5.3 Restricted Quantification: Many-Sorted Predicate Logic

In §3.3 formulas (26) and (27) were given as translations of sentences (25) and (26), respectively:

(24) All teachers are friendly.

(25) Some teachers are friendly.

(26) $\forall x(Tx \rightarrow Fx)$

(27) $\exists x(Tx \wedge Fx)$

We say that the quantifier $\forall x$ is *restricted to* Tx in (26) and that the quantifier $\exists x$ is *restricted to* Tx in (27). More generally: if ϕ is a formula with x as a free variable, then $\forall x$ is said to be restricted to ϕ in $\forall x(\phi \rightarrow \ldots)$ and $\exists x$ is said to be restricted to ϕ in $\exists x(\phi \wedge \ldots)$. The same applies if the whole formula is a subformula of some other formula. If you examine the translation examples we have given so far, you will see that quantifiers are nearly always restricted. Expressions like *everyone* and *someone* are among the few which can be rendered as unrestricted quantifiers, and even then only if the sentence doesn't say anything about any entities other than people, since that is the condition under which we can restrict the domain to people. If it mentions things other than people, then restricted quantifiers are needed—two in formula (29), which is a translation of (28):

(28) Everyone gave Danny something.

(29) $\forall x(Px \rightarrow \exists y(Ty \wedge Gxyd))$

The quantifier $\forall x$ has been restricted to P (for people), and the quantifier $\exists y$ has been restricted to T (for things). So the domain includes both people and things.

It would perhaps be more natural to split the domain into different subdomains, thus distinguishing among people, other living things, and all other things, for example. Typographically different variables could be used, these being interpreted within the different subdomains. So, for example, we could have x, y, and z as variables for the subdomain containing just people; k, l, and m as variables for other living things; and u, v, and w as variables referring to anything else (subscripts being added to any of these variables in case they threaten to run out). It will then also be necessary to say in what subdomains the various constants have their interpretations. In this way, sentence (28) can be translated as (30):

(30) $\forall x \exists u Gxud$

Note that (30) has the unrestricted quantifiers $\forall x$ and $\exists u$ instead of the restricted quantifiers $\forall x(Px \rightarrow$ and $\exists y(Ty \wedge$ of (29). The price to be paid for this simplification is that defining the language and the syntax becomes more complicated.

The logic resulting from the above modifications is called *many-sorted predicate logic*. In defining a language for many-sorted predicate logic, we have to specify what sorts there are, what their respective variables are, and to what sort each of the constants belongs. In formulating the semantics, we must divide all the domains into different subdomains too, one for each sort. It is then not too difficult to give a truth definition, so it is left to the reader.

It is doubtful that much is to be gained by introducing new languages for the above purpose. Any many-sorted language for predicate logic can be turned into a language for standard predicate logic by adding a number of unary predicate letters, one for each sort. The variables can then be interpreted over the whole domain, the predicates taking over the job of referring to the different sorts. Taking (30) as a formula in many-sorted logic, for example, and introducing P and T to refer to the sorts corresponding to x and u, respectively, (29) can be recovered (or at least some variant which, in spite of having different variables from those of (29), has the same interpretation). Similarly, any model for many-sorted predicate logic can easily be turned into a model for standard predicate logic.

But many-sorted predicate logic has some advantages. Consider sentences (31) and (32), for instance.

(31) Mont Blanc gave Danny something.

(32) Everyone gave Mont Blanc something.

Translating these into standard predicate logic we obtain the following two sentences:

(33) $\exists y(Ty \wedge Gmyd)$

(34) $\forall x(Px \rightarrow \exists y(Ty \wedge Gxym))$

But in many-sorted logic, we can also choose to block the translation of (31) and (32). We may choose to require that things only be given to or by people, for example, by accepting Ghst as a formula only if h and t are the right kinds of constants or variables—those which refer to people. But this approach raises a great many problems, beginning with giving exact specifications for the sort or sorts of variables which each of the predicates may accept as its arguments. For this reason, it seems not very satisfactory as an approach to such sentences. But it returns in a somewhat more satisfactory form (in vol. 2) in the *logic of types*, where the different sorts (*types*) distinguish between expressions which have wholly different *functions*. Something similar is also to be found in second-order logic (see §5.4). Another way of dealing with some people's uneasiness with (31) and (32) is not to bar them from being translated into formulas but to arrange for the formulas to receive neither *true* nor *false* as a truth value in the semantics. The logical system which then arises, many-

valued logic (see §5.5), is not a variant of standard predicate logic but a true deviation from it. These kinds of problems have incidentally also been the subject of lively debate in linguistics, centering on examples like the well-known (35):

(35) Colorless green ideas sleep furiously.

