Modern Logic—The Origin and Development of Mathematical Logic

The modern revolution in Logic that has so comprehensively transformed our understanding of the subject began in the middle of the nineteenth century with one man—George Boole. Of course, as with the modern forms of many Classical sciences and branches of Philosophy, many other people—including logicians, mathematicians, and philosophers—made contributions to this particular revolution, both prior to and following the publication of Boole’s work. Such people as the Czech logician Bernard Bolzano (whose work, however, went largely unnoticed in his own lifetime), the British mathematician Sir William Hamilton, the British logician Augustus De Morgan, the American philosopher Charles S. Peirce, and of course the greatest European logician of the nineteenth century—the German logician Gottlob Frege—made such fundamental contributions to the subject that modern Logic certainly would not be what it is today if they had not achieved what they did. Nevertheless, given the historical contingencies that ultimately shaped the actual development of modern logic, George Boole is seen as playing such a key role that the main core of Logic (to which he and these other logicians gave form) is today commonly known as Boolean Algebra.

Throughout this explanation of the development of many aspects of Modern Logics by the Classical logicians, we have been modifying the structure and symbols of the Classical system to transform Classical argument forms into their Modern counterparts. For the most part, however, we have merely assumed the relevant relations between the two forms or expression, without actually explaining them. Before closing this topic, then, we must make explicit what has thus far remained merely implicit: the syntactic and Logical relationship between the syllogism and its Modern counterpart.

To begin, we recall that Aristotle’s “instructions” for interpreting a syllogism suggests that we see it as a type of Immediate Inference (my words, not his) in which the two Premises are reduced to a single Proposition, by means of Conjunction. Following this suggestion, the expanded syllogistic expressions below:
If:

Premise 1: 

(P ⊆ ∀M)

and:

Premise 2: 

(M ⊆ ∀S)

then:

Conclusion: 

P ⊆ ∀S

If:

Premise 1: 

(M ⊆ ∀P)

and:

Premise 2: 

(M ⊆ ∀S)

then:

P ⊆ ∀S

become instances of formal Implication similar (but not restricted) to the Rule of Formal Implication (aka Modus Ponens). In Modern form, then, the syllogism above become:

\[
\left[ (P \subseteq \forall M) \land (M \subseteq \forall S) \right] \Rightarrow (P \subseteq \forall S)
\]

and:

\[
\left[ (M \subseteq \forall P) \land (M \subseteq \forall S) \right] \Rightarrow (P \subseteq \forall S)
\]

in which the two Premises [Propositions inside indexed parentheses, thus: (...)₁] are combined (by “And”, whose symbol is \( \land \)) into a single Formal Implication bearing a syntactic structure known as the Conjunctive Normal Form. In addition, the Formal Implication that the Valid Syllogistic Form is intended to express is explicitly represented by its own symbol: ‘\( \Rightarrow \)’, which is read “formally implies”. Conjunction and Implication are the fundamental Logical relations of Aristotle’s Syllogistics, as well as of Modern Deductive Logic. In the Syllogistics, Conjunction is the syntactic relation between Premise 1 and Premise 2; in modern Deduction, Conjunction is a basic Logical Operator (not unlike ‘+’ in Arithmetic) for which Propositions (such as Premise 1 and Premise 2) serve as Operands (not unlike n, m, and x in the following operation: ‘\( n + m = x \)’). In Classical Logic, it is tacitly assumed that the function and “meaning” of ‘and’ is understood; as a natural syntactic element in Greek (and all) grammar, its use is taken for granted. In modern Logic, however, we find it convenient to define our basic Operators explicitly, and in terms of the various possible assignment of Logical Values—True or False. As a result, we define a conjunction as a compound proposition of two component propositions composed by means of a Conjunctive Operator (e.g. And).

