Natural Deduction for Predicate Logic

5

Fundamentals

5-1. REVIEW AND OVERVIEW

Let's get back to the problem of demonstrating argument validity. You know how to construct derivations which demonstrate the validity of valid sentence logic arguments. Now that you have a basic understanding of quantified sentences and what they mean, you are ready to extend the system of sentence logic derivations to deal with quantified sentences.

Let's start with a short review of the fundamental concepts of natural deduction: To say that an argument is valid is to say that in every possible case in which the premises are true, the conclusion is true also. The natural deduction technique works by applying **truth preserving** rules. That is, we use rules which, when applied to one or two sentences, license us to draw certain conclusions. The rules are constructed so that in any case in which the first sentence or sentences are true, the conclusion drawn is guaranteed to be true also. Certain rules apply, not to sentences, but to subderivations. In the case of these rules, a conclusion which they license is guaranteed to be true if all the sentences reiterated into the subderivation are true.

A derivation begins with no premises or one or more premises. It may include subderivations, and any subderivation may itself include a subderivation. A new sentence, or conclusion, may be added to a derivation if one of the rules of inference licenses us to draw the conclusion from previous premises, assumptions, conclusions, or subderivations. Because

these rules are truth preserving, if the original premises are true in a case, the first conclusion drawn will be true in that case also. And if this first conclusion is true, then so will the next. And so on. Thus, altogether, in any case in which the premises are all true, the final conclusion will be true.

The only further thing you need to remember to be able to write sentence logic derivations are the rules themselves. If you are feeling rusty, please refresh your memory by glancing at the inside front cover, and review chapters 5 and 7 of Volume I, if you need to.

Now we are ready to extend our system of natural deduction for sentence logic to the quantified sentences of predicate logic. Everything you have already learned will still apply without change. Indeed, the only fundamental conceptual change is that we now must think in terms of an expanded idea of what constitutes a **case.** For sentence logic derivations, truth preserving rules guarantee that if the premises are true for an assignment of truth values to sentence letters, then conclusions drawn will be true for the same assignment. In predicate logic we use the same overall idea, except that for a "case" we use the more general idea of an **interpretation** instead of an assignment of truth values to sentence letters. Now we must say that if the premises are true in an interpretation, the conclusions drawn will be true in the same interpretation.

Since interpretations include assignment of truth values to any sentence letters that might occur in a sentence, everything from sentence logic applies as before. But our thinking for quantified sentences now has to extend to include the idea of interpretations as representations of the case in which quantified sentences have a truth value.

You will remember each of our new rules more easily if you understand why they work. You should understand why they are truth preserving by thinking in terms of interpretations. That is, you should try to understand why, if the premises are true in a given interpretation, the conclusion licensed by the rule will inevitably also be true in that interpretation.

Predicate logic adds two new connectives to sentence logic: the universal and existential quantifiers. So we will have four new rules, an introduction and elimination rule for each quantifier. Two of these rules are easy and two are hard. Yes, you guessed it! I'm going to introduce the easy rules first.

5-2. THE UNIVERSAL ELIMINATION RULE

Consider the argument

Everyone is blond.	$(\forall x)Bx$
Adam is blond.	Ba

Intuitively, if everyone is blond, this must include Adam. So if the premise is true, the conclusion is going to have to be true also. In terms of interpretations, let's consider any interpretation you like which is an interpretation of the argument's sentences and in which the premise, '(∀x)Bx', is true. The definition of truth of a universally quantified sentence tells us that '(∀x)Bx' is true in an interpretation just in case all of its substitution instances are true in the interpretation. Observe that 'Ba' is a substitution instance of '(∀x)Bx'. So in our arbitrarily chosen interpretation in which '(∀x)Bx' is true, 'Ba' will be true also. Since 'Ba' is true in any interpretation in which '(∀x)Bx' is true, the argument is valid.

(In this and succeeding chapters I am going to pass over the distinction between some**one** and some**thing**, as this complication is irrelevant to the material we now need to learn. I could give examples of things instead of people, but that makes learning very dull.)

The reasoning works perfectly generally:

> *Universal Elimination Rule:* If **X** is a universally quantified sentence, then you are licensed to conclude any of its substitution instances below it. Expressed with a diagram, for any name, **s**, and any variable, **u**,

[(∀**u**)(. . . **u** . . .)]	
.	
.	
.	
((. . . **s** . . .))	∀E

Remember what the box and the circle mean: If on a derivation you encounter something with the form of what you find in the box, the rule licenses you to conclude something of the form of what you find in the circle.

Here is another example:

Everyone loves Eve.	(∀x)Lxe	1	(∀x)Lxe	*P*
Adam loves Eve.	Lae	2	Lae	1, ∀E

In forming the substitution instance of a universally quantified sentence, you must be careful always to put the same name everywhere for the substituted variable. Substituting 'a' for 'x' in '(∀x)Lxx', we get 'Laa', **not** 'Lxa'. Also, be sure that you substitute your name only for the occurrences of the variable which are **free** after deleting the initial quantifier. Using the name 'a' again, the substitution instance of '(∀x)(Bx ⊃ (∀x)Lxe)' is 'Ba ⊃ (∀x)Lxe'. The occurrence of 'x' in 'Lxe' is bound by the second

'(∀x)', and so is still bound after we drop the first '(∀x)'. If you don't understand this example, you need to review bound and free variables and substitution instances, discussed in chapter 3.

When you feel confident that you understand the last example, look at one more:

(∀x)(Gx ⊃ Kx)	1	(∀x)(Gx ⊃ Kx)	*P*
Gf	2	Gf	*P*
Kf	3	Gf ⊃ Kf	1, ∀E
	4	Kf	2, 3, ⊃E

EXERCISES

5–1. Provide derivations which demonstrate the validity of these arguments. Remember to work from the conclusion backward, seeing what you will need to get your final conclusions, as well as from the premises forward. In problem (d) be sure you recognize that the premise is a universal quantification of a conditional, while the conclusion is the very different conditional with a universally quantified antecedent.

a) (∀x)(Px & Dx) / Pk

b) (∀x)(Px & Dx) / Pd & Dk

c) (∀x)(Dx ⊃ Kx), (∀x)Dx / Ka

d) (∀x)(Mx ⊃ A) / (∀x)Mx ⊃ A

e) (∀x)(Fx v Hx), (∀x)(Fx ⊃ Dx), (∀x)(Hx ⊃ Dx) / Dp & Db

f) (∀x)(~Bx v Lcx) / (∀x)Bx ⊃ Lcd

g) (∀x)(Lxx ⊃ Lxh), ~Lmh / ~(∀x)Lxx

h) (∀x)(Rxx v Rxk), (∀y)~Ryk / Rcc & Rff

5–3. THE EXISTENTIAL INTRODUCTION RULE

Consider the argument

Adam is blond.	Ba
Someone is blond.	(∃x)Bx

Intuitively, this argument is valid. If Adam is blond, there is no help for it: Someone is blond. Thinking in terms of interpretations, we see that this argument is valid according to our new way of making the idea of validity precise. Remember how we defined the truth of an existentially quantified sentence in an interpretation: '(∃x)Bx' is true in an interpretation if and only if at least one of its substitution instances is true in the interpretation. But 'Ba' **is** a substitution instance of '(∃x)Bx'. So, in any interpretation in which 'Ba' is true, '(∃x)Bx' is true also, which is just what we mean by saying that the argument "Ba. Therefore (∃x)Bx." is valid.

