j. John admires himself.
k. If John gambles, then he will hurt himself.
l. Although John and Mary love each other deeply, they make each other very unhappy.

3.2 Quantifying Expressions: Quantifiers

Besides connectives, predicate logic also deals with quantifying expressions. Consider a sentence like:

(21) All teachers are friendly.

Aristotle saw a sentence like this as a relationship between two predicates: in this case between *being a teacher* and *being friendly*. He distinguished four different ways of linking two predicates *A* and *B*. Besides *all A are B*, of which the above is an instance, he had *some A are B*, *all A are not-B*, and *some A are not-B*.

If you just consider properties, then this works quite nicely. But as soon as you move from predicates to relations, and from simple quantification to sentences in which more than one quantifying expression appears, things become more difficult. It would not be easy to say what kind of relationship is expressed by sentence (22) between the relation *admires* and the people being talked about:

(22) Everyone admires someone.

And even if we could manage this sentence somehow, there are always even more complex ones, like (23) and (24):

(23) Everyone admires someone who admires everyone.

(24) No one admires anyone who admires everyone who admires someone.

It would seem that we are in need of a general principle with which the role of quantifying expressions can be analyzed.

Let us first examine sentences in which just a single predicate appears.

(25) Peter is friendly.

(26) No one is friendly.

We translate (25) as Vp: the entity which we refer to as p is said to possess the property which we refer to as V. Now it would not be correct to treat (26) the same way, using a constant *n* for the x in Vx. There simply isn't anyone called *no one* of whom we could say, truthfully or untruthfully, that he is friendly. Expressions whose semantic functions are as different as *Peter* and *no one* cannot be dealt with in the same way. It happens that the syntactic characteristics of *Peter* and *no one* are not entirely the same in natural language either. Compare, for example, the phrases *none of you* and *Peter of you*, or *no one except John* and *Peter except John*.

In (25) it is said of Peter that he has a particular property. We could also turn things around and say that the predicate *friendly* is said to have the property of applying to Peter. This is not the way things are done in predicate logic, but there are richer logical systems which work this way, which can be an advantage in the logical analysis of natural language (see vol. 2). It seems more natural to turn things around in dealing with (26), since there is no one to whom the property of being friendly is attributed, and it is thus better to say that this sentence states something about the property *friendly*, namely, that it applies to none of the entities to which it might in principle apply. Likewise, in a sentence such as

(27) Someone is friendly.

we also have a statement about the property *friendly*, namely, that there is at least one among the entities to which it might in principle apply to which it does in fact apply. Instead of having to say *the entities to which the predicates might in principle apply*, we can make things easier for ourselves by collectively calling these entities the *universe of discourse*. This contains all the things which we are talking about at some given point in time. The sentence

(28) Everyone is friendly.

can with this terminology be paraphrased as: every entity in the domain of discourse has the property *friendly*. The domain is in this case all human beings, or some smaller group of human beings which is fixed in the context in which the sentence appears. Note that the choice of domain can affect the truth values of sentences. It is highly probable that sentence (28) is untrue if we include every single human being in our domain of discourse, but there are certainly smaller groups of human beings for whom (28) is true.

We shall introduce two new symbols into the formal languages, the *universal quantifier* $\forall$ and the *existential quantifier* $\exists$. Each quantifier always appears together with a variable. This combination of a quantifier plus a variable (for example, $\forall x$ or $\exists y$) is conveniently also referred to as a quantifier (universal or existential). $\forall x \ldots$ means: for every entity x in the domain we have $\ldots$; and $\exists x \ldots$ means: there is at least one entity in the domain such that $\ldots$; $\forall x\phi$ is called the *universal generalization* of ϕ, and $\exists x\phi$ is its *existential generalization*.

