More on Natural Deduction for Predicate Logic

6

6–1. MULTIPLE QUANTIFICATION AND HARDER PROBLEMS

In chapter 5 I wanted you to focus on understanding the basic rules for quantifiers. So there I avoided the complications that arise when we have sentences, such as '(∀x)(∀y)(Px & Py)', which stack one quantifier on top of another. Such sentences involve no new principles. It's just a matter of keeping track of the main connective. For example, '(∀x)(∀y)(Px & Qy)' is a universally quantified sentence, with '(∀x)' as the main connective. You practiced forming substitution instances of such sentences in chapter 3. The substitution instance of '(∀x)(∀y)(Px & Qy)' formed with 'a' (a sentence you could write when applying ∀E) is '(∀y)(Pa & Qy)'.

You will see how to deal with such sentences most quickly by just looking at a few examples. So let's write a derivation to establish the validity of

(∀x)(∀y)(Px & Qy)
―――――――――――
(∀x)Px & (∀x)Qx

1	(∀x)(∀y)(Px & Qy)	*P*
2	(∀y)(Pâ & Qy)	1, ∀E
3	Pâ & Qb̂	2, ∀E
4	Pâ	3, &E
5	Qb̂	3, &E
6	(∀x)Px	4, ∀I
7	(∀x)Qx	5, ∀I
8	(∀x)Px & (∀x)Qx	6, 7, &I

In line 2 I applied ∀E by forming the substitution instance of 1 using the name 'a'. Then in line 3 I formed a substitution instance of the universally quantified line 2.

Let's look at an example of multiple existential quantification. The basic ideas are the same. But observe that in order to treat the second existential quantifier, we must start a sub-sub-derivation:

(∃x)(∃y)(Px & Qy)
―――――――――――
(∃x)Px & (∃x)Qx

```
 1 | (∃x)(∃y)(Px & Qy)              P
   |------------------
 2 |  a| (∃y)(Pa & Qy)              A
   |   |--------------
 3 |   |  b| Pa & Qb                A
   |   |   |---------
 4 |   |   | Pa                     3, &E
 5 |   |   | Qb                     3, &E
 6 |   |   | (∃x)Px                 4, ∃I
 7 |   |   | (∃x)Qx                 5, ∃I
 8 |   |   | (∃x)Px & (∃x)Qx        6, 7, &I
   |   |  3
 9 |   | (∃x)Px & (∃x)Qx            2, 3–8, ∃E
   |  2
10 | (∃x)Px & (∃x)Qx                1, 2–9, ∃E
  1
```

In line 2 I wrote down '(∃y)(Pa & Qy)', a substitution instance of line 1, formed with 'a', substituted for 'x', which is the variable in the main connective, '(∃x)', of line 1. Since I plan to appeal to ∃E in application to line 1, I make '(∃y)(Pa & Qy)' the assumption of a subderivation with 'a' an isolated name. I then do the same thing with '(∃y)(Pa & Qy)', but because this is again an existentially quantified sentence to which I will want to apply ∃E, I must make my new substitution instance, 'Pa & Qb', the assumption of a sub-sub-derivation, this time with 'b' the isolated name.

In the previous example, I would have been allowed to use 'a' for the second as well as the first substitution instance, since I was applying ∀E. But, in the present example, when setting up to use two applications of ∃E, I must use a new name in each assumption. To see why, let's review what conditions must be satisfied to correctly apply ∃E to get line 9. I must have an existentially quantified sentence (line 2) and a subderivation (sub-sub-derivation 3), the assumption of which is a substitution instance of the existentially quantified sentence. Furthermore, the name used in forming the substitution instance must be isolated to the subderivation. Thus, in forming line 3 as a substitution instance of line 2, I can't use 'a'. I use the name 'b' instead. The 'a' following 'P' in line 3 does not violate the requirement. 'a' got into the picture when we formed line 2, the substitution instance of line 1, and you will note that 'a' is indeed isolated to subderivation 2, as required, since sub-sub-derivation 3 is **part** of subderivation 2.