You can probably see the form of reasoning which is at play here: From a sentence with a name we can infer what we will call an *Existential Generalization* of that sentence. '(∃x)Bx' is an existential generalization of 'Ba'. We do have to be a little careful in making this notion precise because we can get tripped up again by problems with free and bound variables. What would you say is a correct existential generalization of '(∀x)Lax'? In English: If Adam loves everyone, then we know that someone loves everyone. But we have to use two different variables to transcribe 'Someone loves everyone': '(∃y)(∀x)Lyx'. If I start with '(∀x)Lax', and replace the 'a' with 'x', my new occurrence of 'x' is bound by that universal quantifier. I will have failed to generalize **existentially** on 'a'.

Here is another example for you to try: Existentially generalize

(i) Ba ⊃ (∀x) Lax
 2 3 45

If I drop the 'a' at 2 and 4, write in 'x', and preface the whole with '(∃x)', I get

(ii) (∃x)(Bx ⊃ (∀x)Lxx) **Wrong**
 1 2 3 45

The 'x' at 4, which replaced one of the 'a's, is bound by the universally quantified 'x' at 3, not by the existentially quantified 'x' at 1, as we intend in forming an **existential** generalization. We have to use a new variable. A correct existential generalization of 'Ba ⊃ (∀x)Lax' is

(iii) (∃y)(By ⊃ (∀x) Lyx)
 1 2 3 45

as are

(iv) (∃y)(By ⊃ (∀x)Lax)
 1 2 3 45

and

(v) (∃y)(Ba⊃(∀x)Lyx)
 1 2 3 45

Here is how you should think about this problem: Starting with a closed sentence, (. . . **s** . . .), which uses a name, **s**, take out one or more of the occurrences of the name **s**. For example, take out the 'a' at 4 in (i). Then look to see if the vacated spot is already in the scope of one (or more) quantifiers. In (i) to (v), the place marked by 4 is in the scope of the '(∀x)' at 3. So you can't use 'x'. You must perform your existential generalization with some variable which is not already bound at the places at which you replace the name. After taking out one or more occurrences of the name, **s**, in (. . . **s** . . .), replace the vacated spots with a variable (the **same** variable at each spot) which is not bound by some quantifier already in the sentence.

Continuing our example, at this point you will have turned (i) into

(vi) Ba ⊃ (∀x)Lya

You will have something of the form (. . . **u** . . .) in which **u** is **free**: 'y' is free in (vi). At this point you must have an **open sentence**. Now, at last, you can apply your existential quantifier to the resulting open sentence to get the closed sentence (∃**u**)(. . . **u** . . .).

To summarize more compactly:

> (∃**u**)(. . . **u** . . .) is an *Existential Generalization* of (. . . **s** . . .) with respect to the name **s** if and only if (∃**u**)(. . . **u** . . .) results from (. . . **s** . . .) by
>
> a) Deleting any number of occurrences of **s** in (. . . **s** . . .),
> b) Replacing these occurrences with a variable, **u**, which is **free** at these occurrences, and
> c) Applying (∃**u**) to the result.

(In practice you should read (a) in this definition as "Deleting one or more occurrences of **s** in (. . . **s** . . .)." I have expressed (a) with "any number of" so that it will correctly treat the odd case of vacuous quantifiers, which in practice you will not need to worry about. But if you are interested, you can figure out what is going on by studying exercise 3–3.)

It has taken quite a few words to set this matter straight, but once you see the point you will no longer need the words.

With the idea of an existential generalization, we can accurately state the rule for existential introduction:

> *Existential Introduction Rule:* From any sentence, **X**, you are licensed to conclude any existential generalization of **X** anywhere below. Expressed with a diagram,

(. . . s . . .)	
.	Where (∃**u**)(. . . **u** . . .)
.	is an existential
.	generalization
(∃**u**)(. . . **u** . . .) ∃I	of (. . . s . . .).

Let's look at a new example, complicated only by the feature that it involves a second name which occurs in both the premise and the conclusion:

Adam loves Eve.	Lae
Adam loves someone.	(∃x)Lax

'(∃x)Lax' is an existential generalizaton of 'Lae'. So ∃I applies to make the following a correct derivation:

1	Lae	*P*
2	(∃x)Lax	1, ∃I

To make sure you have the hang of rule ∃I, we'll do one more example. Notice that in this example, the second premise has an atomic sentence letter as its consequent. Remember that predicate logic is perfectly free to use atomic sentence letters as components in building up sentences.

Argument			
Ka	1	Ka	*P*
(∃x)Kx ⊃ P	2	(∃x)Kx ⊃ P	*P*
P	3	(∃x)Kx	1, ∃I
	4	P	2, 3, ⊃E

In line 4 I applied ⊃E to lines 2 and 3. ⊃E applies here in exactly the same way as it did in sentence logic. In particular ⊃E and the other sentence logic rules apply to sentences the components of which may be quantified sentences as well as sentence logic sentences.

Now let's try an example which applies both our new rules:

Argument			
(∀x)Lxx	1	(∀x)Lxx	*P*
(∃x)Lxx	2	Laa	1, ∀E
	3	(∃x)Lxx	2, ∃I

In addition to illustrating both new rules working together, this example illustrates something else we have not yet seen. In past examples, when I applied ∀E I instantiated a universally quantified sentence with a

name which already occurred somewhere in the argument. In this case no name occurs in the argument. But if a universally quantified sentence is true in an interpretation, all of its substitution instances must be true in the interpretation. And every interpretation must have at least one object in it. So a universally quantified sentence must always have at least one substitution instance true in an interpretation. Since a universally quantified sentence always has at least one substitution instance, I can introduce a name into the situation with which to write that substitution instance, if no name already occurs.

To put the point another way, because every interpretation always has at least one object in it, I can always introduce a name to refer to some object in an interpretation and then use this name to form my substitution instance of the universally quantified sentence.

Good. Let's try yet another example:

Argument			
(∀x)(Cx ⊃ Mx)	1	(∀x)(Cx ⊃ Mx)	*P*
Cd	2	Cd	*P*
(∃x)Mx	3	Cd ⊃ Md	1, ∀E
	4	Md	2, 3, ⊃E
	5	(∃x)Mx	4, ∃I

Notice that although the rules permit me to apply ∃I to line 2, doing so would not have gotten me anywhere. To see how I came up with this derivation, look at the final conclusion. You know that it is an existentially quantified sentence, and you know that ∃I permits you to derive such a sentence from an instance, such as 'Md'. So you must ask yourself: Can I derive such an instance from the premises? Yes, because the first premise says about everything that if it is C, then it is M. And the second premise says that d, in particular, is C. So applying ∀E to 1 you can get 3, which, together with 2, gives 4 by ⊃E.

