More about Quantifiers 3

3–1. SOME EXAMPLES OF MULTIPLE QUANTIFICATION

All of the following are sentences of predicate logic:

(1) $(\forall x)(\forall y)Lxy$
(2) $(\exists x)(\exists y)Lxy$
(3) $(\exists x)(\forall y)Lxy$
(4) $(\exists x)(\forall y)Lyx$
(5) $(\forall x)(\exists y)Lxy$
(6) $(\forall x)(\exists y)Lyx$

Let's suppose that 'L' stands for the relation of loving. What do these sentences mean?

Sentence (1) says that everybody loves everybody (including themselves). (2) says that somebody loves somebody. (The somebody can be oneself or someone else.) Sentences (3) to (6) are a little more tricky. (3) says that there is one person who is such that he or she loves everyone. (There is one person who is such that, for all persons, the first loves the second—think of God as an example.) We get (4) from (3) by reversing the order of the 'x' and 'y' as arguments of 'L'. As a result, (4) says that there is one person who is loved by everyone. Notice what a big difference the order of the 'x' and 'y' makes.

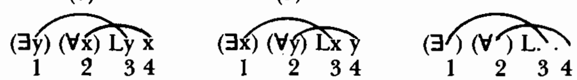
Next, (5) says that everyone loves someone: Every person is such that there is one person such that the first loves the second. In a world in which (5) is true, each person has an object of their affection. Finally we get (6) out of (5) by again reversing the order of 'x' and 'y'. As a result, (6) says that everyone is loved by someone or other. In a world in which (6) is true no one goes unloved. But (6) says something significantly weaker than (3). (3) say that there is **one** person who loves everyone. (6) says that each person gets loved, but Adam might be loved by one person, Eve by another, and so on.

Can we say still other things by further switching around the order of the quantifiers and arguments in sentences (3) to (6)? For example, switching the order of the quantifiers in (6) gives

(7) $(\exists y)(\forall x)Lyx$

Strictly speaking, (7) is a new sentence, but it does not say anything new because it is logically equivalent to (3). It is important to see why this is so:

(7) (3)



These diagrams will help you to see that (7) and (3) say exactly the same thing. The point is that there is nothing special about the variable 'x' or the variable 'y'. Either one can do the job of the other. What matters is the pattern of quantifiers and variables. These diagrams show that the pattern is the same. All that counts is that the variable marked at position 1 in the existential quantifier is tied to, or, in logicians' terminology, *Binds* the variable at position 3; and the variable at position 2 in the universal quantifier binds the variable at position 4. Indeed, we could do without the variables altogether and indicate what we want with the third diagram. This diagram gives the pattern of variable binding which (7) and (3) share.

3–2. QUANTIFIER SCOPE, BOUND VARIABLES, AND FREE VARIABLES

In the last example we saw that the variable at 3 is bound by the quantifier at 1 and the variable at 4 is bound by the quantifier at 2. This case contrasts with that of a variable which is not bound by any quantifier, for example

(8) Lxa ⊃ (∃x)Lxb
1 2 3

(9) (∃x)Lxb ⊃ Lxa
1 2 3

In (8), the occurrence of 'x' at 3 is bound by the quantifier at 2. However, the occurrence of 'x' at 1 is not bound by any quantifier. Logicians say that the occurrence of 'x' at 1 is *Free*. In (9), the occurrence of 'x' at 3 is free because the quantifier at 1 binds only variables in the shortest full sentence which follows it. Logicians call the shortest full sentence following a quantifier the quantifier's *Scope*. In (9), the 'x' at 3 is not in the scope of the quantifier at 1. Consequently, the quantifier does not bind 'x' at 3.

All the important ideas of this section have now been presented. We need these ideas to understand clearly how to apply the methods of derivations and truth trees when quantifiers get stacked on top of each other. All we need do to complete the job is to give the ideas an exact statement and make sure you know how to apply them in more complicated situations.

Everything can be stated in terms of the simple idea of scope. A quantifier is a connective. We use a quantifier to build longer sentences out of shorter ones. In building up sentences, a quantifier works just like the negation sign: It apples to the shortest full sentence which follows it. This shortest full following sentence is the quantifier's scope:

> The *Scope* of a quantifier is the shortest full sentence which follows it. Everything inside this shortest full following sentence is said to be in the scope of the quantifier.

