# Chapter 2

# First-Order Logic

We next consider first-order logic, which is also called quantification theory and predicate calculus.[1] Initially we discuss first-order logic without equality, and then we discuss first-order logic with equality in §26.

## §20. The Language of $\mathcal{F}$

Having discussed in Chapter 1 how to combine statements using propositional connectives, we next turn to the problem of formally representing the statements themselves in a way that shows more of their structure. We shall focus on those aspects of the structure of statements which are most important for representing reasoning involving these statements.

Statements make assertions about entities (such as people, animals, plants, colors, qualities, numbers, abstractions, ideas, feelings, etc.), which are designated by nouns and pronouns. For convenience we shall refer to these entities as *objects* or *individuals*, slightly extending the usual meanings of these words in English. In our formal language we shall introduce *terms* (which will be defined below) to denote (serve as names for) these individuals.

Statements can be regarded as asserting that individuals have certain properties, or that individuals are related to each other in certain ways. For example, "Jack will go to the picnic" expresses the assertion that the individual Jack has the property of being an individual who will go to the

[1] Frege [Frege, 1879] is generally credited with first developing those aspects of first-order logic which go beyond propositional calculus, though others independently developed similar ideas somewhat later, and Frege's notation never found general favor. See [Church, 1956, pp. 288–294] for more information about the history of first-order logic.

picnic. "112 is prime" expresses the assertion (which happens to be false) that the individual 112 has the property of being a prime number. "Jill votes for Alice" expresses the assertion that the individuals "Jill" and "Alice" are related by the relation "votes for". "$5 < 112$" expresses the assertion that the individuals 5 and 112 are related by the relation " $<$ ". Another way of saying this is to assert that the relation " $<$ " *holds* between 5 and 112.

Since it is sometimes complicated to analyze a sentence in a natural language such as English or Chinese, it is convenient to have a quite simple and uniform syntax to represent the assertion that the entity $t$ has the property $P$, or that the relation $R$ holds between the entities $s$ and $t$. We shall use $Pt$ to represent the former, and $Rst$ to represent the latter. (Sometimes it will be convenient to introduce commas or parentheses and write these as $P(t)$ and $R(s,t)$, but the commas and parentheses are in principle unnecessary.) Thus, if we let "Prime" be the property of being a prime number, we can express "112 is prime" as "Prime(112)". Similarly, we can express "Jill votes for Alice" as "VotesFor(Jill, Alice)". We call names for properties and relations *predicates*, and they play such a fundamental role in first-order logic that it is sometimes called predicate calculus.

In order to be able to deal adequately with reasoning involving individuals, one must be able to represent statements containing words like "all", "every", and "some". To illustrate the relevant ideas, let us suppose that a class of girls holds an election for class president. Each girl is given a ballot containing the names of all the girls in the class, and is allowed to vote for as many members of the class as she pleases. Suppose we want to express in a formal language the assertion that everyone votes for Alice. One way one might try to formalize this is to introduce a term "everyone" into the formal language, and express the assertion as "VotesFor(everyone, Alice)". In a similar way, to express the assertion that Jill votes for someone, one might introduce a term "someone" into the formal language, and write "VotesFor(Jill, someone)". If such statements were permitted in the formal language, it would be very natural to also permit "VotesFor(everyone, someone)". What should this mean? One obvious possibility is "Everyone votes for someone". However, there is another possibility: "There is someone for whom everyone votes". Note that these express quite distinct assertions; the latter statement asserts that everyone votes for the same person, while the former statement does not.

Contemplation of this and other examples (particularly those dealing with more complicated statements) leads to the conclusion that a better way of formally representing assertions involving words like "all", "every", and "some" is needed. The solution that has been developed is to use *indi-*