Sentence (35) violates *selection restrictions*. If one thinks selection restrictions are a syntactic matter, one takes the first alternative outlined above, deeming sentences like (31), (32), and (35) syntactically ill-formed. If one thinks they belong to semantics, one will take the second. There are also those who think that an explanation for what is unsatisfactory about sentences like (31), (32), and (35) is to be sought outside of grammar altogether.

A many-sorted predicate logic offers only a minimal solution to what some people feel is unnatural about the way quantifiers are handled in translating (25) as (26), (25) as (27), and (28) as (29). What about sentences like (36)?

(36) All wealthy people gave Danny something.

This translates into standard logic as (37), or, equivalently, as (38):

(37) $\forall x((Px \wedge Wx) \rightarrow \exists y(Ty \wedge Gxyd))$

(38) $\forall x(Px \rightarrow (Wx \rightarrow \exists y(Ty \wedge Gxyd)))$

The quantifier $\forall x$ is restricted to $Px \wedge Wx$ in (37), and it is restricted twice in (38), first to Px and then to Wx, which amounts to the same thing. So in order to 'cover up' these restrictions in a many-sorted logic we would have to introduce some more sorts, for Wx and $Px \wedge Wx$. Besides not being very elegant, distinguishing between people and wealthy people by introducing special sorts for them is a bad precedent. It is not clear where the division into increasingly more special sorts should stop. Perhaps whatever is considered unnatural about translations like (37) and (38) is more easily removed if it is remembered that what matters about translations is not the formulas themselves but their meanings. Translation is actually indirect interpretation, in which formulas function as intermediates between sentences and their meanings. Formulas are *notations* for meanings. And notations are neither natural nor unnatural, they are just more or less useful. In this case it might be better to introduce a notation by writing $\forall x^{\phi}(\psi)$ instead of $\forall x(\phi \rightarrow \psi)$, and $\exists x^{\phi}(\psi)$ instead of $\exists x(\phi \wedge \psi)$. Then (26), (27), (29), (33), and (37) can be rewritten as (39), (40), (41), (42), and (43), respectively:

(39) $\forall x^{Tx}(Fx)$

(40) $\exists x^{Tx}(Fx)$

(41) $\forall x^{Px}\exists y^{Ty}(Gxyd)$

(42) $\exists y^{Ty}(Gmyd)$

(43) $\forall x^{Px \wedge Wx} \exists y^{Ty}(Gxyd)$

A formula like (38) retains a restricted quantifier in this notation, as is apparent from (44):

(44) $\forall x^{Px}(Wx \rightarrow \exists y^{Ty}(Gxyd))$

One solution for this would be to shorten such formulas even more. One could, for example, shorten (44) to (45):

(45) $\forall x^{Px, Wx} \exists y^{Ty}(Gxyd)$

In formulating such shorthands, one must make sure that the original formula can always be recovered from its abbreviation. For this reason, (37) is abbreviated to (43) and (38) is abbreviated to (45). With more complicated sentences such as (46), it is questionable which of the two is more readable, the standard translation (47) or its abbreviation, (48):

(46) He who has a dog that bites someone, is sad.

(47) $\forall x(Px \rightarrow (\exists y(Dy \wedge Hxy \wedge \exists z(Pz \wedge Byz)) \rightarrow Sx))$

(48) $\forall x^{Px, \exists y^{Dy \wedge Hxy}(\exists z^{Pz}(Byz))}(Sx)$

By way of conclusion, a word on inference relations. Since its models hardly differ from the standard ones, many-sorted logic is not very new as far as its semantics is concerned. As for syntax, the system of natural deduction can easily be modified for our purposes by introducing separate introduction and elimination rules for the quantifiers of each sort. The soundness and completeness theorems for standard predicate logic are then inherited by many-sorted logic.