\[
\begin{array}{c|c|c|c|}
\text{p} & \text{q} & \text{p} \land \text{q} \\
\hline
\text{T} & \text{T} & \text{T} \\
\text{T} & \text{F} & \text{F} \\
\text{F} & \text{T} & \text{F} \\
\text{F} & \text{F} & \text{F} \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|}
\text{p} & \text{q} & \text{p} \Rightarrow \text{q} \\
\hline
\text{T} & \text{T} & \text{T} \\
\text{T} & \text{F} & \text{F} \\
\text{F} & \text{T} & \text{T} \\
\text{F} & \text{F} & \text{T} \\
\end{array}
\]
As it turns out, in the Deductive System that we shall develop here (as in modern Deductive Logic in general), the notion of Validity is no longer necessary, for we can (and, here, will) develop a Deductive System in which we shall ensure that all Propositions are either Tautologous—which means (as we shall soon see) that the Propositions are always True—or are equivalent to (that is, “are somehow the same as”) Tautologous Propositions (so that these Propositions, too, will be always True). And in fact we could develop our Deductive System (and here, in part, do develop our Deductive System) in such a way as to totally eliminate the need for the concepts of True and False altogether. Such systems are said to be wholly Syntactical, because they are lacking any Semantics whatsoever (recalling that the rules of Grammar are nothing but the rules of Syntactics and the rules of Semantics). In this type of system, the notions of Validity and Soundness are totally irrelevant, because all the Propositions are Formally True (thus eliminating the question of Soundness), and all Arguments are necessarily Tautologous (which entails and thus eliminates the need for concerns of Validity).

Two Appositions: ≻p/p and ¬p

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>+</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>m</th>
<th>n</th>
<th>m × n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>m</th>
<th>n</th>
<th>m + n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ∧ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p / q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ∨ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>≻</th>
<th>¬</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Due to the ultimate nature of the Logical system that Boole developed, Boolean Algebra constitutes perhaps the most straightforward means of approaching what is now called a symbolic deductive system. These systems, each of which is basically an artificial language, contain three fundamental components—a Lexicon, a Grammar, and a Logic, the last of which is, as should be apparent, the deductive apparatus itself. The other two of these three components, the Lexicon (a collection of words) and the Grammar (a set of rules), are the main components of every language on Earth, either natural or artificial. In the English language, for example, the Lexicon consists of all of the words that anyone might use when speaking English, whether these words are in the dictionary or not (and yes, some words are so new—or so temporary—that they are not in any dictionary). And the Grammar of English is, as I am sure most of you remember, the lists of vocabulary and the set of rules for word-formation and sentence-formation that are taught in what we appropriately refer to as grammar school. In contrast to these, the deductive system is not present (strictly speaking) in natural English, or in any other natural language, for that matter, because, although English-language speakers (and most likely speakers of all other languages) certainly try to think and talk logically, no actual set of rules in any natural language represents a true deductive system (where the word ‘system’ is critical). Rather, what we find in natural languages is merely a loosely-knit collection of (mostly unspoken) logical rules that are typically used without anyone really thinking too much about them (and thus without our being very systematic). Only in our artificially constructed deductive systems, such as we will develop here, do we find the Logic itself laid out explicitly before us; and of course this is done intentionally, for the express purpose of investigating Logic.

Due to the nature of Logic itself, particularly with respect to the grammatical component, artificial languages can be constructed in several different ways. We could, on the one hand, take what is called the proof-theoretical approach, an approach based upon the syntactical component of grammar; and this is perhaps the purest form of a deductive system. In this approach, the symbols of the Logic, as well as the Logical expression that are developed from these symbols, are surprisingly enough not given any meaning whatsoever. Rather, we are only given rules for manipulating the symbols—syntactical rules, as they are called—that allow us to form 1) what can be called primary expressions, to be used as the first line of a Proof; and 2) secondary expressions, which are formed from the primary expressions by means of a set of syntactical rules known as Transformation Rules, and are used to complete the Proof. Using this syntactical approach, even a computer could work out most of the details of the Logic, since—even though computers cannot understand anything—no understanding of the actual symbols is required, only manipulation according to the syntactical rules. On the other hand, we could take the model-theoretical approach to deductive systems, in which we are given...
not only the rules for manipulating the symbolic expressions—that is, syntactical rules—but also a set of values that constitute the meaning of the expressions—this is the semantics of the symbols. This latter, semantic approach, which requires an understanding of what the symbols mean, is more suitable to an introductory level investigation of Logic, since it provides a means for human beings to understand what is going on in the Logic. And this is in fact the approach that we shall use here, although after we have developed our semantically-based Deductive System (where the capitalized form indicates our deductive system specifically) an example of the system in its purely syntactical form will be provided, for the sake of completeness.

