1. INTRODUCTION

Having discussed the grammar of predicate logic and its relation to English, we now turn to the problem of argument validity in predicate logic.

Recall that, in Chapter 5, we developed the technique of formal derivation in the context of sentential logic – specifically System SL. This is a technique to deduce conclusions from premises in sentential logic. In particular, if an argument is valid in sentential logic, then we can (in principle) construct a derivation of its conclusion from its premises in System SL, and if it is invalid, then we cannot construct such a derivation.

In the present chapter, we examine the corresponding deductive system for predicate logic – what will be called System PL (short for ‘predicate logic’). As you might expect, since the syntax (grammar) of predicate logic is considerably more complex than the syntax of sentential logic, the method of derivation in System PL is correspondingly more complex than System SL.

On the other hand, anyone who has already mastered sentential logic derivations can also master predicate logic derivations. The transition primarily involves (1) getting used to the new symbols and (2) practicing doing the new derivations (just like in sentential logic!). The practical converse, unfortunately, is also true. Anyone who hasn't already mastered sentential logic derivations will have tremendous difficulty with predicate logic derivations. Of course, it's still not too late to figure out sentential derivations!

2. THE RULES OF SENTENTIAL LOGIC

We begin by stating the first principle of predicate logic derivations. To wit,

Every rule of System SL (sentential logic) is also a rule of System PL (predicate logic).

The converse is not true; as we shall see in later sections, there are several rules peculiar to predicate logic, i.e., rules that do not arise in sentential logic.

Since predicate logic adopts all the derivation rules of sentential logic, it is a good idea to review the salient features of sentential logic derivations.

First of all, the derivation rules divide into two categories; on the one hand, there are inference rules, which are upward-oriented; on the other hand, there are show rules, which are downward-oriented.

There are numerous inference rules, but they divide into four basic categories.
(I1) Introduction Rules (In-Rules):
&I, ∨I, ←I, XI

(I2) Simple Elimination Rules (Out-Rules):
&O, ∨O, →O, ←O, XO

(I3) Negation Elimination Rules (Tilde-Out-Rules):
¬&O, ¬∨O, ¬→O, ¬←O

(I4) Double Negation, Repetition

In addition, there are four show-rules.

(S1) Direct Derivation
(S2) Conditional Derivation
(S3) Indirect Derivation (First Form)
(S4) Indirect Derivation (Second Form)

As noted at the beginning of the current section, every rule of sentential logic is still operative in predicate logic. However, when applied to predicate logic, the rules of sentential logic look somewhat different, but only because the syntax of predicate logic is different. In particular, instead of formulas that involve only sentential letters and connectives, we are now faced with formulas that involve predicates and quantifiers. Accordingly, when we apply the sentential logic rules to the new formulas, they look somewhat different.

For example, the following are all instances of the arrow-out rule, applied to predicate logic formulas.

1. Fa → Ga
   Fa
   ________
   Ga

2. ∀xFx → ∀xGx
   ∀xFx
   ________
   ∀xGx

3. Fa → Ga
   ~Ga
   ________
   ~Fa

4. ∀x(Fx → Gx) → ∃xFx
   ~∃xFx
   ________
   ~∀x(Fx → Gx)
Thus, in moving from sentential logic to predicate logic, one must first become accustomed to applying the old inference rules to new formulas, as in examples (1)-(4).

The same thing applies to the show rules of sentential logic, and their associated derivation strategies, which remain operative in predicate logic. Just as before, to show a conditional formula, one uses conditional derivation; similarly, to show a negation, or disjunction, or atomic formula, one uses indirect derivation. The only difference is that one must learn to apply these strategies to predicate logic formulas.

For example, consider the following show lines.

(1) SHOW: $Fa \rightarrow Ga$
(2) SHOW: $\forall x Fx \rightarrow \forall x Gx$
(3) SHOW: $\neg Fa$
(4) SHOW: $\neg \exists x (Fx \& Gx)$
(5) SHOW: $Rab$
(6) SHOW: $\forall x Fx \lor \forall x Gx$

Every one of these is a formula for which we already have a ready-made derivation strategy. In each case, either the formula is atomic, or its main connective is a sentential logic connective.