Here's another way to see the point. I write line 3 as a substitution instance of line 2. Since I will want to apply ∃E, the name I use must be isolated to subderivation 3. If I tried to use 'a' in forming the substitution instance of line 2, I would have had to put an 'a' (the "isolation flag") to the left of scope line 3. I would then immediately see that I had made a mistake. 'a' as an isolation flag means that 'a' can occur only to the right. But 'a' already occurs to the left, in line 2. Since I use 'b' as my new name in subderivation 3, I use 'b' as the isolation flag there. Then the 'a' in line 3 causes no problem: All occurrences of 'a' are to the right of scope line 2, which is the line flagged by 'a'.

All this is not really as hard to keep track of as it might seem. The scope lines with the names written at the top to the left (the isolation flags) do all the work for you. 'a' can only appear to the right of the scope line on which it occurs as an isolation flag. 'b' can only occur to the right of the scope line on which it occurs as an isolation flag. That's all you need to check.

Make sure you clearly understand the last two examples before continuing. They fully illustrate, in a simple setting, what you need to understand about applying the quantifier rules to multiply quantified sentences.

Once you have digested these examples, let's try a hard problem. The new example also differs from the last two in that it requires repeated use of a quantifier introduction rule instead of repeated use of a quantifier elimination rule. In reading over my derivation you might well be baffled as to how I figured out what to do at each step. Below the problem I explain the informal thinking I used in constructing this derivation, so that you will start to learn how to work such a problem for yourself.

$$\frac{(\forall x)Px \supset (\exists x)Qx}{(\exists x)(\exists y)(Px \supset Qy)}$$

1	(∀x)Px ⊃ (∃x)Qx	*P*
2	~(∃x)(∃y)(Px ⊃ Qy)	*A*
3	~Pa	A
4	~Qb ⊃ ~Pa	3, W
5	Pa ⊃ Qb	4, CP
6	(∃y)(Pa ⊃ Qy)	5, ∃I
7	(∃x)(∃y)(Px ⊃ Qy)	6, ∃I
8	~(∃x)(∃y)(Px ⊃ Qy)	2, *R*
9	Pâ	3–8, RD
10	(∀x)Px	9, ∀I
11	(∃x)Qx	*A*
12	b Qb	*A*
13	Pa ⊃ Qb	12, W
14	(∃y)(Pa ⊃ Qy)	13, ∃I
15	(∃x)(∃y)(Px ⊃ Qy)	14, ∃I
16	(∃x)(∃y)(Px ⊃Qy)	11, 12–15, ∃E
17	~(∃x)(∃y)(Px ⊃ Qy)	2, *R*
18	~(∃x)Qx	11–17, ~*I*
19	(∀x)Px ⊃ (∃x)Qx	1, *R*
20	(∃x)Qx	10, 19, ⊃E
21	(∃x)(∃y)(Px ⊃ Qy)	2–20, RD

My basic strategy is reductio, to assume the opposite of what I want to prove. From this I must get a contraction with the premise. The premise is a conditional, and a conditional is false only if its antecedent is true and its consequent is false. So I set out to contradict the original premise by deriving its antecedent and the negation of its consequent from my new assumption.

To derive (∀x)Px (line 10), the premise's antecedent, I need to derive Pâ. I do this by assuming ~Pa from which I derive line 7, which contradicts line 2. To derive ~(∃x)Qx (line 18), the negation of the premise's consequent, I assume (∃x)Qx (line 11), and derive a contradiction, so that I can use ~I. This proceeds by using ∃E, as you can see in lines 11 to 16.

Now it's your turn to try your hand at the following exercises. The problems start out with ones much easier than the last example—and gradually get harder!