To begin, then, we start with the lexicon, which is nothing but a collection of symbols of a particular type, and here the collection will contain all of the symbols that we will use (much like all of the words of English) in the construction of the expressions, the “sentences” or Propositions (strictly speaking), of our artificial language. Next comes the Grammar, which—as we know from the English grammar we learned in school—is a set of Rules for actually constructing the expressions or Propositions of the Logic from the symbols in the Lexicon. In natural languages, a Lexicon and a Grammar—the Symbols and Rules of a language system—allow for complex speech itself, by providing Symbols to be spoken (that is, the Words) and Rules for speaking. For our purpose, which is to construct an artificial language, the Lexicon, as listed in the summary below, will consist of only two types of symbols: 1) a potentially infinite number of more or less identical Symbols called Propositional Variables, and 2) three distinct Symbols called Operational Constants. (For the sake of completeness, the Metalogical Symbols are included in the summary below, but these are, strictly speaking, not a part of the system.)

The Deductive System: Lexicon and Grammar

1) The Lexicon (P,R):
   — Operands: A Class P = (p, q, r, s, t, . . .) of Propositional Variables; and
   — Operators: A Class R = (≈, ~, ˅, ˄) of Constants, all defined on P.
   — Metalogical Symbols: a Class M = ((, ), [, ], {, }, . . . )

2) The Grammar:
   2a) The Syntactics of Propositional Forms (Formation Rules): For p ∈ P and q ∈ P,
       1) ≈ p is a Simple Propositional Form
       2) ~ p is a Simple Propositional Form
       3) p ˅ q is a Compound Propositional Form
       4) p ˄ q is a Compound Propositional Form

   2b) The Semantics of Logical Operators (Logical Matrices): Defining Logical Value (T/F)
Duality of Logical Values: All Propositions have a Logical Value of either T xOr F

The first set, the Propositional Variables, given in the summary as the class P, contains an unspecified (and potentially infinite) number of lower-case letters of the English alphabet—letters such p, q, r, s, t, and so on. Each of these letters, when used in our deductions, represents any simple Proposition whatsoever (and the word “any” here is what makes the symbols variables). In contrast, the second set of Symbols, the Operational Constants, given as the class R (for ‘Relations’), are used consistently to represent three basic Logical operations (and the word “consistently” here is the reason that these symbols are called Operational Constants, although typically they are simply called Operators). As we shall see in more detail when we turn to our Grammar, the three Operational Constants or Operators represent: 1) a Monadic, Unary, or One-Place operation called Negation, which is represented in English by the word ‘Not’ and in our Deductive System by the symbol ‘~’; 2) a Dyadic, Binary or Two-Place operation called Disjunction, represented in English by the word ‘Or’ and in our system by the symbol ‘∨’; and finally, 3) a second Binary operation, called Conjunction, which in English is represented the word ‘And’, and in our Logic by the symbol ‘∧’. Since these three Operational Constants are typically called Operators, the Propositional Variables themselves—the things that are operated upon by these three Logical Operators—can themselves called Logical Operands (or more simply just Operands). Operands that stand alone—such as p, or t, or any single alphabetic Symbol—are called simple Propositions; and Operands in combination with Operators—as in such expressions as ~p or “p and q”—are called compound Propositions.