The formulas in (1) and (2) are conditionals, so we use conditional derivation, as follows.

(1) SHOW: $Fa \rightarrow Ga$
   $Fa$ As
   SHOW: $Ga$ ??

(2) SHOW: $\forall x Fx \rightarrow \forall x Gx$
   $\forall x Fx$ As
   SHOW: $\forall x Gx$ ??

The formulas in (3) and (4) are negations, so we use indirect derivation of the first form, as follows.

(3) SHOW: $\neg Fa$
   $Fa$ As
   SHOW: $\times$ ??

(4) SHOW: $\neg \exists x (Fx \& Gx)$
   $\exists x (Fx \& Gx)$ As
   SHOW: $\times$ ??

The formula in (5) is atomic, so we use indirect derivation, supposing that a direct derivation doesn't look promising.
Finally, the formula in (6) is a disjunction, so we use indirect derivation, along with tilde-wedge-out, as follows.

(6) SHOW: $\forall x Fx \lor \forall x Gx$  
\[
\sim (\forall x Fx \lor \forall x Gx) \quad \text{ID}
\]
\[
\sim \forall x Fx \quad \sim \forall x Gx
\]

In conclusion, since predicate logic subsumes sentential logic, all the derivation techniques we have developed for the latter can be transferred to predicate logic. On the other hand, given the additional logical apparatus of predicate logic, in the form of quantifiers, we need additional derivation techniques to deal successfully with predicate logic arguments.

3. THE RULES OF PREDICATE LOGIC: AN OVERVIEW

If we confined ourselves to the rules of sentential logic, we would be unable to derive any interesting conclusions from our premises. All we could derive would be conclusions that follow purely in virtue of sentential logic. On the other hand, as noted at the beginning of Chapter 6, there are valid arguments that can't be shown to be valid using only the resources of sentential logic.

Consider the following (valid) arguments.

\[
\forall x (Fx \rightarrow Hx) \quad \text{every Freshman is Happy}
\]
\[
Fc \quad \text{Chris is a Freshman}
\]
\[
\sim \sim Fc \quad \text{Chris is Happy}
\]

\[
\forall x (Sx \rightarrow Px) \quad \text{every Snake is Poisonous}
\]
\[
\forall x ([Sx \& Px] \rightarrow Dx) \quad \text{every Poisonous Snake is Dangerous}
\]
\[
Sm \quad \text{Max is a Snake}
\]
\[
Dm \quad \text{Max is Dangerous}
\]

In either example, if we try to derive the conclusion from the premises, we are stuck very quickly, for we have no means of dealing with those premises that are universal formulas. They are not conditionals, so we can't use arrow-out; they are not conjunctions, so we can't use ampersand-out, etc., etc.

Sentential logic does not provide a rule for dealing with such formulas, so we need special rules for the added logical structure of predicate logic.
In choosing a set of rules for predicate logic, one goal is to follow the general pattern established in sentential logic. In particular, according to this pattern, for each connective, we have a rule for introducing that connective, and a rule for eliminating that connective. Also, for each two-place connective, we have a rule for eliminating negations of formulas with that connective. In sentential logic, with the exception of the conditional for which there is no introduction rule, every connective has both an in-rule and an out-rule, and every connective has a tilde-out-rule. There is no arrow-in inference rule; rather, there is an arrow show-rule, namely, conditional derivation.

In regard to derivations, moving from sentential logic to predicate logic basically involves adding two sets of one-place connectives; on the one hand, there are the universal quantifiers – \( \forall x, \forall y, \forall z \); on the other hand, there are the existential quantifiers – \( \exists x, \exists y, \exists z \). So, following the general pattern for rules, just as we have three rules for each sentential connective, we correspondingly have three rules for universals, and three rules for existentials, which are summarized as follows.

