2 Review of First-Order Logic

First-order logic (also called *predicate logic*) is an extension of propositional logic that is much more useful than propositional logic. It was created as a way of formalizing common mathematical reasoning. You should have seen first-order logic previously. This section is only review.

In first-order logic, you start with a nonempty set of values called the *universe of discourse* U. Logical statements talk about properties of values in U and relationships among those values.

2.1 Predicates

In place of propositional variables, first-order logic uses *predicates*.

Definition 2.1. A *predicate* P takes zero or more parameters $x_1, x_2, \ldots, x_n$ and yields either true or false. First-order formula $P(x_1, \ldots, x_n)$ is the value of predicate P with parameters $x_1, \ldots, x_n$. A predicate with no parameters is a propositional variable. If P takes no parameters then P is a first-order formula.

Suppose that U is the set of all integers. Here are some examples of predicates. There is no standard collection of predicates that are always used. Rather, each of these is like a function definition in a computer program; different programs contain different functions.

- We might define $\text{even}(n)$ to be true if n is even. For example $\text{even}(4)$ is true and $\text{even}(5)$ is false.
- We might define $\text{greater}(x, y)$ to be true if $x > y$. For example, $\text{greater}(7, 3)$ is true and $\text{greater}(3, 7)$ is false.
- We might define $\text{increasing}(x, y, z)$ to be true if $x < y < z$. For example, $\text{increasing}(2, 4, 6)$ is true and $\text{increasing}(2, 4, 2)$ is false.

2.2 Terms

A *term* is an expression that stands for a particular value in U. The simplest kind of term is a *variable*, which can stand for any value in U.

A *function* takes zero or more parameters that are members of U and yields a member of U. Here are examples of functions that might be defined when U is the set of all integers.

- A function with no parameters is called a *constant*. We might define function zero with no parameters to be the constant 0.
- We might define successor(n) to be $n+1$. For example, successor(2) = 3.
- We might define sum(m, n) to be $m + n$. For example, sum(5, 7) = 12.
- We might define largest(a, b, c) to be the largest of a, b and c. For example, largest(3, 9, 4) = 9 and largest(4, 4, 4) = 4.

Definition 2.2. A *term* is defined as follows.

1. A *variable* is a term. We use single letters such as x and y for variables.
2. If f is a function that takes no parameters then f is a term (standing for a value in U).
3. If f is a function that takes $n > 0$ parameters and $t_1, \ldots, t_n$ are terms then $f(t_1, \ldots, t_n)$ is a term.

For example, sum(sum(x, y), successor(z)) is a term.

The meaning of a term should be clear, provided the values of variables are known. Term sum(x, y) stands for the result that function sum yields on parameters (x, y) (the sum of x and y).

2.3 First-order formulas

Definition 2.3. A *first-order formula* is defined as follows.

1. **T** and **F** are first-order formulas.
2. If P is a predicates that takes no parameters then P is a first-order formula.
3. If $t_1, \ldots, t_n$ are terms and P is a predicate that takes $n > 0$ parameters, then $P(t_1, \ldots, t_n)$ is a first-order formula. It is true if $P(v_1, \ldots, v_n)$ is true, where v_1 is the value of term t_1, v_2 is the value of term t_2, etc.
4. If t_1 and t_2 are terms then $t_1 = t_2$ is a first-order formula. (It is true if terms t_1 and t_2 have the same value.)
5. If A and B are first-order formulas and x is a variable then each of the following is a first-order formula.
   (a) (A)
   (b) $\neg A$
   (c) $A \vee B$
   (d) $A \wedge B$
   (e) $\forall x\, A$
   (f) $\exists x\, A$

The meaning of parentheses, **T**, **F**, $\neg$, $\vee$ and $\wedge$ are the same as in propositional logic. Symbols $\forall$ and $\exists$ are called *quantifiers*. You read $\forall x$ as "for all x, and $\exists x$ as "for some x" or "there exists an x". They have the following meanings.