These two sets of Symbols of the Lexicon, then—the Operators and the Operands—are the only two lexical sets we need to produce the kinds of Propositions we will concern ourselves with in our study of Modern Logic. All the same, having symbols for simple Propositions (which is what are Operands represent) will not do us much good if we do not actually know how to put these simple symbols together to form other, more complex compound Propositions. In addition, none of our Propositions, either simple or compound, will have any natural utility whatsoever if we do not give them some sort of Logical meaning. Accordingly, having stipulated our Lexicon with its two types of Symbols, we must now consider the Grammar of our Deductive System. As with the Lexicon, which contains two types of Symbols, this Grammar will consist of two types of Rules (loosely related to the two types of Symbols): Rules of Syntactics and Rules of Semantics. The Rules of Syntactics (which are also known as Rules of Formation) tell us precisely how to combine the two types of symbols—the Propositional Variables and the Operational Constants, the Operators and Operands—to form not only basic Propositions (as a first step) but also increasingly complex compound Propositions. And the Rules of Semantics will allow us to assign Logical “meaning” (or Logical Value—T for True and F for False) to all of our Propositions, whether they are simple or compound.
Beginning, then, with the former set of Rules, the Rules of Syntactics, we need four such rules, one for each of the four Logical Operators, and these rules—also called Rules of Formation—are typically given in the following form:

1) *The Rule of Position*:  
   If \( p \) is any proposition, then \( \equiv p \) (that is, “\( p \)”) is a Proposition;

2) *The Rule of Negation*:  
   If \( p \) is any proposition, then \( \sim p \) (that is, “\( \text{Not} \, p \)”) is a Proposition;

3) *The Rule of Disjunction*:  
   If \( p \) and \( q \) are any propositions, then \( p \lor q \) (“\( p \, \text{or} \, q \)”) is a Proposition;

4) *The Rule of Conjunction*:  
   If \( p \) and \( q \) are any propositions, then \( p \land q \) (“\( p \, \text{and} \, q \)”) is a Proposition.

These four Rules of Syntactics, as simple as they might seem, are all that we need in order to construct ever more complex compound Propositions; of course, what they tell us is often more than what meets the eye. Note, for instance, that the first two Operators, Position and Negation, require only one Operand, whereas the other two Operators, \( \lor \) and \( \land \), each require two; accordingly, the first two Operators are called Monadic or Unary Operators, while the next two are called Dyadic or Binary Operators. Moreover, since \( p \land q \) is a Proposition (albeit, a compound Proposition), we could, if we so desired, denote \( p \land q \) by some other Operand-symbol (that is, any symbol other than the letters ‘\( p \)’ or ‘\( q \)’, which we are already using). For instance, we could let the letter ‘\( r \)’ represent the Proposition \( p \land q \), and then we can apply our Rules of Syntactics to this new version of \( p \land q \)—that is, to ‘\( r \)’ itself—and use this ‘\( r \)’ to form a different, more complex compound Proposition involving this ‘\( r \)’ and any other Proposition. Maybe we would come up with something like ‘\( r \lor \sim t \)’, nor need we stop here; the Rules of Syntactics can be applied repeatedly in this manner, thereby allowing us to construct increasingly compound Propositions, regardless of the original or final complexity of Propositions involved. So, for example, we could easily construct propositions of such complexity as:

\[
\sim[\sim(p \lor q) \land (\sim r \land s)]
\]

and then, if we so choose, we could represent everything inside the square brackets by ‘\( t \)’ (with the final result being \( \sim t \)), and then use this abbreviated form of the formula above to form yet another Proposition, such as:

\[
\sim t \lor u
\]
which, as we know, actually stands for:

\[ \neg[\neg(p \lor q) \land \neg(r \land s)] \lor u \]

In this manner, we can construct any Proposition of any complexity whatsoever, as long as we form Propositions according to our three Rules of Formation. These three Formation Rules thus provide us with virtually infinite power to create new, increasingly complex compound Propositions from pre-existing Propositions, either simple or compound, *ad infinitum*. Any Proposition constructed according to these Rules of Syntactics, which tells us how to correctly *form* our Propositions, are called Well-Formed Formulas.