We can now define 'bound' and 'free' in terms of scope:

> A variable, **u**, is *Bound* just in case it occurs in the scope of a quantifier, (∀**u**) or (∃**u**).
>
> A variable, **u**, is *Free* just in case it is not bound; that is, just in case it does not occur in the scope of any quantifier, (∀**u**) or (∃**u**).

Clearly, a variable gets bound only by using a quantifier expressed with the same variable. 'x' can never be bound by quantifiers such as '(∀y)' or '(∃z)'.

Occasionally, students ask about the variables that occur within the quantifiers—the 'x' in '(∃x)' and in '(∀x)'. Are they bound? Are they free? The answer to this question is merely a matter of convention on which nothing important turns. I think the most sensible thing to say is that the variable within a quantifier is part of the quantifier symbol and so does not count as either bound or free. Only variables outside a quantifier can be either bound or free. Some logicians prefer to deal with this question by defining the scope of a quantifier to include the quantifier itself as well as the shortest full sentence which follows it. On this convention one would say that a variable within a quantifier always binds itself.

These definitions leave one fine point unclear. What happens if the variable **u** is in the scope of **two** quantifiers that use **u**? For example, consider

(10) (∃x)[(∀x)Lxa ⊃ Lxb]
1 2 3 4

The occurrence of 'x' at 3 is in the scope of both the 'x' quantifiers. Which quantifier binds 'x' at 3?

To get straight about this, think through how we build (10) up from atomic constituents. We start with the atomic sentences 'Lxa' and 'Lxb'. Because atomic sentences have no quantifiers, 'x' is free in both of these atomic sentences. Next we apply '(∀x)' to 'Lxa', forming '(∀x)Lxa', which we use as the antecedent in the conditional

(11) (∀x)Lxa ⊃ Lxb
2 3 4

In (11), the occurrence of 'x' at 3 is bound by the quantifier at 2. The occurrence of 'x' at 4 is free in (11).

Finally, we can clearly describe the effect of '(∃x)' when we apply it to (11). '(∃x)' binds just the **free** occurrences of 'x' in (11). The occurrence at 4 is free and so gets bound by the new quantifier. The occurrence at 3 is already bound, so the new quantifier can't touch it. The following diagram describes the overall effect:

(10) (∃x)[(∀x)Lxa ⊃ Lxb]
1 2 3 4

First, the occurrence at 3 is bound by the quantifier at 2. Then the occurrence at 4 is bound by the quantifier at 1. The job being done by the 2–3 link is completely independent of the job being done by the 1–4 link.

Let's give a general statement to the facts we have uncovered:

> A quantifier (∀**u**) or (∃**u**) binds all and only all **free** occurrences of **u** in its scope. Such a quantifier does not bind an occurrence of **u** in its scope which is already bound by some other quantifier in its scope.

We can make any given case even clearer by using different variables where we have occurrences of a variable bound by different quantifiers. So, for example, (10) is equivalent to

(12) (∃x)[(∀z)Lza ⊃ Lxb]

In (12), there can be no confusion about which quantifier binds which variable—we keep track of everything by using different variables. Why, then, didn't we just resolve to use different variables from the beginning and save ourselves a lot of trouble? We could have done that, but then the definition of the sentences of predicate logic would have been much more complicated. Either way, we have work to do. Besides, the formulation I have presented here is the one traditionally used by logicians and so the one you will need to know if you study more logic.