We are now in a position to translate (28) as $\forall x Vx$ (or equivalently, as $\forall y Vy$ or as $\forall z Vz$, since variables have no meaning of their own), to translate (27) as $\exists x Vx$ (or as $\exists y Vy$ or $\exists z Vz$), (26) as $\neg\exists x Vx$, and *everyone is unfriendly* as $\forall x \neg Vx$.

It turns out that under this interpretation *no one is friendly* and *everyone is unfriendly* have the same meaning, since $\neg\exists x Vx$ and $\forall x \neg Vx$ are equivalent sentences in predicate logic. Later we shall find this analysis of *everyone* and *someone* a bit simplistic, but it will do for the cases we have discussed.

We will now build up the translation of (22), an example of a sentence which contains two quantifying expressions, in several steps. We use the key

(29) Axy: x admires y.

We replace the x in *x admires y* by *Plato* and thus obtain a *propositional function*: see 74, 78

(30) Plato admires y.

This would be translated as Apy and expresses the property of *being admired by Plato*. If we wish to say that someone has this property, this can be done by translating

(31) Plato admires someone.

as ∃yApy. Replacing *Plato* by *x* in (31), we obtain the propositional function

(32) x admires someone.

This again expresses a property, namely, that of *admiring someone*, and would be translated as ∃yAxy. Finally, by universally quantifying this formula we obtain the formula ∀x∃yAxy, which says that everyone in the domain has the property expressed by (32). So ∀x∃yAxy will serve as a translation of (22); (23) and (24) are best left until we have dealt with the notion of *formulas of predicate logic*.

We shall first discuss how the four forms which Aristotle distinguished can be represented by means of quantifiers. The following can be formed with *teacher* and *friendly* ((33) = (21)):

(33) All teachers are friendly.

(34) Some teachers are friendly.

(35) All teachers are unfriendly.

(36) Some teachers are unfriendly.

The material implication, as the reader may already suspect from what was said when it was first introduced, is rather useful in translating (33). For if (33) is true, then whatever Peter does for a living, we can be quite sure that (37) is true.

(37) If Peter is a teacher, then Peter is friendly.

In (37), the *if . . . then* is understood to be the material implication. This can be seen very simply. If he happens to be a teacher, then, assuming (33) to be true, he must also be friendly, so (37) is true. And if he does not happen to be a teacher, then according to the truth table, (37) must be true too, whether he is friendly or not.

If, on the other hand, (33) is not true, then there must be at least one unfriendly teacher, say John, and then (38) is untrue.

(38) If John is a teacher, then John is friendly.

It should now be clear that (33) is true just in case it is true that for every person x, if x is a teacher, then x is friendly. This means that we now have the following translation for (33):

(39) ∀x(Tx → Fx)

The reader should be warned at this stage that (39) would also be true if there were no teachers at all. This does not agree with what Aristotle had to say on the matter, since he was of the opinion that *all A are B* implies that there are at least some As. He allowed only nonempty 'terms' in his syllogisms.

Sentence (34) would be translated into predicate logic as (40):

(40) ∃x(Tx ∧ Fx)

Translation (40) is true if and only if there is at least one person in the domain who is a teacher and who is friendly. Some nuances seem to be lost in translating (34) like this; (34) seems to say that there are more friendly teachers than just one, whereas a single friendly teacher is all that is needed for (40) to be true. Also, as a result of the communtativity of ∧, (40) means the same as (41), which is the translation of (42): ?

(41) ∃x(Tx ∧ Fx)

(42) Some friendly people are teachers.

It could be argued that it is unrealistic to ignore the asymmetry which is present in natural language. But for our purposes, this translation of (34) will do. In §3.7 we will see that it is quite possible to express the fact that there are several friendly teachers by introducing the relation of identity. Sentences (35) and (36) are now no problem; (36) can be rendered as (43), while (35) becomes (44).

(43) ∃x(Tx ∧ ¬Fx)

(44) ∀x(Tx → ¬Fx)

Sentences (45) and (46) mean the same as (35), and both can be translated as (47):

(45) No teachers are friendly.