EXERCISES

6–1. Provide derivations to establish the validity of the following argument:

a) $\dfrac{(\exists x)Lxx}{(\exists x)(\exists y)Lxy}$

Note that the argument, $\dfrac{(\forall x)Lxx}{(\forall x)(\forall y)Lyx}$, is invalid. Prove that this argument is invalid by giving a counterexample to it (that is, an interpretation in which the premise is true and the conclusion is false). Explain why you can't get from (∀x)Lxx to (∀x)(∀y)Lxy by using ∀E and ∀I as you can get from (∃x)Lxx to (∃x)(∃y)Lxy by using ∃E and ∃I.

b) $\dfrac{(\forall x)(\forall y)Lxy}{(\forall x)Lxx}$

Note that the argument, $\dfrac{(\exists x)(\exists y)Lxy}{(\exists x)Lxx}$, is invalid. Prove that this argument is invalid by giving a counterexample to it. Explain why you can't get from (∃x)(∃y)Lxy to (∃x)Lxx by using ∃E and ∃I as you can get from (∀x)(∀y)Lxy to (∀x)Lxx by using ∀E and ∀I.

c) $\dfrac{(\forall x)(\forall y)Lxy}{(\forall y)(\forall x)Lxy}$ d) $\dfrac{(\exists x)(\exists y)Lxy}{(\exists y)(\exists x)Lxy}$

e) $\dfrac{(\exists x)(\forall y)Lxy}{(\forall y)(\exists x)Lxy}$

Note that the converse argument, $\dfrac{(\forall y)(\exists x)Lxy}{(\exists x)(\forall y)Lxy}$, is invalid. Prove this by providing a counterexample.

f) $\dfrac{(\forall x)Px \,\&\, (\forall x)Qx}{(\forall x)(\forall y)(Px \,\&\, Qy)}$ g) $\dfrac{(\exists x)Px \,\&\, (\exists x)Qx}{(\exists x)(\exists y)(Px \,\&\, Qy)}$ h) $\dfrac{(\forall x)Px \vee (\forall x)Qx}{(\forall x)(\forall y)(Px \vee Qy)}$

i) $\dfrac{(\exists x)Px \vee (\exists x)Qx}{(\exists x)(\exists y)(Px \vee Qy)}$ j) $\dfrac{(\exists x)(\exists y)(Px \vee Qy)}{(\exists x)Px \vee (\exists x)Qx}$ k) $\dfrac{(\forall x)(\forall y)(Lxy \supset \sim Lxy)}{(\forall x)\sim Lxx}$

l) $\dfrac{(\forall x)(\forall y)(Px \supset Qy)}{(\exists x)Px \supset (\forall x)Qx}$ m) $\dfrac{(\exists x)(\exists y)(Px \supset Qy)}{(\forall x)Px \supset (\exists x)Qx}$ n) $\dfrac{(\exists x)(\forall y)(Px \supset Qy)}{(\forall x)Px \supset (\forall x)Qx}$

o) (∀x)(∃y)(Px ⊃ Qy)
—————————
(∃x)Px ⊃ (∃x)Qx

p) (∀x)Px ⊃ (∀x)Qx
—————————
(∃x)(∀y)(Px ⊃ Qy)

q) (∀x)(∀y)(Px v Qy)
—————————
(∀x)Px v (∀x)Qy

r) (∃x)(∀y)Jxy
(∃y)(∃z))(Hzy & ~Py)
(∀z)(∀w)[(Jzw & ~Pw) ⊃ Gz]
—————————
(∃z)Gz

6–2. SOME DERIVED RULES

Problem 5–7(q) posed a special difficulty:

(∀x)~Fx
—————
~(∃x)Fx

1		(∀x)~Fx	P
2		(∃x)Fx	A
3	a	Fa	A
4		(∀x)~Fx	1, R
5		~Fa	4, ∀E

? ? ?

We would like to apply ~I to derive ~(∃x)Fx. To do this, we need to get a contradiction in subderivation 2. But we can use the assumption of subderivation 2 only by using ∃E, which requires starting subderivation 3, which uses 'a' as an isolated name. We do get a sentence and its negation in subderivation 3, but these sentences use the isolated name 'a', so that we are not allowed to use ∃E to bring them out into subderivation 2 where we need the contradiction. What can we do?

