First Order Logic:
1. Syntax

May 19, 2020

Syntax: Alphabet

Note object level symbols and expressions versus meta–level text and formulae.

The alphabet of FOL consists of several non–intersecting alphabets:

1. fixed "standard" symbols:
   - logical connectives: $\{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow\}$,
   - parentheses: $\{(,)\}$,
   - quantifiers $(\forall, \exists)$
2. variables $(x, y, z, \ldots)$
3. function symbols $(f, g, h, \ldots)$: $\mathcal{F} = \bigcup_{k \in \mathbb{N}} \mathcal{F}_k$ (k: arity)

   0-ary functions are usually called *constants* $(a, b, c, \ldots)$
4. predicate symbols $(P, Q, R, \ldots)$: $\mathcal{P} = \bigcup_{k=0}^{\infty} \mathcal{P}_k$ (k: arity)

   0-ary predicates behave like the propositional variables $(A, B, C, \ldots)$

Remark: The "nonstandard" symbols listed above are examples of typical use, however they may be used with different roles in certain formulae. The structure of the formula determines the role of each symbol.

Syntax: Two languages

The language of FOL consists in terms and formulae.

Syntax: Terms

Note meta–level variables denoting object level elements versus object level and meta–level texts.

1. A constant is a term.
2. A variable is a term.
3. If f is an n-ary function symbol, and t_1, ..., t_n are terms then $f[t_1,...,t_n]$ is a term.
4. These are all terms.

Remark: Similar to the definition of syntax in propositional logic, the fourth clause above can be expressed more formally by specifying the set of terms as being the smallest set which has the first three properties, or alternatively the predicate "is term" as the strongest predicate with the three properties. The latter gives us the inductive principles for proving properties of terms and for constructing functions over the set of terms.

Syntax: Formulae

1. $\mathbb{T}$ and $\mathbb{F}$ are formulae;
   If P is an n-ary predicate symbol and t_1, ..., t_n are terms then $P[t_1,...,t_n]$ is a formula.
2. The propositional logical connectives are used like in propositional logic. (If φ is a formula then $(\neg\varphi)$ is a formula, etc.)
3. If φ is a formula and x is a variable, then $(\underset{x}{\forall}\varphi)$ and $(\underset{x}{\exists}\varphi)$ are formulae.
4. These are all formulae. (Similar remark as by terms.)

The formulae defined at point **1** are called atoms.
Atoms and negated atoms are also called literals.

In $(\underset{x}{\forall}\varphi)$ and $(\underset{x}{\exists}\varphi)$, we say:

- x is the quantified variable,
- φ is the scope of the quantifier,
- every occurrence of x in φ is bound (by the respective quantifier).

An occurrence in a formula φ of a variable x which is not bound by any quantifier is called free. We also say: x occurs free in φ.

A formula without free occurrences of variables is called closed.

Syntax: Examples

Identify constants, variables (free, bound), quantifiers, function symbols, predicate symbols, atoms, terms, formulae:

1. $\underset{x}{\forall}\ x + 1 \geq x$

   $\underset{x}{\forall}\ G[p[x, 1], x]$

2. $\neg\left(\underset{x}{\exists}\ 0 = x^{-1}\right)$

   $\neg\left(\underset{x}{\exists}\ E[0, f[x]]\right)$

3. $\underset{x}{\forall}\ \underset{y}{\exists}\left(P[y, f[x]]\ \wedge\ \underset{z}{\forall}(P[z, f[x]] \Rightarrow P[y, z])\right)$

   $\underset{y}{\exists}\left(P[y, f[x]]\ \wedge\ \underset{z}{\forall}(P[z, f[x]] \Rightarrow P[y, z])\right)$

   $P[y, f[x]]\ \wedge\ \underset{z}{\forall}(P[z, f[x]] \Rightarrow P[y, z])$

Syntax: Examples (constants and variables)

Constants and variables (free, bound).

1. $\underset{x}{\forall}\ x + 1 \geq x$

$\underset{x}{\forall}\ G[p[x, 1], x]$

2. $\neg\left(\underset{x}{\exists}\ 0 = x^{-1}\right)$

$\neg\left(\underset{x}{\exists}\ E[0, f[x]]\right)$

3. $\underset{x}{\forall}\ \underset{y}{\exists}\left(P[y, f[x]]\ \wedge\ \underset{z}{\forall}\left(P[z, f[x]] \Rightarrow P[y, z]\right)\right)$

$\underset{y}{\exists}\left(P[y, f[x]]\ \wedge\ \underset{z}{\forall}\left(P[z, f[x]] \Rightarrow P[y, z]\right)\right)$

$P[y, f[x]]\ \wedge\ \underset{z}{\forall}\left(P[z, f[x]] \Rightarrow P[y, z]\right)$

Syntax: Examples (functions and predicates)