Let's look at another, slightly more complicated, example to make sure you have put this all together. Draw in the lines which show which quantifier binds which variable in the following:

(13) (∃x)[(∃x)(Bx v Lxa) ⊃ (Bx & Lxb)]

If you are having trouble, think through how (13) gets built up from its parts. In

(14) (∃x)(Bx v Lxa) ⊃ (Bx & Lxb)
 2 3 4 5 6

the quantifier at 2 applies only to the shortest full sentence which follows it, which ends before the '⊃'. So the occurrences of 'x' at 3 and 4 are both bound by the quantifier at 2. The two occurrences of 'x' at 5 and 6 are not in the scope of a quantifier and are both free. So when we apply the second '(∃x)' to all of (14), the new '(∃x)' binds only the 'x's which are still free in (14), namely, the 'x's which occur at 5 and 6. In sum, the pattern of binding is

(13) (∃x)[(∃x)(Bx v Lxa) ⊃ (Bx & Lxb)]
 1 2 3 4 5 6

We can make this pattern even clearer by writing the sentence equivalent to (13):

(15) (∃x)[(∃z)(Bz v Lza) ⊃ (Bx & Lxb)]

In practice, of course, it is much better to use sentences such as (15) and (12) instead of the equivalent (13) and (10), which are more difficult to interpret.

EXERCISES

3–1. In the following sentences draw link lines to show which quantifiers bind which variables and say which occurrences of the variables are bound and which are free:

a) Lzz
 12

b) (∀y)(∀z)Lzy
 12

c) (∀z)(Bz ⊃ Lxz)
 1 23

d) (∃x)[Lxz & (∀y)(Lxy v Lzx)]
 12 34 56

e) (∀x)(Lax & Bx) ≡ (Lxx ⊃ (∃x)Bx)
 1 2 34 5

f) (∀x)[Lyx ⊃ (Bx ⊃ (∃x)Lyx)]
 12 3 45

3–3. CORRECT DEFINITIONS OF SUBSTITUTION INSTANCE AND TRUTH IN AN INTERPRETATION

In chapter 2 I gave an incorrect definition of 'substitution instance.' I said that we get the substitution instance of (∀**u**) (. . . **u** . . .) with **s** substituted for **u** by simply dropping the initial (**u**) and writing in **s** wherever we find **u** in (. . . **u** . . .). This is correct as long as neither a second (∀**u**) nor a (∃**u**) occurs within the scope of the initial (∀**u**), that is, within the sentence (. . . **u** . . .). Since I used only this kind of simple sentence in chapter 2, there we could get away with the simple but incorrect defintion. But now we must correct our definition so that it will apply to any sentence. Before reading on, can you see how multiple quantification can make trouble for the simple definition of substitution instance, and can you see how to state the definition correctly?

To correct the definition of substitution instance, all we have to do is to add the qualification that the substituted occurrences of the variable be **free**:

> For any universally quantified sentence (∀**u**) (. . . **u** . . .), the *Substitution Instance* of the sentence, with the name **s** substituted for the variable **u**, is (. . . **s** . . .), the sentence formed by dropping the initial universal quantifier and writing **s** for all **free** occurrences of **u** in (. . . **u** . . .).

> For any existentially quantified sentence (∃**u**) (. . . **u** . . .), the *Substitution Instance* of the sentence, with the name **s** substituted for the variable **u**, is (. . . **s** . . .), the sentence formed by dropping the initial existential quantifier and writing **s** for all **free** occurrences of **u** in (. . . **u** . . .).

For example, look back at (13). Its substitution instance with 'c' substituted for 'x' is

(16) (∃x)(Bx v Lxa) ⊃ (Bc & Lcb)
 2 3 4 5 6

The occurrences of 'x' at 3 and 4 are not free in the sentence which results from (13) by dropping the initial quantifier. So we don't substitute 'c' for 'x' at 3 and 4. We substitute 'c' only at the free occurrences, which were at 5 and 6.

Can you see why, when we form substitution instances, we pay attention only to the occurrences which are free after dropping the outermost quantifier? The occurrences at 3 and 4, bound by the '(∃x)' quantifier at 2, have nothing to do with the outermost quantification. When forming substitution instances of a quantified sentence, we are concerned only with the outermost quantifier and the occurrences which it binds.

To help make this clear, once again consider (15), which is equivalent to (13). In (15), we have no temptation to substitute 'c' for 'z' when forming the 'c'-substitution instance for the sentence at a whole. (15) says that there is some x such that so on and so forth about x. In making this true for some specific x, say c, we do not touch the occurrences of 'z'. The internal 'z'-quantified sentence is just part of the so on and so forth which is asserted about x in the quantified form of the sentence, that is, in (15). So the internal 'z'-quantified sentence is just part of the so on and so forth which is asserted about c in the substitution instance of the sentence. Finally, (13) says exactly what (15) says. So we treat (13) in the same way.