(46) It is not the case that some teachers are friendly.

(47) ¬∃x(Tx ∧ Fx)

Indeed, the precise formulation of the semantics of predicate logic is such that (44) and (47) are equivalent. The definitions of the quantifiers are such that ∀x¬ϕ always means the same as ¬∃xϕ. This is reflected in the fact that (48) and (49) have the same meaning:

(48) Everyone is unfriendly.

(49) No one is friendly.

This means that (47) must be equivalent to $\forall x \neg(Tx \wedge Fx)$. And according to propositional logic, this formula must once again be equivalent to (44), since $\neg(\phi \wedge \psi)$ is equivalent to $\phi \rightarrow \neg\psi$.

Exercise 2

Translate the following sentences into predicate logic. Preserve as much of the structure as possible and give in each case the key and the domain of discourse.

a. Everybody loves Marion.
b. Some politicians are honest.
c. Nobody is a politician and not ambitious.
d. It is not the case that all ambitious people are not honest.
e. All blond authors are clever.
f. Some best-selling authors are blind.
g. Peter is an author who has written some best-selling books.

3.3 Formulas

Certain problems arise in defining the formulas of predicate logic which we didn't have with propositional logic. To begin with, it is desirable that the notions of *sentence* and of *formula* do not coincide. We wish to have two kinds of formulas: those which express propositions, which may be called *sentences*, and those which express properties or relations, which may be called *propositional functions*. So we shall first give a general definition of *formula* and then distinguish the sentences among them.

Another point is that it is not as obvious which expressions are to be accepted as formulas as it was in the case of propositional logic. If A and B are unary predicate letters, then $\forall xAx$, $\forall y(Ay \rightarrow By)$, and $Ax \wedge By$ are clearly the sorts of expressions which we wish to have among the formulas. But what about $\forall xAy$ and $\forall x(Ax \wedge \exists xBx)$? One decisive factor in choosing a definition is simplicity. A simple definition makes it easier to think about formulas in general and facilitates general statements about them. If ϕ is a formula, we simply choose to accept $\forall x\phi$ and $\exists x\phi$ as formulas too. We shall see that the eventuality that the variable x does not even occur in ϕ need not cause any complications in the interpretation of $\forall x\phi$ and $\exists x\phi$: $\forall xAy$ is given the same interpretation as Ay, and the same applies to $\exists xAy$. In much the same way, $\forall x(Ax \wedge \exists xBx)$ receives the same interpretation as $\forall x(Ax \wedge \exists yBy)$. We shall see that all formulas which may be recognized as such admit of interpretation. This is primarily of theoretical importance. When translating formulas from natural language into predicate logic, we shall of course strive to keep the formulas as easily readable as possible.

Each language L of predicate logic has its own stock of constants and predicate letters. Each of the predicate letters has its own fixed arity. Besides these,

there are also the symbols which all languages of predicate logic have in common: the connectives, the quantifiers $\forall$ and $\exists$, and as auxiliary signs, the brackets and an infinite supply of variables. Any given formula will, of course, contain only a finite number of the latter, but we do not wish to place an upper limit on the length of formulas, and we therefore can't have any finite upper limit to the number of variables either. Together these symbols form the *vocabulary* of L. Given this vocabulary, we define the formulas of language L as follows (compare definition 1 in §2.3):

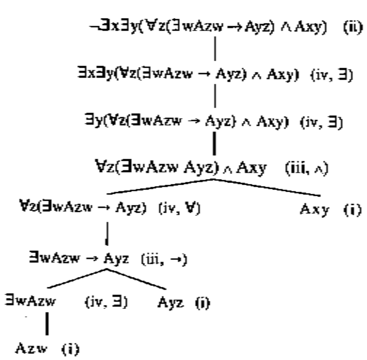
Definition 1

(i) If A is an *n*-ary predicate letter in the vocabulary of L, and each of $t_1, \ldots, t_n$ is a constant or a variable in the vocabulary of L, then $At_1, \ldots, t_n$ is a formula in L.
(ii) If ϕ is a formula in L, then $\neg\phi$ is too.
(iii) If ϕ and ψ are formulas in L, then so are $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, and $(\phi \leftrightarrow \psi)$.
(iv) If ϕ is a formula in L and x is a variable, then $\forall x\phi$ and $\exists x\phi$ are formulas in L.
(v) Only that which can be generated by the clauses (i)–(iv) in a finite number of steps is a formula in L.