**Universal Rules**

1. Universal Derivation (UD)
2. Universal-Out (\( \forall O \))
3. Tilde-Universal-Out (\( \sim \forall O \))

**Existential Rules**

1. Existential-In (\( \exists I \))
2. Existential-Out (\( \exists O \))
3. Tilde-Existential-Out (\( \sim \exists O \))

Thus, predicate logic employs six rules, in addition to all of the rules of sentential logic. Notice carefully, that five of the rules are inference rules (upward-oriented rules), but one of them (universal derivation) is a show-rule (downward-oriented rule), much like conditional derivation. Indeed, universal derivation plays a role in predicate logic very similar to the role of conditional derivation in sentential logic.

[Note: Technically speaking, Existential-Out (\( \exists O \)) is an assumption rule, rather than a true inference rule. See Section 10 for an explanation.]
4. UNIVERSAL OUT

The first, and easiest, rule we examine is universal-elimination (universal-out, for short). As its name suggests, it is a rule designed to decompose any formula whose main connective is a universal quantifier (i.e., \(\forall x\), \(\forall y\), or \(\forall z\)).

The official statement of the rule goes as follows.

**Universal-Out (\(\forall O\))**

If one has an available line that is a universal formula, which is to say that it has the form \(\forall v F[v]\), where \(v\) is any variable, and \(F[v]\) is any formula in which \(v\) occurs free, then one is entitled to infer any substitution instance of \(F[v]\).

In symbols, this may be pictorially summarized as follows.

\[
\begin{align*}
\forall O: \quad & \forall v F[v] \\
\implies & \quad F[n]
\end{align*}
\]

Here,

1. \(v\) is any variable (x, y, z);
2. \(n\) is any name (a-w);
3. \(F[v]\) is any formula, and \(F[n]\) is the formula that results when \(n\) is substituted for every occurrence of \(v\) that is free in \(F[v]\).

In order to understand this rule, it is best to look at a few examples.

**Example 1:** \(\forall xFx\)

This is by far the easiest example. In this \(v\) is \(x\), and \(F[v]\) is \(Fx\). To obtain a substitution instance of \(Fx\) one simply replaces \(x\) by a name, any name. Thus, all of the following follow by \(\forall O\):

\(Fa, Fb, Fc, Fd, etc.\)

**Example 2:** \(\forall yRyk\)

This is almost as easy. In this \(v\) is \(y\), and \(F[v]\) is \(Ryk\). To obtain a substitution instance of \(Ryk\) one simply replaces \(y\) by a name, any name. Thus, all of the following follow by \(\forall O\):

\(Rak, Rbk, Rck, Rdk, etc.\)
In both of these examples, the intuition behind the rule is quite straightforward. In Example 1, the premise says that everything is an F; but if everything is an F, then any particular thing we care to mention is an F, so a is an F, b is an F, c is an F, etc. Similarly, in Example 2, the premise says that everything bears relation R to k (for example, everyone respects Kay); but if everything bears R to k, then any particular thing we care to mention bears R to k, so a bears R to k, b bears R to k, etc.

In examples 1 and 2, the formula F[v] is atomic. In the remaining examples, F[v] is molecular.

**Example 3:** \(\forall x(Fx \rightarrow Gx)\)

In this v is x, and F[v] is Fx \(\rightarrow\) Gx. To obtain a substitution instance, we replace both occurrences of x by a name, the same name for both occurrences. Thus, all of the following follow by \(\forall O\).

\[F_a \rightarrow G_a, F_b \rightarrow G_b, F_c \rightarrow G_c, \text{ etc.}\]

In this example, the intuition underlying the rule may be less clear than in the first two examples. The premise may be read in many ways in English, some more colloquial than others.

\[
\begin{align*}
(r1) & \text{ every F is G} \\
(r2) & \text{ everything is G if it's F} \\
(r3) & \text{ everything is such that: if it is F, then it is G.}
\end{align*}
\]

The last reading (r3) says that everything has a certain property, namely, that if it is F then it is G. But if everything has this property, then any particular thing we care to mention has the property. So a has the property, b has the property, etc. But to say that a has the property is simply to say that if a is F then a is G; to say that b has the property is to say that if b is F then b is G. Both of these are applications of universal-out.