We break this impasse by using the fact that from a contradiction you can prove **anything**. Be sure you understand this general fact before we apply it to resolving our special problem. Suppose that in a derivation you have already derived **X** and ~**X**. Let **Y** be **any** sentence you like. You can then derive **Y**:

.
.
.
X
~**X**

~**Y**	*A*
X	*R*
~**X**	*R*
Y	RD

We can use this general fact to resolve our difficulty in the following way. Since anything follows from the contradiction of 'Pa' and '~Pa', we can use this contradiction to derive a **new** contradiction, 'A &~A', which does not use the name 'a'. ∃E then licenses us to write 'A &~A' in derivation 2 where we need the contradiction.

To streamline our work, we will introduce several new derived rules. The first is the one I have just proved, that any sentence, **Y**, follows from a contradiction:

Contradiction

X
~**X**

Y CD

In practice, I will always use a standard contradiction, 'A & ~A', for **Y**. I will also use a trivial reformulation of the rules ~I and Rd expressed in terms of a conjunction of a sentence and its negation where up to now these rules have, strictly speaking, been expressed only in terms of a sentence and the negation of the sentence on separate lines:

Negation Introduction

X
—————
Y & ~**Y**

~**X** ~I

Reductio

~**X**
—————
Y & ~**Y**

X RD

These derived rules enable us to deal efficiently with problem 5–7(q) and ones like it:

(∀x)~Fx
―――――
~(∃x)Fx

1	(∀x)~Fx	P
2	| (∃x)Fx	A
3	| | a | Fa	A
4	| | | (∀x)~Fx	1, R
5	| | | ~Fa	4, ∀E
6	| | | A & ~A	3, 4, CD
7	| A & ~A	2, 3–6, ∃E
8	~(∃x)Fx	2–7, ~I

Let's turn now to four more derived rules, ones which express the rules of logical equivalence, ~∀ and ~∃, which we discussed in chapter 3. There we proved that they are correct rules of logical equivalence. Formulated as derived rules, you have really done the work of proving them in problems 5–4(q) and (r) and 5–7(q) and (r). To prove these rules, all you need do is to copy the derivations you provided for those problems, using an arbitrary open sentence (. . . **u** . . .), with the free variable **u**, instead of the special case with the open sentence 'Px' or 'Fx' with the free variable 'x'.

Negated Quantifier Rules

| ~(∀**u**)(. . . **u** . . .)
| (∃**u**)~(. . . **u** . . .) ~∀

| (∃**u**)~(. . . **u** . . .)
| ~(∀**u**)(. . . **u** . . .) ∃~

| ~(∃**u**)(. . . **u** . . .)
| (∀**u**)~(. . . **u** . . .) ~∃

| (∀**u**)~(. . . **u** . . .)
| ~(∃**u**)(. . . **u** . . .) ∀~

A word of caution in using these negated quantifier rules: Students often rush to apply them whenever they see the opportunity. In many cases you may more easily see how to get a correct derivation by using these rules than if you try to make do without the rules. But often, if you

work hard and are ingenious, you can produce more elegant derivations without using the quantifier negation rules. In the following exercises, use the rules so that you have some practice with them. But in later exercises, be on the lookout for clever ways to produce derivations without the quantifier negation rules. Instructors who are particularly keen on their students learning to do derivations ingeniously may require you to do later problems without the quantifier negation rules. (These comments do not apply to the derived contradiction rule and derived forms of ~I and RD rules. These rules just save work which is invariably boring, so you should use them whenever they will shorten your derivations.)

