assignments in specifying the truth conditions of quantified formulas: $g[x]g'$ iff g and g' assigning the same value to all variables other than x.

6 Semantics

If φ is a formula, $|\varphi|_{\mathcal{A},g}$ denotes its truth value. We define the truth conditions of formulas in terms of U, ϱ and g recursively as follows.

A1 $|P(t_1, \ldots, t_n)|_{\mathcal{A},g} = \begin{cases} \mathbf{t} \text{ if } \langle |t_1|_{\mathcal{A},g}, \ldots, |t_1|_{\mathcal{A},g} \rangle \in P^{\mathcal{A}}; \\ \mathbf{f} \text{ otherwise.} \end{cases}$

A2 $|t_1 = t_2|_{\mathcal{A},g} = \begin{cases} \mathbf{t} \text{ if } |t_1|_{\mathcal{A},g} = |t_2|_{\mathcal{A},g}; \\ \mathbf{f} \text{ otherwise.} \end{cases}$

F1 (a) $|(\varphi \wedge \psi)|_{\mathcal{A},g} = \begin{cases} \mathbf{t} \text{ if } |\varphi|_{\mathcal{A},g} = \mathbf{t} \text{ and } |\varphi|_{\mathcal{A},g} = \mathbf{t}; \\ \mathbf{f} \text{ otherwise.} \end{cases}$

(b) $|(\varphi \vee \psi)|_{\mathcal{A},g} = \begin{cases} \mathbf{f} \text{ if } |\varphi|_{\mathcal{A},g} = \mathbf{f} \text{ and } |\varphi|_{\mathcal{A},g} = \mathbf{f}; \\ \mathbf{t} \text{ otherwise.} \end{cases}$

(c) $|(\varphi \rightarrow \psi)|_{\mathcal{A},g} = \begin{cases} \mathbf{f} \text{ if } |\varphi|_{\mathcal{A},g} = \mathbf{t} \text{ and } |\varphi|_{\mathcal{A},g} = \mathbf{f}; \\ \mathbf{t} \text{ otherwise.} \end{cases}$

(d) $|(\varphi \leftrightarrow \psi)|_{\mathcal{A},g} = \begin{cases} \mathbf{t} \text{ if } |\varphi|_{\mathcal{A},g} = |\varphi|_{\mathcal{A},g}; \\ \mathbf{f} \text{ otherwise.} \end{cases}$

(e) $|\neg \varphi|_{\mathcal{A},g} = \begin{cases} \mathbf{t} \text{ if } |\varphi|_{\mathcal{A},g} = \mathbf{f}; \\ \mathbf{f} \text{ otherwise.} \end{cases}$

$$\text{F2}\quad \text{(a)}\ |\forall x\ \varphi|_{\mathcal{A},g} = \begin{cases} \mathbf{t} \text{ if } |\varphi|_{\mathcal{A},g'} = \mathbf{t} \text{ for all } g' \text{ such that } g[x]g'; \\ \mathbf{f} \text{ otherwise.} \end{cases}$$

$$\text{(b)}\ |\exists x\ \varphi|_{\mathcal{A},g} = \begin{cases} \mathbf{t} \text{ if } |\varphi|_{\mathcal{A},g'} = \mathbf{t} \text{ for some } g' \text{ such that } g[x]g'; \\ \mathbf{f} \text{ otherwise.} \end{cases}$$

If $|\varphi|_{\mathcal{A},g} = \mathbf{t}$, we say that φ is true in the structure $\mathcal{A}$ under the assignment g. If φ is true in $\mathcal{A}$ under every assignment g, we simply say that φ is true in $\mathcal{A}$.

If every member of a set Γ of formulas is true in a structure $\mathcal{A}$ under an assignment g, we say that $\mathcal{A}$ models Γ under g:

$$\mathcal{A} \models_g \Gamma \underset{\text{def}}{\iff} \text{for every } \varphi \in \Gamma,\ |\varphi|_{\mathcal{A},g} = \mathbf{t}$$

If $\mathcal{A}$ models Γ under every g, we simply say that $\mathcal{A}$ models Γ:

$$\mathcal{A} \models \Gamma \underset{\text{def}}{\iff} \text{for every } g, \mathcal{A} \models_g \Gamma$$

Modeling a single formula is a special case:

$$\mathcal{A} \models_g \varphi \underset{\text{def}}{\iff} \mathcal{A} \models_g \{\varphi\}$$

and

$$\mathcal{A} \models \varphi \underset{\text{def}}{\iff} \mathcal{A} \models \{\varphi\}$$

7 Some derived semantic concepts

Logical truth A formula φ is a logical truth (or a *valid formula* or a *tautology*) if and only if it is true in any structure, under every assignment; that is, $\mathcal{A} \models \varphi$ for all $\mathcal{A}$. In short we write $\models \varphi$; or sometimes $\Rightarrow \varphi$.

Logical equivalence The formulas φ and ψ are equivalent if and only if they have the same truth value in any structure, under every assignment; that is, $|\varphi|_{\mathcal{A},g} = |\psi|_{\mathcal{A},g}$ for all U, ϱ and g. In symbols: $\varphi \mathrel{=\!\!|\!\models} \psi$ or $\varphi \Leftrightarrow \psi$.

Logical consequence A formula φ is a consequence of a set of formulas Γ if and only if every structure that makes all the members of Γ true under every assignment also makes φ true under every assignment; that is, if $\mathcal{A} \models \Gamma$, then $\mathcal{A} \models \varphi$. In symbols: $\Gamma \models \varphi$.

Logical consequence: an alternative definition The above is the standard definition of logical consequence in mathematical logic. Sometimes linguists and philosophers prefer an alternative version, which we will call context-dependent consequence or C-consequence (this is *ad hoc* terminology). The alternative definition is as follows:

A formula φ is a C-consequence of a set of formulas Γ if and only if for every structure and assignment, if they make all the members of Γ true, they also make φ true; that is, if $\mathcal{A} \models_g \Gamma$, then $\mathcal{A} \models_g \varphi$. In symbols: $\Gamma \models_{\mathrm{C}} \varphi$.

The two definitions coincide in closed formulas, but sometimes differ in open formulas. For example, the universal generalization

$$\varphi \models \forall x\ \varphi$$

is valid, while its alternative

$$\varphi \models_{\mathrm{C}} \forall x\ \varphi$$

is valid only if x has no free occurrence in φ.