**Example 4:** \(\forall x \exists y R_{xy}\)

Here, v is x, and F[v] is \(\exists y R_{xy}\). To obtain a substitution instance of \(\exists y R_{xy}\), one replaces the one and only occurrence of x by a name, any name. Thus, the following all follow by \(\forall O\).

\[\exists y Ray, \exists y Rby, \exists y Rcy, \exists y Rdy, \text{ etc.}\]

The premise says that everything bears relation R to something or other. For example, it translates the English sentence ‘everyone respects someone (or other)’. But if everyone respects someone (or other), then anyone you care to mention respects someone, so a respects someone, b respects someone, etc.

**Example 5:** \(\forall x(Fx \rightarrow \forall x Gx)\)

Here, v is x, and F[v] is Fx \(\rightarrow\) \(\forall x Gx\). To obtain a substitution instance, one replaces every free occurrence of x in Fx \(\rightarrow\) \(\forall x Gx\) by a name. In this example, the
first occurrence is free, but the remaining two are not, so we only replace the first occurrence. Thus, the following all follow by $\forall O$.

$$Fa \rightarrow \forall xGx, \ Fb \rightarrow \forall xGx, \ Fc \rightarrow \forall xGx,$$ etc.

This example is complicated by the presence of a second quantifier governing the same variable, so we have to be especially careful in applying $\forall O$. Nevertheless, one's intuitions are not violated. The premise says that if anyone is an F then everyone is a G (recall the distinction between 'if any' and 'if every'). From this it follows that if a is an F then everyone is a G, and if b is an F then everyone is a G, etc. But that is precisely what we get when we apply $\forall O$ to the premise.

5. POTENTIAL ERRORS IN APPLYING UNIVERSAL-OUT

There are basically two ways in which one can misapply the rule universal-out: (1) improper substitution; (2) improper application.

In the case of improper substitution, the rule is applied to an appropriate formula, namely, a universal, but an error is made in performing the substitution. Refer to the Appendix concerning correct and incorrect substitution instances. The following are a few examples of improper substitution.

1. $\exists xRxx$ ; to infer $Rax, Rab, Rba$ WRONG!!!
2. $\exists x(Fx \rightarrow Gx)$; to infer $Fa \rightarrow Gb, Fb \rightarrow Gc$ WRONG!!!
3. $\exists x(Fx \rightarrow \exists xGx)$; to infer $Fa \rightarrow \exists xGa, Fa \rightarrow \exists xGa$ WRONG!!!

In the case of improper application, one attempts to apply the rule to a line that does not have the appropriate form. Universal-out, as its name is intended to suggest, applies to universal formulas, not to atomic formulas, or existentials, or negations, or conditionals, or biconditional, or conjunctions, or disjunctions.

Recall, in this connection, a very important principle.

**INFORMATION RULES APPLY EXCLUSIVELY TO WHOLE LINES, NOT TO PIECES OF LINES.**

The following are examples of improper application of universal-out.

4. $\exists xFx \rightarrow \exists xGx$
   - to infer $Fa \rightarrow \exists xGx$ WRONG!!!
   - to infer $\exists xFx \rightarrow Gx$ WRONG!!!
   - to infer $Fa \rightarrow Gb$ WRONG!!!
In each case, the error is the same – specifically, applying universal-out to a formula that does not have the appropriate form. Now, the formula in question is not a universal, but is rather a conditional; so the appropriate elimination rule is not universal-out, but rather arrow-out (which, of course, requires an additional premise).

\[ \neg \forall xFx \]

to infer \( \neg Fa, \neg Fb, \) or \( \neg Fc \)  \text{WRONG!!!} 

Once again, the error involves applying universal-out to a formula that is not a universal. In this case, the formula is a negation. Later, we will have a rule – tilde-universal-out – designed specifically for formulas of this form.

