logic and the syntax of natural language. It also becomes clear that another syntax need not necessarily lead to another semantic interpretation, though of course the details of how the semantics is set up will have to be adjusted because of the direct relationship between the way a formula is constructed and the way its interpretation is constructed. Another advantage, which doesn't really have much to do with subordinating and coordinating connectives, is the following: By not introducing the connectives syncategorematically but as independent expressions with their own semantic interpretations, the way the principle of compositionality works is more clearly displayed.

Exercise 16

How can the binary connective $\wedge$ (conjunction) be treated 'stepwise' in the same way as implication?

3 Predicate Logic

3.1 Atomic Sentences

A language for predicate logic, as before, consists of logical constants, logical variables, and auxiliary symbols. Among the logical constants we have the familiar connectives, and brackets are still to be found among the auxiliary signs, but both categories will be expanded by the introduction of various new symbols. The propositional letters have disappeared, since the idea of predicate logic is to subject simple statements to a deeper analysis. The simple statements we are thinking of are, first of all, individual statements with a clear subject-predicate structure, like:

(1) Plato is a man.

(2) Socrates is mortal.

(3) The chicken is cackling.

(4) This kettle leaks.

Each of the sentences has one part which refers to a property (being a man, being mortal, cackling, and leaking) and another part which refers to some entity (Plato, Socrates, the chicken, and this kettle). Accordingly, in predicate logic we have (*individual*) *constants* which are always interpreted in such a way that they refer to an entity (that is, an individual or an object) and *predicate constants* or *predicate letters* which are always interpreted such that they refer to all kinds of properties which entities (of some particular sort) may or may not have. Note that individual constants and predicate constants are *logical* variables (see §1.3). We shall use lowercase letters for individual constants, for the time being a–v, though later we shall restrict ourselves to the letters a, b, and c. We shall use capital letters for predicate letters, and both will have subscripts where necessary. A well-formed formula corresponding to a sentence can be made by prefixing a predicate letter to a constant. If we have some particular interpretation of the sentences in mind, then we may choose suggestive letters. So (1)–(4) might, for example, be represented as M_1p_1, M_2s, Cc, and Lk, respectively.

Until the end of the nineteenth century, statements with the subject-predicate structure were the only individual statements which were taken seriously. Besides these, however, there are other kinds of individual statements which from a logical point of view cannot profitably be analyzed in terms of subjects and predicates. Sentences which say that two entities bear some particular relationship to each other are a case in point. Here are some examples:

(5) Casper is bigger than John.

(6) Peter is plucking the chicken.

(7) Alcibiades admires Socrates.

There are, of course, instances in which it is useful to distinguish subjects and predicates in such sentences. For example, in linguistics, (5) is often parsed as consisting of a subject, *Casper,* and a predicate, *is bigger than John.* But if one is interested in studying reasoning, then another approach seems preferable, at least for the moment (there are richer logical systems, like higher-order logic with lambda abstraction [see vol. 2], which allow an approach closer to the subject-predicate analysis). For example:

(8) Casper is bigger than John.
John is bigger than Peter.
———
Casper is bigger than Peter.

This is a valid argument, given the meaning of *is bigger than.* However, that cannot be shown if we analyze the premises and conclusion in the subject-predicate schema. For the premises would then contain a different predicate, referring to a different property. The first premise would be translated as Jc, with c translating *Casper,* and J standing for *is bigger than John,* whereas the second would come out as Pj, with j for *John* and P for *is bigger than Peter,* and the conclusion would read Pc. But the argument schema:

$$\begin{array}{l} \mathrm{Jc} \\ \mathrm{Pj} \\ \hline \mathrm{Pc} \end{array}$$

cannot be shown to be valid. What we need is an analysis of (8) which treats the relation *is bigger than* as a logical unit of its own. For it is a general property of the relation *is bigger than* which makes (8) valid: that where the first of three things is bigger than the second, and the second is bigger than the third, the first will always also be bigger than the third. So in order to show the validity of (8), we need to be able to express this general property of *is bigger than,* to treat it as an extra (hidden) premise of the argument in (8) (see §4.1). And we need to be able to express that in the premises and the conclusion of (8), this is the relation that is involved.

For this reason, languages of predicate logic also include symbols which stand for relations between two entities. Sentences (5), (6), and (7) can thus

be translated as B_1cj, Pp_2c, and Aas, respectively; B, P, and A are the translations of *being bigger than, plucking,* and *admiring,* respectively. And (8) can now be turned into the schema:

$$\begin{array}{l} B_1cj \\ B_1jp \\ \hline B_1cp \end{array}$$

This can be shown to be valid, once we add the extra premise mentioned above, which can also be expressed by using the apparatus of predicate logic. We can also use symbols for relations between three entities (like *lies between . . . and*), and so on. All of these symbols are called predicate constants, or predicate letters. Each predicate letter has its own fixed arity: there are unary predicate letters which stand for the properties of entities, there are binary predicate letters which stand for relations between pairs of entities, and so on. In general, n-ary predicates may be introduced for any whole number n larger than zero.

An *atomic sentence* is obtained by writing n (not necessarily different) constants after an n-ary predicate letter. If A is a quaternary predicate letter, for instance, and a, b, c, and d are constants, then Aabcd, Adabc, Addaa, and Abbbc are all atomic sentences. The notation with the predicate letter first is called, as with functions, prefix notation. There are a few relations which are conventionally written in infix notation, one of these being the identity relation, for which we shall introduce the logical constant = shortly. We write a = b and not =ab.

The order of the entities can make a difference for some relations: if Casper is bigger than John, then John is not bigger than Casper. So the order in which the constants are placed after a predicate letter is important: B_1cj and B_1jc express different things. This must not be forgotten when writing keys to translations of natural language sentences: (9), for example, is insufficient as a key.

(9) B_2: lies between
b: Breda, t: Tilburg, e: Eindhoven

This is because it is not clear from (9) whether the formula B_2bte stands for sentence (10) or sentence (11).

(10) Tilburg is between Breda and Eindhoven

(11) Breda is between Tilburg and Eindhoven

So apparently we have to find some way to fix the order of the entities in keys to translations. *Variables* are useful for this purpose. Variables are referred to by x, y, z, and w, and subscripts may be added if we run out of letters. Variables will be seen to have an even more important role to play when we come to the analysis of expressions which quantify. In themselves, variables never have a meaning; they just mark places in sentences. We can use this in giving keys to translations. Instead of (9) we use (12):

(12) B_2xyz: x is between y and z.
b: Breda; t: Tilburg; e: Eindhoven

Unlike (9), (12) leaves no ambiguities in the meanings of sentences which can be formed from these letters; B_2bte is the translation of (11), and B_2tbe is that of (10). The less explicit keys in the above can now be given in the following form:

(13)

Lx	: x leaks	j	: John
M_1x	: x is a man	p_1	: Plato
M_2x	: x is mortal	p_2	: Peter
Cx	: x is cackling	s	: Socrates
B_1xy	: x is bigger than y	c_1	: the chicken
Pxy	: x is plucking y	k	: this kettle
Axy	: x admires y	c_2	: Casper
		a	: Alcibiades

The key (13) gives all the translations we had for the sentences (1)–(7).

We now return to B_2 in order to emphasize that variables do not have any meaning of their own but simply serve as markers.

(14) B_2yxz: y is between x and z.

(15) B_2zxy: z is between x and y.

(16) B_2zyx: z is between y and x.

(17) B_2xyz: y is between x and z.

Key (14) is just the same as (12): (12) and (14) give identical readings to atomic sentences B_2bte, B_2tbe, and so on. And both (15) and (16) give the same results as (12) and (14) too. But key (17) is essentially different, since it gives B_2bte as the translation of (10) and B_2tbe as the translation of (11).

Combining key (13) with the use of the connectives of propositional logic, we can translate some more complicated sentences from natural language, as can be seen in (18):

(18)

	Sentence	*Translation*
(a)	John is bigger than Peter or Peter is bigger than John.	$Bjp_2 \vee Bp_2j$
(b)	If the chicken is cackling, then Casper is plucking it.	$Cc_1 \rightarrow Pc_2c_1$
(c)	If John is cackling, then Casper is bigger than John.	$Cj \rightarrow B_1c_2j$
(d)	If Peter admires Casper, then he is not plucking him.	$Ap_2c_2 \rightarrow \neg Pp_2c_2$
(e)	Alcibiades admires himself.	Aaa
(f)	Casper and John are plucking each other.	$Pjc_2 \wedge Pc_2j$
(g)	If Socrates is a man, then he is mortal.	$M_1s \rightarrow M_2s$
(h)	Socrates is a mortal man.	$M_1s \wedge M_2s$

In (18) we also see how words which refer back to entities already mentioned, like personal and reflexive pronouns, can be handled in predicate logic. Possessive pronouns are a bit more difficult. Expressions beginning with a possessive pronoun generally refer to some particular object; the context determines which one. As such they are just like expressions beginning with *the, this,* etc., and all such expressions will in the meantime be translated as individual constants without being subjected to any further analysis. In §5.2, where we discuss so-called definite descriptions, we shall have more to say about them.

Sentences (b), (d), and (g) in (18) of course all have readings for which the given translations are incorrect; contexts can be thought of in which *it* and *he* refer to entities other than *the chicken, Casper, Peter,* and *Socrates*. In translating these kinds of sentences, we just choose the most natural interpretation.

Unlike the theory of types, which will be discussed in volume 2, predicate logic does not enable us to distinguish between (18e) and *Alcibiades admires Alcibiades*. Sentences (19) and (20) cannot be distinguished either:

(19) If Onno teases Peter, then he pleases him.

(20) If Onno teases Peter, then Onno pleases Peter.

Both (19) and (20), given the obvious translation key, are rendered as $Top \rightarrow Pop$.

Note that the simple sentence (18h) has been translated as the conjunction of two atomic sentences. This is in order to make the logical properties of the sentence as explicit as possible, which is the aim of such translations. Logically speaking, sentence (18h) expresses two things about Socrates: that he is a man and that he is mortal.

Exercise 1

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

a. John is nicer than Peter.
b. Charles is nice, but Elsa isn't.
c. Peter went with Charles on Marion's new bicycle to Zandvoort.
d. If Peter didn't hear the news from Charles, he heard it from Elsa.
e. Charles is boring or irritating.
f. Marion is a happy woman.
g. Bee is a best-selling author.
h. Charles and Elsa are brother and sister or nephew and niece.
i. John and Peter are close friends.

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 . . . means: for every entity x in the domain we have . . . ; and $\exists$x . . . means: there is at least one entity in the domain such that . . . ; $\forall x\phi$ 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 $\forall$xVx (or equivalently, as $\forall$yVy or as $\forall$zVz, since variables have no meaning of their own), to translate (27) as $\exists$xVx (or as $\exists$yVy or $\exists$zVz), (26) as $\neg\exists$xVx, 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 $\neg\exists$xVx 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:*

(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 $\exists$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 $\exists$yAxy. Finally, by universally quantifying this formula we obtain the formula $\forall$x$\exists$yAxy, which says that everyone in the domain has the property expressed by (32). So $\forall$x$\exists$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) $\forall x(Tx \rightarrow Fx)$

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

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 $\wedge$, (40) means the same as (41), which is the translation of (42):

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

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

(44) $\forall x(Tx \rightarrow \neg Fx)$

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

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 $\forall x\neg\phi$ always means the same as $\neg\exists x\phi$. 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)

$\neg\exists x\exists y(\forall z(\exists wAzw \rightarrow Ayz) \wedge Axy)$ (ii)

$\exists x\exists y(\forall z(\exists wAzw \rightarrow Ayz) \wedge Axy)$ (iv, $\exists$)

$\exists y(\forall z(\exists wAzw \rightarrow Ayz) \wedge Axy)$ (iv, $\exists$)

$\forall z(\exists wAzw\ Ayz) \wedge Axy$ (iii, $\wedge$)

$\forall z(\exists wAzw \rightarrow Ayz)$ (iv, $\forall$) Axy (i)

$\exists wAzw \rightarrow Ayz$ (iii, $\rightarrow$)

$\exists wAzw$ (iv, $\exists$) Ayz (i)

Azw (i)

(51) $\neg\exists x\exists y(\forall z(\exists wAzw \rightarrow Ayz) \wedge Axy)$

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)
```
      Axy (i)
    /   |   \
   A    x    y
```

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, $\exists x\exists y(\forall z(\exists wAzw \rightarrow Ayz) \wedge Axy)$, $\exists y(\forall z(\exists wAzw \rightarrow Ayz) \wedge Axy)$, $\forall z(\exists wAzw \rightarrow Ayz) \wedge Axy$, $\forall z(\exists wAzw \rightarrow Ayz)$, Axy, $\exists wAzw \rightarrow Ayz$, $\exists wAzw$, 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
$\forall z$	$\exists wAzw \rightarrow Ayz$
$\exists y$	$\forall z(\exists wAzw \rightarrow Ayz) \wedge Axy$
$\exists x$	$\exists y(\forall z(\exists wAzw \rightarrow Ayz) \wedge Axy)$

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

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

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 $\forall x(Ax \wedge \exists xBx)$. 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 $Ax \wedge \exists xBx$, 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; $\forall x(Ax \wedge \exists xBx)$ is a sentence, but $Ax \wedge \exists xBx$ 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) whethrer it is a sentence.