1. $\forall x\, A$ is true of A is true for all values of x in U.
2. $\exists x\, A$ is true if A is true for at least one value of x in U.

By convention, quantifiers have higher precedence than all of the operators $\wedge$, $\vee$, etc.

Examples of first-order formulas are:

1. $P(\text{sum}(x, y))$ says that, if $v = \text{sum}(x, y)$, then $P(v)$ is true. Its value (true or false) depends on the meanings of predicate P and function sum, as well as on the values of variables x and y.

2. $\forall x(\text{greater}(x, x))$ says that $\text{greater}(x, x)$ is true for every value x in U. Using the meaning of $\text{greater}(a, b)$ given above, $\forall x(\text{greater}(x, x))$ is clearly false, since no x can be greater than itself.

3. $\neg\forall x(\text{greater}(x, x))$ says that $\forall x(\text{greater}(x, x))$ is false. That is true.

4. $\exists y(y = \text{sum}(y, y))$ says that there exists a value y where $y = y + y$. That is true since $0 = 0 + 0$.

5. $\forall x(\exists y(\text{greater}(y, x)))$ says that, for every value v of x, first-order formula $\exists y(\text{greater}(y, v))$ is true. That is true. If $v = 100$, then choose $y = 101$, which is larger than 100. If $v = 1000$, choose $y = 1001$. If $v = 1{,}000{,}000$, choose $y = 1{,}000{,}001$.

6. $\exists y(\forall x(\text{greater}(y, x)))$ says that there exists a value v of y so that $\forall x(\text{greater}(v, x))$. That is false. There is no single value v that is larger than every integer x.

Operators $\rightarrow$, $\leftrightarrow$ and $\equiv$ have the same meanings in first-order logic as in propositional logic.

2.4 Sentences

Example 1 above uses variable x and y, and its value cannot be determined without knowing the values of x and y. It only makes sense if the values of x and y have already been specified. Think of them as similar to global variables in a function definition in a computer program.

The other examples above do not depend on any variable values. They manage their own variables, and are similar to a function definition that only uses local variables.

We say that variable x is *bound* if it occurs inside A in a first-order formula of the form $\forall x\, A$ or $\exists x\, A$.

Definition 2.4. A first-order formula is a *sentence* if all of its variables are bound.

Table 2-1. Some valid equivalences
$\exists x\, P(x) \vee \neg\exists x\, P(x)$
$\forall x\, P(x) \;\wedge\; \exists y\, Q(y) \;\equiv\; \exists y\, Q(y) \;\wedge\; \forall x\, P(x)$
$\neg(\forall x\, A) \;\equiv\; \exists x(\neg A)$
$\neg(\exists x\, A) \;\equiv\; \forall x(\neg A)$
$\forall x(A \wedge B) \;\equiv\; \forall x\, A \wedge \forall x\, B$
$\forall x\, A \to \exists x\, A$

2.5 Validity

Recall that a propositional formula is *valid* if it is true for all values of the variables that it contains. There is a similar concept of validity for first-order formulas.

Definition 2.5. Suppose that S is a sentence of first-order logic. (That is, it does not contain any unbound variables.) We say that S is *valid* if it is true regardless of the universe of discourse and the meanings of the predicates and functions that it mentions.

One way to get a valid first-order formula is to substitute first-order formulas into a propositional tautology. The following table lists two valid first-order formulas found in that way. Table 2-1 lists a few valid first-order equivalences, the first two of which are examples of substituting a first-order formula into a propositional equivalence.

2.6 Notation

First-order logic notation is usually extended to include common mathematical notation. For example, we write $x > y$ rather than greater(x, y), and $x + y$ rather than sum(x, y). Constants such as 0, 1 and 200 are also usually allowed. Instead of writing even(x), we write "x is even". For example,

$$\forall x(x \text{ is even } \wedge y \text{ is even } \to x + y \text{ is even})$$

is true. Those notational changes make first-order logic more readable.

prev **next**