EXERCISES

6–2.

a) (∀x)Px
(∀x)~Qx
―――――
~(∃x)(Px ≡ Qx)

b) (∀x)(Fx ⊃ Gx)
(∀x)(Gx ⊃ Hx)
~(∃x)Hx
―――――
~(∃x)Fx

c) ~(∀x)(∀y)Lxy
―――――
(∃x)(∃y)~Lxy

d) ~(∃x)(∃y)Lxy
―――――
(∀x)(∀y)~Lxy

e) ~(∃x)(Px v Qx)
―――――
(∀x)~Px & (∀x)~Qx

f) ~(∀x)(Px & Qx)
―――――
(∃x)~Px v (∃x)~Qx

g) (∀x)[~(∃y)Rxy & ~(∃y)Ryx]
―――――
(∀x)(∀y)~Rxy

h) (∃x)[Px ⊃ (∀y)(Py ⊃ Qy)]
~(∃x)Qx
―――――
~(∀x)Px

i) (∃y)(∃z)[(∀x)~Rxy v (∀x)~Rxz]
―――――
~(∀y)(∀z)(∃x)(Rxy & Rxz)

6–3. LOGICAL TRUTH, CONTRADICTIONS, INCONSISTENCY, AND LOGICAL EQUIVALENCE

This section straightforwardly applies concepts you have already learned for sentence logic. We said that a sentence of sentence logic is a logical truth if and only if it is true in all cases, that is, if and only if it comes out true for all assignments of truth values to sentence letters. The concept of logical truth is the same in predicate logic if we take our cases to be interpretations of a sentence:

> A closed predicate logic sentence is a *Logical Truth* if and only if it is true in all its interpretations.

Proof of logical truth also works just as it did for sentence logic, as we discussed in section 7–3 of Volume I. A derivation with no premises shows all its conclusions to be true in all cases (all assignments of truth values to sentence letters in sentence logic, all interpretations in predicate logic). A brief reminder of the reason: If we have a derivation with no premises we can always tack on unused premises at the beginning of the derivation. But any case which makes the premises of a derivation true makes all the derivation's conclusions true. For any case you like, tack on a premise in which that case is true. Then the derivation's conclusions will be true in that case also:

> A derivation with no premises shows all its conclusions to be logical truths.

Contradictions in predicate logic also follow the same story as in sentence logic. The whole discussion is the same as for logical truth, except that we replace "true" with "false":

> A closed predicate logic sentence is a *Contradiction* if and only if it is false in all its interpretations.
>
> To demonstrate a sentence, **X**, to be a contradiction, demonstrate its negation, ~**X**, to be a logical truth. That is, construct a derivation with no premises, with ~**X** as the final conclusion.

If you did exercise 7–5 (in volume I), you learned an alternative test for contradictions, which also works in exactly the same way in predicate logic:

> A derivation with a sentence, **X**, as its only premise and two sentences, **Y** and ~**Y**, as conclusions shows **X** to be a contradiction.

Exercise 7–8 (volume I) dealt with the concept of inconsistency. Once more, the idea carries directly over to predicate logic. I state it here, together with several related ideas which are important in more advanced work in logic:

> A collection of closed predicate logic sentences is *Consistent* if there is at least one interpretation which makes all of them true. Such an interpretation is called a *Model* for the consistent collection of sentences. If there is no interpretation which makes all of the sentences in the collection true (if there is no model), the collection is *Inconsistent*.
>
> A finite collection of sentences is inconsistent if and only if their conjunction is a contradiction.
>
> To demonstrate that a finite collection of sentences is inconsistent, demonstrate their conjunction to be a contradiction. Equivalently, provide a derivation with all of the sentences in the collection as premises and a contradiction as the final conclusion.

Finally, in predicate logic, the idea of logical equivalence of closed sentences works just as it did in sentence logic. We have already discussed this in section 3–4:

> 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.

Exercise 4–3 (volume I) provides the key to showing logical equivalence, as you already saw if you did exercise 7–9 (volume I). Two sentences are logically equivalent if in any interpretation in which the first is true the second is true, and in any interpretation in which the second is true the first is true. (Be sure you understand why this characterization comes to the same thing as the definition of logical equivalence I just gave.) Consequently

> To demonstrate that two sentences, **X** and **Y**, are logically equivalent, show that the two arguments, "**X**. Therefore **Y**." and "**Y**. Therefore **X**." are both valid. That is, provide two derivations, one with **X** as premise and **Y** as final conclusion and one with **Y** as premise and **X** as final conclusion.