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

As we have mentioned, a formula with free variables is called a *propositional function*. If we take the formula $Tx \to Fx$ with its one free variable x and replace x with the constant j, then we obtain a sentence, namely, $Tj \to Fj$. So $Tx \to Fx$ 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 $Tx \to Fx$ with c as its argument is the sentence $Tc \to Fc$. 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) $\forall z(Axz \to Ayz)$

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

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
$Axx \wedge \exists xBx$	$Acc \wedge \exists xBx$
$\forall xBy$	$\forall xBy$
$\exists x\exists yAxy \to Bx$	$\exists x\exists yAxy \to Bc$
$\forall x\forall yAyy \to Bx$	$\forall x\forall yAyy \to Bc$

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

When restricted to A, a quantifier $\exists x$ becomes $\exists x(Ax \wedge$; and a quantifier $\forall x$ becomes $\forall x(Ax \to$. 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 reasons for this. First, it is the usual way of doing things in the literature. And second, the concept of a set plays an essential role in the semantics of logical systems which are more complex than predicate logic (and which we shall come to in volume 2).

Actually, we have already run across sets in the domains and ranges of functions. To put it as generally as possible, a *set* is a collection of entities. There is a sense in which its membership is the only important thing about a set, so it does not matter how the collection was formed, or how we can discover what entities belong to it. Take the domain of a function, for instance. Whether or not this function attributes a value to any given entity depends on just one thing—the membership of this entity in the domain. The central importance of membership is expressed in the *principle of extensionality* for sets. According to this principle, a set is completely specified by the entities which belong to it. Or, in other words, no two different sets can contain exactly the same members. For example, the set of all whole numbers larger than 3 and smaller than 6, the set containing just the numbers 4 and 5, and the set of all numbers which differ from 4.5 by exactly 0.5 are all the same set. An entity a which belongs to a set A is called an *element* or a *member* of A. We say that A *contains* a (as an element). This is written $a \in A$. We write $a \notin A$ if a is not an element of A.

Finite sets can be described by placing the names of the elements between set brackets: so $\{4, 5\}$ is the set described above, for example; $\{0, 1\}$ is the set of truth values; $\{p, q, p \to q, \neg(p \to q)\}$ is the set of all subformulas of $\neg(p \to q)$; $\{x, y, z\}$ is the set of all variables which are free in the formula $\forall w((Axw \wedge Byw) \to Czw)$. So we have, for example $0 \in \{0, 1\}$ and $y \in \{x, y, z\}$. There is no reason why a set may not contain just a single element, so that $\{0\}$, $\{1\}$, and $\{x\}$ are all examples of sets. Thus $0 \in \{0\}$, and to put it generally, $a \in \{0\}$ just in case $a = 0$. It should be noted that a set containing some single thing is not the same as that thing itself; in symbols, $a \neq \{a\}$. It is obvious that $2 \neq \{2\}$, for example, since 2 is a number, while $\{2\}$ is a set. Sets with no elements at all are also allowed; in view of the principle of extensionality, there can be only one such empty set, for which we have the notation $\varnothing$. So there is no a such that $a \in \varnothing$. Since the only thing which matters is the membership, the order in which the elements of a set are given in the brackets notation is irrelevant. Thus $\{4, 5\} = \{5, 4\}$ and $\{z, x, y\} = \{x, y, z\}$, for example. Nor does it make any difference if some elements are written more than once: $\{0, 0\} = \{0\}$ and $\{4, 4, 5\} = \{4, 5\}$. A similar notation is also used for some infinite sets, with an expression between the brackets which suggests what elements are included. For example, $\{1, 2, 3, 4, \ldots\}$ is the set of positive whole numbers; $\{0, 1, 2, 3, \ldots\}$ is the set of *natural numbers;* $\{\ldots -2, -1, 0, 1, 2, \ldots\}$ is the set of all whole numbers; $\{0, 2, 4, 6, \ldots\}$ is the set of even natural numbers; and $\{p, \neg p, \neg\neg p, \neg\neg\neg p, \ldots\}$ is the set of all formulas in which only p and $\neg$ occur. We shall refer to the set of natural numbers as N for convenience, to the whole numbers as Z, and (at least in this section) to the even numbers as E.

If all of the elements of a set A also happen to be elements of a set B, then we say that A is a *subset* of B, which is written $A \subseteq B$. For example, we have $\{x, z\} \subseteq \{x, y, z\}$; $\{0\} \subseteq \{0, 1\}$; $E \subseteq N$, and $N \subseteq Z$. Two borderline cases of this are $A \subseteq A$ for every set A and $\varnothing \subseteq A$ (since the empty set has no elements at all, the requirement that all of its elements are elements of A is fulfilled vacuously). Here are a few properties of $\in$ and $\subseteq$ which can easily be verified:

(86) if $a \in A$ and $A \subseteq B$, then $a \in B$
if $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$
$a \in A$ iff $\{a\} \subseteq A$
$a \in A$ and $b \in A$ iff $\{a, b\} \subseteq A$
if $A \subseteq B$ and $B \subseteq A$, then $A = B$

The last of these, which says that two sets that are each other's subsets are equal, emphasizes once more that it is the membership of a set which determines its identity.

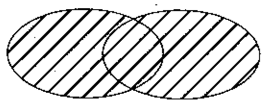
Often we will have cause to specify a subset of some set A by means of a property G, by singling out all of A's elements which have this property G. The set of natural numbers which have the property of being both larger than 3 and smaller than 6 is, for example, the set $\{4, 5\}$. Specifying this set in the manner just described, it would be written as $\{4, 5\} = \{x \in N \mid x > 3 \text{ and } x < 6\}$. The general notation for the set of all elements of A which have the property G is $\{x \in A \mid G(x)\}$. A few examples have been given as (87):

(87) $N = \{x \in Z \mid x \geqslant 0\}$
$E = \{x \in N \mid \text{there is an } y \in N \text{ such that } x = 2y\}$
$\{0\} = \{x \in \{0, 1\} \mid x + x = x\}$
$\{0, 1, 4, 9, 16, 25, \ldots\} = \{x \in N \mid \text{there is an } y \in N \text{ such that } x = y^2\}$
$\varnothing = \{x \in \{4, 5\} \mid x + x = x\}$
$\{0, 1\} = \{x \in \{0, 1\} \mid x \times x = x\}$
$\{p \to q\} = \{\phi \in \{p, q, p \to q, \neg(p \to q)\} \mid \phi \text{ is an implication}\}$

The above specification of E is also abbreviated as: $\{2y \mid y \in N\}$. Analogously, we might also write $\{y^2 \mid y \in N\}$ for the set $\{0, 1, 4, 9, 16, 25, \ldots\}$. Using this notation, the fact that f is a function from A *onto* B can easily be expressed by $\{f(a) \mid a \in A\} = B$. Another notation for the set of entities with some property G is $\{x \mid G(x)\}$. We can, by way of example, define $P = \{X \mid X \subseteq \{0, 1\}\}$; in which case P is set $\{\varnothing, \{0\}, \{1\}, \{0, 1\}\}$. Note that sets are allowed to have other sets as members.

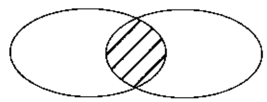
The *union* $A \cup B$ of two sets A and B can now be defined as the set $\{x \mid x \in A \vee x \in B\}$. So $A \cup B$ is the set of all things which appear in either or both of A and B. Analogously, the *intersection* $A \cap B$ of A and B is defined as $\{x \mid x \in A \wedge x \in B\}$, the set of all things which appear in both A and B. By means of Venn diagrams, $A \cup B$ and $A \cap B$ can be represented graphically as in figures (88) and (89), respectively.

(88) A B



$A \cup B$

(89) A B



$A \cap B$

Defining sets by means of $\{x \mid G(x)\}$ can, however, cause considerable difficulty if no restrictions are placed on the reservoir from which the entities satisfying G are to be drawn. In fact, if we assume that $\{x \mid G(x)\}$ is always a set for every property G, then we get caught in the *Russell paradox,* which caused a great deal of consternation in mathematics around the turn of the century. A short sketch of the paradox now follows. Given the above assumption, we have to accept $\{x \mid x = x\}$ as a set. V is the universal set containing everything, since every entity is equal to itself. Now if V contains everything, then in particular, $V \in V$; so $x \in x$ is a property which some special sets like V have, but which most sets do not have; $0 \notin 0$ because 0 is not a set; $\{0\} \notin \{0\}$ because $\{0\}$ has just one element, 0, and $0 \neq \{0\}$; $N \notin N$, since N has only whole numbers as its elements, and not sets of these, etc. Now consider the set R of all these entities, which according to our assumption, is defined by $R = \{x \mid x \notin x\}$. Then either R is an element of itself or not, and this is where the paradox comes in. If we suppose that $R \in R$, then R must have the property which determines membership in R, whence $R \notin R$. So apparently $R \in R$ is impossible. But if $R \notin R$, then R has the property which determines membership in R, and so it must be the case that $R \in R$. So $R \notin R$ is also impossible.

In modern set theory, axioms determine which sets can be defined by means of which others. In this manner, many sets may be defined in the manner of $\{x \mid G(x)\}$, without giving rise to the Russell paradox. One price which must be paid for this is that the class $V = \{x \mid x = x\}$ can no longer be accepted as a set: it is too big for this. This is one of the reasons why we cannot simply include everything in our domain when translating into predicate logic.

There are occasions when the fact that the order of elements in a set does not matter is inconvenient. We sometimes need to be able to specify the sequential order of a group of entities. For this reason, we now introduce the notion of *finite sequences* of entities. The finite sequence beginning with the numeral 4, ending with 5, and containing just two entities, for example, is written as $\langle 4, 5\rangle$. Thus, we have $\langle 4, 5\rangle \neq \langle 5, 4\rangle$ and $\langle z, x, y\rangle \neq \langle x, y, z\rangle$. Other than with sets, with finite sequences it makes a difference if an entity appears a number of times: $\langle 4, 4, 5\rangle \neq \langle 4, 5\rangle$ and $\langle 4, 4, 4\rangle \neq \langle 4, 4\rangle$: the length of the sequences $\langle 4, 4, 5\rangle$ and $\langle 4, 4, 4\rangle$ is 3, while the length of $\langle 4, 5\rangle$ and $\langle 4, 4\rangle$ is 2, the length of a sequence being the number of entities appearing in it. Finite sequences of two entities are also called *ordered pairs,* finite sequences of three entities are called ordered triples, and ordered sequences of *n* entities are called ordered *n*-tuples. The set of all ordered pairs which can be formed from a set A is written A^2, A^3 is written for that of all ordered 3-tuples, and so on. More formally: $A^2 = \{\langle a, b\rangle \mid a \in A \text{ and } b \in A\}$; $A^3 = \{\langle a_1, a_2, a_3\rangle \mid a_1 \in A \text{ and } a_2 \in A \text{ and } a_3 \in A\}$, and so on. For example, $\langle 2, 3\rangle \in N^2$ and $\langle 1, 1, 1\rangle \in N^3$ and $\langle -1, 2, -3, 4\rangle \in Z^4$. The general notation A^n is used for the set of ordered *n*-tuples of elements of A; A^1 and A are identical.

This enables us to treat a binary function f with A as its domain as a unary function with A^2 as its domain. Instead of writing f(a, b), we can then write: $f(\langle a, b\rangle)$.

3.6 The Semantics of Predicate logic

The semantics of predicate logic is concerned with how the meanings of sentences, which just as in propositional logic, amount to their truth values, depend on the meanings of the parts of which they are composed. But since the parts need not themselves be sentences, or even formulas—they may also be predicate letters, constants, or variables—we will not be able to restrict ourselves to truth values in interpreting languages of predicate logic. We will need functions other than the valuations we encountered with in propositional logic, and ultimately the truth values of sentences will have to reduce to the interpretations of the constants and predicate letters and everything else which appears in them. Valuations, however, retain a central role, and it is instructive to start off just with them and to build up the rest of the apparatus for the interpretation of predicate logic from there. One first attempt to do this is found in the following definition, in which valuations are extended to the languages of predicate logic. It turns out that this is in itself not enough, so remember that the definition is only preliminary.