Clause (i) yields the *atomic formulas*. These are formulas like Bxyz, Mp, and Apx. Formulas formed according to (iv) are called *universal* and *existential* formulas, respectively.

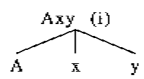
Just as in propositional logic, we leave off the outer brackets of formulas and just talk about predicate-logical formulas where it doesn't matter what language L we are dealing with. Here too there is a characteristic construction tree associated with each formula. Formula (51), for instance, has the construction tree represented in figure (50):

(50)



(51) $\neg\exists x\exists y(\forall z(\exists wAzw \to Ayz) \land Axy)$

This tree could be added to in order to show how the atomic formulas appearing in it have been built up from predicate letters, variables, and constants, as in figure (52):

(52)


But for our purposes these details are unnecessary. Just as in propositional logic, the subformulas of a formula are those formulas which appear in its construction tree. Formula (51) has, for example, itself, $\exists x\exists y(\forall z(\exists wAzw \to Ayz) \land Axy)$, $\exists y(\forall z(\exists wAzw \to Ayz) \land Axy)$, $\forall z(\exists wAzw \to Ayz) \land Axy$, $\forall z(\exists wAzw \to Ayz)$, Axy, $\exists wAzw \to Ayz$, $\exists wAzw$, Azw, and Ayz as its subformulas. And just as in propositional logic, it can be shown that the subformulas of a formula ϕ are just those strings of consecutive symbols taken from ϕ which are themselves formulas.

In order to decide which formulas are to be called sentences, but also in order to be able to interpret formulas in the first place, it is essential to be able to say how much of a given formula is governed by any quantifier appearing in it. We shall deal with this in the next few definitions.

Definition 2

If $\forall x\psi$ is a subformula of ϕ, then ψ is called the *scope* of this particular occurrence of the quantifier $\forall x$ in ϕ. The same applies to occurrences of the quantifier $\exists x$.

As a first example, the scopes of the quantifiers occurring in (51) have been summarized in (53):

(53)

Quantifier	*Scope*
$\exists w$	Azw
$\forall z$	$\exists wAzw \to Ayz$
$\exists y$	$\forall z(\exists wAzw \to Ayz) \land Axy$
$\exists x$	$\exists y(\forall z(\exists wAzw \to Ayz) \land Axy)$

We distinguish between different *occurrences* of a quantifier in definition 2 because there are formulas like (54):

(54) $\forall xAx \land \forall xBx$

In (54), one and the same quantifier appears more than once. The first occurrence of $\forall x$ in (54) has Ax as its scope, while the second occurrence has Bx as its scope. What this means is that the first occurrence of $\forall x$ only governs the x in Ax, while the second occurrence governs the x in Bx. We shall now incorporate this distinction into the following general definition:

Definition 3

(a) An occurrence of a variable x in the formula ϕ (which is not part of a quantifier) is said to be *free in* ϕ if this occurrence of x does not fall within the scope of a quantifier $\forall x$ or a quantifier $\exists x$ appearing in ϕ.
(b) If $\forall x\psi$ (or $\exists x\psi$) is a subformula of ϕ and x is free in ψ, then this occurrence of x is said to be *bound* by the quantifier $\forall x$ (or $\exists x$).

It will be clear that either an occurrence of a variable x in a formula is free or it is bound by a quantifier $\forall x$ or $\exists x$.