EXERCISES

5–2. Provide derivations which demonstrate the validity of the following arguments:

a) Na / (∃x)(Nx v Gx)

b) (∀x)(Kx & Px) / (∃x)Kx & (∃x)Px

c) (∀x)(Hx ⊃ ~Dx), Dg / (∃x)~Hx

d) (∀x)Ax & (∀x)Txd / (∃x)(Ax & Txd)

e) Fa v Nh / (∃x)Fx v (∃x)Nx

f) (∀x)(Sx v Jx) / (∃x)Sx v (∃x)Jx

g) $$\begin{array}{c} (\exists x)Rxa \supset (\forall x)Rax \\ Rea \\ \hline (\exists x)Rax \end{array}$$

h) $$\begin{array}{c} Lae \vee Lea \\ (\exists x)Lax \supset A \\ (\exists x)Lxa \supset A \\ \hline A \end{array}$$

i) $$\begin{array}{c} (\exists x)Jx \supset Q \\ (\forall x)Jx \\ \hline Q \end{array}$$

j) $$\begin{array}{c} (\forall x)(Max \vee Mex) \\ \sim(\exists x)Max \vee Bg \\ \sim(\exists x)Mex \vee Bg \\ \hline (\exists x)Bx \end{array}$$

k) $$\begin{array}{c} (\forall x)(Kxx \equiv Px) \\ (\forall x)[Kjx \& (Px \supset Sx)] \\ \hline (\exists x)Sx \end{array}$$

l) $$\begin{array}{c} (\forall x)(\sim Oxx \vee Ix) \\ (\forall x)(Ix \supset Rxm) \\ \hline (\forall x)Oxx \supset (\exists x)Rxm \end{array}$$

5-4. THE EXISTENTIAL ELIMINATION AND UNIVERSAL INTRODUCTION RULES: BACKGROUND IN INFORMAL ARGUMENT

Now let's go to work on the two harder rules. To understand these rules, it is especially important to see how they are motivated. Let us begin by looking at some examples of informal deductive arguments which present the kind of reasoning which our new rules will make exact. Let's start with this argument:

> Everyone likes either rock music or country/western.
> Someone does not like rock.
> ______________________________
> Someone likes country/western.

Perhaps this example is not quite as trivial as our previous examples. How can we see that the conclusion follows from the premises? We commonly argue in the following way. We are given the premise that someone does not like rock. To facilitate our argument, let us suppose that this person (or one of them if there are more than one) is called Doe. (Since I don't know this person's name, I'm using 'Doe' as the police do when they book a man with an unknown name as 'John Doe.') Now, since according to the first premise, everyone likes either rock or country/western, this must be true, in particular, of Doe. That is, either Doe likes rock, or he or she likes country/western. But we had already agreed that Doe does not like rock. So Doe must like country/western. Finally, since Doe likes country/western, we see that someone likes country/western. But that was just the conclusion we were trying to derive.

What you need to focus on in this example is how I used the name 'Doe'. The second premise gives me the assumption that someone does not like rock. So that I can talk about this someone, I give him or her a name: 'Doe'. I don't know anything more that applies to just this person, but I do have a fact, the first premise, which applies to everyone. So I can use this fact in arguing about Doe, even though I really don't know who Doe is. I use this general fact to conclude that Doe, whoever he or she might be, does like country/western. Finally, before I am done, I acknowledge that I really don't know who Doe is, in essence by saying: Whoever this person Doe might be, I know that he or she likes country/western. That is, what I really can conclude is that there is someone who likes country/western.

Now let's compare this argument with another:

> (1) Everyone either likes rock or country/western.
> (2) Anyone who likes country/western likes soft music.
> ______________________________
> (3) Anyone who doesn't like rock likes soft music.

This time I have deliberately chosen an example which might not be completely obvious so that you can see the pattern of reasoning doing its work.

The two premises say something about absolutely everyone. But it's hard to argue about 'everyone'. So let us think of an arbitrary example of a person, named 'Arb', to whom these premises will then apply. My strategy is to carry the argument forward in application to this arbitrarily chosen individual. I have made up the name 'Arb' to emphasize the fact that I have chosen this person (and likewise the name) perfectly arbitrarily. We could just as well have chosen any person named by any name.

To begin the argument, the first premise tells us that

> (4) Either Arb likes rock, or Arb likes country/western.

The second premise tells us that

> (5) If Arb does like country/western, then Arb likes soft music.

Now, let us make a further assumption about Arb:

> (6) (Further Assumption): Arb doesn't like rock.

From (6) and (4), it follows that

> (7) Arb likes country/western.

And from (7) and (5), it follows that

> (8) Arb likes soft music.

Altogether we see that Arb's liking soft music, (8), follows from the further assumption, (6), with the help of the original premises (1) and (2) (as applied through this application to Arb, in (4) and (5)). Consequently, from the original premises it follows that

(9) If Arb doesn't like rock, then Arb likes soft music.

All this is old hat. Now comes the new step. The whole argument to this point has been conducted in terms of the person, Arb. But Arb could have been anyone, or equally, we could have conducted the argument with the name of anyone at all. So the argument is perfectly general. What (9) says about Arb will be true of anyone. That is, we can legitimately conclude that

(3) Anyone who doesn't like rock likes soft music.

which is exactly the conclusion we were trying to reach.

We have now seen two arguments which use "stand-in" names, that is, names that are somehow doing the work of "someone" or of "anyone". Insofar as both arguments use stand-in names, they seem to be similar. But they are importantly different, and understanding our new rules turns on understanding how the two arguments are different. In the second argument, Arb could be anyone—absolutely anyone at all. But in the first argument, Doe could not be anyone. Doe could only be the person, or one of the people, who does not like rock. 'Doe' is "partially arbitrary" because we are careful not to assume anything we don't know about Doe. But we do know that Doe is a rock hater and so is not just anyone at all. Arb, however, could have been anyone.

We must be very careful not to conflate these two ways of using stand-in names in arguments. Watch what happens if you do conflate the ways:

Someone does not like rock. (Invalid)
Everyone does not like rock.

The argument is just silly. But confusing the two functions of stand-in names could seem to legitimate the argument, if one were to argue as follows: Someone does not like rock. Let's call this person 'Arb'. So Arb does not like rock. But Arb could be anyone, so everyone does not like rock. In such a simple case, no one is going to blunder in this way. But in more complicated arguments it can happen easily.

To avoid this kind of mistake, we must find some way to clearly mark the difference between the two kinds of argument. I have tried to bring out the distinction by using one kind of stand-in name, 'Doe', when we are talking about the existence of some particular person, and another kind of stand-in name, 'Arb', when we are talking about absolutely any arbitrary individual. This device works well in explaining that a stand-in name can function in two very different ways. Unfortunately, we cannot incorporate this device in natural deduction in a straightforward way simply by using two different kinds of names to do the two different jobs.

Let me try to explain the problem. (You don't need to understand the problem in detail right now; detailed understanding will come later. All you need at this point is just a glimmer of what the problem is.) At the beginning of a derivation a name can be arbitrary. But then we might start a subderivation in which the name occurs, and although arbitrary from the point of view of the outer derivation, the name might **not** be arbitrary from the point of view of the subderivation. This can happen because in the original derivation nothing special, such as hating rock, is assumed about the individual. But inside the subderivation we might make such a further assumption about the individual. **While the further assumption is in effect, the name is not arbitrary,** although it can become arbitrary again when we discharge the further assumption of the subderivation. In fact, exactly these things happened in our last example. If, while the further assumption (6) was in effect, I had tried to generalize on statements about Arb, saying that what was true of Arb was true of anyone, I could have drawn all sorts of crazy conclusions. Look back at the example and see if you can figure out for yourself what some of these conclusions might be.