Definition 5

A valuation for a language L of predicate logic is a function with the sentences in L as its domain and $\{0, 1\}$ as its range, and such that:

(i) $V(\neg\phi) = 1$ iff $V(\phi) = 0$;
(ii) $V(\phi \wedge \psi) = 1$ iff $V(\phi) = 1$ and $V(\psi) = 1$;
(iii) $V(\phi \vee \psi) = 1$ iff $V(\phi) = 1$ or $V(\psi) = 1$;

(iv) $V(\phi \to \psi) = 1$ iff $V(\phi) = 0$ or $V(\psi) = 1$;
(v) $V(\phi \leftrightarrow \psi) = 1$ iff $V(\phi) = V(\psi)$;
(vi) $V(\forall x\phi) = 1$ iff $V([c/x]\phi) = 1$ for all constants c in L;
(vii) $V(\exists x\phi) = 1$ iff $V([c/x]\phi) = 1$ for at least one constant c in L.

The idea is that $\forall x\phi$ is true just in case $[c/x]\phi$ is true for every c in L, and that $\exists x\phi$ is true just in case $[c/x]\phi$ is true for at least one c in L. This could be motivated with reference to (90) and (91). For (90) is true just in case every substitution of the name of an individual human being into the open space in (91) results in a true sentence. And (92) is true just in case there is at least one name the substitution of which into (91) results in a true sentence.

(90) Everyone is friendly.

(91) . . . is friendly.

(92) Someone is friendly.

One thing should be obvious right from the start: in formal semantics, as in informal semantics, it is necessary to introduce a *domain of discourse*. For (90) may very well be true if the inhabitants of the Pacific state of Hawaii are taken as the domain, but untrue if all human beings are included. So in order to judge the truth value of (90), it is necessary to know what we are talking about, i.e., what the domain of discourse is. Interpretations of a language L of predicate logic will therefore always be with reference to some domain set D. It is usual to suppose that there is always at least one thing to talk about—so by convention, the domain is not empty.

3.6.1 *Interpretation Functions*

We will also have to be more precise about the relationship between the constants in L and the domain D. For if we wish to establish the truth value of (90) in the domain consisting of all inhabitants of Hawaii, then the truth value of *Liliuokalani is friendly* is of importance, while the truth value of *Gorbachev is friendly* is of no importance at all, since Liliuokalani is the name of an inhabitant of Hawaii (in fact she is, or at least was, one of its queens), while Gorbachev, barring unlikely coincidences, is not. Now it is a general characteristic of a proper name in natural language that it refers to some fixed thing. This is not the case in formal languages, where it is necessary to stipulate what the constants refer to. So an interpretation of L will have to include a specification of what each constant in L refers to. In this manner, constants refer to entities in the domain D, and as far as predicate logic is concerned, their meanings can be restricted to the entities to which they refer. The interpretation of the constants in L will therefore be an attribution of some entity in D to each of them, that is, a function with the set of constants in L as its domain and D as its range. Such functions are called *interpretation functions*.

I(c) is called the *interpretation* of a constant c, or its *reference* or its *denotation*, and if e is the entity in D such that I(c) = e, then c is said to be one of e's *names* (e may have several different names).

Now we have a domain D and an interpretation function I, but we are not quite there yet. It could well be that

(93) Some are white.

is true for the domain consisting of all snowflakes without there really being any English sentence of the form *a is white* in which *a* is the name of a snowflake. For although snowflakes tend to be white, it could well be that none of them has an English name. It should be clear from this that definition 5 does not work as it is supposed to as soon as we admit domains with unnamed elements. So two approaches are open to us:

A. We could stick to definition 5 but make sure that all objects in our domains have names. In this case, it will sometimes be necessary to add constants to a language if it does not contain enough constants to give a unique name to everything in some domain that we are working with.

B. We replace definition 5 by a definition which will also work if some entities lack names.

We shall take both approaches. Approach B seems preferable, because of A's intuitive shortcomings: it would be strange if the truth of a sentence in predicate logic were to depend on a contingency such as whether or not all of the entities being talked about had a name. After all, the sentences in predicate logic do not seem to be saying these kinds of things about the domains in which they are interpreted. But we shall also discuss A, since this approach, where it is possible, is simpler and is equivalent to B.

3.6.2 *Interpretation by Substitution*

First we shall discuss **approach A**, which may be referred to as *the interpretation of quantifiers by substitution*. We shall now say more precisely what we mean when we say that each element in the domain has a name in L. Given the terminology introduced in §2.4, we can be quite succinct: the interpretation function I must be a function from the constants in L *onto* D. This means that for every element d in D, there is at least one constant c in L such that I(c) = d, i.e., c is a name of d. So we will only be allowed to make use of the definition if I is a function onto D.

But even this is not wholly satisfactory. So far, the meaning of predicate letters has only been given syncategorematically. This can be seen clearly if the question is transplanted into natural language: definition 5 enables us to know the meaning of the word *friendly* only to the extent that we know which sentences of the form *a is friendly* are true. If we want to give a direct, categorematic interpretation of *friendly*, then the interpretation will have to be such that the truth values of sentences of the form *a is friendly* can be deduced from it. And that is the requirement that can be placed on it, since we have restricted the meanings of sentences to their truth values. As a result, the only thing which matters as far as sentences of the form *a is friendly* are concerned is their truth values. An interpretation which establishes which people are friendly and which are not will satisfy this requirement. For example, *Gorbachev is friendly* is true just in case Gorbachev is friendly, since *Gorbachev* is one name for the man Gorbachev. Thus we can establish which people are friendly and which are not just by taking the set of all friendly people in our domain as the interpretation of *friendly*. In general then, as the interpretation I(A) of a unary predicate letter A we take the set of all entities e in D such that for some constant a, Aa is true and I(a) = e. So I(A) = {I(a) | Aa is true} or, in other words, Aa is true just in case I(a) ∈ I(A).

Interpreting A as a set of entities is not the only approach open to us. We might also interpret A as a property and determine whether a given element of D has this property. Indeed, this seems to be the most natural interpretation. If it is a predicate letter, we would expect A to refer to a property. What we have done here is to take, not properties themselves, but the sets of all things having them, as the interpretations of unary predicate letters. This approach may be less natural, but it has the advantage of emphasizing that in predicate logic the only thing we need to know in order to determine the truth or falsity of a sentence asserting that something has some property is which of the things in the domain have that property. It does not matter, for example, how we know this or whether things could be otherwise. As far as truth values are concerned, anything else which may be said about the property is irrelevant. If the set of friendly Hawaiians were to coincide precisely with the set of bald ones, then in this approach, *friendly* and *bald* would have the same meaning, at least if we took the set of Hawaiians as our domain. We say that predicate letters are *extensional* in predicate logic. It is characteristic of modern logic that such restrictions are explored in depth and subsequently relaxed. More than extensional meaning is attributed to expressions, for example, in *intensional* logical systems, which will be studied in volume 2.

To continue with approach A, and assuming that I is a function onto D as far as the constants are concerned, we turn to the interpretations of binary predicate letters. Just as with unary predicates, the interpretation of any given binary predicate B does not have to do anything more than determine the d and e in D for which Bab is true if I(a) = d and I(b) = e. This can be done by interpreting B as a set of ordered pairs ⟨d, e⟩ in D^2 and taking Bab to be true if I(a) = d and I(b) = e. The interpretation must consist of ordered pairs, because the order of a and b matters. The interpretation of B is, in other words, a subset of D^2, and we have I(B) = {⟨I(a), I(b)⟩ | Bab is true} or equivalently, Bab is true just in case ⟨I(a), I(b)⟩ ∈ I(B). Here too it may seem more intuitive to interpret B as a relation on D and to say that Bab is true if and only if I(a) and I(b) bear this relation to each other. For reasons already mentioned, however, we prefer the extensional approach and interpret a binary predicate letter not as a relation itself but as the set of ordered pairs of domain elements which (in the order they have in the pairs) have this relation to each other. And we thus have the principle of extensionality here too: two relations which hold for the same ordered pairs are identical. Ternary predicates and predicates of all higher arities are given an analogous treatment. If C is a ternary predicate letter, then I(C) is a subset of D^3, and if C is an *n*-ary predicate, then I(C) is a subset of D^n. We shall now summarize all of this in the following two definitions:

Definition 6

A *model* **M** for a language L of predicate logic consists of a domain D (this being a nonempty set) and an interpretation function I which is defined on the set of constants and predicate letters in the vocabulary of L and which conforms to the following requirements:

(i) if c is a constant in L, then I(c) ∈ D;
(ii) if B is an *n*-ary predicate letter in L, then I(B) ⊆ D^n.

Definition 7

If **M** is a model for L whose interpretation function I is a function of the constants in L onto the domain D, then V_M, *the valuation V based on* **M**, is defined as follows:

(i) If $Aa_1 \ldots a_n$ is an atomic sentence in L, then $V_M(Aa_1 \ldots a_n) = 1$ if and only if $\langle I(a_1), \ldots, I(a_n)\rangle \in I(A)$.
(ii) $V_M(\neg\phi) = 1$ iff $V_M(\phi) = 0$.
(iii) $V_M(\phi \wedge \psi) = 1$ iff $V_M(\phi) = 1$ and $V_M(\psi) = 1$.
(iv) $V_M(\phi \vee \psi) = 1$ iff $V_M(\phi) = 1$ or $V_M(\psi) = 1$.
(v) $V_M(\phi \to \psi) = 1$ iff $V_M(\phi) = 0$ or $V_M(\psi) = 1$.
(vi) $V_M(\phi \leftrightarrow \psi) = 1$ iff $V_M(\phi) = V_M(\psi)$.
(vii) $V_M(\forall x\phi) = 1$ iff $V_M([c/x]\phi) = 1$ for all constants c in L.
(viii) $V_M(\exists x\phi) = 1$ iff $V_M([c/x]\phi) = 1$ for at least one constant c in L.

If $V_M(\phi) = 1$, then ϕ is said to be *true* in model **M**.

If the condition that I be a function onto D is not fulfilled, then approach B will still enable us to define a suitable valuation function V_M, though this function will no longer fulfill clauses (vii) and (viii) of definition 7. Before showing how this can be done, we shall first give a few examples to illustrate method A.

Example 1

We turn the key to a translation into a model.
Key: Lxy: x loves y; domain: Hawaiians.
We take H, the set of all Hawaiians, as the domain of model **M**. Besides the binary predicate L, our language must contain enough constants to give each

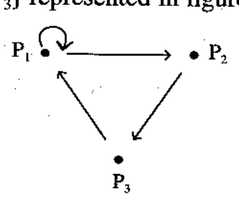
Hawaiian a name; $a_1, \ldots, a_{1,000,000}$ should be enough. Now for each *i* from 1 to 1,000,000 inclusive, a_i must be interpreted as a Hawaiian: $I(a_i) \in H$, and this in such a way that for each Hawaiian h there is some a_h which is interpreted as that Hawaiian, that is, for which $I(a_h) = h$. The interpretation of L is the following subset of H^2, i.e., the set of pairs of Hawaiians: $\{\langle d, e\rangle | d$ loves $e\}$. Lets us now determine the truth value of $\exists x\exists y(Lxy \wedge Lyx)$, which is the translation of *some people love each other.* Suppose that John loves Mary, that Mary's love for John is no less, that $I(a_{26})$ is Mary, and that $I(a_{27})$ is John. Then $\langle I(a_{26}), I(a_{27})\rangle \in I(L)$, and $\langle I(a_{27}), I(a_{26})\rangle \in I(L)$. According to definition 7i, we have $V_M(La_{26}a_{27}) = 1$ and $V_M(La_{27}a_{26}) = 1$, so that according to definition 7iii, we have $V_M(La_{26}a_{27} \wedge La_{27}a_{26}) = 1$. One application of definition 7viii now gives us $V_M(\exists y(La_{26}y \wedge Lya_{26})) = 1$, and a second gives us $V_M(\exists x\exists y(Lxy \wedge Lyx)) = 1$. Of course, it doesn't matter at all which constants are interpreted as which people. We could have shown that $V_M(\exists x\exists y(Lxy \wedge Lyx)) = 1$ just as well if $I(a_2)$ had been John and $I(a_9)$ had been Mary. This is a general fact: the truth of a sentence lacking constants is in any model independent of the interpretations of the constants in that model—with the proviso that everything in the domain has a name. A comment such as this should of course be proved, but we do not have the space here.

It is perhaps worth pointing out at this stage that semantics is not really concerned with finding out which sentences are in fact true and which are false. One's ideas about this are unlikely to be influenced much by the analysis given here. Essentially, semantics is concerned with *the ways the truth values of sentences depend on the meanings of their parts and the ways the truth values of different sentences are related.* This is analogous to the analysis of the notion of grammaticality in linguistics. It is assumed that it is clear which expressions are grammatical and which are not; the problem is to conceive a systematic theory on the subject.

The following examples contain a few extremely simple mathematical structures. We shall leave off the index **M** in V_M if it is clear what model the valuation is based on.

Example 2

The language we will interpret contains three constants, a_1, a_2, and a_3, and the binary predicate letter R. The domain D of the model is the set of points $\{P_1, P_2, P_3\}$ represented in figure (94).

