Syntax of FOL

The *syntax* of a language defines the way in which basic elements of the language may be put together to form clauses of that language. In the case of FOL, the basic ingredients are (besides the logic operators): *variables*, *function symbols*, and *predicate symbols*. Each function and predicate symbol is associated with an *arity* $n \geq 0$.

Definition 1 (Terms) *We inductively define the set of terms as the smallest set such that:*

(i) every variable is a term;

(ii) if f is a function symbol of arity k and $t_1, \ldots, t_k$ are terms, then $f(t_1, \ldots, t_k)$ is also a term.

Function symbols of arity 0 are better known as *constants*.

Syntax of FOL (2)

Definition 2 (Formulas) *We inductively define the set of formulas as the smallest set such that:*

(i) if P is a predicate symbol of arity k and $t_1 \ldots, t_k$ are terms, then $P(t_1, \ldots, t_k)$ is a formula;

(ii) if φ and ψ are formulas, so are $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$, and $\varphi \leftrightarrow \psi$;

(iii) if x is a variable and φ is a formula, then $(\forall x)\varphi$ and $(\exists x)\varphi$ are also formulas.

Also recall: *precedence hierarchy of connectives*, *subformulas*, *ground terms*, *atoms*, *literals*, *quantifier scope*, *bound* and *free variables*, *closed formulas* (aka *sentences*).

Substitution

Definition 3 (Substitution) *Let φ be a formula, x a variable, and t a term. Then $\varphi[x \leftarrow t]$ denotes the formula we get by replacing each free occurrence of x in φ by t.*

Semantics of FOL

The *semantics* of a language defines the meaning of clauses in that language. In the case of FOL, we do this through the notion of models (and variable assignments).

Definition 4 (Models) *A model is a pair $\mathcal{M} = (\mathcal{D}, \mathcal{I})$, where $\mathcal{D}$ (the 'domain') is a non-empty set of objects and $\mathcal{I}$ (the 'interpretation function') is mapping each n-place function symbol f to some n-ary function $f^{\mathcal{I}} : \mathcal{D}^n \to \mathcal{D}$ and each n-place predicate symbol P to some n-ary relation $P^{\mathcal{I}} : \mathcal{D}^n \to \{true, false\}$.*

Observation. This definition also covers the cases of 0-place function symbols (constants) and predicate symbols.
Because: (1) $\mathcal{D}^0 = \varnothing$ and (2) a function from $\varnothing$ into a set X is just an element of X.

Semantics of FOL (2)

Definition 5 (Assignments) *A variable assignment over a domain $\mathcal{D}$ is a function g from the set of variables to $\mathcal{D}$.*

Definition 6 (Valuation of terms) *We define a valuation function $val_{\mathcal{I},g}$ over terms as follows:*

$$
\begin{aligned}
val_{\mathcal{I},g}(x) &= g(x) \quad \text{for variables } x \\
val_{\mathcal{I},g}(f(t_1,\ldots,t_n)) &= f^{\mathcal{I}}(val_{\mathcal{I},g}(t_1),\ldots,val_{\mathcal{I},g}(t_n))
\end{aligned}
$$

Definition 7 (Assignment variants) *Let g and g' be assignments over $\mathcal{D}$ and let x be a variable, Then g' is called an x-variant of g iff $g(y) = g'(y)$ for all variables $y \neq x$.*

Semantics of FOL (3)

Definition 8 (Satisfaction relation) *We write $\mathcal{M}, g \models \varphi$ to say that the formula φ is satisfied in the model $\mathcal{M} = (\mathcal{I}, \mathcal{D})$ under the assignment g. The relation $\models$ is defined inductively as follows:*

(i) $\mathcal{M}, g \models P(t_1,\ldots,t_n)$ *iff* $P^{\mathcal{I}}(val_{\mathcal{I},g}(t_1),\ldots,val_{\mathcal{I},g}(t_n)) = true$;

(ii) $\mathcal{M}, g \models \neg\varphi$ *iff not* $\mathcal{M}, g \models \varphi$;

(iii) $\mathcal{M}, g \models \varphi \wedge \psi$ *iff* $\mathcal{M}, g \models \varphi$ *and* $\mathcal{M}, g \models \psi$;

(iv) $\mathcal{M}, g \models \varphi \vee \psi$ *iff* $\mathcal{M}, g \models \varphi$ *or* $\mathcal{M}, g \models \psi$;

(v) $\mathcal{M}, g \models (\forall x)\varphi$ *iff* $\mathcal{M}, g' \models \varphi$ *for all x-variants g' of g; and*

(vi) $\mathcal{M}, g \models (\exists x)\varphi$ *iff* $\mathcal{M}, g' \models \varphi$ *for some x-variant g' of g.*

Semantics of FOL (4)

Observe that in the case of *closed* formulas φ the variable assignment g does not matter (we just write $\mathcal{M} \models \varphi$).

Satisfiability. A closed formula φ is called *satisfiable* iff it has a model, i.e. there is a model $\mathcal{M}$ with $\mathcal{M} \models \varphi$.

Validity. A closed formula φ is called *valid* iff for every model $\mathcal{M}$ we have $\mathcal{M} \models \varphi$. We write $\models \varphi$. (Observe: φ is valid iff $\neg\varphi$ is not satisfiable.)

Consequence relation. Let φ be a closed formula and let Δ be a set of closed formulas. We write $\Delta \models \varphi$ iff whenever $\mathcal{M} \models \psi$ holds for all $\psi \in \Delta$ then also $\mathcal{M} \models \varphi$ holds. (Observe: $\Delta \models \varphi$ holds iff $\Delta \cup \{\neg\varphi\}$ is not satisfiable.)

(The discipline of *Automated Deduction* is essentially concerned with the study of algorithms to prove (or disprove) $\Delta \models \varphi$.)

Abbreviations

$$
\begin{array}{rcll}
\varphi \rightarrow \psi & \equiv & \neg\varphi \vee \psi & \\
\varphi \leftrightarrow \psi & \equiv & (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi) & \\
\top & \equiv & P \vee \neg P & \text{for an arbitrary 0-place predicate symbol } P \\
\bot & \equiv & \neg\top &
\end{array}
$$