The moral is that you must be able to recognize the major connective of a formula; is it an atomic formula, a conjunction, a disjunction, a conditional, a biconditional, a negation, a universal, or an existential? Otherwise, you can't apply the rules successfully, and hence you can't construct proper derivations.

Of course, sometimes misapplying a rule produces a valid conclusion. Take the following example.

\[ \forall xFx \rightarrow \forall xGx \]

to infer \( \forall xFx \rightarrow Ga \)

to infer \( \forall xFx \rightarrow Gb \)

e tc.

All of these inferences correspond to valid arguments. But many arguments are valid! The question, at the moment, is whether the inference is an instance of universal out. These inferences are not. In order to show that \( \forall xFx \rightarrow Ga \) follows from \( \forall xFx \rightarrow \forall xGx \), one must construct a derivation of the conclusion from the premise.

In the next section, we examine this particular derivation, as well as a number of others that employ our new tool, universal-out.

### 6. EXAMPLES OF DERIVATIONS USING UNIVERSAL-OUT

Having figured out the universal-out rule, we next look at examples of derivations in which this rule is used. We start with the arguments at the beginning of Section 3.

**Example 1**

\[ \forall x(Fx \rightarrow Hx) \] Pr  
\[ Fc \] Pr  
\[ \text{SHOW: } Hc \] DD  
\[ Fc \rightarrow Hc \] 1,\( \forall O \)  
\[ Hc \] 2,4,\( \rightarrow O \)
Example 2

(1) \( \forall x (Sx \rightarrow Px) \) \hspace{1cm} \text{Pr}
(2) \( \forall x ([Sx & Px] \rightarrow Dx) \) \hspace{1cm} \text{Pr}
(3) \( \text{Sm} \) \hspace{1cm} \text{Pr}
(4) \( \text{SHOW: } \text{Dm} \) \hspace{1cm} \text{DD}
(5) \( \text{Sm} \rightarrow \text{Pm} \) \hspace{1cm} 1, \forall \text{O}
(6) \( (\text{Sm} & \text{Pm}) \rightarrow \text{Dm} \) \hspace{1cm} 2, \forall \text{O}
(7) \( \text{Pm} \) \hspace{1cm} 3,5, \rightarrow \text{O}
(8) \( \text{Sm} & \text{Pm} \) \hspace{1cm} 3,7, \&I
(9) \( \text{Dm} \) \hspace{1cm} 6,8, \rightarrow \text{O}

The above two examples are quite simple, but they illustrate an important strategic principle for doing derivations in predicate logic.

**REDUCE THE PROBLEM TO A POINT WHERE YOU CAN APPLY RULES OF SENTENTIAL LOGIC.**

In each of the above examples, we reduce the problem to the point where we can finish it by applying arrow-out.

Notice in the two derivations above that the tool – namely, universal-out – is specialized to the job at hand. According to universal-out, if we have a line of the form \( \forall v F[v] \), we are entitled to write down any instance of the formula \( F[v] \). So, for example, in line (4) of the first example, we are entitled to write down \( Fa \rightarrow Ha \), \( Fb \rightarrow Hb \), as well as a host of other formulas. But, of all the formulas we are entitled to write down, only one of them is of any use – namely, \( Fc \rightarrow Hc \).

Similarly, in the second example, we are entitled by universal-out to instantiate lines (1) and (2) respectively to any name we choose. But of all the permitted instantiations, only those that involve the name \( m \) are of any use.

To say that one is permitted to do something is quite different from saying that one must do it, or even that one should do it. At any given point in a game (say, chess), one is permitted to make any number of moves, but most of them are stupid (supposing one's goal is to win). A good chess player chooses good moves from among the legal moves. Similarly, a good derivation builder chooses good moves from among the legal moves. In the first example, it is certainly true that \( Fa \rightarrow Ga \) is a permitted step at line (4); but it is pointless because it makes no contribution whatsoever to completing the derivation.