EXERCISES

6–3. Provide derivations which show that the following sentences are logical truths:

a) (∀x)(∀y)Lxy ⊃ (∃x)(∃y)Lxy
b) (∀x)(Gx v ~Gx)
c) (∀x)(∃y)(Ax & By) ⊃ (∃x)(Ax & Bx)
d) (∃y)[Ky & (∀x)(Dx ⊃ Rxy)] ⊃ (∀x)[Dx ⊃ (∃y)(Ky & Rxy)]
e) (∃x)(∀y)(Fy ⊃ Fx)
f) (∀x)(∃y)(Fy ⊃ Fx)
g) (∃x)(∀y)(Fx ⊃ Fy)

6–4. Provide derivations which show that the following sentences are contradictions:

a) $(\forall x)(Ax \supset Bx) \mathbin{\&} (\exists x)(\sim Bx \mathbin{\&} (\forall y)Ay)$
b) $(\forall x)(Rxb \supset \sim Rxb) \mathbin{\&} (\exists x)Rxb$
c) $(\forall x)[(\forall y)Lxy \mathbin{\&} (\exists y)\sim Lyx]$
d) $(\forall x)(\exists y)(Mx \mathbin{\&} \sim My)$
e) $(\forall x)(\exists y)(\forall w)(\exists z)(Lxw \mathbin{\&} \sim Lyz)$

6–5. Provide derivations which show that the following collections of sentences are inconsistent:

a) $(\forall x)Kx$, $(\forall y)\sim(Ky \vee Lya)$
b) $(\forall x)(\exists y)Rxy$, $(\exists x)(\forall y)\sim Rxy$
c) $(\exists x)Dx$, $(\forall x)(Dx \supset (\forall y)(\forall z)Ryz)$, $(\exists x)(\exists y)\sim Rxy$
d) $(\exists x)(\exists y)(Rxx \mathbin{\&} \sim Ryy \mathbin{\&} Rxy)$, $(\forall x)(\forall y)(Rxy \supset Ryx)$, $(\forall x)(\forall y)\forall z)[(Rxy \mathbin{\&} Ryz) \supset Rxz)]$

6–6. a) List the pairs of sentences which are shown to be logically equivalent by the examples in this chapter and any of the derivations in exercises 6–1 and 6–8.

b) Write derivations which show the following three arguments to be valid. (You will see in the next part of this exercise that there is a point to your doing these trivial derivations.)

$$\begin{array}{l} (\forall x)Rxa \\ \hline (\exists x)Rxa \end{array} \qquad \begin{array}{l} (\forall x)Rxx \\ \hline (\exists x)Rxx \end{array} \qquad \begin{array}{l} (\forall x)Px \\ \hline (\exists x)Px \end{array}$$

c) Note that the three derivations you provided in your answer to (b) are essentially the same. From the point of view of these derivations, 'Rxa' and 'Rxx' are both open sentences which we could have just as well have written as **P(u)**, an arbitrary (perhaps very complex) open sentence with **u** as its only free variable. In many of the problems in 5–5 and 5–7, I threw in names and repeated variables which played no real role in the problem, just as in the first two derivations in (b) above. (I did so to keep you on your toes in applying the new rules.) Find the problems which, when recast in the manner illustrated in (b) above, do the work of proving the following logical equivalences. Here, **P(u)** and **Q(u)** are arbitrary open sentences with **u** as their only free variable. **A** is an arbitrary closed sentence.

