

## syn.1 Introduction

fol:syn:int:  
sec

In order to develop the theory and metatheory of first-order logic, we must first define the syntax and semantics of its expressions. The expressions of first-order logic are terms and **formulas**. Terms are formed from **variables**, **constant symbols**, and **function symbols**. **Formulas**, in turn, are formed from **predicate symbols** together with terms (these form the smallest, “atomic” **formulas**), and then from atomic **formulas** we can form more complex ones using logical connectives and quantifiers. There are many different ways to set down the formation rules; we give just one possible one. Other systems will chose different symbols, will select different sets of connectives as primitive, will use parentheses differently (or even not at all, as in the case of so-called Polish notation). What all approaches have in common, though, is that the formation rules define the set of terms and **formulas** *inductively*. If done properly, every expression can result essentially in only one way according to the formation rules. The inductive definition resulting in expressions that are *uniquely readable* means we can give meanings to these expressions using the same method—inductive definition.

Giving the meaning of expressions is the domain of semantics. The central concept in semantics is that of satisfaction in a **structure**. A **structure** gives meaning to the building blocks of the language: a **domain** is a non-empty set of objects. The quantifiers are interpreted as ranging over this domain, **constant symbols** are assigned elements in the domain, **function symbols** are assigned functions from the **domain** to itself, and **predicate symbols** are assigned relations on the **domain**. The **domain** together with assignments to the basic vocabulary constitutes a **structure**. **Variables** may appear in **formulas**, and in order to give a semantics, we also have to assign **elements** of the **domain** to them—this is a variable assignment. The satisfaction relation, finally, brings these together. A **formula** may be satisfied in a **structure**  $\mathfrak{M}$  relative to a variable assignment  $s$ , written as  $\mathfrak{M}, s \models \varphi$ . This relation is also defined by induction on the structure of  $\varphi$ , using the truth tables for the logical connectives to define, say, satisfaction of  $\varphi \wedge \psi$  in terms of satisfaction (or not) of  $\varphi$  and  $\psi$ . It then turns out that the variable assignment is irrelevant if the **formula**  $\varphi$  is a **sentence**, i.e., has no free variables, and so we can talk of **sentences** being simply satisfied (or not) in **structures**.

On the basis of the satisfaction relation  $\mathfrak{M} \models \varphi$  for sentences we can then define the basic semantic notions of validity, entailment, and satisfiability. A sentence is valid,  $\models \varphi$ , if every structure satisfies it. It is entailed by a set of **sentences**,  $\Gamma \models \varphi$ , if every **structure** that satisfies all the **sentences** in  $\Gamma$  also satisfies  $\varphi$ . And a set of sentences is satisfiable if some **structure** satisfies all **sentences** in it at the same time. Because **formulas** are inductively defined, and satisfaction is in turn defined by induction on the structure of **formulas**, we can use induction to prove properties of our semantics and to relate the semantic notions defined.

**Photo Credits**

**Bibliography**