Natural deduction has the job of accurately representing valid reasoning which uses stand-in names, but in a way which won't allow the sort of mistake or confusion I have been pointing out. Because the confusion can be subtle, the natural deduction rules are a little complicated. The better you understand what I have said in this section, the quicker you will grasp the natural deduction rules which set all this straight.

EXERCISES

5–3. For each of the two different uses of stand-in names discussed in this section, give a valid argument of your own, expressed in English, which illustrates the use.

5–5. THE UNIVERSAL INTRODUCTION RULE

Here is the intuitive idea for universal introduction, as I used this rule in the soft music example: If a name, as it occurs in a sentence, is completely arbitrary, you can *Universally Generalize* on the name. This means that you rewrite the sentence with a variable written in for all occurrences of the arbitrary name, and you put a universal quantifier, written with the same variable, in front. To make this intuition exact, we have to say exactly when a name is arbitrary and what is involved in universal generalization. We must take special care because universal generalization differs importantly from existential generalizaton.

Let's tackle arbitrariness first. When does a name **not** occur arbitrarily? Certainly not if some assumption is made about (the object referred to by) the name. If some assumption is made using a name, then the name can't refer to absolutely anything. If a name occurs in a premise or assumption, the name can refer only to things which satisfy that premise or assumption. So a name does not occur arbitrarily when the name appears in a premise or an assumption, and it does not occur arbitrarily as long as such a premise or assumption is in effect.

The soft music example shows these facts at work. I'll use 'Rx' for 'x likes rock.', 'Cx' for 'x likes country/western.', and 'Sx' for 'x likes soft music.' Here are the formalized argument and derivation which I am going to use to explain these ideas:

```
(∀x)(Rx v Cx)
(∀x)(Cx ⊃ Sx)
--------------
(∀x)(~Rx ⊃ Sx)
```

```
 1 | (∀x)(Rx v Cx)      P
 2 | (∀x)(Cx ⊃ Sx)      P
   |----------------
 3 | Ra v Ca            1, ∀E
 4 | Ca ⊃ Sa            2, ∀E
 5 |    | ~Ra           A
   |    |-------
 6 |    | Ra v Ca       3, R
 7 |    | Ca            5, 6, vE
 8 |    | Ca ⊃ Sa       4, R
 9 |    | Sa            7, 8, ⊃E
10 | ~Ra ⊃ Sa           5–9, ⊃I
11 | (∀x)(~Rx ⊃ Sx)     10, ∀I
```

Where does 'a' occur arbitrarily in this example? It occurs arbitrarily in lines 3 and 4, because at these lines no premise or assumption using 'a' is in effect. We say that these lines are *Not Governed* by any premise or assumption in which 'a' occurs. In lines 5 through 9, however, 'a' does not occur arbitrarily. Line 5 is an assumption using 'a'. In lines 5 through 9, the assumption of line 5 is in effect, so these lines are governed by the assumption of line 5. (We are going to need to say that a premise or assumption always governs itself.) In all these lines something special is being assumed about the thing named by 'a', namely, that it has the property named by '~R'. So in these lines the thing named by 'a' is not just any old thing. However, in line 10 we discharge the assumption of line 5. So in line 10 'a' again occurs arbitrarily. Line 10 is only governed by the premises 1 and 2, in which 'a' does not occur. Line 10 is not governed by the assumption of line 5.

I am going to introduce a device to mark the arbitrary occurrences of a name. If a name occurs arbitrarily we will put a hat on it, so it looks like this: â. Marking all the arbitrary occurrences of 'a' in the last derivation makes the derivation look like this:

```
 1 | (∀x)(Rx v Cx)      P
 2 | (∀x)(Cx ⊃ Sx)      P
   |----------------
 3 | Râ v Câ            1, ∀E
 4 | Câ ⊃ Sâ            2, ∀E
 5 |    | ~Ra           A
   |    |-------
 6 |    | Ra v Ca       3, R
 7 |    | Ca            5, 6, vE
 8 |    | Ca ⊃ Sa       4, R
 9 |    | Sa            7, 8, ⊃E
10 | ~Râ ⊃ Sâ           5–9, ⊃I
11 | (∀x)(~Rx ⊃ Sx)     10, ∀I
```

Read through this copy of the derivation and make sure you understand why the hat occurs where it does and why it does not occur where it doesn't. If you have a question, reread the previous paragraph, remembering that a hat on a name just means that the name occurs arbitrarily at that place.

I want to be sure that you do not misunderstand what the hat means. A name with a hat on it is not a new kind of name. A name is a name is a name, and two occurrences of the same name, one with and one without a hat, are two occurrences of the same name. A hat on a name is a kind of flag to remind us that at that point the name is occurring arbitrarily. Whether or not a name occurs arbitrarily is not really a fact just about the name. It is a fact about the relation of the name to the derivation in which it occurs. If, at an occurrence of a name, the name is governed by a premise or assumption which uses the same name, the name does not occur there arbitrarily. It is not arbitrary there because the thing it refers to has to satisfy the premise or assumption. Only if a name is not governed by any premise or assumption using the same name is the name arbitrary, in which case we mark it by dressing it with a hat.

Before continuing, let's summarize the discussion of arbitrary occurrence with an exact statement:

> Suppose that a sentence, **X**, occurs in a derivation or subderivation. That occurrence of **X** is *Governed* by a premise or assumption, **Y**, if and only if **Y** is a premise or assumption of **X**'s derivation, or of any outer derivation of **X**'s derivation (an outer derivation, or outer-outer derivation, and so on). In particular, a premise or assumption is always governed by itself.

A name *Occurs Arbitrarily* in a sentence of a derivation if that occurrence of the sentence is not governed by any premise or assumption in which the name occurs. To help us remember, we mark an arbitrary occurrence of a name by writing it with a hat.

The idea for the universal introduction rule was that we would *Universally Generalize* on a name that occurs arbitrarily. We have discussed arbitrary occurrence. Now on to universal generalization.

The idea of a universal generalization differs in one important respect from the idea of an existential generalization. To see the difference, you must be clear about what we want out of a generalization: We want a new quantified sentence which follows from a sentence with a name.

For the existential quantifier, '(∃x)Lxx', '(∃x)Lax', and '(∃x)Lxa' all follow from 'Laa'. From the fact that Adam loves himself, it follows that Adam loves someone, someone loves Adam, and someone loves themself.

Now suppose that the name 'â' occurs arbitrarily in 'Lââ'. We know that "Adam" loves himself, where Adam now could be just anybody at all. What universal fact follows? **Only** that '(∀x)Lxx', that everyone loves themself. It does **not** follow that '(∀x)Lâx' or '(∀x)Lxâ'. That is, it does not follow that Adam loves everyone or everyone loves Adam. Even though 'Adam' occurs arbitrarily, '(∀x)Lâx' and '(∀x)Lxâ' make it sound as if someone ("Adam") loves everyone and as if someone ("Adam") is loved by everyone. These surely do not follow from 'Lââ'. But ∃I would license us to infer these sentences, respectively, from '(∀x)Lâx' and from '(∀x)Lxâ'.