(94) 

The constants are interpreted as follows: $I(a_1) = P_1$; $I(a_2) = P_2$; and $I(a_3) = P_3$. The interpretation of R is the relation holding between any two not neces-

sarily different points with an arrow pointing from the first to the second. So the following interpretation of R can be read from figure (94): $I(R) = \{\langle P_1, P_1\rangle, \langle P_1, P_2\rangle, \langle P_2, P_3\rangle, \langle P_3, P_1\rangle\}$. Representing this by means of a key, Rxy: there is an arrow pointing from x to y. It is directly obvious that $V(Ra_1a_2) = 1$, $V(Ra_2a_3) = 1$, $V(Ra_3a_1) = 1$, and $V(Ra_1a_1) = 1$; in all other cases, $V(Rbc) = 0$, so that, for example, $V(Ra_2a_1) = 0$ and $V(Ra_3a_3) = 0$. We shall now determine the truth value of $\forall x\exists yRxy$ (which means *every point has an arrow pointing away from it*).

(a) $V(\exists yRa_1y) = 1$ follows from $V(Ra_1a_2) = 1$ with definition 7viii;
(b) $V(\exists yRa_2y) = 1$ follows from $V(Ra_2a_3) = 1$ with definition 7viii;
(c) $V(\exists yRa_3y) = 1$ follows from $V(Ra_3a_1) = 1$ with definition 7viii.

From (a), (b), and (c), we can now conclude that $V(\forall x\exists yRxy) = 1$ with definition 7vii. The truth value of $\forall x\exists yRyx$ (which means *every point has an arrow pointing to it*) can be determined in just the same way:

(d) $V(\exists yRya_1) = 1$ follows from $V(Ra_1a_1) = 1$ with definition 7viii;
(e) $V(\exists yRya_2) = 1$ follows from $V(Ra_1a_2) = 1$ with definition 7viii;
(f) $V(\exists yRya_3) = 1$ follows from $V(Ra_2a_3) = 1$ with definition 7viii.

From (d), (e), and (f), we conclude that $V(\forall x\exists yRyx) = 1$ with definition 7vii.

Finally, we shall determine the truth value of $\exists x\forall yRxy$ (which means: *there is a point from which arrows go to all other points*):

(g) $V(\forall yRa_1y) = 0$ follows from $V(Ra_1a_3) = 0$ with definition 7vii;
(h) $V(\forall yRa_2y) = 0$ follows from $V(Ra_2a_2) = 0$ with definition 7vii;
(i) $V(\forall yRa_3y) = 0$ follows from $V(Ra_3a_3) = 0$ with definition 7vii.

From (g), (h), and (i), we can now conclude that $V(\exists x\forall yRxy) = 0$ with definition 7viii.

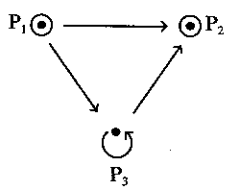
Example 3

We consider a language with a unary predicate letter E, a binary predicate letter L, and constants $a_0, a_1, a_2, a_3, \ldots$. We take N, the set $\{0, 1, 2, 3, \ldots\}$ of natural numbers, as our domain. We choose $V(a_i) = i$ for every i and interpret E as the set of even numbers, so that $I(E) = \{0, 2, 4, 6, \ldots\}$. We interpret L as $<$, so that $I(L) = \{\langle m, n\rangle | m$ less than $n\}$. As true sentences we then have, for example, Ea_2, La_4a_5, and $\forall x\exists y(Lxy \wedge \neg Ey)$ (these mean *2 is even, 4 is less than 5,* and *for every number there is a larger number which is odd,* respectively). We shall expand on the last of these. Consider any number m. This number must be either even or odd.

If m is even, then m + 1 is odd, so that $V(Ea_{m+1}) = 0$ and $V(\neg Ea_{m+1}) = 1$. We also have $V(La_ma_{m+1}) = 1$, since $m < m + 1$. From this we may conclude that $V(La_ma_{m+1} \wedge \neg Ea_{m+1}) = 1$, and finally that $V(\exists y(La_my \wedge \neg Ey)) = 1$.

If, on the other hand, m is odd, then m + 2 is odd too, so that $V(Ea_{m+2}) = 0$ and $V(\neg Ea_{m+2}) = 1$. We also have $V(La_ma_{m+2}) = 1$, since $m < m + 2$, and thus $V(La_m a_{m+2} \wedge \neg Ea_{m+2}) = 1$, so that we have $V(\exists y(La_m y \wedge \neg Ey)) = 1$ in this case as well. Since this line of reasoning applies to an arbitrary number m, we have for every a_m: $V(\exists y(La_m y \wedge \neg Ey)) = 1$. Now we have shown that $V(\forall x \exists y(Lxy \wedge \neg Ey)) = 1$.

(95)



Exercise 7

Model **M** is given in figure (95). The language has three constants a_1, a_2, and a_3 interpreted as the points P_1, P_2, and P_3, a unary predicate letter A interpreted as the predicate that applies to a point if it has a circle around it, and a binary predicate letter R to be interpreted as in example 2.

(a) Describe exactly the interpretation function I of the model **M**.

(b) Determine on the basis of their meaning the truth or falsity of the following sentences on model **M** and then justify this in detail, using definition 7:

- (i) $\exists x \exists y \exists z(Rxy \wedge Ay \wedge Rxz \wedge \neg Az)$.
- (ii) $\forall x Rxx$.
- (iii) $\forall x(Rxx \leftrightarrow \neg Ax)$.
- (iv) $\exists x \exists y(Rxy \wedge \neg Ax \wedge \neg Ay)$.
- (v) $\forall x(Rxx \rightarrow \exists y(Rxy \wedge Ay))$.
- (vi) $\forall x(Ax \rightarrow \exists y\, Rxy)$.
- (vii) $\exists x \exists y(Rxy \wedge \neg Ryx \wedge \exists z(Rxz \wedge Rzy))$.

3.6.3 *Interpretation by means of assignments*

We have now come to the explication of **approach B.** To recapitulate: we have a language L, a domain D, and an interpretation function I which maps all of L's constants into D but which is not necessarily a function *onto* D. That is, we have no guarantee that everything in the domain has some constant as its name. This means that the truth of sentences $\exists x\phi$ and $\forall x\phi$ can no longer be reduced to that of sentences of the form $[c/x]\phi$. Actually, this reduction is not that attractive anyway, if we wish to take the principle of compositionality strictly. This principle requires that the meaning (i.e., the truth value) of an expression be reducible to that of its composite parts. But sentences $\exists x\phi$ and $\forall x\phi$ do not have sentences of the form $[c/x]\phi$ as their component parts, because they are obtained by placing a quantifier in front of a formula ϕ, which normally has a free variable x and therefore is not even another sentence. What this means is that we will have to find some way to attach meanings to

formulas in general; we can no longer restrict ourselves to the special case of sentences.

We have reserved the name *propositional function* for formulas with free variables, in part because sentences can be obtained by replacing the free variables with constants, and in part because a formula with free variables does not seem to express a proposition but rather a property or a relation. But we could also take a different view and say that formulas with free variables express propositions just as much as sentences do, only these propositions are about unspecified entities. This would be why they are suited to express properties and relations.

In order to see how a meaning can be attached to these kinds of formulas, let us return again to (96) (=(93)):

(96) Some are white.

This was to be interpreted in the domain consisting of all snowflakes. What we want to do is determine the truth value of (96) with reference to the meaning of *x is white* interpreted in the domain consisting of all snowflakes. Now x, as we have emphasized, has no meaning of its own, so it must not refer to some fixed entity in the domain as if it were a constant. This may be compared with the way pronouns refer in sentences like *he is white* and *she is black*. But precisely for this reason, it may make sense to consider x as the *temporary* name of some entity. The idea is to consider model **M** together with an extra attribution of denotations to x and all the other variables; x will receive a temporary interpretation as an element in D. It is then quite easy to determine the truth value of (96): (96) is true if and only if there is some attribution of a denotation in the domain of all snowflakes to x, such that *x is white* becomes a true sentence. In other words, (96) is true just in case there is some snowflake which, if it is given the name x, will turn *x is white* into a true sentence—and that is exactly what we need.

The meaning of

(97) They are all black.

in the domain consisting of all snowflakes can be handled in much the same way: (97) is true if and only if *every* attribution of a denotation to x in this domain turns *x is black* into a true sentence. Analyzing this idea brings up more technical problems than most things we have encountered so far.

In order to determine the truth value of a sentence like $\exists x \exists y(Hxy \wedge Hyx)$, it is necessary to work back (in two steps) to the meaning of its subformula $Hxy \wedge Hyx$, which has two free variables. Obviously since no limitation is placed on the length of formulas, such subformulas can contain any number of free variables. This means that we must deal with the meanings of formulas with any number of free variables in order to determine the truth values of sentences. What matters is the truth value of a formula once all of its free variables have been given a temporary denotation, but it turns out that it is easiest to give all free variables a denotation at the same time. It is unnecessarily difficult to keep track of what free variables each formula has and to assign denotations to them. What we do is use certain functions called *assignments* which have the set of all variables in the language as their domain, and D, the domain of the model, as their range.

We will now describe the truth values a model **M** gives to the formulas of L under an assignment g by means of a valuation function $V_{M,g}$. This function will be defined by modifying conditions (i)–(viii) of definition 7 above.

The complications begin with clause (i). There is no problem as long as we deal with an atomic formula containing only variables and no constants: we are then dealing with $V_{M,g}(Ax_1 \ldots x_n)$, and it is clear that we wish to have $V_{M,g}(Ax_1 \ldots x_n) = 1$ if and only if $\langle g(x_1), \ldots, g(x_n)\rangle \in I(A)$, since the only difference from the earlier situation is that we have an assignment g attributing denotations to variables instead of an interpretation I attributing denotations to constants. But it becomes more difficult to write things up properly for formulas of the form $At_1 \ldots t_n$, in which $t_1, \ldots, t_n$ may be either constants or variables. What we do is introduce *term* as the collective name for the constants and variables of L. We first define what we mean by $[\![t]\!]_{M,g}$, the interpretation of a term t in a model **M** under an assignment g.

Definition 8

$[\![t]\!]_{M,g} = I(t)$ if t is a constant in L, and
$[\![t]\!]_{M,g} = g(t)$ if t is a variable.

Now we can generalize (i) in definition 7 to:

$$V_{M,g}(At_1 \ldots t_n) = 1 \text{ iff } \langle [\![t_1]\!]_{M,g}, \ldots, [\![t_n]\!]_{M,g}\rangle \in I(A).$$

It is clear that the value of $V_{M,g}(At_1 \ldots t_n)$ does not depend on the value of g(y) if y does not appear among the terms $t_1, \ldots, t_n$.

Clauses (ii) to (vi) in definition 7 can be transferred to the definition of $V_{M,g}$ without modification. The second clause we have to adapt is (viii), the clause for $V_{M,g}(\exists y\phi)$. Note that ϕ may have free variables other than y. Let us return to the model given in example 1, only this time for a language lacking constants. We take Lxy as our ϕ. Now how is $V_{M,g}(\exists yLxy)$ to be defined? Under an assignment g, x is treated as if it denotes g(x), so $\exists yLxy$ means that g(x) loves someone. So the definition must result in $V_{M,g}(\exists yLxy) = 1$ if and only if there is a $d \in H$ such that $\langle g(x), d\rangle \in I(L)$. The idea was to reduce the meaning of $\exists yLxy$ to the meaning of Lxy. But we cannot take $V_{M,g}(\exists yLxy) = 1$ if and only if $V_{M,g}(Lxy) = 1$, since $V_{M,g}(Lxy) = 1$ if and only if $\langle g(x), g(y)\rangle \in I(L)$, that is, if and only if g(x) loves g(y). For it may well be that g(x) loves someone without this someone being g(y). The existential quantifier forces us to consider assignments other than g which only differ from g in the value which they assign to y, since the denotation of x may clearly not be changed.