| | |
|---|---|
| VotesFor(Jill, Alice) | Jill votes for Alice. |
| $\forall x$ VotesFor$(x, \mathit{Alice})$ | Everyone votes for Alice. |
| $\forall y$ VotesFor$(\mathit{Jill}, y)$ | Jill votes for everyone. |
| $\forall x$ VotesFor$(x, x)$ | Everyone votes for herself. |
| $\forall x \forall y$ VotesFor$(x, y)$ | Everyone votes for everyone. |
| $\forall x \exists y$ VotesFor$(x, y)$ | Everyone votes for someone. |
| $\exists y \forall x$ VotesFor$(x, y)$ | There is someone for whom everyone votes. |
| $\exists x \forall y$ VotesFor$(x, y)$ | There is someone who votes for everyone. |
| $\forall y \exists x$ VotesFor$(x, y)$ | Everyone gets someone's vote. |
| $\exists x \exists y$ VotesFor$(x, y)$ | Someone votes for someone. |
| $\exists x$ VotesFor$(x, x)$ | Someone votes for herself. |
| $\exists x$ VotesFor$(x, \mathit{Alice})$ | Someone votes for Alice. |
| $\exists y$ VotesFor$(\mathit{Jill}, y)$ | Jill votes for someone. |

Figure 2.1: Statements about voting in symbolic form and in English.

*vidual variables* to refer to arbitrary (unspecified) individuals, and express the assertion that "Everyone votes for Alice" as "For every $x$, $x$ votes for Alice", and express the assertion that "Jill votes for someone" as "There is a $y$ such that Jill votes for $y$." Just as we introduced symbols for propositional connectives, we introduce the symbol $\forall$ to abbreviate the phrase "for every" (or "for all") and the symbol $\exists$ and write "$\exists y$" to abbreviate the phrase "there is a $y$ such that" (where $y$ is any individual variable). Now we can express "Everyone votes for someone" as "$\forall x \exists y$ VotesFor$(x, y)$" and "There is someone for whom everyone votes" as "$\exists y \forall x$ VotesFor$(x, y)$". With this symbolic representation, the distinction between these assertions is made very clear. In Figure 2.1 we list a variety of statements about voting in symbolic form and in English. Note how the symbolic representations of these statements make it easy to focus on the essential logical differences between them.

Expressions of the form $\forall \mathbf{x}$ and $\exists \mathbf{x}$ are called *quantifiers*. They play such a fundamental role in first-order logic that it is sometimes called quantification theory. The reason we actually use the name "first-order logic" will become evident when we discuss higher-order logic at the beginning of section §50.

Just as $\vee$ can be defined in terms of $\sim$ and $\wedge$, $\exists$ can be defined in terms of $\sim$ and $\forall$. We can see this from the following example. Suppose our class of girls decides to vote on some issue, but each girl has the option of voting or abstaining (not voting). If we write "$x$ votes" as "Votes$(x)$", then "$\sim$Votes$(x)$" means "$x$ does not vote" or "$x$ abstains". Therefore

"$\forall$x$\sim$Votes($x$)" means "everyone abstains", and "$\sim \forall x \sim$Votes($x$)" means "not everyone abstains", which is of course equivalent to the assertion that "someone votes". From this and other examples it can be seen that $\exists x Px$ always has the same logical meaning as $\sim \forall x \sim Px$, provided that there is at least one individual.[2] This provides a contextual definition of $\exists$ in terms of $\sim$ and $\forall$. Therefore, for the sake of economy, $\exists$ will not be a primitive symbol of the formal system $\mathcal{F}$ of first-order logic which we shall soon introduce.

Sometimes we wish to refer to individuals by more complex expressions than we have used so far, such as "Jill's father", "$x + 3$", or "the value of the diamond". We can do this in a uniform way by having in our language notations for functions which map individuals to individuals. If we let f($x$) mean "the father of $x$", g($x, y$) mean "$x + y$", and V($x$) mean "the value of x", then we can represent "Jill's father" as f(Jill), "$x + 3$" as g($x$, 3), and "the value of the diamond" as V(the diamond).

The symbols we will use to refer to individuals, functions, and predicates will be *variables* or *constants*. Variables are used to refer to arbitrary elements of the appropriate type, whereas constants denote particular elements. Thus, variables may occur in quantifiers, but constants may not.

Now we are ready to start describing the system $\mathcal{F}$ of first-order logic. The *primitive symbols* of $\mathcal{F}$ are the following:

(a) Improper symbols: [ ] $\sim$ $\vee$ $\forall$
(b) Individual variables: $u\ v\ w\ x\ y\ z\ u_1\ v_1\ w_1\ \ldots\ u_2\ \ldots$
(c) $n$-ary function variables: $f^n\ g^n\ h^n\ f_1^n\ g_1^n\ h_1^n\ \ldots$ for each natural number $n \geq 1$.
(d) Propositional variables: $p\ q\ r\ s\ p_1\ q_1\ r_1\ s_1\ \ldots$
(e) $n$-ary predicate variables: $P^n\ Q^n\ R^n\ S^n\ P_1^n\ Q_1^n\ R_1^n\ S_1^n \ldots$ for each natural number $n \geq 1$.