Now let's straighten out the definition of truth of a sentence in an interpretation. Can you guess what the problem is with our old definition? I'll give you a clue. Try to determine the truth value of 'Lxe' in the interpretation of figure 2–1. You can't do it! Nothing in our definition of an interpretation gives us a truth value for an atomic sentence with a free variable. An interpretation only gives truth values for atomic sentences which use no variables. You will have just as much trouble trying to determine the truth value of '(∀x)Lxy' in any interpretation. A substitution instance of '(∀x)Lxy' will still have the free variable 'y', and no interpretation will assign such a substitution instance a truth value.

Two technical terms (mentioned in passing in chapter 1) will help us in talking about our new problem:

> A sentence with one or more free variables is called an *Open Sentence*.
>
> A sentence which is not open (i.e., a sentence with no free variables) is called a *Closed Sentence*.

In a nutshell, our problem is that our definitions of truth in an interpretation do not specify truth values for open sentences. Some logicians deal with this problem by treating all free variables in an open sentence as if they were universally quantified. Others do what I will do here: We simply say that open sentences have no truth value.

If you think about it, this is really very natural. What, anyway, is the truth value of the English "sentence" 'He is blond.', when nothing has been said or done to give you even a clue as to who 'he' refers to? In such a situation you can't assign any truth value to 'He is blond.' 'He is blond.' functions syntactically as a sentence—it has the form of a sentence. But

there is still something very problematic about it. In predicate logic we allow such open sentences to function syntactically as sentences. Doing this is very useful in making clear how longer sentences get built up from shorter ones. But open sentences never get assigned a truth value, and in this way they fail to be full-fledged sentences of predicate logic.

It may seem that I am dealing with the problem of no truth value for open sentences by simply ignoring the problem. In fact, as long as we acknowledge up-front that this is what we are doing, saying that open sentences have no truth value is a completely adequate way to proceed.

We have only one small detail to take care of. As I stated the definitions of truth of quantified sentences in an interpretation, the definitions were said to apply to any quantified sentences. But they apply only to **closed** sentences. So we must write in this restriction:

> A universally quantified closed sentence is true in an interpretation just in case all of the sentence's substitution instances, formed with names in the interpretation, are true in the interpretation.
>
> An existentially quantified closed sentence is true in an interpretation just in case some (i.e., one or more) of the sentence's substitution instances, formed with names in the interpretation, are true in the interpretation.

These two definitions, together with the rules of valuation given in chapters 1 and 4 of volume I for the sentence logic connectives, specify a truth value for any **closed** sentence in any of our interpretations.

You may remember that in chapter 1 in volume I we agreed that sentences of logic would always be true or false. Sticking by that agreement now means stipulating that only the closed sentences of predicate logic are real sentences. As I mentioned in chapter 1 in this volume, some logicians use the phrase *Formulas*, or *Propositional Functions* for predicate logic open sentences, to make the distinction clear. I prefer to stick with the word 'sentence' for both open and closed sentences, both to keep terminology to a minimum and to help us keep in mind how longer (open and closed) sentences get built up from shorter (open and closed) sentences. But you must keep in mind that only the closed sentences are full-fledged sentences with truth values.

EXERCISES

3–2. Write a substitution instance using 'a' for each of the following sentences:

a) (∀y)(∃x)Lxy b) (∃z)[(∀x)Bx v Bz]
c) (∃x)[Bx ≡ (∀x)(Lax v Bx)]
d) (∀y)[(∃x)(Bx ⊃ By) & (∀x)(By ⊃ Bx)]
e) (∀y){(∃x)Bx v [(∃y)By ⊃ Lyy]}

f) (∀y)(∃x)[(Rxy ⊃ Dy) ⊃ Ryx]
g) (∀x)(∀y)(∀z){[Sxy v (Hz ⊃ Lxz)] ≡ (Scx & Hy)}
h) (∃x)(∀z){(Pxa ⊃ Kz) & (∃y)[(Pxy v Kc) & Pxx]}
i) (∃z)(∀y){[(∃x)Mzx v (∃x)(Mxy ⊃ Myz)] & (∃x)Mzx}
j) (∀x){[(∀x)Rxa ⊃ Rxb] v [(∃x)(Rcx v Rxa) ⊃ Rxx]}