Definition 3 is a little more complicated than may seem necessary, and this is because we allow formulas such as $\forall x(Ax \land \exists xBx)$. In this formula, the x in Bx is bound by the $\exists x$, while the x in Ax is bound by the $\forall x$. According to definition 2, the x in Bx also occurs within the scope of the $\forall x$. But this occurrence of x is not bound by the $\forall x$, because it is not free in $Ax \land \exists xBx$, the scope of $\forall x$, which is what clause (b) of definition 3 requires. In practice we will tend to avoid situations in which bound variables occur within the scope of quantifiers with the same variable, but definition 1 does not exclude them. The funny thing about the other strange formula we have mentioned, $\forall xAy$, is that the quantifier $\forall x$ does not bind any variables at all. These kinds of formulas we shall tend to avoid as well, but definition 1 does not exclude them either.

Now we can define what we mean by *sentence* in predicate logic:

Definition 4

A *sentence* is a formula in L which lacks free variables.

$\forall xAy$ is not a sentence, for example, because the occurrence of the variable y is free; $\forall x(Ax \land \exists xBx)$ is a sentence, but $Ax \land \exists xBx$ is not, since the first occurrence of x is free.

Exercise 3

For each of the following formulas of the predicate calculus, indicate:
(a) whether it is a negation, a conjunction, a disjunction, an implication, a universal formula, or an existential formula;
(b) the scope of the quantifiers;
(c) the free variables;
(d) whether it is a sentence.

(i) $\exists x(Axy \land Bx)$
(ii) $\exists xAxy \land Bx$
(iii) $\exists x\exists yAxy \to Bx$
(iv) $\exists x(\exists yAxy \to Bx)$
(v) $\neg\exists x\exists yAxy \to Bx$
(vi) $\forall x\neg\exists yAxy$
(vii) $\neg Bx \to (\neg\forall y(\neg Axy \lor Bx) \to Cy)$
(viii) $\exists x(Axy \lor By)$
(ix) $\exists xAxx \lor \exists yBy$
(x) $\exists x(\exists yAxy \lor By)$
(xi) $\forall x\forall y((Axy \land By) \to \exists wCxw)$
(xii) $\forall x(\forall yAyx \to By)$
(xiii) $\forall x\forall yAyy \to Bx$

As we have mentioned, a formula with free variables is called a ***propositional function***. If we take the formula $Tx \rightarrow Fx$ with its one free variable x and replace x with the constant j, then we obtain a sentence, namely, $Tj \rightarrow Fj$. So $Tx \rightarrow Fx$ can indeed be seen as a function: it has as its domain the constants of the language L which we are working in, and the sentences in L as its range. If c is a constant, then the value of the propositional function $Tx \rightarrow Fx$ with c as its argument is the sentence $Tc \rightarrow Fc$. Analogously, the function corresponding to a formula with two free variables is binary. For example, formula (55), the translation of *y admires all those whom x admires*, has sentence (56) as its value when fed the arguments p and j:

(55) $\forall z(Axz \rightarrow Ayz)$

(56) $\forall z(Apz \rightarrow Ajz)$

This is the translation of *John admires all those whom Peter admires*. The following notation is often useful in this connection. If ϕ is a formula, c is a constant, and x is a variable, then $[c/x]\phi$ is the formula which results when all free occurrences of x in ϕ are replaced with occurrences of c. The examples given in table (57) should make this clear. The formulas $[y/x]\phi$ and $[x/c]\phi$ can be defined in exactly the same way.

