CS 395T
Spring 1996

First- and Second-Order Logic

Vladimir Lifschitz

1. Signatures and Interpretations. A *signature* is a set of *function constants* and *predicate constants*, with a nonnegative integer, called the *arity*, assigned to each of them. The function constants of arity 0 are called *object constants*. The predicate constants of arity 0 are called *propositional constants*.

For instance, the signature used in the blocks world example consists of the object constants A, B and C, the predicate constant On of arity 2, and the predicate constant $Ontable$ of arity 1.

An *interpretation* I of a signature σ consists of the following components:

- a nonempty set $|I|$, called the *universe* of I,
- for every object constant a in σ, an element $I[\![a]\!]$ of $|I|$,
- for every function constant f in σ of arity $n > 0$, a function $I[\![f]\!]$ from $|I|^n$ into $|I|$,
- for every propositional constant P in σ, an element $I[\![P]\!]$ of $\{\mathsf{f}, \mathsf{t}\}$,
- for every predicate constant P in σ of arity $n > 0$, a subset $I[\![P]\!]$ of $|I|^n$.

2. Syntax of First-Order Logic. Assume that we have chosen an infinite sequence of symbols called *object variables*. Let σ be a signature. The *terms* of σ are defined recursively, as follows: object constants and object variables are *terms*; if f is an n-ary function constant ($n > 0$) and $t_1, \dots, t_n$ are *terms* then $f(t_1, \dots, t_n)$ is a *term*. The *atomic formulas* of σ are

- propositional constants,
- expressions of the form $P(t_1, \dots, t_n)$, where P is an n-ary predicate constant ($n > 0$) and $t_1, \dots, t_n$ are terms,

- expressions of the form $(t_1 = t_2)$, where t_1 and t_2 are terms.

The *formulas* of σ are built from atomic formulas using the connectives $\neg$, $\wedge$ and the universal quantifier $\forall$. The other propositional connectives and the existential quantifier will be treated as abbreviations:

$$\begin{array}{l} A \vee B \text{ stands for } \neg(\neg A \wedge \neg B), \\ A \supset B \text{ stands for } \neg(A \wedge \neg B), \\ A \equiv B \text{ stands for } (A \supset B) \wedge (B \supset A), \\ \exists x A \text{ stands for } \neg \forall x \neg A. \end{array}$$

A *ground term* is a term without variables. A *sentence* is a formula without free variables. A *first-order theory* is a set of sentences, called the *axioms*.

3. Semantics of First-Order Logic. Let I be an interpretation of a signature σ. Extend the signature by adding to it, for every element $\xi \in |I|$, a new object constant a^ξ, called the *name* of ξ. We extend I to the new signature by defining $I[\![a^\xi]\!] = \xi$. For any ground term t of the extended signature, its *value* $I[\![t]\!]$ is defined recursively, as follows: if t is an object constant then $I[\![t]\!]$ is already defined; otherwise,

$$I[\![f(t_1, \ldots, t_n)]\!] = I[\![f]\!](I[\![t_1]\!], \ldots, I[\![t_n]\!]).$$

The relation $I \models A$ ("I satisfies A") between an interpretation I and a sentence A of the extended signature is defined recursively, as follows:

- if P is a propositional constant, then $I \models P$ iff $I(P) = \mathsf{t}$,
- $I \models P(t_1, \ldots, t_n)$ iff $\langle I[\![t_1]\!], \ldots, I[\![t_n]\!]\rangle \in I[\![P]\!]$,
- $I \models t_1 = t_2$ iff $I[\![t_1]\!] = I[\![t_2]\!]$,
- $I \models \neg A$ iff $I \not\models A$,
- $I \models A \wedge B$ iff $I \models A$ and $I \models B$,
- $I \models \forall x A(x)$ iff, for every name a^ξ, $I \models A(a^\xi)$.

A sentence A is *logically valid* if every interpretation satisfies A. A *model* of a first-order theory T is an interpretation that satisfies all axioms of T.

