8.2. **Functions**

Functions were briefly mentioned in Section 2.1. I now give a more formal treatment, but one that relies upon a convenient simplification.

We said originally that a function would be defined upon a certain domain of objects, so that if you take any object from that domain as the input to the function, then the function will yield just one object as its output for that input. In the received terminology, the inputs to a function are called its arguments, and the output for a given input is called the value of the function for that argument. So the chief characteristic of a function, then, is that it has one and only one value for each of its appropriate arguments. But usually we allow that not *everything* need be counted as an appropriate argument for the function, i.e. that the function need not be defined for all arguments whatever, but only for arguments of a suitable kind. The simplification to be imposed is that we shall *not* allow this; on the contrary, all functions are to be taken as 'defined everywhere', or in other words every function is to have one and only one value for any object whatever taken as argument, i.e. any object from the whole domain that our quantifiers range over. (Another way of saying the same thing is that all our functions are to be *total*, rather than *partial*, functions.) It must be admitted that this assumption is not very realistic, and it introduces a problem which

will underlie most of the rest of this chapter. But the reason for imposing it is that it enables us to extend our formal languages to include functional expressions in a very simple way.

A function may take any number of arguments. The simplest kind of function is a one-place function, which takes just one object at a time as argument, and yields a value for that one argument. An example from arithmetic would be the function expressed by 'the square of . . .'; an example from every day would be the function expressed by 'the father of . . .' . But there are also two-place functions, such as that expressed by 'the sum of . . .' or 'the product of . . .' in arithmetic, and again there are three-place functions, and so on. (Functions of more than two places are usually complex, and put together from several simpler functions. For example, the arithmetical expression '$(x+y)\cdot z$' indicates a three-place function, which we could express in words as 'the result of multiplying the sum of x and y by z'. But there is nothing to stop one introducing simple three-place functions if there is a need for them.) In the other direction, we can, if we wish, speak of zero-place functions, but this is just a new-fangled name for a familiar item. For a zero-place function is something which cannot take any argument, but just has a single value, and that is to say that it has the same role as a name. Names, then, can be regarded as expressing zero-place functions if one wishes, but we already have a perfectly good notation for names, and do not need a new one. We do, however, need a new notation to represent other functions.

We shall use the letters

$$f,g,h,f_1,g_1,h_1,f_2,\ldots$$

as schematic function-letters. They take the place of particular expressions for functions, just as our schematic name-letters and predicate-letters take the place of particular names and predicates. For official purposes we shall regard each function-letter as furnished with a superscripted numeral (greater than 0) to show how many places that function has. But, as with predicate-letters, we shall always omit these superscripts in practice, since the rest of the notation will convey this information. The arguments to the function will be written in a pair of round brackets to the right of the function-letter, separated by commas where there is more than one argument. So our function-letters will appear in contexts such as

$$f(a), \quad g(a,b), \quad g(f(a),b).$$

The last example should be noted. If we start with the name a, and supply it as argument to the one-place function f, then the resulting expression $f(a)$ is

in effect another name, but a complex one. (We can read '$f(a)$' as 'the f of a'.) So *it* can then be supplied as argument to another function, or to the same function again, and so on indefinitely. In this way we can now form very complex names indeed. And by starting with a variable in place of a name-letter we can also form what might be called 'complex variables', though that is not the usual terminology.

For example, consider the arithmetical expression

$$3x^2 + 2x + 1.$$

Suppose that we use

$f(x,y)$ for $x+y$
$g(x,y)$ for $x \cdot y$.

Then this expression can be analysed as

$$f(f(g(3,g(x,x)),g(2,x)),1).$$

If you put in a particular numeral in place of the variable x, then the whole expression becomes a complex name of some number. But if you leave x as a variable, then you have an expression that behaves just like a complex name, except that it contains a free variable within it. So you obtain an open sentence by putting this expression into the gap of a one-place predicate, for example, the predicate '$\ldots = 0$'. Then you can form a closed sentence by adding a quantifier to bind the free variable, as in

$$\exists x[f(f(g(3,g(x,x)),g(2,x)),1) = 0].$$

Returning to the familiar notation, this just says

$$\exists x[3x^2 + 2x + 1 = 0].$$

Evidently much of school mathematics is concerned with procedures for discovering the truth-value of sentences such as these.

Of course, in the particular case nothing is gained by using the new letters $f,g,\ldots$ to re-express what is already expressed perfectly well by the familiar notation of mathematics. The example does illustrate how in mathematics one does use functional expressions in quite complex ways, but the main purpose of the letters $f,g,\ldots$ is not to 'abbreviate' particular functional expressions, but to act as schematic letters standing in for *any* such expressions. They allow us to frame general logical laws that apply to all functions without exception. To see how they do this we must see how such schematic letters can be added to the logical framework that we have already.