Worse, â is still arbitrary in '(∀x)Lâx'. So if we could infer '(∀x)Lâx' from 'Lââ', we could then argue that in '(∀x)Lâx', 'â' could be anyone. We would then be able to infer '(∀y)(∀x)Lyx', that everyone loves everyone! But from 'Lââ' we should only be able to infer '(∀x)Lxx', that everyone loves themself, not '(∀y)(∀x)Lyx', that everyone loves everyone.

We want to use the idea of existential and universal generalizations to express valid rules of inference. The last example shows that, to achieve this goal, we have to be a little careful with sentences in which the same name occurs more than once. If **s** occurs more than once in (. . . **s** . . .), we may form an **existential** generalization by generalizing on any number of the occurrences of **s**. But, to avoid the problem I have just described and to get a valid rule of inference, we must insist that a **universal** generalization of (. . . **s** . . .), with respect to the name, **s**, must leave no instance of **s** in (. . . **s** . . .).

In other respects the idea of universal generalization works just like existential generalization. In particular, we must carefully avoid the trap of trying to replace a name by a variable already bound by a quantifier. This idea works exactly as before, so I will proceed immediately to an exact statement:

The sentence (∀**u**)(. . . **u** . . .) results by *Universally Generalizing* on the name **s** in (. . . **s** . . .) if and only if one obtains (∀**u**)(. . . **u** . . .) from (. . . **s** . . .) by

a) Deleting **all** occurrences of **s** in (. . . **s** . . .),
b) Replacing these occurrences with a variable, **u**, which is **free** at these occurrences, and
c) Applying (∀**u**) to the result.

(∀**u**)(. . . **u** . . .) is then said to be the *Universal Generalization of (. . . s . . .) with Respect to the Name s.*

With these definitions, we are at last ready for an exact statement of the universal introduction rule:

Universal Introduction Rule: If a sentence, **X**, appears in a derivation, and if at the place where it appears a name, **ŝ**, occurs arbitrarily in **X**, then you are licensed to conclude, anywhere below, the sentence which results by universally generalizing on the name **ŝ** in **X**. Expressed with a diagram:

| (. . . ŝ . . .)
| (∀**u**)(. . . **u** . . .) ∀I

Where ŝ occurs arbitrarily in (. . . ŝ . . .) and (∀**u**)(. . . **u** . . .) is the universal generalization of (. . . ŝ . . .) with respect to ŝ.

Let's look at two simple examples to illustrate what can go wrong if you do not follow the rule correctly. The first example is the one we used to illustrate the difference between existential and universal generalization:

Everyone loves themself.
Everyone loves Adam.

(Invalid!)

1	(∀x)Lxx	*P*
2	Lââ	1, ∀E
3	(∀x)Lxâ	**Mistaken attempt to apply ∀I to 2. 3 is not a universal generalization of 2.**

The second example will make sure you understand the requirement that ∀I applies only to an arbitrary occurrence of a name:

Adam is blond.
Everyone is blond.

(Invalid!)

1	Ba	*P*
2	(∀x)Bx	**Mistaken attempt to apply ∀I to 1. 'a' is not arbitrary at 1.**

The problem here is that the premise assumes something special about the thing referred to by 'a', that it has the property referred to by 'B'. We can universally generalize on a name—that is, apply ∀I—**only** when nothing special is assumed in this way, that is, when the name is arbitrary. You will see this even more clearly if you go back to our last formalization of the soft music example and see what sorts of crazy conclusions you could draw if you were to allow yourself to generalize on occurrences of names without hats.

Let's consolidate our understanding of ∀I by working through one more example. Before reading on, try your own hand at providing a derivation for

(∀x)(Lax & Lxa)
(∀x)(Lax ≡ Lxa)

If you don't see how to begin, use the same overall strategy we developed in chapter 6 of volume I. Write a skeleton derivation with its premise and final conclusion and ask what you need in order to get the final, or target, conclusion.

```
1 | (∀x)(Lax & Lxa)    P
  |----------------
  | ?
  |
  | ?
  |
  | (∀x)(Lax ≡ Lxa)
```

We could get our target conclusion by ∀I if we had a sentence of the form 'Lab̂ ≡ Lb̂a'. Let's write that in to see if we can make headway in this manner:

```
1 | (∀x)(Lax & Lxa)    P
  |----------------
  | ?
  |
  | ?
  |
  | Lab̂ ≡ Lb̂a
  | (∀x)(Lax ≡ Lxa)    ∀I
```


'Lab̂ ≡ Lb̂a' is now our target conclusion. As a biconditional, our best bet is to get it by ≡I from 'Lab ⊃ Lba' and 'Lba ⊃ Lab'. (I didn't write hats on any names because, as I haven't written the sentences as part of the derivation, I am not yet sure which sentences will govern these two conditionals.) The conditionals, in turn, I hope to get from two subderivations, one each starting from one of the antecedents of the two conditionals:

```
1 | (∀x)(Lax & Lxa)        P
  |----------------
  | ?
  |     | Lab              A
  |     |----
  |     | ?
  |     |
  |     | Lba
  |
  | Lab̂ ⊃ Lb̂a              ⊃I
  |
  | ?
  |     | Lba              A
  |     |----
  |     | ?
  |     |
  |     | Lab
  |
  | Lb̂a ⊃ Lab̂              ⊃I
  | Lab̂ ≡ Lb̂a              ≡I
  | (∀x)(Lax ≡ Lxa)        ∀I
```

Notice that 'b' gets a hat wherever it appears in the main derivation. There, 'b' is not governed by any assumption in which 'b' occurs. But 'b' occurs in the assumptions of both subderivations. So in the subderivations 'b' gets no hat. Finally, 'a' occurs in the original premise. That by itself rules out putting a hat on 'a' anywhere in the whole derivation, which includes all of its subderivations.

Back to the question of how we will fill in the subderivations. We need to derive 'Lba' in the first and 'Lab' in the second. Notice that if we apply ∀E to the premise, using 'b' to instantiate 'x', we get a conjunction with exactly the two new target sentences as conjuncts. We will be able to apply &E to the conjunction and then simply reiterate the conjuncts in the subderivations. Our completed derivation will look like this:

1	(∀x)(Lax & Lxa)	*P*
2	Lab̂ & Lb̂a	1, ∀E
3	Lab̂	2, &E
4	Lb̂a	2, &E
5	Lab	*A*
6	Lba	4, *R*
7	Lab̂ ⊃ Lb̂a	5–6, ⊃I
8	Lba	*A*
9	Lab	3, *R*
10	Lb̂a ⊃ Lab̂	8–9, ⊃I
11	Lab̂ ≡ Lb̂a	7, 10, ≡I
12	(∀x)(Lax ≡ Lxa)	11, ∀I

Once more, notice that 'b' gets a hat in lines 2, 3, and 4. In these lines no premise or assumption using 'b' is operative. But in lines 5, 6, 8, and 9, 'b' gets no hat, even though exactly the same sentences appeared earlier (lines 3 and 4) with hats on 'b'. This is because when we move into the subderivations an assumption goes into effect which says something special about 'b'. So in the subderivations, off comes the hat. As soon as this special assumption about 'b' is discharged, and we move back out of the subderivation, no special assumption using 'b' is in effect, and the hat goes back on 'b'.

