

syn.1 First-Order Languages

fol:syn:fol:
sec

Expressions of first-order logic are built up from a basic vocabulary containing *variables*, *constant symbols*, *predicate symbols* and sometimes *function symbols*. From them, together with logical connectives, quantifiers, and punctuation symbols such as parentheses and commas, *terms* and *formulas* are formed.

Informally, *predicate symbols* are names for properties and relations, *constant symbols* are names for individual objects, and *function symbols* are names for mappings. These, except for the *identity predicate* $=$, are the *non-logical symbols* and together make up a language. Any first-order language \mathcal{L} is determined by its non-logical symbols. In the most general case, \mathcal{L} contains infinitely many symbols of each kind.

explanation

In the general case, we make use of the following symbols in first-order logic:

1. Logical symbols
 - a) Logical connectives: \neg (negation), \wedge (conjunction), \vee (disjunction), \rightarrow (conditional), \leftrightarrow (biconditional), \forall (universal quantifier), \exists (existential quantifier).
 - b) The propositional constant for falsity \perp .
 - c) The propositional constant for truth \top .
 - d) The two-place identity predicate $=$.
 - e) A denumerable set of variables: v_0, v_1, v_2, \dots
2. Non-logical symbols, making up the *standard language* of first-order logic
 - a) A denumerable set of n -place predicate symbols for each $n > 0$: $A_0^n, A_1^n, A_2^n, \dots$
 - b) A denumerable set of constant symbols: c_0, c_1, c_2, \dots
 - c) A denumerable set of n -place function symbols for each $n > 0$: $f_0^n, f_1^n, f_2^n, \dots$
3. Punctuation marks: $(,)$, and the comma.

Most of our definitions and results will be formulated for the full standard language of first-order logic. However, depending on the application, we may also restrict the language to only a few *predicate symbols*, *constant symbols*, and *function symbols*.

Example syn.1. The language \mathcal{L}_A of arithmetic contains a single two-place predicate symbol $<$, a single constant symbol 0 , one one-place function symbol $!$, and two two-place function symbols $+$ and \times .

Example syn.2. The language of set theory \mathcal{L}_Z contains only the single two-place predicate symbol \in .

Example syn.3. The language of orders \mathcal{L}_{\leq} contains only the two-place predicate symbol \leq .

Again, these are conventions: officially, these are just aliases, e.g., $<$, \in , and \leq are aliases for A_0^2 , o for c_0 , \prime for f_0^1 , $+$ for f_0^2 , \times for f_1^2 .

intro You may be familiar with different terminology and symbols than the ones we use above. Logic texts (and teachers) commonly use either \sim , \neg , and $!$ for “negation”, \wedge , \cdot , and $\&$ for “conjunction”. Commonly used symbols for the “conditional” or “implication” are \rightarrow , \Rightarrow , and \supset . Symbols for “biconditional,” “bi-implication,” or “(material) equivalence” are \leftrightarrow , \Leftrightarrow , and \equiv . The \perp symbol is variously called “falsity,” “falsum,” “absurdity,” or “bottom.” The \top symbol is variously called “truth,” “verum,” or “top.”

It is conventional to use lower case letters (e.g., a , b , c) from the beginning of the Latin alphabet for constant symbols (sometimes called names), and lower case letters from the end (e.g., x , y , z) for variables. Quantifiers combine with variables, e.g., x ; notational variations include $\forall x$, $(\forall x)$, (x) , Πx , \bigwedge_x for the universal quantifier and $\exists x$, $(\exists x)$, (Ex) , Σx , \bigvee_x for the existential quantifier.

explanation We might treat all the propositional operators and both quantifiers as primitive symbols of the language. We might instead choose a smaller stock of primitive symbols and treat the other logical operators as defined. “Truth functionally complete” sets of Boolean operators include $\{\neg, \vee\}$, $\{\neg, \wedge\}$, and $\{\neg, \rightarrow\}$ —these can be combined with either quantifier for an expressively complete first-order language.

You may be familiar with two other logical operators: the Sheffer stroke $|$ (named after Henry Sheffer), and Peirce’s arrow \downarrow , also known as Quine’s dagger. When given their usual readings of “nand” and “nor” (respectively), these operators are truth functionally complete by themselves.

Photo Credits

Bibliography