On the one hand, if there is an assignment g′ which differs from g only in the value it assigns to y and such that $V_{M,g'}(Lxy) = 1$, then $\langle g'(x), g'(y)\rangle \in I(L)$, and thus, because $g(x) = g'(x)$, $\langle g(x), g'(y)\rangle \in I(L)$. So for some $d \in H$, $\langle g(x), d\rangle \in I(L)$. On the other hand, if there is some $d \in H$ such that $\langle g(x), d\rangle \in I(L)$, then it can easily be seen that there is always an assignment g′ such that $V_{M,g'}(Lxy) = 1$. Choose g′, for example, the assignment obtained by taking g and then just changing the value assigned to y to d. Then $\langle g'(x), g'(y)\rangle \in I(L)$, and so $V_{M,g'}(Lxy) = 1$. This argument can be repeated for any given formula, so now we can give a first version of the new clause for existential formulas. It is this: $V_{M,g}(\exists yLxy) = 1$ if and only if there is a g′ which differs from g only in its value for y and for which $V_{M,g'}(Lxy) = 1$. So g′ is uniquely determined by g, and the value g′ is assigned to the variable y. This means that we can adopt the following notation: we write g[y/d] for g′ if this assignment assigns d to y and assigns the same values as g to all the other variables. (Note that c in the notation $[c/x]\phi$ refers to a constant in L, whereas the d in g[y/d] refers to an entity in the domain; the first expression refers to the result of a syntactic operation, and the second does not.) The assignments g[y/d] and g tend to differ. But that is not necessarily the case, since they are identical if $g(y) = d$. So now we can give the final version of the new clause for existential formulas. It is this:

$$V_{M,g}(\exists y\phi) = 1 \text{ iff there is a } d \in D \text{ such that } V_{M,g[y/d]}(\phi) = 1.$$

A similar development can be given for the new clause for the universal quantifier. So now we can complete this discussion of the B approach by giving the following definition. It is well known as Tarski's truth definition, in honor of the mathematician A. Tarski who initiated it; it is a generalization of definition 7. Although clauses (ii)–(vi) are not essentially changed, we give the definition in full for ease of reference.

Definition 9

If **M** is a model, D is its domain, I is its interpretation function, and g is an assignment into D, then

(i) $V_{M,g}(At_1 \ldots t_n) = 1$ iff $\langle [\![t_1]\!]_{M,g}, \ldots, [\![t_n]\!]_{M,g}\rangle \in I(A)$;
(ii) $V_{M,g}(\neg\phi) = 1$ iff $V_{M,g}(\phi) = 0$;
(iii) $V_{M,g}(\phi \wedge \psi) = 1$ iff $V_{M,g}(\phi) = 1$ and $V_{M,g}(\psi) = 1$;
(iv) $V_{M,g}(\phi \vee \psi) = 1$ iff $V_{M,g}(\phi) = 1$ or $V_{M,g}(\psi) = 1$;
(v) $V_{M,g}(\phi \to \psi) = 1$ iff $V_{M,g}(\phi) = 0$ or $V_{M,g}(\psi) = 1$;
(vi) $V_{M,g}(\phi \leftrightarrow \psi) = 1$ iff $V_{M,g}(\phi) = V_{M,g}(\psi)$;
(vii) $V_{M,g}(\forall x\phi) = 1$ iff for all $d \in D$, $V_{M,g[x/d]}(\phi) = 1$;
(viii) $V_{M,g}(\exists x\phi) = 1$ iff there is at least one $d \in D$ such that $V_{M,g[x/d]}(\phi) = 1$.

We now state a few facts about this definition which we shall not prove. First, the only values of g which $V_{M,g}(\phi)$ is dependent on are the values which g assigns to variables which occur as free variables in ϕ; so ϕ has the same value for every g in the extreme case in which ϕ is a sentence. This means that for sentences ϕ we can just write $V_M(\phi)$. Consequently, it holds for sentences ϕ that if ϕ is true with respect to some g, then it is true with respect to all g. If all elements of the domain of **M** have names, then for any sentence ϕ, approach A and approach B give the same values for $V_M(\phi)$. In such cases then, either can be taken. We shall now return to the examples given in connection with approach A, and reconsider them with B.

Example 1

There is just a single binary predicate letter L in the language; the domain is H, the set of all Hawaiians; $I(L) = \{\langle d, e\rangle \in H^2 | d \text{ loves } e\}$, and John and Mary are two members of the domain who love one another. We now define g(x) = John and g(y) = Mary; we complete g by assigning the other variables at random. Then $V_{M,g}(Lxy) = 1$, since $\langle [\![x]\!]_{M,g}, [\![y]\!]_{M,g}\rangle = \langle g(x), g(y)\rangle = \langle \text{John}, \text{Mary}\rangle \in I(L)$. Analogously, $V_{M,g}(Lyx) = 1$, so that we also have $V_{M,g}(Lxy \wedge Lyx) = 1$. This means that $V_{M,g}(\exists y(Lxy \wedge Lyx)) = 1$, since g = g[y/Mary], and that $V_{M,g}(\exists x\exists y(Lxy \wedge Lyx)) = 1$ too, since g = g[x/John].

Example 2

There is just a single binary predicate letter R in the language; the domain is $\{P_1, P_2, P_3\}$; $I(R) = \{\langle P_1, P_2\rangle, \langle P_1, P_1\rangle, \langle P_2, P_3\rangle, \langle P_3, P_1\rangle\}$. Now for an arbitrary g we have:

if $g(x) = P_1$, then $V_{M,g[y/P_2]}(Rxy) = 1$, since $\langle P_1, P_2\rangle \in I(R)$;
if $g(x) = P_2$, then $V_{M,g[y/P_3]}(Rxy) = 1$, since $\langle P_2, P_3\rangle \in I(R)$;
if $g(x) = P_3$, then $V_{M,g[y/P_1]}(Rxy) = 1$, since $\langle P_3, P_1\rangle \in I(R)$.

This means that for every g there is a $d \in \{P_1, P_2, P_3\}$ such that $V_{M,g[y/d]}(Rxy) = 1$. This means that $V_{M,g}(\exists yRxy) = 1$. Since this holds for an arbitrary g, we may conclude that $V_{M,g[x/d]}(\exists yRxy) = 1$ for every $d \in D$. We have now shown that $V_{M,g}(\forall x\exists yRxy) = 1$. That $V_{M,g}(\forall x\exists yRyx) = 1$ can be shown in the same way.

Now for the truth value of $\exists x\forall yRxy$. For arbitrary g, we have:

if $g(x) = P_1$, then $V_{M,g[y/P_3]}(Rxy) = 0$, since $\langle P_1, P_3\rangle \notin I(R)$;
if $g(x) = P_2$, then $V_{M,g[y/P_2]}(Rxy) = 0$, since $\langle P_2, P_2\rangle \notin I(R)$;
if $g(x) = P_3$, then $V_{M,g[y/P_3]}(Rxy) = 0$, since $\langle P_3, P_3\rangle \notin I(R)$.

This means that for every g there is a $d \in \{P_1, P_2, P_3\}$ such that $V_{M,g[y/d]}(Rxy) = 0$. From this it is clear that for every g we have $V_{M,g}(\forall yRxy) = 0$, and thus that for every $d \in D$, $V_{M,g[y/d]}(\forall yRxy) = 0$; and this gives $V_{M,g}(\exists x\forall yRxy) = 0$.

Example 3

The language contains a unary predicate letter E and a binary predicate letter L. The domain of our model **M** is the set N, $I(E) = \{0, 2, 4, 6, \ldots\}$, and $I(L) = \{\langle m, n\rangle | m < n\}$. Now let g be chosen at random. Then there are two possibilities:

(a) g(x) is an even number. In that case g(x) + 1 is odd, so that $g(x) + 1 \notin I(E)$, from which it follows that $V_{M,g[y/g(x)+1]}(Ey) = 0$ and that $V_{M,g[y/g(x)+1]}(\neg Ey) = 1$. Furhermore, $\langle g(x), g(x) + 1\rangle \in I(L)$, and therefore $V_{M,g[y/g(x)+1]}(Lxy) = 1$, so that we have $V_{M,g[y/g(x)+1]}(Lxy \wedge \neg Ey) = 1$.

(b) g(x) is an odd number. In that case, g(x) + 2 is an odd number too. From this it follows, as in (a), that $V_{M,g[y/g(x)+2]}(Lxy \wedge \neg Ey) = 1$. In both cases, then, there is an $n \in N$ such that $V_{M,g[y/n]}(Lxy \wedge \neg Ey) = 1$. This means that for every g, $V_{M,g}(\exists y(Lxy \wedge \neg Ey)) = 1$, from which it is clear that $V_{M,g}(\forall x\exists y(Lxy \wedge \neg Ey)) = 1$.

Exercise 8

Work out exercise 7bi, iii, and v again, now according to approach B (definition 9).

3.6.4 *Universal Validity*

In predicate logic as in propositional logic, we speak of *contradictions*, these being sentences ϕ such that $V_M(\phi) = 0$ for all models **M** in the language from which ϕ is taken. Here are some examples of contradictions: $\forall x(Ax \wedge \neg Ax)$, $\forall xAx \wedge \exists y\neg Ay$, $\exists x\forall y(Ryx \leftrightarrow \neg Ryy)$ (the last one is a formalization of Russell's paradox).

Formulas ϕ such that $V_M(\phi) = 1$ for all models **M** for the language from which ϕ is taken are called *universally valid* formulas (they are not normally called tautologies). That ϕ is universally valid is written as $\models\phi$. Here are some examples of universally valid formulas (more will follow later): $\forall x(Ax \vee \neg Ax)$, $\forall x(Ax \wedge Bx) \rightarrow \forall xAx$, $(\forall x(Ax \vee Bx) \wedge \exists x\neg Ax) \rightarrow \exists xBx$.

And in predicate logic as in propositional logic, sentences ϕ and ψ are said to be *equivalent* if they always have the same truth values, that is, if for every model **M** for the language from which ϕ and ψ are taken, $V_M(\phi) = V_M(\psi)$. On approach B, this can be generalized to: two formulas ϕ and ψ are equivalent if for every model **M** for the language from which they are taken and every assignment g into **M**, $V_{M,g}(\phi) = V_{M,g}(\psi)$. As an example of a pair of equivalent sentences, we have $\forall xAx$, $\forall yAy$, as can easily be checked. More generally, are $\forall x\phi$ and $\forall y([y/x]\phi)$ always equivalent? Not when y occurs free in ϕ; obviously $\exists xLxy$ is not equivalent to $\exists yLyy$: somebody may love y without anybody loving him- or herself.

It might be thought though, that $\forall x\phi$ and $\forall y([y/x]\phi)$ are equivalent for any ϕ in which y does not occur free. This is, however, not the case, as can be seen from the fact that $\forall x\exists yAxy$ and $\forall y\exists yAyy$ are not equivalent. In $\forall y\exists yAyy$, the quantifier $\forall y$ does not bind any variable y, and therefore $\forall y\exists yAyy$ is equivalent to $\exists yAyy$. But clearly $\forall x\exists yAxy$ can be true without $\exists yAyy$ being true. Everyone has a mother, for example, but there is no one who is his or her own mother. The problem, of course, is that y has been substituted for a free variable x within the range of the quantifier $\forall y$. If we want to turn the above into a theorem, then we need at least one restriction saying that this may not occur. The following definition enables us to formulate such restrictions more easily:

Definition 10

y is *free (for substitution) for x in* ϕ if x does not occur as a free variable within the scope of any quantifier $\forall y$ or $\exists y$ in ϕ.

For example, y will clearly be free for x in ϕ if y doesn't appear in ϕ. In general, it is not difficult to prove (by induction on the complexity of ϕ) that for ϕ in which y does not occur free, ϕ and $\forall y([y/x]\phi)$ are indeed equivalent if y is free for x in ϕ.

In predicate logic as in propositional logic, substituting equivalent subformulas for each other does not affect equivalence. We will discuss this in §4.2, but we use it in the following list of pairs of equivalent formulas:

(a) $\forall x\neg\phi$ is equivalent to $\neg\exists x\phi$. This is apparent from the fact that $V_{M,g}(\forall x\neg\phi) = 1$ iff for every $d \in D_M$, $V_{M,g[x/d]}(\neg\phi) = 1$; iff for every $d \in D_M$, $V_{M,g[x/d]}(\phi) = 0$; iff it is not the case that there is a $d \in D_M$ such that $V_{M,g[x/d]}(\phi) = 1$; iff it is not the case that $V_{M,g}(\exists x\phi) = 1$; iff $V_{M,g}(\exists x\phi) = 0$; iff $V_{M,g}(\neg\exists x\phi) = 1$.

(b) $\forall x\phi$ is equivalent to $\neg\exists x\neg\phi$, since $\forall x\phi$ is equivalent to $\forall x\neg\neg\phi$, and thus, according to (a), to $\neg\exists x\neg\phi$ too.

(c) $\neg\forall x\phi$ is equivalent to $\exists x\neg\phi$, since $\exists x\neg\phi$ is equivalent to $\neg\neg\exists x\neg\phi$, and thus, according to (b), to $\neg\forall x\phi$ too.

(d) $\neg\forall x\neg\phi$ is equivalent to $\exists x\phi$. According to (c), $\neg\forall x\neg\phi$ is equivalent to $\exists x\neg\neg\phi$, and thus to $\exists x\phi$.

(e) $\forall x(Ax \wedge Bx)$ is equivalent to $\forall xAx \wedge \forall xBx$, since $V_{M,g}(\forall x(Ax \wedge Bx)) = 1$ iff for every $d \in D_M$: $V_{M,g[x/d]}(Ax \wedge Bx) = 1$; iff for every $d \in D_M$: $V_{M,g[x/d]}(Ax) = 1$ and $V_{M,g[x/d]}(Bx) = 1$; iff for every $d \in D_M$: $V_{M,g[x/d]}(Ax) = 1$, while for every $d \in D_M$: $V_{M,g[x/d]}(Bx) = 1$; iff $V_{M,g}(\forall xAx) = 1$ and $V_{M,g}(\forall xBx) = 1$; iff $V_{M,g}(\forall xAx \wedge \forall xBx)) = 1$.