First we add function-letters to the vocabulary of our language. Using

these letters, we now add to the formation rules (p. 78) a recursive characterization of what is to count as a *term*, which goes like this:

(1) A name-letter is a term.
(2) A variable is a term.
(3) If θ^n is an n-place function-letter ($n>0$), and if $\tau_1,...,\tau_n$ is a series of n terms (not necessarily distinct), then $\theta^n(\tau_1,...,\tau_n)$ is a term.
(4) There are no other terms.

The other formation rules remain as before, except that when complex terms become available it increases clarity if the predicate-letters are supplied with brackets and commas, as the function-letters are. This means that the clause for atomic formulae should now be rephrased in this way:

> If Φ^n is an n-place predicate-letter (or is the two-place identity predicate I^2), and if $\tau_1,...,\tau_n$ is a series of n terms, not necessarily distinct, then $\Phi^n(\tau_1,...,\tau_n)$ is a formula.

(Of course, we can for simplicity omit these extra brackets and commas when there is no need for them.) It is to be observed that formulae given by this rule are still called *atomic* formulae, since they do not have any proper subformulae, but nevertheless they may now be very complicated, if the terms that they contain are complicated.

The intended interpretation for a function-letter is, of course, that it be interpreted as a function defined on the domain $\mathcal{D}$ of the interpretation, i.e. a function yielding an object in that domain as value for each object, or series of objects, from the domain as argument(s). Where θ^n is a function-letter, we use $|\theta^n|_I$ for the value that the interpretation I assigns to θ^n. Then what we have just said is that $|\theta^n|_I$ is to be a function from $\mathcal{D}_I^n$ into $\mathcal{D}_I$. The relevant clause for evaluating expressions containing function-letters is simply

$$|\theta^n(\tau_1,...,\tau_n)|_I = |\theta^n|_I(|\tau_1|_I,...,|\tau_n|_I).$$

This merely spells out the original intention in an obvious way.

It turns out, then, that the result of admitting function letters is that the language becomes very much more complicated, but the rules of inference are scarcely affected. There is one small liberalization, but that is all. However, the reason why we have been able to keep these rules so simple is that we have been relying on the assumption noted at the beginning of this section, namely the assumption that every function is always defined for every argument. So far as concerns our ordinary and everyday use of functional expressions, this assumption is wholly unrealistic. For example, 'the father of . . .' is very naturally viewed as a functional expression, but we certainly do not suppose that absolutely everything has a father. Rather, we say that this function is defined only for certain kinds of argument (e.g. persons) but not for others (e.g. stones). The case is the same in mathematics too, where functions have a very important role to play, as we have seen. Naturally, in arithmetic we are only concerned with whether such functions as addition, subtraction, multiplication, and so on, are defined *for numbers*; we do not bother about whether they happen to be defined for other things too. This is compatible with the proposed logic, provided that our intended domain of quantification contains only the numbers, as in arithmetic it will do. However, not all arithmetical functions are defined even for all numbers as arguments. As every schoolboy knows, an exception is 'x divided by y', since division by zero is not defined, and all kinds of fallacies result from ignoring this point.

Where we have a function that is not defined for certain arguments, it is always possible to introduce a surrogate function that is defined for all arguments, by stipulating arbitrarily what value the function is to have in the cases hitherto undefined. For example, suppose that $f(x,y)$ abbreviates 'the number which results upon dividing x by y'. Then, as we have said, $f(x,y)$ is not defined for $y=0$, and it is equally not defined if x or y is not a number at all. But we could introduce the surrogate function f' by setting

$$f'(x,y) = \begin{cases} f(x,y), \text{ if } x \text{ and } y \text{ are both numbers, and } y \neq 0 \\ 0 \text{ otherwise.} \end{cases}$$

Then we have

$$f'(4,2) = f'(6,3) = 2.$$

And in addition

$$f'(4,0) = f'(\text{the moon},3) = f'(\text{the moon, the sun}) = 0.$$

But one has no sympathy with such surrogate functions. It is very much more natural to say that there is *no* number which can be obtained upon dividing 4 by 0, and *nothing* which counts as the result of dividing the moon by the sun. And it is surely perverse to suppose that the 'nothing' which comes from the latter division is 'the same thing' as the perfectly good number which results upon dividing 0 by 4.

The assumption that all functions are everywhere defined is, then, something that one would much rather do without. Yet we cannot easily abandon it, for if we do, then the simple rules of inference for function-letters must be abandoned too. This is because the rules of inference for name-letters are based upon the assumption that a name does always stand for something, as we noted way back in Section 3.1. And our present rules of inference treat functional expressions such as $f(a)$ as complex name-symbols, not differing in any important way from a simple name-letter. Consequently, these rules just assume that $f(a)$ does always stand for something, and without this assumption they would not be correct. Now you might say that the initial assumption about names is unrealistic, and we should seriously consider whether we can do without it. I shall take up this question from Section 4 onwards. But you might instead say that the present trouble arises only because functional expressions have been treated as if they were names, and that this is the point that needs revision. I take up this suggestion in the next section.

EXERCISES

(These exercises are exercises in applied logic, using function symbols.)