An $n$-ary symbol is called *unary* or *monadic* if $n = 1$, and it is called *binary* if $n = 2$. There are denumerably many variables of each type. Individual variables may be regarded as 0-ary function variables, and propositional variables may be regarded as 0-ary predicate variables. In addition, there may be finitely or infinitely many individual, function, propositional, or predicate constants. We also admit systems in which there are no variables of certain types; however, individual variables must always be present, and there must be predicate variables or at least one predicate constant. Each choice of constants and variables determines a different formulation of the

[2]There is little need for a logical system to discuss nothing, so, as is customary, our logical system will be designed with the tacit assumption that there is at least one individual under discussion.

system $\mathcal{F}$, of course; thus we use $\mathcal{F}$ ambiguously as the name of any one of a variety of quite similar systems.

The *terms* and *wffs* of $\mathcal{F}$ are defined inductively by the following *formation rules*:

(a) Each individual variable or constant is a term.
(b) If $\mathbf{t}_1, \ldots, \mathbf{t}_n$ are terms and $\mathbf{f}^n$ is an $n$-ary function variable or constant, then $\mathbf{f}^n\mathbf{t}_1 \ldots \mathbf{t}_n$ is a term.
(c) If $\mathbf{t}_1, \ldots, \mathbf{t}_n$ are terms and $\mathbf{P}^n$ is an $n$-ary predicate variable or constant, then $\mathbf{P}^n\mathbf{t}_1 \ldots \mathbf{t}_n$ is a wff. Also, each propositional variable standing alone is a wff. (Wffs of these forms are called *atomic* or *elementary* wffs.)
(d) If $\mathbf{A}$ is a wff, so is $\sim\mathbf{A}$.
(e) If $\mathbf{A}$ and $\mathbf{B}$ are wffs, so is $[\mathbf{A} \vee \mathbf{B}]$.
(f) If $\mathbf{A}$ is a wff and $\mathbf{x}$ is an individual variable, then $\forall\mathbf{x}\mathbf{A}$ is a wff.

Sometimes $\forall\mathbf{x}\mathbf{A}$ is written as $(\forall\mathbf{x})\mathbf{A}$ or simply as $(\mathbf{x})\mathbf{A}$. We shall often omit the numerical superscripts from function and predicate symbols in terms and wffs, since the context will show what they must be.

We leave it to the reader (exercises X2006 and X2007) to formulate definitions of the sets of terms and wffs in the style of the definition of the set of wffs in §10 and to formulate and prove analogues of Theorem 1000 (the Principle of Induction on the Construction of a Wff) for terms and wffs of $\mathcal{F}$.

As in the case of propositional calculus, we shall use **A, B, C, D, E**, etc., as syntactical variables ranging over wffs. Similarly, we use **u, v, w, x, y, z**, etc., as syntactical variables for individual variables.

While in first-order logic only individual variables may be quantified, in second-order logic the other types of variables may be quantified too. Since variables designate arbitrary elements of the appropriate type, it is normally appropriate to substitute arbitrary expressions of the appropriate type for variables, under suitable conditions. Such substitution for constants is generally not appropriate since it would have the effect of implying that all elements have the special properties of the element denoted by the constant in question.

For the sake of economy, starting with §25 we shall restrict our attention to formulations of $\mathcal{F}$ in which there are no variables other than individual variables.

We adopt from §10 the conventions for regarding $[\mathbf{A} \wedge \mathbf{B}]$, $[\mathbf{A} \supset \mathbf{B}]$, and $[\mathbf{A} \equiv \mathbf{B}]$ as abbreviations for wffs (now wffs of $\mathcal{F}$). In addition,

http://www.springer.com/978-1-4020-0763-7

An Introduction to Mathematical Logic and Type Theory
To Truth Through Proof
Andrews, P.B.
2002, XVIII, 390 p., Hardcover
ISBN: 978-1-4020-0763-7