(f) $\forall x(\phi \wedge \psi)$ is equivalent to $\forall x\phi \wedge \forall x\psi$. This is a generalization of (e), and its proof is the same.

(g) $\exists x(\phi \vee \psi)$ is equivalent to $\exists x\phi \vee \exists x\psi$, since $\exists x(\phi \vee \psi)$ is equivalent to $\neg\forall x\neg(\phi \vee \psi)$, and thus to $\neg\forall x(\neg\phi \wedge \neg\psi)$ (de Morgan) and thus, according to (f), to $\neg(\forall x\neg\phi \wedge \forall x\neg\psi)$, and thus to $\neg\forall x\neg\phi \vee \neg\forall x\neg\psi$ (de Morgan), and thus, according to (d), to $\exists x\phi \vee \exists x\psi$.

N.B. $\forall x(\phi \vee \psi)$ is not necessarily equivalent to $\forall x\phi \vee \forall x\psi$. For example, each is male or female in the domain of human beings, but it is not the case that either all are male or all are female. $\exists x(\phi \wedge \psi)$ and $\exists x\phi \wedge \exists x\psi$ are not necessarily equivalent either. What we do have, and can easily prove, is:

(h) $\forall x(\phi \vee \psi)$ is equivalent to $\phi \vee \forall x\psi$ if x is not free in ϕ, and to $\forall x\phi \vee \psi$ if x is not free in ψ. Similarly:

(k) $\exists x(\phi \wedge \psi)$ is equivalent to $\exists x\phi \wedge \psi$ if x is not free in ψ, and to $\phi \wedge \exists x\phi$ if x is not free in ϕ.

(l) $\forall x(\phi \to \psi)$ is equivalent to $\phi \to \forall x\psi$ if x is not free in ϕ, since $\forall x(\phi \to \psi)$ is equivalent to $\forall x(\neg\phi \vee \psi)$ and thus, according to (h), to $\neg\phi \vee \forall x\psi$, and thus to $\phi \to \forall x\psi$. An example: *For everyone it holds that if the weather is fine, then he or she is in a good mood* means the same as *If the weather is fine, then everyone is in a good mood.*

(m) $\forall x(\phi \to \psi)$ is equivalent to $\exists x\phi \to \psi$ if x is not free in ψ, since $\forall x(\phi \to \psi)$ is equivalent to $\forall x(\neg\phi \vee \psi)$ and thus, according to (h), to $\forall x\neg\phi \vee \psi$, and thus, according to (a), to $\neg\exists x\phi \vee \psi$, and thus to $\exists x\phi \to \psi$. An example: *For everyone it holds that if he or she puts a penny in the slot, then a package of chewing gum drops out* means the same as *If someone puts a penny in the machine, then a package of chewing gum rolls out.*

(n) $\exists x\exists y(Ax \wedge By)$ is equivalent to $\exists xAx \wedge \exists yBy$, since $\exists x\exists y(Ax \wedge By)$ is equivalent to $\exists x(Ax \wedge \exists yBy)$, given (k), and with another application of (k), to $\exists xAx \wedge \exists yBy$.

(o) $\exists x\phi$ is equivalent to $\exists y([y/x]\phi)$ if y does not occur free in ϕ and y is free for x in ϕ, since $\exists x\phi$ is equivalent to $\neg\forall x\neg\phi$, according to (d). This in turn is equivalent to $\neg\forall y([y/x]\neg\phi)$, for y is free for x in ϕ if y is free for x in $\neg\phi$. And $\neg\forall y(\exists y/x]\neg\phi)$, finally, is equivalent to $\exists y([y/x]\phi)$ by (d), since $\neg([y/x]\phi)$ and $[y/x]\neg\phi$ are one and the same formula.

(p) $\forall x\forall y\phi$ is equivalent to $\forall y\forall x\phi$, as can easily be proved.

(q) $\exists x\exists y\phi$ is equivalent to $\exists y\exists x\phi$, on the basis of (d) and (p).

(r) $\exists x\exists yAxy$ is equivalent to $\exists x\exists yAyx$. According to (o), $\exists x\exists yAxy$ is equivalent to $\exists x\exists zAxz$, with another application of (o), to $\exists w\exists zAwz$, with (q), to $\exists z\exists wAwz$, and applying (o) another two times, to $\exists x\exists yAyx$.

In predicate logic too, for sentences ϕ and ψ, $\vDash\phi \leftrightarrow \psi$ iff ϕ and ψ are equivalent. And if $\vDash\phi \leftrightarrow \psi$, then both $\vDash\phi \to \psi$ and $\vDash\psi \to \phi$. But it is quite possible that $\vDash\phi \to \psi$ without ϕ and ψ being fully equivalent.

Here are some examples of universally valid formulas (proofs are omitted):

(i)	$\forall x\phi \to \exists x\phi$	(vi)	$\exists x\forall y\phi \to \exists y\forall x\phi$
(ii)	$\forall x\phi \to [t/x]\phi$	(vii)	$\forall xAxx \to \forall x\exists yAxy$
(iii)	$[t/x]\phi \to \exists x\phi$	(viii)	$\exists x\forall yAxy \to \exists xAxx$
(iv)	$(\forall x\phi \wedge \forall x\psi) \to \forall x(\phi \wedge \psi)$	(ix)	$\forall x(\phi \to \psi) \to (\forall x\phi \to \forall x\psi)$
(v)	$\exists x(\phi \wedge \psi) \to (\exists x\phi \wedge \exists x\psi)$	(x)	$\forall x(\phi \to \psi) \to (\exists x\phi \to \exists x\psi)$

Exercise 9

Prove of (i), (ii), (v) and (vii) of the above formulas that they are universally valid: prove (i) and (v) using approach A, assuming that all elements of a model have a name; prove (ii) and (vii) using approach B.

Exercise 10 ◇

Find as many implications and nonimplications as you can in the set of all possible formulas of the form Rxy prefixed by two quantifiers Q_1x, Q_2y (not necessarily in that order).

3.6.5 *Rules*

In order to discover universally valid formulas we may use certain *rules*. First, there is *modus ponens:*

(i) If $\models\phi$ and $\models\phi \rightarrow \psi$, then $\models\psi$.

It is not difficult to see that this rule is correct. For suppose that $\models\phi$ and $\models\phi \rightarrow \psi$, but that $\not\models\psi$. It follows from $\not\models\psi$ that there is some model **M** with $V_M(\psi) = 0$, and it follows from $\models\phi$ that $V_M(\phi) = 1$, and thus that $V_M(\phi \rightarrow \psi) = 0$, which contradicts $\models\phi \rightarrow \psi$. Here are some more rules:

(ii) If $\models\phi$ and $\models\psi$, then $\models\phi \wedge \psi$.
(iii) If $\models\phi \wedge \psi$, then $\models\phi$.
(iv) If $\models\phi$, then $\models\phi \vee \psi$.
(v) If $\models\phi \rightarrow \psi$, then $\models\neg\psi \rightarrow \neg\phi$.
(vi) $\models\neg\neg\phi$ iff $\models\phi$.

Such rules can be reduced to modus ponens. Take (v), for example, and suppose $\models\phi \rightarrow \psi$. It is clear that $\models(\phi \rightarrow \psi) \rightarrow (\neg\psi \rightarrow \neg\phi)$, since this formula has the form of a propositional tautology (theorem 13 in §4.2.2 shows that substitutions into tautologies like this are universally valid). Then with modus ponens it follows that $\models\neg\psi \rightarrow \neg\phi$. Here is a different kind of rule:

(vii) $\models\phi$ iff $\models\forall x([x/c]\phi)$, if x is free for c in ϕ.

Intuitively this is clear enough: if ϕ is universally valid and c is a constant appearing in ϕ, then apparently the truth of ϕ is independent of the interpretation given to c (ϕ holds for an 'arbitrary' c), so we might as well have a universal quantification instead of c.

Proof of (vii):

$\Leftarrow$: Suppose $\models\forall x([x/c]\phi)$. From example (ii) at the end of §3.6.4, we may conclude that $\models\forall x([x/c]\phi) \rightarrow [c/x][x/c]\phi$, and $[c/x][x/c]\phi$ is the same formula as ϕ (since x is free for c in ϕ). Now $\models\phi$ follows with modus ponens.

$\Rightarrow$: Suppose $\models\phi$, while $\not\models\forall x([x/c]\phi)$. Then apparently there is a model **M** with $V_M([x/c]\phi) = 0$. This means that there is an assignment g into **M** such that $V_{M,g}([x/c]\phi) = 0$. If we now define **M′** such that **M′** is the same as **M** (the same domain, the same interpretations), except that $I_{M'}(c) = g(x)$, then it is clear that $V_{M'}(\phi) = 0$, since x appears as a free variable in $[x/c]\phi$ at precisely the same points at which c appears in ϕ, because x is free for c in ϕ. This, however, cannot be the case, since ϕ is universally valid, so $\not\models\forall x([x/c]\phi)$ cannot be the case either. □

Rule (vii) now opens all kinds of possibilities. From $\models(Ac \wedge Bc) \rightarrow Ac$ (by substitution into a taulogy), it now follows that $\models\forall x((Ax \wedge Bx) \rightarrow Ax)$. And applying (ix) in §3.6.4 and modus ponens to this result, we obtain $\models\forall x(Ax \wedge Bx) \rightarrow \forall xAx$.

3.7 Identity

It is often useful in languages for predicate logic to have a binary predicate letter which expresses *identity*, the *equality* of two things. For this reason, we now introduce a new logical constant, =, which will always be interpreted as the relation of identity. The symbol =, of course, has been used many times in this book as an informal equality symbol derived from natural language or, if the reader prefers, as a symbol which is commonly added to natural language in order to express equality. We will continue to use = in this informal way, but this need not lead to any confusion.

A strong sense of the notion *identity* is intended here: by a = b we do not mean that the entities to which a and b refer are identical in the sense that they resemble each other very closely, like identical twins, for example. What we mean is that they are the same, so that a = b is true just in case a and b refer to the same entity. To put this in terms of valuations, we want $V_M(a = b) = 1$ in any model **M** just in case I(a) = I(b). (The first = in the sentence was in a formal language, the object language; the other two were in natural language, the metalanguage.)

The right valuations can be obtained if we stipulate that I will always be such that: $I(=) = \{\langle d, e\rangle \in D^2 | d = e\}$, or a shorter notation: $I(=) = \{\langle d, d\rangle | d \in D\}$. Then, with approach A, we have $V_M(a = b) = 1$ iff $\langle I(a), I(b)\rangle \in I(=)$ iff I(a) = I(b). And with method B, we have $V_{M,g}(a = b) = 1$ iff $\langle [\![a]\!]_{M,g}, [\![b]\!]_{M,g}\rangle \in I(=)$ iff $[\![a]\!]_{M,g} = [\![b]\!]_{M,g}$ iff I(a) = I(b).

The identity symbol can be used for more than just translations of sentences like *The morning star is the evening star* and *Shakespeare and Bacon are one and the same person*. Some have been given in (98):

(98)

Sentence	*Translation*
John loves Mary, but Mary loves someone else.	$Ljm \wedge \exists x(Lmx \wedge x \neq j)$
John does not love Mary but someone else.	$\neg Ljm \wedge \exists x(Ljx \wedge x \neq m)$

John loves no one but Mary.	$\forall x(Ljx \leftrightarrow x = m)$
No one but John loves Mary.	$\forall x(Lxm \leftrightarrow x = j)$
John loves everyone except Mary.	$\forall x(Ljx \leftrightarrow x \neq m)$
Everyone loves Mary except John.	$\forall x(Lxm \leftrightarrow x \neq j)$

The keys to the translations are the obvious ones and have been left out. In all cases, the domain is one with just people in it. We shall always write $s \neq t$ instead of $\neg(s = t)$.

If the domain in the above examples were to include things other than people, then $\forall x(Hx \rightarrow$ would have to be substituted for $\forall x$ in all the translations, and $\exists x(Hx \wedge$ for $\exists x$. Quite generally, if a sentence says that of all entities which have some property A, only a bears the relation R to b, then that sentence can be translated as $\forall x(Ax \rightarrow (Rxb \leftrightarrow x = a))$; but if a sentence states that all entities which have A bear R to b except the one entity a, then that sentence can be translated as $\forall x(Ax \rightarrow (Rxb \leftrightarrow x \neq a))$. We can also handle more complicated sentences, such as (99):

(99) Only John loves no one but Mary.

Sentence (99) can be rendered as $\forall x(\forall y(Lxy \leftrightarrow y = m) \leftrightarrow x = j)$. That this is correct should be fairly clear if it is remembered that $\forall y(Lxy \leftrightarrow y = m)$ says that x loves no one but Mary.