(57)	ϕ	$[c/x]\phi$
	Axy	Acy
	Axx	Acc
	$\forall xAxx$	$\forall xAxx$
	Ay	Ay
	Acx	Acc
	$Axx \wedge \exists xBx$	$Acc \wedge \exists xBx$
	$\forall xBy$	$\forall xBy$
	$\exists x\exists yAxy \rightarrow Bx$	$\exists x\exists yAxy \rightarrow Bc$
	$\forall x\forall yAyy \rightarrow Bx$	$\forall x\forall yAyy \rightarrow Bc$

Exercise 4

The *quantifier depth* of a predicate-logical formula is the maximal length of a 'nest' of quantifiers $Q_1x(\ldots(Q_2y(\ldots(Q_3z(\ldots$ occurring in it. E.g., both $\exists x\forall yRxy$ and $\exists x(\forall yRxy \wedge \exists zSxz)$ have quantifier depth 2. Give a precise definition of this notion using the inductive definition of formulas.

3.4 Some more quantifying expressions and their translations

Besides the expressions *everyone, someone, all, some, no one*, and *no* which we have discussed, there are a few other quantifying expressions which it is relatively simple to translate into predicate logic. To begin with, *every* and *each* can be treated as *all*, while *a few* and *one or more* and *a number of* can

be treated as *some*. In addition, translations can also be given for *everything, something*, and *nothing*. Here are a few examples:

(58) Everything is subject to decay.
Translation: $\forall xVx$.
Key: Vx: x is subject to decay.
Domain: everything on earth.

(59) John gave something to Peter.
Translation: $\exists x(Tx \wedge Gjxp)$.
Key: Tx: x is a thing; Gxyz: x gave y to z.
Domain: people and things.

The translation of (59) is perhaps a bit more complicated than seems necessary; with a domain containing both people and things, however, $\exists xGjxp$ would translate back into English as: *John gave Peter someone or something*. We say that the quantifier $\exists x$ is *restricted to* T in $\exists x(Tx \wedge Gjxp)$. Suppose we wish to translate a sentence like

(60) Everyone gave Peter something.

Then these problems are even more pressing. This cannot as it is be translated as $\forall y\exists x(Tx \wedge Gyxp)$, since this would mean: *everyone and everything gave Peter one or more things*. The quantifier $\forall y$ will have to be restricted too, in this case to P (key: Px: x is a person). We then obtain:

(61) $\forall y(Py \rightarrow \exists x(Tx \wedge Gyxp))$

When restricted to A, a quantifier $\exists x$ becomes $\exists x(Ax \wedge$; and a quantifier $\forall x$ becomes $\forall x(Ax \rightarrow$. The reasons for this were explained in the discussion of *all* and *some*. Sentence (61) also serves as a translation of:

(62) All people gave Peter one or more things.

Here is an example with *nothing:*

(63) John gave Peter nothing.

Sentence (63) can be seen as the negation of (59) and can thus be translated as $\neg\exists x(Tx \wedge Gjxp)$.

The existential quantifier is especially well suited as a translation of *a(n)* in English.

(64) John gave Peter a book.

Sentence (64), for example, can be translated as $\exists x(Bx \wedge Gjxp)$; Bx: x is a book, being added to the key. This shows that $\exists x(Tx \wedge Gjxp)$ can also function as a translation of

(65) John gave Peter a thing.

This means that the sentence *John gave Peter a book* is true just in case *John gave Peter one or more books* is. In *John gave Peter a book,* there is a strong suggestion that exactly one book changed hands, but the corresponding suggestion is entirely absent in sentences (66) and (67), for example.

(66) Do you have a pen?

(67) He has a friend who can manage that.

We conclude that semantically speaking, the existential quantifier is a suitable translation for the indefinite article. Note that there is a usage in which *a(n)* means something entirely different:

(68) A whale is a mammal.

Sentence (68) means the same as *Every whale is a mammal* and must therefore be translated as $\forall x(Wx \rightarrow Mx)$, with Wx: x is a whale, Mx: x is a mammal as the key and all living creatures as the domain. This is called the generic usage of the indefinite article *a(n)*.