3–3. If **u** does not occur free in **X**, the quantifiers (∀**u**) and (∃**u**) are said to occur *Vacuously* in (∀**u**)**X** and (∃**u**)**X**. Vacuous quantifiers have no effect. Let's restrict our attention to the special case in which **X** is closed, so that it has a truth value in any of its interpretations. The problem I want to raise is how to apply the definitions for interpreting quantifiers to vacuously occurring quantifiers. Because truth of a quantified sentence is defined in terms of substitution instances of (∀**u**)**X** and (∃**u**)**X**, when **u** does not occur free in **X**, we most naturally treat this vacuous case by saying that **X** counts as a substitution instance of (∀**u**)(**X**) and (∃**u**)(**X**). (If you look back at my definitions of 'substitution instance', you will see that they really say this if by 'for all free occurrences of **u**' you understand 'for no occurrences of **u**' when **u** does not occur free in **X** at all. In any case, this is the way you should understand these definitions when **u** does not occur free in **X**.) With this understanding, show that (∀**u**)**X**, (∃**u**)**X**, and **X** all have the same truth value in any interpretation of **X**.

3–4. a) As I have defined interpretation, every object in an interpretation has a name. Explain why this chapter's definitions of truth of existentially and universally quantified sentences would not work as intended if interpretations were allowed to have unnamed objects.

b) Explain why one might want to consider interpretations with unnamed objects.

In part II we will consider interpretations with unnamed objects and revise the definitions of truth of quantified sentences accordingly.

3–4. SOME LOGICAL EQUIVALENCES

The idea of logical equivalence transfers from sentence logic to predicate logic in the obvious way. In sentence logic two sentences are logically equivalent if and only if in all possible cases the sentences have the same truth value, where a possible case is just a line of the truth table for the sentence, that is, an assignment of truth values to sentence letters. All we have to do is to redescribe possible cases as interpretations:

> Two closed predicate logic sentences are *Logically Equivalent* if and only if in each of their interpretations the two sentences are either both true or both false.

Notice that I have stated the definition only for closed sentences. Indeed, the definition would not make any sense for open sentences because open sentences don't have truth values in interpretations. Nonetheless, one can extend the idea of logical equivalence to apply to open sentences. That's a good thing, because otherwise the law of substitution of logical equivalents would break down in predicate logic. We won't be making much use of these further ideas in this book, so I won't hold things up with the details. But you might amuse yourself by trying to extend the definition of logical equivalence to open sentences in a way which will make the law of substitution of logical equivalents work in just the way you would expect.

Let us immediately take note of two equivalences which will prove very useful later on. By way of example, consider the sentence, 'No one loves Eve', which we transcribe as '~(∃x)Lxe', that is, as 'It is not the case that someone loves Eve'. How could this unromantic situation arise? Only if **everyone didn't** love Eve. In fact, saying '~(∃x)Lxe' comes to the same thing as saying '(∀x)~Lxe'. If there is not a single person who does love Eve, then it has to be that everyone does not love Eve. And conversely, if positively everyone does not love Eve, then not even one person does love Eve.

There is nothing special about the example I have chosen. If our sentence is of the form ~(∃**u**)(. . . **u** . . .), this says that there is not a single **u** such that so on and so forth about **u**. But this comes to the same as saying about each and every **u** that so on and so forth is not true about **u**, that is, that (∀**u**)~(. . . **u** . . .).

We can easily prove the equivalence of ~(∃**u**)(. . . **u** . . .) and (∀**u**)~(. . . **u** . . .) by appealing to De Morgan's laws. We have to prove that these two sentences have the same truth value in each and every interpretation. In any one interpretation, ~(∃**u**)(. . . **u** . . .) is true just in case the negation of the disjunction of the instances

~[(. . . a . . .) v (. . . b . . .) v (. . . c . . .) v . . .]

is true in the interpretation, where we have included in the disjunction all the instances formed using names which name things in the interpretation. By De Morgan's laws, this is equivalent to the conjunction of the negation of the instances

~(. . . a . . .) & ~(. . . b . . .) & ~(. . . c . . .) & . . .

which is true in the interpretation just in case (∀**u**)~(. . . **u** . . .) is true in the interpretation. Because this is true in all interpretations, we see that

> Rule ~∃: ~(∃**u**)(. . . **u** . . .) is logically equivalent to (∀**u**)~(. . . **u** . . .).