$(\forall \mathbf{u})(\mathbf{P}(\mathbf{u}) \mathbin{\&} \mathbf{Q}(\mathbf{u}))$	is logically equivalent to	$(\forall \mathbf{u})\mathbf{P}(\mathbf{u}) \mathbin{\&} (\forall \mathbf{u})\mathbf{Q}(\mathbf{u})$
$(\exists \mathbf{u})(\mathbf{P}(\mathbf{u}) \vee \mathbf{Q}(\mathbf{u}))$	is logically equivalent to	$(\exists \mathbf{u})\mathbf{P}(\mathbf{u}) \vee (\exists \mathbf{u})\mathbf{Q}(\mathbf{u})$
$\mathbf{A} \supset (\forall \mathbf{u})\mathbf{P}(\mathbf{u})$	is logically equivalent to	$(\forall \mathbf{u})(\mathbf{A} \supset \mathbf{P}(\mathbf{u}))$
$\mathbf{A} \supset (\exists \mathbf{u})\mathbf{P}(\mathbf{u})$	is logically equivalent to	$(\exists \mathbf{u})(\mathbf{A} \supset \mathbf{P}(\mathbf{u}))$
$(\forall \mathbf{u})\mathbf{P}(\mathbf{u}) \supset \mathbf{A}$	is logically equivalent to	$(\exists \mathbf{u})(\mathbf{P}(\mathbf{u}) \supset \mathbf{A})$
$(\exists \mathbf{u})\mathbf{P}(\mathbf{u}) \supset \mathbf{A}$	is logically equivalent to	$(\forall \mathbf{u})(\mathbf{P}(\mathbf{u}) \supset \mathbf{A})$

d) Prove, by providing a counterexample, that the following two pairs of sentences are not logically equivalent. (A counterexample is an interpretation in which one of the two sentences is true and the other is false.)

$(\forall x)(Px \vee Qx)$	is not logically equivalent to	$(\forall x)Px \vee (\forall x)Qx$
$(\exists x)(Px \mathbin{\&} Qx)$	is not logically equivalent to	$(\exists x)Px \mathbin{\&} (\exists x)Qx$

e) Complete the work done in 6–1(c) and (d) to show that the following pairs of sentences are logically equivalent. (**R** is an arbitrary open sentence with **u** and **v** as its only two free variables.)

$(\forall \mathbf{u})(\forall \mathbf{v})\mathbf{R}(\mathbf{u}, \mathbf{v})$	is logically equivalent to	$(\forall \mathbf{v})(\forall \mathbf{u})\mathbf{R}(\mathbf{u}, \mathbf{v})$
$(\exists \mathbf{u})(\exists \mathbf{v})\mathbf{R}(\mathbf{u}, \mathbf{v})$	is logically equivalent to	$(\exists \mathbf{v})(\exists \mathbf{u})\mathbf{R}(\mathbf{u}, \mathbf{v})$

6–7. Here are some harder arguments to prove valid by providing derivations. In some cases it is easier to find solutions by using the derived rules for negated quantifiers. But in every case you should look for elegant solutions which do not use these rules.

a) $$\begin{array}{l} (\forall x)[(\exists y)(Lxy \vee Lyx) \supset Lxx] \\ (\exists x)(\exists y)Lxy \\ \hline (\exists x)Lxx \end{array}$$
(Everyone who loves or is loved by someone loves themself. Someone loves someone. Therefore, someone loves themself.)

b) $$\begin{array}{l} (\forall x)(Hx \supset Ax) \\ \hline (\forall x)[(\exists y)(Hy \mathbin{\&} Txy) \supset (\exists y)(Ay \mathbin{\&} Txy)] \end{array}$$
(Horses are animals. Therefore horses' tails are animals' tails.)

c) $$\begin{array}{l} (\forall x)(\forall y)[(\exists z)Lyz \supset Lxy] \\ (\exists x)(\exists y)Lxy \\ \hline (\forall x)(\forall y)Lxy \end{array}$$
(Everyone loves a lover. Someone loves someone. Therefore, everyone loves everyone.)