You may well wonder why I bother with the hats in lines like 2, 3, 4, 7, and 10, on which I am never going to universally generalize. The point is that, so far as the rules go, I am permitted to universally generalize on 'b' in these lines. In this problem I don't bother, because applying ∀I to these lines will not help me get my target conclusion. But you need to develop awareness of just when the formal statement of the ∀I rule allows you to apply it. Hence you need to learn to mark those places at which the rule legitimately could apply.

Students often have two more questions about hats. First, ∀I permits you to universally generalize on a name with a hat. But you can **also** apply ∃I to a name with a hat. Now that I have introduced the hats, the last example in section 5–3 should really look like this:

1	(∀x)Lxx	*P*
2	Lââ	1, ∀E
3	(∃x)Lxx	2, ∃I

If everyone loves themself, then Arb loves him or herself, whoever Arb may be. But then someone loves themself. When a name occurs arbitrarily, the name can refer to anything. But then it **also** refers to something. You can apply **either ∀I or ∃I** to a hatted name.

It is also easy to be puzzled by the fact that a name which is introduced in the assumption of a subderivation, and thus does not occur arbitrarily there, can occur arbitrarily after the assumption of the subderivation has been discharged. Consider this example:

1	(∃x)Px ⊃ (∀x)Qx	*P*
2	Pa	*A*
3	(∃x)Px	2, ∃I
4	(∃x)Px ⊃ (∀x)Qx	1, *R*
5	(∀x)Qx	3, 4, ⊃E
6	Qa	5, ∀E
7	Pâ ⊃ Qâ	2–6, ⊃I
8	(∀x)(Px ⊃ Qx)	7, ∀I

In the subderivation something is assumed about 'a', namely, that it has the property P. So, from the point of view of the subderivation, 'a' is not arbitrary. As long as the assumption of the subderivation is in effect, 'a' cannot refer to just anything. It can only refer to something which is P. But after the subderivation's assumption has been discharged, 'a' is arbitrary. Why? The rules tell us that 'a' is arbitrary in line 7 because line 7 is not governed by any premises or assumptions in which 'a' occurs. But to make this more intuitive, notice that I could have just as well constructed the same subderivation using the name 'b' instead of 'a', using ⊃E to write 'Pb̂ ⊃ Qb̂' on line 7. Or I could have used 'c', 'd', or any other name. This is why 'a' is arbitrary in line 7. I could have arrived at a conditional in line 7 using any name I liked instead of using 'a'.

Some students get annoyed and frustrated by having to learn when to put a hat on a name and when to leave it off. But it's worth the effort to learn. Once you master the hat trick, ∀I is simple: You can apply ∀I whenever you have a name with a hat. Not otherwise.

EXERCISES

5–4. There is a mistake in the following derivation. Put on hats where they belong, and write in the justification for those steps which are justified. Identify and explain the mistake.

1	(∀x)(Bx ⊃ Cx)	P
2	Be ⊃ Ce	
3	Be	A
4	Be ⊃ Ce	
5	Ce	
6	(∀x)Ce	
7	Be ⊃ (∀x)Cx	

5–5. Provide derivations which establish the validity of the following arguments. Be sure you don't mix up sentences which are a quantification of a sentence formed with a '&', a 'v', or a '⊃' with compounds formed with a '&', a 'v', or a '⊃', the components of which are quantified sentences. For example, '(∀x)(Px & Qa)' is a universally quantified sentence to which you may apply ∀E. '(∀x)Px & Qa' is a conjunction to which you may apply &E but not ∀E.

a) (∀x)(Fx & Gx)
———
(∀x)Fx

b) (∀x)(Mx ⊃ Nx)
(∀x)Mx
———
(∀x)Nx

c) A
———
(∀x)(A v Nx)

d) (∀x)Hx & (∀x)Qx
———
(∀x)(Hx & Qx)

e) (∀x)(Kxm & Kmx)
———
(∀x)Kxm & (∀x)Kmx

f) (∀x)(Fx v Gx)
(∀x)(Fx ⊃ Gx)
———
(∀x)Gx

g) (∀x)~Px v C
———
(∀x)(~Px v C)

h) (∀x)(Rxb ⊃ Rax)
———
(∀x)Rxb ⊃ (∀x)Rax

i) (∀x)(Gxh ⊃ Gxm)
———
(∀x)(~Gxm ⊃ ~Gxh)

j) (∀x)(Mx ⊃ Nx)
(∀x)(Nx ⊃ Ox)
———
(∀x)(Mx ⊃ Ox)

k) T ⊃ (∀x)Mdx
———
(∀x)(T ⊃ Mdx)

l) (∀x)(Hff ⊃ Lxx)
———
Hff ⊃ (∀x)Lxx

m) (∀x)Px v (∀x)Qx
———
(∀x)(Px v Qx)

n) (∀x)Hx
(∃x)Hx ⊃ (∀x)(Hx ⊃ Jx)
———
(∀x)Jx

o) (∀x)(Sx ≡ Ox)
———
(∀x)Sx ≡ (∀x)Ox

p) (∃x)Px ⊃ A
———
(∀x)(Px ⊃ A)

q) ~(∃x)Px
———
(∀x)~Px

r) ~(∀x)Px
———
(∃x)~Px

s) (∀x)Px ⊃ A
———
(∃x)(Px ⊃ A)

t) ~(∀x)(Jx ⊃ ~Kx)
———
(∃x)(Jx & Kx)

u) ~(∃x)Qx v H
———
(∀x)(~Qx v H)

v) ~(∃x)Dx
———
(∀x)(Dx ⊃ Kx)

5–6. THE EXISTENTIAL ELIMINATION RULE

∀I and ∃E are difficult rules. Many of you will have to work patiently over this material a number of times before you understand them clearly. But if you have at least a fair understanding of ∀I, we can proceed to ∃E because ultimately these two rules need to be understood together.

Let's go back to the first example in section 5–4: Everyone likes either rock music or country/western. Someone does not like rock. So someone likes country/western. I will symbolize this as

(∀x)(Rx v Cx)
(∃x)~Rx
———
(∃x)Cx

In informally showing this argument's validity, I used 'Doe', which I will now write just as 'd', as a stand-in name for the unknown "someone" who does not like rock. But I must be careful in at least two respects:

i) I must not allow myself to apply ∀I to the stand-in name, 'd'. Otherwise, I could argue from '(∃x)~Rx' to '~Rd' to '(∀x)~Rx'. In short, I have to make sure that such a name never gets a hat.

ii) When I introduce the stand-in name, 'd', I must not be assuming anything else about the thing to which 'd' refers other than that '~R' is true of it.