One of Frege's discoveries was that the meanings of numerals can be expressed by means of the quantifiers of predicate logic and identity. The principle behind this is illustrated in (100), the last three rows of which contain sentences expressing the numerals *one, two,* and *three*. For any natural number *n*, we can express the proposition that there are *at least n* things which have some property A by saying that there are *n* mutually different things which have A. That there are *at most n* different things which have A can be expressed by saying that of any $n + 1$ (not necessarily different) things which have A, at least two must be identical. That there are *exactly n* entities with A can now be expressed by saying that there are at least, and at most, *n* entities with A. So, for example, $\exists xAx \wedge \forall x\forall y((Ax \wedge Ay) \rightarrow x = y)$ can be used to say that there is exactly one x such that Ax. But shorter formulas that have the same effect can be found if we follow the procedure illustrated in (100). We say that there are *n* different entities and that any entity which has the property A must be one of these.

(100)

There is at least one x such that Ax.	$\exists xAx$
There are at least two (different) x such that Ax.	$\exists x\exists y(x \neq y \wedge Ax \wedge Ay)$
There are at least three (different) x such that Ax.	$\exists x\exists y\exists z(x \neq y \wedge x \neq z \wedge y \neq z \wedge Ax \wedge Ay \wedge Az)$
There is at most one x such that Ax.	$\forall x\forall y((Ax \wedge Ay) \rightarrow x = y)$
There are at most two (different) x such that Ax.	$\forall x\forall y\forall z((Ax \wedge Ay \wedge Az) \rightarrow (x = y \vee x = z \vee y = z))$
There are at most three (different) x such that Ax.	$\forall x\forall y\forall z\forall w((Ax \wedge Ay \wedge Az \wedge Aw) \rightarrow (x = y \vee x = z \vee x = w \vee y = z \vee y = w \vee z = w))$
There is exactly one x such that Ax.	$\exists x\forall y(Ay \leftrightarrow y = x)$
There are exactly two x such that Ax.	$\exists x\exists y(x \neq y \wedge \forall z(Az \leftrightarrow (z = x \vee z = y)))$
There are exactly three x such that Ax.	$\exists x\exists y\exists z(x \neq y \wedge x \neq z \wedge y \neq z \wedge \forall w(Aw \leftrightarrow (w = x \vee w = y \vee w = z)))$

This procedure is illustrated for a unary predicate letter A, but it works just as well for formulas ϕ. The formula $\exists x\forall y([y/x]\phi \leftrightarrow y = x)$, for example, says that there is exactly one thing such that ϕ, with the proviso that y must be a variable which is free for x in ϕ and does not occur free in ϕ. Sometimes a special notation is used for a sentence expressing *There is exactly one x such that* ϕ, $\exists x\forall y([y/x]\phi \leftrightarrow y = x)$ being abbreviated as $\exists!x\phi$.

We now give a few examples of sentences which can be translated by means of =. We do not specify the domains, since any set which is large enough will do.

(101) There is just one queen.

Translation: $\exists x\forall y\ (Qy \leftrightarrow y = x)$.
Key: Qx: x is a queen.

(102) There is just one queen, who is the head of state.

Translation: $\exists x(\forall y(Qy \leftrightarrow y = x) \wedge x = h)$.
Key: Qx: x is a queen; h: the head of state.
(This should be contrasted with $\exists!x(Qx \wedge x = h)$, which expresses that only one person is a governing queen, although there may be other queens around.)

(103) Two toddlers are sitting on a fence.

Translation: $\exists x(Fx \wedge \exists y_1\exists y_2(y_1 \neq y_2 \wedge \forall z((Tz \wedge Szx) \leftrightarrow$

$(z = y_1 \vee z = y_2))))$.
Key: Tx: x is a toddler; Sxy: x is sitting on y; Fx: x is a fence.

(104) If two people fight for something, another will win it.

Translation: $\forall x\forall y\forall z((Px \wedge Py \wedge x \neq y \wedge Tz \wedge Fxyz) \rightarrow \exists w(Pw \wedge w \neq x \wedge w \neq y \wedge Wwz))$.
Key: Px: x is a person; Tx: x is a thing; Fxyz: x and y fight for z; Wxy: x wins y.

Exercise 11

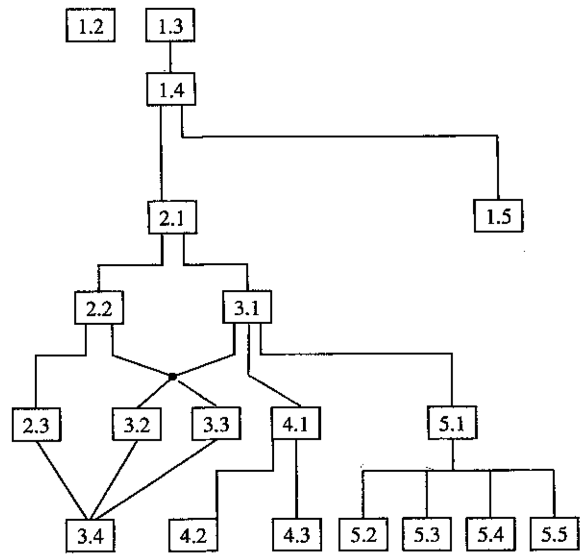
(a) No man is more clever than himself.
(b) For every man there exists another who is more clever.
(c) There is some man who is more clever than everybody except himself.
(d) There is somebody who is more clever than anybody except himself, and that is the prime minister.
(e) There are at least two queens.
(f) There are at most two queens.
(g) There are no queens except Beatrix.
(h) If two people make an exchange, then one of the two will be badly off.
(i) Any person has two parents.
(j) Mary only likes men.
(k) Charles loves no one but Elsie and Betty.
(l) Charles loves none but those loved by Betty.
(m) Nobody understands somebody who loves nobody except Mary.
(n) I help only those who help themselves.
(o) Everybody loves exactly one person.
(p) Everybody loves exactly one other person.
(q) Everybody loves a different person.
(r) All people love only themselves.
(s) People who love everybody but themselves are altruists.
(t) Altruists love each other.
(u) People who love each other are happy.

Exercise 12

(a) In many books, the dependencies between the different chapters or sections is given in the introduction by a figure. An example is figure a taken from Chang and Keisler's *Model Theory* (North-Holland, 1973). One can read figure a as a model having as its domain the set of sections {1.1, 1.3, . . . , 5.4, 5.5} in which the binary predicate letter R has been interpreted as dependency, according to the key: Rxy: y depends on x. Section 4.1, for example, depends on §3.1, but also on §§2.1, 1.4, and 1.3. For example, $\langle 2.1, 3.1\rangle \in I(R)$, and $\langle 1.4, 5.3\rangle \in I(R)$, but $\langle 2.2, 4.1\rangle \notin I(R)$.

Determine the truth values of the sentences below in the model on the basis of their meaning. Do not give all details. (With method A that is not possible anyway, since the entities in the model have not been named.)

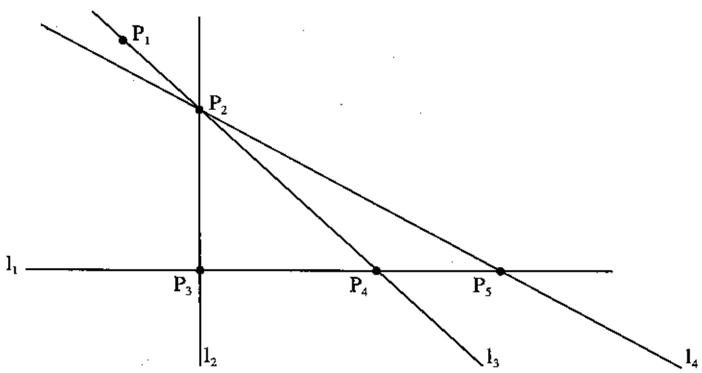
a.



(i) $\exists xRxx$
(ii) $\exists x\exists y(x \neq y \wedge Rxy \wedge Ryx)$
(iii) $\exists x(\neg\exists yRyx \wedge \neg\exists yRxy)$
(iv) $\exists x\exists y(x \neq y \wedge \forall z(\neg\exists wRzw \leftrightarrow (z = x \vee z = y)))$
(v) $\exists x\exists y\exists z(y \neq z \wedge \forall w(Rxw \leftrightarrow (w = y \vee w = z)))$
(vi) $\exists x\exists y(x \neq y \wedge \exists zRxz \wedge \exists zRyz \wedge \forall z(Rxz \leftrightarrow Ryz))$
(vii) $\exists x_1\exists x_2\exists x_3\exists x_4\exists x_5\exists x_6\exists x_7(Rx_1x_2 \wedge Rx_2x_3 \wedge Rx_3x_4 \wedge Rx_4x_5 \wedge Rx_5x_6 \wedge Rx_6x_7)$
(viii) $\forall x_1\forall x_2\forall x_3((x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3 \wedge \neg Rx_1x_2 \wedge \neg Rx_2x_1 \wedge \neg Rx_1x_3 \wedge \neg Rx_3x_1 \wedge \neg Rx_2x_3 \wedge \neg Rx_3x_2) \rightarrow \neg\exists y(Rx_1y \wedge Rx_2y \wedge Rx_3y))$
(ix) $\forall x\forall y((x \neq y \wedge \neg Rxy \wedge \neg Ryx) \rightarrow \neg\exists z\exists w(z \neq w \wedge \neg Rzw \wedge \neg Rwz \wedge Rxz \wedge Ryz \wedge Rxw \wedge Ryw))$

(b) Consider the model given in figure b. Its domain consists of the points and the lines in the figure. Hence $D = \{P_1, P_2, P_3, P_4, P_5, l_1, l_2, l_3, l_4\}$. The language contains the unary predicate letter P with the points as its interpretation; the unary predicate letter L with the lines as its interpretation; the binary predicate letter O with, as its interpretation, *lie on* (key: Oxy: the point x lies on the line y); and the ternary predicate letter B with, as its interpretation, *lie between* (key: Bxyz: y lies between x and z, i.e., $I(B) = \{\langle P_1, P_2, P_4\rangle, \langle P_4, P_2, P_1\rangle, \langle P_3, P_4, P_5\rangle, \langle P_5, P_4, P_3\rangle\}$).

b.



As in (a), determine the truth value in the model of the sentences below on the basis of their meaning.

(i) $\forall x(Lx \leftrightarrow \exists yOyx)$
(ii) $\forall x\forall y((Lx \wedge Ly) \rightarrow \exists z(Pz \wedge Ozx \wedge Ozy))$
(iii) $\forall x\forall y((Px \wedge Py) \rightarrow \exists z(Lz \wedge Oxz \wedge Oyz))$
(iv) $\exists x\exists y\forall z(Pz \rightarrow (Ozx \vee Ozy))$
(v) $\exists x\exists y_1\exists y_2\exists y_3\,(y_1 \neq y_2 \wedge y_1 \neq y_3 \wedge y_2 \neq y_3 \wedge \forall z((Pz \wedge Ozx) \leftrightarrow (z = y_1 \vee z = y_2 \vee z = y_3)))$
(vi) $\exists x_1\exists y_1\exists x_2\exists y_2(x_1 \neq x_2 \wedge y_1 \neq y_2 \wedge Ox_1y_1 \wedge Ox_1y_2 \wedge Ox_2y_1 \wedge Ox_2y_2)$
(vii) $\forall x\forall y\forall z(Bxyz \rightarrow Bzyx)$
(viii) $\forall x(Lx \rightarrow \exists y\exists z\exists w(Oyx \wedge Ozx \wedge Owx \wedge Byzw))$
(ix) $\forall x\forall y\forall z((x \neq y \wedge x \neq z \wedge y \neq z \wedge \exists w(Oxw \wedge Oyw \wedge Ozw)) \rightarrow (Bxyz \vee Byzx \vee Bzxy))$
(x) $\forall x(\exists y_1\exists y_2(y_1 \neq y_2 \wedge Oxy_1 \wedge Oxy_2) \rightarrow \exists z_1\exists z_2Bz_1xz_2)$

Exercise 13

There is actually a great deal of flexibility in the semantic schema presented here. Although the main emphasis has been on the case where a formula ϕ is interpreted in a given model ('verification'), there are various other modes of employment. For instance, given only some formula ϕ, one may ask for all models where it holds. Or conversely, given some model **M**, one may try to describe exactly those formulas that are true in it. And given some formulas and some nonlinguistic situation, one may even try to set up an interpretation function that makes the formulas true in that situation: this happens when we learn a foreign language. For instance, given a domain of three objects, what different interpretation functions will verify the following formula?

$\forall x\forall y(Rxy \vee Ryx \vee x = y) \wedge \forall x\forall y(Rxy \rightarrow \neg Ryx)$

Exercise 14 ◇

Formulas can have different numbers of models of different sizes. Show that

(i) $\exists x\forall y(Rxy \leftrightarrow \neg Ryy)$ has no models.