4. Syntax of Second-Order Logic. Assume that for every nonnegative integer n we have chosen an infinite sequence of symbols called *n-ary function variables* and an infinite sequence of symbols called *n-ary predicate variables*. The function variables of arity 0 are called *object variables*. The predicate variables of arity 0 are called *propositional variables*. In examples, variables will be denoted by lower-case letters.

Let σ be a signature. The *terms* of σ are defined recursively, as follows: object constants and object variables are *terms*; if f is an n-ary function constant or variable ($n > 0$) and $t_1, \dots, t_n$ are *terms* then $f(t_1, \dots, t_n)$ is a *term*. The *atomic formulas* of σ are

- propositional constants and variables,
- expressions of the form $P(t_1, \dots, t_n)$, where P is an n-ary predicate constant or variable ($n > 0$) and $t_1, \dots, t_n$ are terms,
- expressions of the form $(t_1 = t_2)$, where t_1 and t_2 are terms.

The *formulas* of σ are built from atomic formulas using the connectives $\neg$, $\wedge$ and the universal quantifier $\forall$; quantifiers can be followed by function or predicate variables of any arity.

Ground terms and *sentences* are defined as in Section 2, with the words "term" and "formula" understood in the sense of second-order logic. A *second-order theory* is a set of sentences in the sense of second-order logic.

For example,

$$p(x) \wedge \neg\exists y(p(y) \wedge On(x, y))$$

is a formula whose free variables are x and p. Call this formula $Bottom(x, p)$. Then

$$\forall p[\exists x\, p(x) \supset \exists x(p(x) \wedge Bottom(x, p))]$$

is a sentence.

5. Semantics of Second-Order Logic. Let I be an interpretation of a signature σ. Extend the signature by adding to it new constants, called *names*, as follows:

- each element of $|I|$ has a name, which is an object constant,

- for every $n > 0$, each function from $|I|^n$ into $|I|$ has a name, which is an n-ary function constant,

- f and t have names, which are propositional constants,

- for every $n > 0$, each subset of $|I|^n$ has a name, which is an n-ary predicate constant.

Then we extend I to the new signature and define, in the same way as in first-order logic, the *value* of a ground term and the *satisfaction relation* $\models$ between an interpretation and a sentence. *Logically valid* sentences and *models* are defined as in first-order logic also.

For instance, the following sentences are logically valid:

$$\exists p \forall x\, p(x),$$
$$\exists f \forall x (f(x) = x),$$
$$\forall xy [x = y \equiv \forall p (p(x) \supset p(y))].$$

6. Derivations in Second-Order Logic. How can we derive new second-order formulas from a given set of second-order assumptions? One method is to apply first-order logic and to treat function and predicate variables of different arities as object variables of different sorts. There are very simple logically valid formulas, however, which cannot be proved by this method, such as the three formulas shown above. Treating predicate variables as object variables does not reflect an important feature of their interpretation: n-ary predicate variables range over *all* subsets of $|M|^n$, not merely over some nonempty collection of subsets.

Thus we need additional logical axioms which guarantee the existence of predicates and functions. For predicates, the second-order logic includes the axiom schema

$$\exists p \forall x (p(x) \equiv F(x)),$$

where p is an n-ary predicate variable, x an n-tuple of distinct object variables, and $F(x)$ a formula without free occurrences of p. For functions, we can postulate

$$\forall x \exists ! y\, p(x, y) \supset \exists f \forall x\, p(x, f(x)),$$

where x is an n-tuple of distinct object variables, y an object variable different from the members of x, p an $(n+1)$-ary predicate variable, f an n-ary function variable, and $\exists!$ is an abbreviation for the combination of connectives "there exists exactly one".

The formal system of second-order logic outlined above is sufficient for formalizing many proofs of second-order formulas. But it is incomplete. For instance, the *axiom of choice*

$$\forall x \exists y\, p(x, y) \supset \exists f \forall x\, p(x, f(x))$$

is a logically valid formula which cannot be proved in this formal system. Gödel's completeness theorem has no counterpart in second-order logic.