It's going to take a few paragraphs to explain how we will meet these two requirements. To help you follow these paragraphs, I'll begin by writing down our example's derivation, which you should not expect to understand until you have read the explanation. Refer back to this example as you read:

(∀x)(Rx v Cx)
(∃x)~Rx
———
(∃x)Cx

1		(∀x)(Rx v Cx)	P
2		(∃x)~Rx	P
3	d	~Rd	A
4		(∀x)(Rx v Cx)	1, R
5		Rd v Cd	4, ∀E
6		Cd	3, 5, vE
7		(∃x)Cx	6, ∃I
8		(∃x)Cx	2, 3–7, ∃E

I propose to argue from the premise, '(∃x)~Rx', by using the stand-in name, 'd'. I will say about the thing named by 'd' what '(∃x)~Rx' says about "someone". But I must be sure that 'd' never gets a hat. How can I guarantee that? Well, names that occur in assumptions can't get hats anywhere in the subderivation governed by the assumption. So we can guarantee that 'd' won't get a hat by introducing it as an assumption of a subderivation and insisting that 'd' **never occur outside** that subderivation. This is what I did in line 3. '~Rd' appears as the subderivation's assumption, and the 'd' written just to the left of the scope line signals the requirement that 'd' be an *Isolated Name.* That is to say, 'd' is isolated in the subderivation the scope line of which is marked with the 'd'. An isolated name may never appear outside its subderivation.

Introducing 'd' in the assumption of a subderivation might seem a little strange. I encounter the sentence, '(∃x)~Rx', on a derivation. I reason: Let's assume that this thing of which '~R' is true is called 'd', and let's record this assumption by starting a subderivation with '~Rd' as its assumption, and see what we can derive. Why could this seem strange? Because if I already know '(∃x)~Rx', no further assumption is involved in assuming that there is something of which '~R' is true. But, in a sense, I do make a new assumption in assuming that this thing is called 'd'. It turns out that this sense of making a special assumption is just what we need.

By making 'd' occur in the assumption of a subderivation, and insisting that 'd' be isolated, that it appear **only** in the subderivation, I guarantee that 'd' never gets a hat. But this move also accomplishes our other requirement: If 'd' occurs only in the subderivation, 'd' cannot occur in any outer premise or assumption.

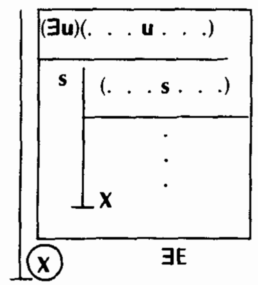
Now let's see how the overall strategy works. Look at the argument's subderivation, steps 3–7. You see that, with the help of reiterated premise 1, from '~Rd' I have derived '(∃x)Cx'. But neither 1 nor the conclusion '(∃x)Cx' uses the name 'd'. Thus, in this subderivation, the fact that I used the name 'd' was immaterial. I could have used any other name not appearing in the outer derivation. The real force of the assumption '~Rd' is that **there exists something of which '~R' is true** (there is someone who does not like rock). But that there exists something of which '~R' is true has already been given to me in line 2! Since the real force of the assumption of line 3 is that there exists something of which '~R' is true, and since I am already given this fact in line 2, I don't really need the assumption 3. I can discharge it. In other words, if I am given the truth of lines 1 and 2, I know that the conclusion of the subderivation, 7, must also be true, and I can enter 7 as a further conclusion of the outer derivation.

It is essential, however, that 'd' not appear in line 7. If 'd' appeared in the final conclusion of the subderivation, then I would not be allowed to discharge the assumption and enter this final conclusion in the outer derivation. For if 'd' appeared in the subderivation's final conclusion, I would

be relying, not just on the assumption that '~R' was true of something, but on the assumption that this thing was named by 'd'.

The example's pattern of reasoning works perfectly generally. Here is how we make it precise:

> A name is *Isolated in a Subderivation* if it does not occur outside the subderivation. We mark the isolation of a name by writing the name at the top left of the scope line of its subderivation. In applying this definition, remember that a sub-sub-derivation of a subderivation counts as part of the subderivation.
>
> *Existential Elimination Rule:* Suppose a sentence of the form (∃**u**)(. . . **u** . . .) appears in a derivation, as does a subderivation with assumption (. . . **s** . . .), a substitution instance of (∃**u**)(. . . **u** . . .). Also suppose that **s** is isolated in this subderivation. If **X** is any of the subderivation's conclusions in which **s** does not occur, you are licensed to draw **X** as a further conclusion in the outer derivation, anywhere below the sentence (∃**u**)(. . . **u** . . .) and below the subderivation. Expressed with a diagram:



Where (. . . **s** . . .) is a substitution instance of (∃**u**) (. . . **u** . . .) and **s** is isolated in the sub-derivation.

When you annotate your application of the ∃E rule, cite the line number of the existentially quantified sentence and the inclusive line numbers of the subderivation to which you appeal in applying the rule.

You should be absolutely clear about three facets of this rule. I will illustrate all three.

> Suppose the ∃E rule has been applied, licensing the new conclusion, **X**, by appeal to a sentence of the form (∃**u**)(. . . **u** . . .) and a subderivation beginning with assumption (. . . **s** . . .):
>
> 1) **s** cannot occur in any premise or prior assumption governing the subderivation,
> 2) **s** cannot occur in (∃**u**)(. . . **u** . . .), and
> 3) **s** cannot occur in **X**.

All three restrictions are automatically enforced by requiring **s** to be isolated in the subderivation. (Make sure you understand why this is correct.) Some texts formulate the ∃E rule by imposing these three requirements separately instead of requiring that s be isolated. If you reach chapter 15, you will learn that these three restrictions are really all the work that the isolation requirement needs to do. But, since it is always easy to pick a name which is unique to a subderivation, I think it is easier simply to require that s be isolated in the subderivation.

Let us see how things go wrong if we violate the isolation requirement in any of these three ways. For the first, consider:

Ca
(∃x)Bx
――――――― (Invalid!)
(∃x)(Cx & Bx)

1	Ca	*P*
2	(∃x)Bx	*P*
3	a | Ba	*A*
4	| Ca	1, *R*
5	| Ca & Ba	3, 4, &I
6	| (∃x)(Cx & Bx)	5, ∃I
7	(∃x)(Cx & Bx)	**Mistaken attempt to apply ∃E to 2 and 3–6. 'a' occurs in premise 1 and is not isolated in the subderivation.**

From the fact that Adam is clever and someone (it may well not be Adam) is blond, it does not follow that any **one** person is both clever and blond.

Now let's see what happens if one violates the isolation requirement in the second way:

(∀x)(∃y)Lxy
――――――― (Invalid!)
(∃x)Lxx

1	(∀x)(∃y)Lxy	*P*
2	(∃y)Lây	1, ∀E
3	a | Laa	*A*
4	| (∃x)Lxx	3, ∃I
5	(∃x)Lxx	**Mistaken attempt to apply ∃E to 2 and 3–4. 'a' occurs in 2 and is not isolated in the subderivation.**

From the fact that everyone loves someone, it certainly does not follow that someone loves themself.

And, for violation of the isolation requirement in the third way:

(∃x)Bx
――――――― **(Invalid)**
(∀x)Bx

1	(∃x)Bx	*P*
2	a | Ba	*A*
3	| Ba	2, *R*
4	Bâ	
5	(∀x)Bx	**Mistaken attempt to apply ∃E to 1 and 2–3. 'a' occurs in 4 and is not isolated in the subderivation.**

From the fact that someone is blond, it will never follow that everyone is blond.