Now consider the sentence 'Not everyone loves Eve,' which we transcribe as '~(∀x)Lxe'. If not everyone loves Eve, then there must be someone who does not love Eve. And if there is someone who does not love Eve, then not everyone loves Eve. So '~(∀x)Lxe' is logically equivalent to '(∃x)~Lxe'.

Pretty clearly, again there is nothing special about the example. ~(∀**u**)(. . . **u** . . .) is logically equivalent to (∃**u**)~(. . . **u** . . .). If it is not the case that, for all **u**, so on and so forth about **u**, then there must be some **u** such that not so on and so forth about **u**. And, conversely, if there is some **u** such that not so on and so forth about **u**, then it is not the case that for all **u**, so on and so forth about **u**. In summary

Rule ~∀: ~(∀**u**)(. . . **u** . . .) is logically equivalent to (∃**u**)~(. . . **u** . . .).

You can easily give a proof of this rule by imitating the proof of the rule ~∃. But I will let you write out the new proof as an exercise.

EXERCISES

3–5. a) Give a proof of the rule of logical equivalence, ~∀. Your proof will be very similar to the proof given in the text for the rule ~∃.

b) The proof for the rule ~∃ is flawed! It assumes that all interpretations have finitely many things in their domain. But not all interpretations are finite in this way. (Exercise 2–5 gives an example of an infinite interpretation.) The problem is that the proof tries to talk about the disjunction of all the substitution instances of a quantified sentence. But if an interpretation is infinite, there are infinitely many substitution instances, and no sentence can be infinitely long. Since I instructed you, in part (a) of this problem, to imitate the proof in the text, probably your proof has the same problem as mine.

Your task is to correct this mistake in the proofs. Give informal arguments for the rules ~∃ and ~∀ which take account of the fact that some interpretations have infinitely many things in their domain.

3–6. In the text I defined logical equivalence for closed sentences of predicate logic. However, this definition is not broad enough to enable us to state a sensible law of substitution of logical equivalents for predicate logic. Let me explain the problem with an example. The following two sentences are logically equivalent:

(1) ~(∀x)(∀y)Lxy

(2) (∃x)(∃y)~Lxy

But we cannot prove that (1) and (2) are logically equivalent with the rule ~∀ as I have stated it. Here is the difficulty. The rule ~∀ tells us that (1) is logically equivalent to

(3) (∃x)~(∀y)Lxy

What we would like to say is that ~(∀y)Lxy is logically equivalent to (∃y)~Lxy, again by the rule ~∀. But the rule ~∀ does not license this because I have defined logical equivalence only for closed sentences and '~(∀y)Lxy' and '(∃y)~Lxy' are open sentences. (Strictly speaking, I should have restricted the ~∀ and ~∃ rules to closed sentences. I didn't because I anticipated the results of this exercise.) Since open sentences are never true or false, the idea of logical equivalence for open sentences does not make any sense, at least not on the basis of the definitions I have so far introduced.

Here is your task:

a) Extend the definition of logical equivalence for predicate logic sentences so that it applies to open as well as closed sentences. Do this in such a way that the law of substitution of logical equivalents will be correct when one open sentence is substituted for another when the two open sentences are logically equivalent according to your extended definition.

b) Show that the law of substitution of logical equivalents works when used with open sentences which are logically equivalent according to your extended definition.

CHAPTER SUMMARY EXERCISES

Here are this chapter's important terms. Check your understanding by writing short explanations for each, saving your results in your notebook for reference and review.

a) Bound Variables
b) Free Variables
c) Scope
d) Closed Sentence
e) Open Sentence
f) Truth of a Sentence in an Interpretation
g) Rule ~∃
h) Rule ~∀