d) $$\begin{array}{l} (\forall x)(\forall y)[(\exists z)(Rzy \mathbin{\&} \sim Rxz) \supset Lxy] \\ \sim(\exists x)Lxx \\ \hline (\forall x)(\forall y)(\sim Ryx \supset \sim Rxy) \end{array}$$

e) $(\forall x)\{(\exists y)Lxy \supset (\exists y)[(\forall z)Lyz \,\&\, Lxy]\}$
$(\exists x)(\exists y)Lxy$

$(\exists x)(\forall y)Lxy$

(Everyone who loves someone loves someone who loves everyone. Someone loves someone. Therefore, someone loves everyone.)

f) $(\forall x)[Px \supset (\forall y)(Hy \supset Rxy)]$
$(\exists x)(Px \,\&\, (\exists y){\sim}Rxy)$

${\sim}(\forall x)Hx$

g) $(\forall x)[(Ex \supset (\forall y)(Hy \supset Wxy)]$
$(\exists x)[Hx \,\&\, (\forall y)(Dy \supset Wxy)]$
$(\forall x)(\forall y)(\forall z)[(Wxy \,\&\, Wyz) \supset Wxz]$

$(\forall x)[Ex \supset (\forall y)(Dy \supset Wxy)]$

(Any elephant weighs more than a horse. Some horse weighs more than any donkey. If a first thing weighs more than a second, and the second weighs more than a third, the first weighs more than the third. Therefore, any elephant weighs more than any donkey.)

h) $(\forall x)(\exists y)(Py \supset Qx)$

$(\exists y)(\forall x)(Py \supset Qx)$

Note that in general a sentence of the form $(\forall x)(\exists y)\mathbf{X}$ does not imply a sentence of the form $(\exists y)(\forall x)\mathbf{X}$ (See problem 6–1(e)). However, in this case, the special form of the conditional makes the argument valid.

i) $(\exists x)Px \supset (\exists x)Qx$

$(\forall x)(\exists y)(Px \supset Qy)$

j) $(\exists x)Px \supset (\forall x)Qx$

$(\forall x)(\forall y)(Px \supset Qy)$

k) $(\forall x)\{Bx \supset [(\exists y)Lxy \supset (\exists y)Lyx]\}$
$(\forall x)[(\exists y)Lyx \supset Lxx]$
${\sim}(\exists x)Lxx$

$(\forall x)(Bx \supset (\forall y){\sim}Lxy)$

(All blond lovers are loved. All those who are loved love themselves. No one loves themself. Therefore, all blonds love no one.)

l) $(\forall x)\{Fx \supset [Hx \,\&\, ({\sim}Cx \,\&\, {\sim}Kx)]\}$
$(\forall x)[(Hx \,\&\, {\sim}(\exists y)Nxy) \supset Dx]$

$(\forall x)(Fx \supset (\exists y)Nxy)$

m) $(\forall y)(Cy \supset Dy)$
$(\forall x)(\exists y)[(Hx \,\&\, Cx) \,\&\, (Gy \,\&\, Ryx)]$
$(\exists x)Dx \supset (\forall y)(\forall z)(Ryz \supset Dy)$

$(\exists x)(Gx \,\&\, Cx)$

n) $(\forall x)(\forall y)[(Rdy \,\&\, Rxd) \supset Rxy]$
$(\forall x)(Bx \supset Rdx)$
$(\exists x)(Bx \,\&\, Rxd)$

$(\exists x)[Bx \,\&\, (\forall y)(By \supset Rxy)]$

CHAPTER REVIEW EXERCISES

Write short explanations in your notebook for each of the following.

a) Contradiction Rule
b) Quantifier Negation Rules
c) Logical Truth of Predicate Logic
d) Test for a Logical Truth
e) Contradiction of Predicate Logic
f) Test for a Contradiction
g) Consistent Set of Sentences
h) Inconsistent Set of Sentences
i) Test for a Finite Set of Inconsistent Sentences
j) Logical Equivalence of Predicate Logic Sentences
k) Test for Logical Equivalence