One more example will illustrate the point about a sub-sub-derivation being part of a subderivation. The following derivation is completely correct:

1	(∀x)(Cx ⊃ ~Bx)	*P*
2	(∃x)Bx	*P*
3	d | Bd	*A*
4	| | Cd	*A*
5	| | (∀x)(Cx ⊃ ~Bx)	1, *R*
6	| | Cd ⊃ ~Bd	5, ∀E
7	| | ~Bd	4, 6, ⊃E
8	| | Bd (3)	3, *R*
9	| ~Cd	4–8, ~I
10	| (∃x)~Cx (2)	9, ∃I
11	(∃x)~Cx (1)	2, 3–10, ∃E

You might worry about this derivation: If 'd' is supposed to be isolated in subderivation 2, how can it legitimately get into sub-sub-derivation 3?

A subderivation is always **part** of the derivation in which it occurs, and the same holds between a sub-sub-derivation and the subderivation in which it occurs. We have already encountered this fact in noting that the premises and assumptions of a derivation or subderivation always apply to the derivation's subderivations, its sub-sub-derivations, and so on.

Now apply this idea about parts to the occurrence of 'd' in sub-sub-derivation 3 above: When I say that a name is isolated in a subderivation I mean that the name can occur in the subderivation **and all its parts,** but the name cannot occur outside the subderivation.

Here is another way to think about this issue: The 'd' at the scope line of the second derivation means that 'd' occurs to the right of the scope line and not to the left. But the scope line of subderivation 3 is not marked by any name. So the notation permits you to use 'd' to the right of this line also.

I hope that you are now beginning to understand the rules for quantifiers. If your grasp still feels shaky, the best way to understand the rules better is to go back and forth between reading the explanations and practicing with the problems. As you do so, try to keep in mind why the rules are supposed to work. Struggle to see why the rules are truth preserving. By striving to understand the rules, as opposed to merely learning them as cookbook recipes, you will learn them better, and you will also have more fun.

EXERCISES

5–6. There is one or more mistakes in the following derivation. Write the hats where they belong, justify the steps that can be justified, and identify and explain the mistake, or mistakes.

1	$(\forall y)(\exists x)Lxy$	*P*
2	$(\exists x)Lxb$	
3	$\quad Lab$	*A*
4	$\quad (\forall y)Lay$	
5	$\quad (\exists x)(\forall y)Lxy$	
6	$(\exists x)(\forall y)Lxy$	

5–7. Provide derivations which establish the validity of the following arguments:

a) $\begin{array}{l}(\exists x)Ix \\ (\forall x)(Ix \supset Jx) \\ \hline (\exists x)Jx\end{array}$

b) $\begin{array}{l}(\exists x)(A \supset Px) \\ \hline A \supset (\exists x)Px\end{array}$

c) $\begin{array}{l}(\exists x)Hmx \\ (\forall x)(\sim Hmx \vee Gxn) \\ \hline (\exists x)Gxn\end{array}$

d) $\begin{array}{l}(\exists x)(Cfx \,\&\, Cxf) \\ \hline (\exists x)Cfx \,\&\, (\exists x)Cxf\end{array}$

e) $\begin{array}{l}(\exists x)(Px \vee Qx) \\ \hline (\exists x)Px \vee (\exists x)Qx\end{array}$

f) $\begin{array}{l}(\exists x)Px \vee (\exists x)Qx \\ \hline (\exists x)(Px \vee Qx)\end{array}$

g) $\begin{array}{l}(\exists x)(Px \supset A) \\ \hline (\forall x)Px \supset A\end{array}$

h) $\begin{array}{l}(\forall x)(Px \supset A) \\ \hline (\exists x)Px \supset A\end{array}$

i) $\begin{array}{l}(\exists x)(Lxa \equiv Lex) \\ (\forall x)Lxa \\ \hline (\exists x)Lex\end{array}$

j) $\begin{array}{l}(\forall x)(Gsx \supset \sim Gxs) \\ \hline (\exists x)Gxs \supset (\exists x)\sim Gsx\end{array}$

k) $\begin{array}{l}(\exists x)(Px \vee Qx) \\ (\forall x)(Px \supset Kx) \\ (\forall x)(Qx \supset Kx) \\ \hline (\exists x)Kx\end{array}$

l) $\begin{array}{l}(\exists x)(\sim Mxt \vee Mtx) \\ (\exists x)(Mtx \supset Axx) \\ \hline (\exists x)(\sim Mxt \vee Axx)\end{array}$

m) $\begin{array}{l}(\exists x)Hxg \vee (\exists x)Nxf \\ (\forall x)(Hxg \supset Cx) \\ (\forall x)(Nxf \supset Cx) \\ \hline (\exists x)Cx\end{array}$

n) $\begin{array}{l}(\forall x)[(Fx \vee Gx) \supset Lxx] \\ (\exists x)\sim Lxx \\ \hline (\exists x)\sim Fx \,\&\, (\exists x)\sim Gx\end{array}$

o) $\begin{array}{l}(\forall x)[Fx \supset (Rxa \vee Rax)] \\ (\exists x)\sim Rxa \\ \hline (\forall x)\sim Rax \supset (\exists x)\sim Fx\end{array}$

p) $\begin{array}{l}(\exists x)Qxj \\ (\exists x)(Qxj \vee Dgx) \supset (\forall x)Dgx \\ \hline (\forall x)(Dgx \vee Qjx)\end{array}$

q) $\begin{array}{l}(\forall x)\sim Fx \\ \hline \sim(\exists x)Fx\end{array}$

r) $\begin{array}{l}(\exists x)\sim Fx \\ \hline \sim(\forall x)Fx\end{array}$

s) $\begin{array}{l}(\forall x)(Jxx \supset \sim Jxf) \\ \hline \sim(\exists x)(Jxx \,\&\, Jxf)\end{array}$

t) $\begin{array}{l}(\exists x)Px \vee Qa \\ (\forall x)\sim Px \\ \hline (\exists x)Qx\end{array}$

u) $\begin{array}{l}A \supset (\exists x)Px \\ \hline (\exists x)(A \supset Px)\end{array}$

5–8. Are you bothered by the fact that ∃E requires use of a subderivation with an instance of the existentially quantified sentence as its assumption? Good news! Here is an alternate version of ∃E which does not require starting a subderivation:

$(\exists \mathbf{u})(\ldots \mathbf{u} \ldots)$	
$(\forall \mathbf{u})[(\ldots \mathbf{u} \ldots) \supset \mathbf{X}]$	
Ⓧ	∃E

Show that, in the presence of the other rules, this version is exchangeable with the ∃E rule given in the text. That is, show that the above is a derived rule if we start with the rules given in the text. And show that if we start with all the rules in the text except for ∃E, and if we use the above rule for ∃E, then the ∃E of the text is a derived rule.

CHAPTER SUMMARY EXERCISES

Here is a list of important terms from this chapter. Explain them briefly and record your explanations in your notebook:

a) Truth Preserving Rule of Inference
b) Sound
c) Complete
d) Stand-in Name
e) Govern
f) Arbitrary Occurrence
g) Existential Generalization
h) Universal Generalization
i) Isolated Name
j) Existential Introduction Rule
k) Existential Elimination Rule
l) Universal Introduction Rule
m) Universal Elimination Rule