By analogy, standing on your head until you have a splitting headache and are sick to your stomach is not against the law; it's just stupid.

In the examples above, the choice of one particular letter over any other letter as the letter of instantiation is natural and obvious. Other times, as you will later see, there are several names floating around in a derivation, and it may not be obvious which one to use at any given place. Under these circumstances, one must primarily use trial-and-error.
Let us look at some more examples. In the previous section, we looked at an argument that was obtained by a misapplication of universal-out. As noted there, the argument is valid, although it is not an instance of universal-out. Let us now show that it is indeed valid by deriving the conclusion from the premises.

**Example 3**

| (1) | \( \forall x Fx \rightarrow \forall x Gx \) | Pr |
| (2) | \( \text{SHOW: } \forall x Fx \rightarrow Ga \) | CD |
| (3) | \( \forall x Fx \) | As |
| (4) | \( \text{SHOW: } Ga \) | DD |
| (5) | \( \forall x Gx \) | 1,3,\( \rightarrow O \) |
| (6) | Ga | 5,\( \forall O \) |

Notice, in particular, that the formula in (2) is a conditional, and is accordingly shown by conditional derivation. You are, of course, already very familiar with conditional derivations; to show a conditional, you assume the antecedent and show the consequent.

The following is another example in which a sentential derivation strategy is employed.

**Example 4**

| (1) | \( \forall x (Fx \rightarrow Hx) \) | Pr |
| (2) | \( \neg Hb \) | Pr |
| (3) | \( \text{SHOW: } \neg \forall x Fx \) | ID |
| (4) | \( \forall x Fx \) | As |
| (5) | \( \text{SHOW: } \neg \) | DD |
| (6) | \( Fb \rightarrow Hb \) | 1,\( \forall O \) |
| (7) | \( Fb \) | 4,\( \forall O \) |
| (8) | \( Hb \) | 6,7,\( \rightarrow O \) |
| (9) | \( \neg \) | 2,8,\( \neg I \) |

In line (3), we have to show \( \neg \forall x Fx \); this is a negation, so we use a tried-and-true strategy for showing negations, namely indirect derivation. To show the negation of a formula, one assumes the formula negated and one shows the generic contradiction, \( \neg \).

We conclude this section by looking at a considerably more complex example, but still an example that requires only one special predicate logic rule, universal-out.
Chapter 8: Derivations in Predicate Logic

Example 5

(1) $\forall x(Fx \to \forall yRxy)$  Pr
(2) $\forall x\forall y(Rxy \to \forall zGz)$  Pr
(3) $\sim Gb$  Pr
(4) SHOW: $\sim Fa$  ID
(5) $Fa$  As
(6) SHOW: $\exists x$  DD
(7) $Fa \to \forall yRay$  1, $\forall O$
(8) $\forall yRay$  5, 7, $\to O$
(9) $Rab$  8, $\forall O$
(10) $\forall y(Ray \to \forall zGz)$  2, $\forall O$
(11) $Rab \to \forall zGz$  10, $\forall O$
(12) $\forall zGz$  9, 11, $\to O$
(13) $Gb$  12, $\forall O$
(14) $\exists x$  3, 13, $\exists I$

If you can figure out this derivation, better yet if you can reproduce it yourself, then you have truly mastered the universal-out rule!

7. EXISTENTIAL IN

Of the six rules of predicate logic that we are eventually going to have, we have now examined only one – universal-out. In the present section, we add one more to the list.

The new rule, existential introduction (existential-in, $\exists I$) is officially stated as follows.

**Existential-In ($\exists I$)**

If formula $F[n]$ is an available line, where $F[n]$ is a substitution instance of formula $F[v]$, then one is entitled to infer the existential formula $\exists vF[v]$.

In symbols, this may be pictorially summarized as follows.

$$\exists I: \begin{array}{c} F[n] \\ \hline \exists vF[v] \end{array}$$

Here,

(1) $v$ is any variable (x, y, z);

(2) $n$ is any name (a-w);