Not all quantifying expressions can be translated into predicate logic. Quantifying expressions like *many* and *most* are cases in point. Subordinate clauses with *who* and *that,* on the other hand, often can. Here are some examples with *who.*

(69) He who is late is to be punished.
Translation: $\forall x(Lx \rightarrow Px)$
Key: Lx: x is late; Px: x is to be punished.
Domain: People

(70) Boys who are late are to be punished.
Translation: $\forall x((Bx \wedge Lx) \rightarrow Px)$, or, given the equivalence of $(\phi \wedge \psi) \rightarrow \chi$ and $\phi \rightarrow (\psi \rightarrow \chi)$ (see exercise 5o in §2.5), $\forall x(Bx \rightarrow (Lx \rightarrow Px))$. Bx: x is a boy must be added to the key to the translation.

The *who* in (69) can without changing the meaning be replaced by *someone who,* as can be seen by comparing (69) and (71):

(71) Someone who is late is to be punished.

This must, of course, not be confused with

(72) Someone, who is late, is to be punished.

Sentences (71) and (69) are synonymous; (71) and (72) are not. In (71), with the restrictive clause *who is late,* the *someone* must be translated as a universal quantifier; whereas in (72), with its appositive relative clause, it must be translated as an existential quantifier, as is more usual. Sentence (71) is thus translated as $\forall x(Lx \rightarrow Px)$, while (72) becomes $\exists x(Lx \wedge Px)$.

Combining personal and reflexive pronouns with quantifying expressions opens some interesting possibilities, of which the following is an example:

(73) Everyone admires himself.

Sentence (73) can be translated as $\forall xAxx$ if the domain contains only humans, while $\forall x(Hx \rightarrow Axx)$ is the translation for any mixed domain.

(74) John has a cat which he spoils.
Translation: $\exists x(Hjx \wedge Cx \wedge Sjx)$.
Key: Hxy: x has y; Cx: x is a cat; Sxy: x spoils y.
Domain: humans and animals.

(75) Everyone who visits New York likes it.
Translation: $\forall x((Hx \wedge Vxn) \rightarrow Lxn)$.
Key: Hx: x is human; Vxy: x visits y; Lxy: x likes y.
Domain: humans and cities.

(76) He who wants something badly enough will get it.

Sentence (76) is complicated by the fact that *it* refers back to *something.* Simply rendering *something* as an existential quantifier results in the following incorrect translation:

(77) $\forall x((Px \wedge \exists y(Ty \wedge Wxy)) \rightarrow Gxy)$
Key: Px: x is a person; Tx: x is a thing; Wxy: x wants y badly enough; Gxy: x will get y.
Domain: people and things.

This translation will not do, since Gxy does not fall within the scope of $\exists y$, so the y in Gxy is free. Changing this to (78) will not help at all:

(78) $\forall x(Px \wedge \exists y(Ty \wedge (Wxy \rightarrow Gxy)))$

This is because what (78) says is that for every person, there is something with a given property, which (76) does not say at all. The solution is to change (76) into

(79) For all persons x and things y, if x wants y badly enough then x will get y.

This can then be translated into predicate logic as

(80) $\forall x(Px \rightarrow \forall y(Ty \rightarrow (Wxy \rightarrow Gxy)))$

Sentences (81) and (82) are two other translations which are equivalent to (80):

(81) $\forall x \forall y((Px \wedge Ty \wedge Wxy) \rightarrow Gxy)$

(82) $\forall y(Ty \rightarrow \forall x(Px \rightarrow (Wxy \rightarrow Gxy)))$

Actually, officially we do not know yet what *equivalence* means in predicate logic; we come to that in §3.6.4. So strictly speaking, we are not yet entitled to leave off the brackets and write $(Px \wedge Ty \wedge Wxy)$ as we did in (81). We will come to this as well. By way of conclusion, we now return to (83) and (84) (=(23) and (24)):

(83) Everyone admires someone who admires everyone.

(84) No one admires anyone who admires everyone who admires someone.