8.2.1. The theory of groups can be presented as having in its vocabulary just identity and a single two-place function $f(x,y)$ which we write as '$x \cdot y$'. The usual laws for identity apply, and in addition these three axioms:

(A1) $\forall xyz(x\cdot(y\cdot z) = (x\cdot y)\cdot z)$
(A2) $\forall xy\exists z(x = z\cdot y)$
(A3) $\forall xy\exists z(x = y\cdot z)$.

In this theory, prove

(1) $a = a\cdot c \vdash c = c\cdot c$. [Use $\exists z(c = z\cdot a)$.]
(2) $c = c\cdot c, d = d\cdot d \vdash c = d$. [Use $\exists z(c = d\cdot z), \exists w(d = w\cdot c)$.]
(3) $a = a\cdot c \vdash b = b\cdot c$. [Use (1), (2), and $\exists z(b = b\cdot z)$.]
(4) $\vdash 1x\forall y(y = y\cdot x)$. [Recall that '$1x$' means 'there is exactly one x such that'. It is easily shown from (3) that there is at least one, and from (2) that there is at most one.]

Given the result (4) we are evidently entitled to introduce a name for the unique entity x such that $\forall y(y = y\cdot x)$. We shall call it '1'. Thus we have

(4′) $\vdash \forall y(y = y\cdot 1)$.

Continue to prove

(5) $a = c\cdot a \vdash c = c\cdot c$. [Similar to (1).]
(6) $\vdash a\cdot 1 = 1\cdot a$. [Use (2), (4), (5).]
(7) $a\cdot b = 1 \vdash b\cdot a = 1$. [The proof goes via $(b\cdot a)\cdot(b\cdot a) = b\cdot a$, and the result follows from this by (2) and (4′).]
(8) $a\cdot b = 1, a\cdot c = 1 \vdash b = c$. [Use (6) and (7).]
(9) $\vdash \forall x1y(x\cdot y = 1)$. [Use (8) and (A3).].

Given this result, we are evidently entitled to introduce a one-place function $f(x)$, to represent the fact that for each x there is one and only one item $f(x)$ such that $x\cdot f(x) = 1$. We shall write $f(x)$ as x^{-1}, so we have

(9′) $\vdash \forall x(x\cdot x^{-1} = 1)$.

I remark incidentally that the constant 1 is called the identity of the group, and the function x^{-1} the inverse function of the group.

8.2.2. (continuing 8.2.1). Suppose that the axioms for a group are given as

$\forall xyz(x\cdot(y\cdot z) = (x\cdot y)\cdot z)$
$\forall x(x = x\cdot 1)$
$\forall x(x\cdot x^{-1} = 1)$.

(These axioms simply *assume* the existence of the constant 1 and the inverse function x^{-1} just proved.) Prove from these axioms the original axioms (A1)–(A3) of Exercise 8.2.1. [You will need to establish a couple of lemmas on the way, but I leave you to find them.]

8.2.3. Consider a theory which is supposed to axiomatize elementary arithmetic. It has in its vocabulary a constant 0, a one-place function $f(x)$ which we write as x', meaning the number after x, and two two-place predicates, namely = and <. We assume the usual laws for identity and in addition these eight axioms:

(A1) $\forall x(x'\neq 0)$.
(A2) $\forall xy(x'=y' \rightarrow x=y)$.
(A3) $\forall x(x\neq 0 \rightarrow \exists y(x=y'))$.
(A4) $\forall xyz(x<y \wedge y<z \rightarrow x<z)$.
(A5) $\forall xy\neg(x<y \wedge y<x)$.
(A6) $\forall x\neg(x<0)$.
(A7) $\forall xy(x<y \rightarrow x'<y')$.
(A8) $\forall xy(x<y' \leftrightarrow (x=y \vee x<y))$.

(*a*) Prove, informally if you wish, that the axioms imply

(9) $\forall x\exists y(x<y \wedge \forall z(x<z \rightarrow y=z \vee y<z))$.

(*b*) Show by an interpretation that the axioms do *not* imply

(10) $\forall xy\,(x\neq y \rightarrow x<y \vee y<x)$.

[Hint: an interpretation that verifies (A1)–(A8) must contain a set of elements corresponding to the natural numbers, i.e. with a first member (to interpret 0) and for each member a next (to interpret x'), and the relation $<$ must be connected on this set. But consider how to add *further* elements to the interpretation, still satisfying axioms (A1)–(A8), but not connected with the elements that correspond to the natural numbers.[1]]

(*c*) Suppose that we add (10) to the axioms (A1)–(A8) as a further axiom. Show that in that case the axiom (A2) becomes superfluous.

(*d*) Show that, even if (10) is added to the axioms, still there is an interpretation in which all the axioms are true and yet this domain does not have the intended structure of the natural numbers. [Hint: your answer to part (*b*) will also answer this.]

I remark as an aside that *no* set of axioms which we can formulate in elementary logic will constrain an interpretation to have just the structure of the natural numbers. (That is a consequence of the compactness theorem; the discussion on pp. 183–4 may give a suggestion as to how it might be proved.)