(ii) $\forall x\forall y(Rxy \vee Ryx \vee x = y) \wedge \forall x\forall y(Rxy \leftrightarrow \neg(Px \leftrightarrow Py))$ has only finite models of size at most two.
(iii) $\forall x\exists yRxy \wedge \forall x\neg Rxx \wedge \exists x\forall y\neg Ryx \wedge \forall x\forall y\forall z((Rxz \wedge Ryz) \rightarrow x = y)$ has only models with infinite domains.

Exercise 15 ◇

Describe all models with finite domains of 1, 2, 3, . . . objects for the conjunction of the following formulas:

$\forall x\neg Rxx$
$\forall x\exists yRxy$
$\forall x\forall y\forall z((Rxy \wedge Rxz) \rightarrow y = z)$
$\forall x\forall y\forall z((Rxz \wedge Ryz) \rightarrow x = y)$

Exercise 16 ◇

In natural language (and also in science), discourse often has changing domains. Therefore it is interesting to study what happens to the truth of formulas in a model when that model undergoes some transformation. For instance, in semantics, a formula is sometimes called *persistent* when its truth is not affected by enlarging the models with new objects. Which of the following formulas are generally persistent?

(i) $\exists xPx$
(ii) $\forall xPx$
(iii) $\exists x\forall yRxy$
(iv) $\neg\forall x\forall yRxy$

3.8 Some Properties of Relations

In § 3.1 we stated that if the first of three objects is larger than the second, and the second is in turn larger than the third, then the first object must also be larger than the third; and this fact can be expressed in predicate logic. It can, for example, be expressed by the formula $\forall x\forall y\forall z((Lxy \wedge Lyz) \rightarrow Lxz)$, for in any model **M** in which L is interpreted as the relation *larger than*, it will be the case that $V_M(\forall x\forall y\forall z((Lxy \wedge Lyz) \rightarrow Lxz)) = 1$. It follows directly from the truth definition that this is true just in case, for any d_1, d_2, $d_3 \in D$, if $\langle d_1, d_2\rangle \in D$ and $\langle d_2, d_3\rangle \in D$, then $\langle d_1, d_3\rangle \in D$. A relation I(R) in a model **M** is said to be *transitive* if $\forall x\forall y\forall z((Rxy \wedge Ryz) \rightarrow Rxz)$ is true in **M.** So *larger than* is a transitive relation. The relations *just as large as* and = are other examples of transitive relations. For the sentence $\forall x\forall y\forall z((x = y \wedge y = z) \rightarrow x = z)$ is true in every model.

There is also a difference between *just as large as* (translated as H) and =, on the one hand, and *larger than*, on the other: $\forall x\forall y(Hxy \rightarrow Hyx)$ and $\forall x\forall y(x = y \rightarrow y = x)$ are always true, but $\forall x\forall y(Lxy \rightarrow Lyx)$ is never true. Apparently the order of the elements doesn't matter with *just as large as* and =, but does matter with *larger than*. If $\forall x\forall y(Rxy \rightarrow Ryx)$ is true in a model

M, then we say that I(R) is *symmetric(al)* in **M;** so *just as large as* and = are symmetric relations. If $\forall x\forall y(Rxy \to \neg Ryx)$ is true in a model **M,** then we say that I(R) is *asymmetric(al)* in **M.** *Larger than* is an *asymmetric(al)* relation. Not every relation is either symmetric or asymmetric; the *brother of* relation, for example, is neither: if John is one of Robin's brothers, then Robin may or may not be one of John's brothers, depending on whether Robin is male or female.

A relation I(R) is said to be *reflexive* in **M,** just in case $\forall xRxx$ is true in **M.** The relations *just as large as* and =, once again, are reflexive, since everything is just as large as and equal to itself. On the other hand, nothing is larger than itself; we say that I(R) is *irreflexive* in **M** just in case $\forall x\neg Rxx$ is true in **M,** so that *larger than* is an irreflexive relation.

There are other comparatives in natural language which are both asymmetrical and irreflexive, such as *thinner than* and *happier than,* for example. Other comparatives, like *at least as large as* and *at least as happy as,* are neither symmetrical nor asymmetrical, though they are both reflexive and transitive. The relations $>$ and $\geqslant$ between numbers are analogous to *larger than* and *at least as large as:* $>$ is transitive, asymmetrical, and irreflexive, whereas $\geqslant$ is transitive and neither symmetrical nor asymmetrical.

But $\geqslant$ has one additional property: if I(R) is $\geqslant$, then $\forall x\forall y((Rxy \wedge Ryx) \to x = y)$ is always true. Relations like this are said to be *antisymmetric(al). At least as large as* is not antisymmetric, since John and Robin can each be just as large as the other without being the same person.

Finally, we say that a relation I(R) is *connected* in a model **M** just in case $\forall x\forall y(Rxy \vee x = y \vee Ryx)$ is true in **M.** The relations $>$ and $\geqslant$ are connected. The relations $\geqslant$, *at least as large as,* and $>$ are too, but note that *larger than* is not connected.

These properties of relations can be illustrated as follows. Just as in example 2 in §3.6.2, we choose the points in a figure as the domain of a model and we interpret R such that $\langle d, e\rangle \in I(R)$ iff there is an arrow pointing from d to e. Then (105) gives what all of the different properties mean for the particular relation I(R). For ease of reference we also include the defining predicate logical formula.

(105)

I(R) is symmetric.	$\forall x\forall y(Rxy \to Ryx)$	If an arrow connects two points in one direction, then there is an arrow in the other direction too.
I(R) is asymmetric.	$\forall x\forall y(Rxy \to \neg Ryx)$	Arrows do not go back and forth between points.
I(R) is reflexive.	$\forall xRxx$	Every point has an arrow pointing to itself.
I(R) is irreflexive.	$\forall x\neg Rxx$	No point has an arrow pointing to itself.
I(R) is transitive.	$\forall x\forall y\forall z((Rxy \wedge Ryz) \to Rxz)$	If an arrow points from the first of three points to the second, and an arrow points from the second to the third, then there is an arrow pointing from the first to the third.
I(R) is antisymmetric.	$\forall x\forall y((Rxy \wedge Ryx) \to x = y)$	Arrows do not go back and forth between different points.
I(R) is connected.	$\forall x\forall y(Rxy \vee x = y \vee Ryx)$	Any two different points are connected by at least one arrow.

The last two cases in (105) will be clearer if it is realized that antisymmetry can just as well be expressed by $\forall x\forall y(x \neq y \to (Rxy \to \neg Ryx))$, and connectedness by $\forall x\forall y(x \neq y \to (Rxy \vee Ryx))$. The difference between asymmetry and antisymmetry is that asymmetry implies irreflexivity. This is apparent from the formulation given above: if an arrow were to run from one point to itself, then there would automatically be an arrow running 'back'. In formulas: if $\forall x\forall y(Rxy \to \neg Ryx)$ is true in a model, then $\forall x(Rxx \to \neg Rxx)$ is true too. And this last formula is equivalent to $\forall x\neg Rxx$.

Finally, we observe that all the properties mentioned here make sense for arbitrary binary relations, whether they serve as the interpretation of some binary predicate constant or not. With respect to natural language expressions of relations, a word of caution is in order. The exact properties of a relation in natural language depend on the domain of discourse. Thus *brother of* is neither symmetric nor asymmetric in the set of all people, but is symmetric in the set of all male people. And *smaller than* is connected in the set of all natural numbers but not in the set of all people.

Exercise 17

Investigate the following relations as to their reflexivity, irreflexivity, symmetry, asymmetry, antisymmetry, transitivity, and connectedness:

(i) the grandfather relation in the set of all people;
(ii) the ancestor relation in the set of all people;
(iii) the relation *smaller than* in the set of all people;
(iv) the relation *as tall as* in the set of all people;
(v) the relation *exactly one year younger than* in the set of all people;
(vi) the relation *north of* in the set of all sites on earth;
(vii) the relation *smaller than* in the set of all natural numbers;
(viii) the relation *divisible by* in the set of all natural numbers;
(ix) the relation *differs from* in the set of all natural numbers.

Exercise 18

There are certain natural *operations* on binary relations that transform them into other relations. One example is *negation*, which turns a relation H into its complement, $-$H; another is *converse*, which turns a relation H into $\breve{H} = \{\langle x, y\rangle \mid \langle y, x\rangle \in H\}$. Such operations may or may not preserve the special properties of the relations defined above. Which of the following are preserved under negation or converse?

(i) reflexivity
(ii) symmetry
(iii) transitivity

3.9 Function symbols

A function is a special kind of relation. A function r from D into D can always be represented as a relation R defined as follows: $\langle d, e\rangle \in I(R)$ iff $r(d) = e$. And then $\forall x\exists!yRxy$ is true in the model in question. Conversely, if $\forall x\exists!yRxy$ is true in some model for a binary relation R, then we can define a function r which assigns the unique e such that $\langle d, e\rangle \in I(R)$ to any domain element d. So unary functions can be represented as binary relations, n-ary functions as $n+1$-ary relations. For example, the sum function + can be represented by means of a ternary predicate letter P. Given a model with the natural numbers as its domain, we then define I(P) such that $\langle n_1, n_2, n_3\rangle \in I(P)$ iff $n_1 + n_2 = n_3$. Then, for example, $\langle 2, 2, 4\rangle \in I(P)$ and $\langle 2, 2, 5\rangle \notin I(P)$.

The commutativity of addition then amounts to the truth of $\forall x\forall y\forall z(Pxyz \rightarrow Pyxz)$ in the model. Associativity is more difficult to express. But it can be done; it is done by the following sentence: $\forall x\forall y\forall z\forall w_1\forall w_2\forall w_3((Pxyw_1 \wedge Pw_1zw_2 \wedge Pyzw_3) \rightarrow Pxw_3w_2)$. This is represented graphically in figure (106):

(106) $(x + y) + z = x + (y + z)$

$$\begin{array}{ccc} w_1 + z & \qquad & x + w_3 \\ w_2 & \qquad & w_2 \end{array}$$

It is clear that expressing the properties of functions by means of predicate letters leads to formulas which are not very readable. It is for this reason that special symbols which are always interpreted as functions are often included in predicate languages, the *function symbols*.

Function symbols, like predicate letters, come in all kinds of arities: they may be unary, binary, ternary, and so forth. But whereas an n-ary predicate letter followed by n terms forms an atomic formula, an n-ary function symbol followed by n terms forms another term, an expression which refers to some entity in the domain of any model in which it is interpreted, just as constants and variables do. Such expressions can therefore play the same roles as constants and variables, appearing in just the same positions in formulas as constants and variables do.

If the addition function on natural numbers is represented by means of the binary function symbol p, then the commutativity and associativity of addition are conveniently expressed by:

(107) $\forall x\forall y(p(x, y) = p(y, x))$

(108) $\forall x\forall y\forall z(p(p(x, y), z) = p(x, p(y, z)))$

So now we have not only simple terms like constants and variables but also composite terms which can be constructed by prefixing function symbols to the right number of other terms. For example, the expressions p(x, y), p(y, x), p(p(x, y), z), p(x, p(y, z)), and p(y, z) appearing in (107) and (108) are all composite terms. Composite terms are built up from simpler parts in much the same way as composite formulas, so they too can be given an inductive definition:

Definition 11

(i) If t is a variable or constant in L, then t is a term in L.
(ii) If f is an n-ary function symbol in L and $t_1, \ldots, t_n$ are terms in L, then $f(t_1, \ldots, t_n)$ is a term in L too.

The definition of the formulas of L does not have to be adapted. Their semantics becomes slightly more complicated, since we now have to begin by interpreting terms. Naturally enough, we interpret an n-ary function symbol f as some n-ary function I(f) which maps D^n, the set of all n-tuples of elements of the domain D of some model we are working with, into D. Variables and constants are interpreted just as before, and the interpretations of composite terms can be calculated by means of the clause:

$$[\![f(t_1, \ldots, t_n)]\!]_{M,g} = (I(f))(\langle[\![t_1]\!]_{M,g}, \ldots, [\![t_n]\!]_{M,g}\rangle).$$

So now we can see why the idea behind definition 8 is useful: it makes generalizing so much easier. In approach A, by the way, we only have to consider terms without variables, in which case $[\![t]\!]_M$ can be defined instead of $[\![t]\!]_{M,g}$.

Our account of predicate logic so far has been biased toward *predicates* of and *relations* among individual objects as the logically simple expressions. In this we followed natural language, which has few (if any) basic, i.e., lexical functional expressions. Nevertheless, it should be stressed that in many applications of predicate logic to *mathematics*, functions are the basic notion rather than predicates. (This is true, for instance, in many fields of algebra.) Moreover, at a higher level, there is much functional behavior in natural language too, as we shall see in a later chapter on type theory (see vol. 2).