The most natural reading of (83) is as (85):

(85) Everyone admires at least one person who admires everyone.

The translation of (85) is put together in the following 'modular' way:

y admires everyone: $\forall zAyz$;
x admires y, and y admires everyone: $Axy \wedge \forall zAyz$;
there is at least one y whom x admires, and y admires everyone: $\exists y(Axy \wedge \forall zAyz)$.
for each x there is at least one y whom x admires, and y admires everyone: $\forall x\exists y(Axy \wedge \forall zAyz)$.

As a first step toward rendering the most natural reading of (84), we translate the phrase *y admires everyone who admires someone* as $\forall z(\exists wAzw \rightarrow Ayz)$. We then observe that (84) amounts to denying the existence of x and y such that both *x admires y* and *y admires everyone who admires someone* hold. Thus, one suitable translation is given by formula $\neg\exists x\exists y(\forall z(\exists wAzw \wedge Ayz) \wedge Axy)$, which we met before as formula (51), and whose construction tree was studied in figure (50).

Perhaps it is unnecessary to point out that these translations do not pretend to do justice to the grammatical forms of sentences. The question of the relation between grammatical and logical forms will be discussed at length in volume 2.

Exercise 5

Translate the following sentences into predicate logic. Retain as much structure as possible and in each case give the key and the domain.

(i) Everything is bitter or sweet.
(ii) Either everything is bitter or everything is sweet.
(iii) A whale is a mammal.
(iv) Theodore is a whale.
(v) Mary Ann has a new bicycle.
(vi) This man owns a big car.
(vii) Everybody loves somebody.
(viii) There is somebody who is loved by everyone.

(ix) Elsie did not get anything from Charles.
(x) Lynn gets some present from John, but she doesn't get anything from Peter.
(xi) Somebody stole or borrowed Mary's new bike.
(xii) You have eaten all my cookies.
(xiii) Nobody is loved by no one.
(xiv) If all logicians are smart, then Alfred is smart too.
(xv) Some men and women are not mature.
(xvi) Barking dogs don't bite.
(xvii) If John owns a dog, he has never shown it to anyone.
(xviii) Harry has a beautiful wife, but she hates him.
(xix) Nobody lives in Urk who wasn't born there.
(xx) John borrowed a book from Peter but hasn't given it back to him.
(xxi) Some people are nice to their bosses even though they are offended by them.
(xxii) Someone who promises something to somebody should do it.
(xxiii) People who live in Amherst or close by own a car.
(xxiv) If you see anyone, you should give no letter to her.
(xxv) If Pedro owns donkeys, he beats them.
(xxvi) Someone who owns no car does own a motorbike.
(xxvii) If someone who cannot make a move has lost, then I have lost.
(xxviii) Someone has borrowed a motorbike and is riding it.
(xxix) Someone has borrowed a motorbike from somebody and didn't return it to her.
(xxx) If someone is noisy, everybody is annoyed.
(xxxi) If someone is noisy, everybody is annoyed at him.

Exercise 6 ◇

In natural language there seem to be linguistic restrictions on how deeply inside subordinate expressions a quantifier can bind. Let us call a formula *shallow* if no quantifier in it binds free variables occurring within the scope of more than one intervening quantifier. For instance, $\exists xPx$, $\exists x\forall yRxy$ are shallow, whereas $\exists x\forall y\exists zRxyz$ is not. Which of the following formulas are shallow or intuitively equivalent to one which is shallow?

(i) $\exists x(\forall yRxy \rightarrow \forall zSzx)$
(ii) $\exists x\forall y(Rxy \rightarrow \forall zTzxy)$
(iii) $\exists x(\forall y\exists uRuy \rightarrow \forall zSzx)$
(iv) $\exists x\forall y\forall z(Rxy \wedge Sxz)$

3.5 Sets

Although it is strictly speaking not necessary, in §3.6 we shall give a set-theoretical treatment of the semantics of predicate logic. There are two rea-