Function symbols and predicate symbols

1. $\underset{x}{\forall}\ x + 1 \geq x$

   $\underset{x}{\forall}\ \mathrm{G}[\mathrm{p}[x, 1], x]$

2. $\neg\left(\underset{x}{\exists}\ 0 = x^{\text{-1}}\right)$

   $\neg\left(\underset{x}{\exists}\ \mathrm{E}[0, \mathrm{f}[x]]\right)$

3. $\underset{x}{\forall}\ \underset{y}{\exists}\left(\mathrm{P}[y, \mathrm{f}[x]] \wedge \underset{z}{\forall}\left(\mathrm{P}[z, \mathrm{f}[x]] \Rightarrow \mathrm{P}[y, z]\right)\right)$

   $\underset{y}{\exists}\left(\mathrm{P}[y, \mathrm{f}[x]] \wedge \underset{z}{\forall}\left(\mathrm{P}[z, \mathrm{f}[x]] \Rightarrow \mathrm{P}[y, z]\right)\right)$

   $\mathrm{P}[y, \mathrm{f}[x]] \wedge \underset{z}{\forall}\left(\mathrm{P}[z, \mathrm{f}[x]] \Rightarrow \mathrm{P}[y, z]\right)$

Syntax: Examples (formulae)

Predicate symbols, atoms, quantifiers, and $\boxed{\text{formulae}}$.

1. $\boxed{\underset{x}{\forall}\ \boxed{x + 1 \geq x}}$

   $\boxed{\underset{x}{\forall}\ \boxed{G[p[x, 1], x]}}$

2. $\boxed{\neg\ \boxed{\underset{x}{\exists}\ \boxed{0 = x^{-1}}}}$

3. $\boxed{\underset{x}{\forall}\ \boxed{\underset{y}{\exists}\ \boxed{\left(\boxed{\mathrm{P}[\mathrm{y},\ \mathrm{f}[\mathrm{x}]]} \wedge \boxed{\underset{z}{\forall}\ \boxed{\left(\boxed{\mathrm{P}[\mathrm{z},\ \mathrm{f}[\mathrm{x}]]} \Rightarrow \boxed{\mathrm{P}[\mathrm{y},\mathrm{z}]}\right)}}\right)}}}$

Syntax: Examples (terms)

Function symbols, $\boxed{\text{terms}}$, predicate symbols, and atoms.

1. $\underset{x}{\forall}\ \boxed{\boxed{x} + \boxed{1}} \geq x$

$\underset{x}{\forall}\ G[\boxed{p[\boxed{x}, \boxed{1}]}, x]$

2. $\neg \underset{x}{\exists}\ \boxed{0} = \boxed{\boxed{x}^{-1}}$

$\neg \underset{x}{\exists}\ E[\boxed{0}, \boxed{f[\boxed{x}]}]$

3. $\underset{x}{\forall} \underset{y}{\exists} \left(P[\boxed{y}, \boxed{f[\boxed{x}]}] \wedge \underset{z}{\forall} \left(P[\boxed{z}, \boxed{f[\boxed{x}]}] \Rightarrow P[\boxed{y}, \boxed{z}] \right) \right)$

Syntax: Definition of Convergence

A typical mathematical formula: "f is convergent to a":

$$\forall_a (C[f,a] \Leftrightarrow \forall_{\epsilon>0} \exists_{\delta>0} \forall_x (|x-a|<\delta \ \Rightarrow \ |f[x]-f[a]|<\epsilon))$$

The notation for quantifiers is an abbreviation of:

$$\forall_a (C[f,a] \Leftrightarrow \forall_\epsilon (\epsilon>0 \ \Rightarrow \exists_\delta (\delta>0 \Rightarrow \forall_x (|x-a|<\delta \ \Rightarrow \ |f[x]-f[a]|<\epsilon))))$$

To express it in first order predicate logic syntax, we replace the special mathematical notation by predicate symbols and function symbols:
We denote $>$ by P, $<$ by Q, $|\ldots|$ by g, and $-$ by h.

$$\begin{aligned}
&(\forall_a (C[f,a] \Leftrightarrow \\
&\quad (\forall_\epsilon (P[\epsilon,0] \Rightarrow \\
&\quad\quad (\exists_\delta (P[\delta,0] \ \wedge (\forall_x (Q[h[x,a],\delta] \Rightarrow \ Q[h[g[f[x],f[a]]],\epsilon]))))))))
\end{aligned}$$

This formula contains all parentheses according to the syntax definition, but in practice we may abbreviate by omitting some unnecessary ones:

$$\begin{aligned}
&\forall_a (C[f,a] \Leftrightarrow \\
&\quad \forall_\epsilon (P[\epsilon,0] \ \Rightarrow \exists_\delta (P[\delta,0] \ \wedge \forall_x (Q[h[x,a],\delta] \ \Rightarrow \ Q[h[g[f[x],f[a]]],\epsilon])))).
\end{aligned}$$