The second part of our Grammar, the Rules of Semantics, consist of displaying, by means of tables of values, the meaning of each of our three Operators. Of course, since we are developing a two-valued Logic here—the Logic of the values True (T) and False (F)—*displaying* what an Operator symbol *means* boils down to giving a *complete* list of the Logical Values—the T’s and F’s, that is—for all of the Propositions involved in the use of the Operator. To do this, we use what are typically (albeit misleadingly) called Truth Tables, the five most important of which are given in Figure 4-1, below. A Truth Table—which is more properly called a Logical Matrix—is a table of rows and columns in which the top row contains the various Propositions involved in our analysis, while the rest of the table—the other rows and columns—lists the Logical Values assigned to each of these Propositions. In the standard terminology of Logical Matrices (or Truth Tables), the *left-most column* of the table is called the Guide Column, and across the top row of this column we list the various Propositions contained in the Proposition to be defined by the matrix. Next in the Guide Column, below the top row of Propositions needed, we list all of the possible Logical Values of each of the simple Propositions that we have listed in the top row. These Logical Values, which are either T or F, are assigned to each of the Propositional Variables used in the top row, with at least one T and one F assigned to each Propositional Variable. With this the Guide Column is complete. In addition, however, we note that Guide Columns are common to *all* Logical Matrices; and each Guide Column will have exactly the same form for all tables *with the same number of simple Propositions* listed in the top row of the Guide Column. In fact, we have a simple mathematical formula for determining the number of Rows in a Guide Column from the number of Simple Propositions listed in the Guide Column. This formula is \( r = 2^n \), where ‘n’ is the number of Simple Propositions listed at the top of the Guide Column and ‘r’ is resulting number of Rows in the Guide Column (this will become more clear when we consider specific matrices).
Basic Propositional Forms:

Basic Simple Propositional Forms — Position and Negation:

Position & Negation:

<table>
<thead>
<tr>
<th>≈ p</th>
<th>~ p</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

Basic Compound Propositional Forms — The Contingency of Material Propositions:

Disjunction:

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ∨ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Conjunction:

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p ∧ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Derived Compound Propositional Forms:

Material Implication:

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p &gt; q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

Material Equivalence:

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p &lt; &gt; q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

Figure 4-1. The Five Basic Logical Matrices

To use a Logical Matrix, then, we first write out the Logical expression we wish to define (or evaluate, as the case may be). To achieve this, each of the Logical Values listed in the Guide Column, under the row of Propositional Variables, is used to determine the Logical Value of the compound Proposition for that Logical possibility (that is, for that row of T’s and F’s in the Guide Column). The determined value for a
specific Logical possibility is then listed (and thereby stipulated and thus *defined*) to the right of the Guide Column values, underneath the compound Proposition that is being defined. For example, with Disjunction, which is illustrated in Figure 4-1, we know from the second Rule of Formation that *two* component Propositions are involved (since Disjunction is a Binary Operator). We thus need to use two Operand symbols (two lower-case letters, that is), say p and q, and these are listed across the top row of the Guide Column in the Logical Matrix for Disjunction (in Figure 4-1). Now there are, of course, four possible combinations of Logical Values for two Propositional Variables, since both variables (p and q, in this case) can be assigned either T of F (and of course, two variables times two values equals four combinations). And this can be confirmed by using our formula, given above, to determine the number of Rows we need in the Guide Column. We have just said that we have two Simple Propositions to list in the top Row of the Guide Column, which means that the ‘n’ in our formula $r = 2^n$ is 2, and our formula becomes $r = 2^2$. And working this out, we see that $r$ equals 4. Thus, in the Figure, we see the variables p and q listed across the top row of the Guide Column, and under each of these we see four more rows, one row for each of the four possible combinations of T and F, assigned two at a time. First, for these, it is possible for each of p and q to be assigned a T, and this is indicated by the two T’s in the first row of the Guide Column; second, it is possible for p to be T and q to be F, as we see in the second row; next, it is possible for p to be F and q to be T, as in the third row; and fourth and finally, it is possible for both p and q to be F, and this is given in the fourth row. And these four combinations of T’s and F’s constitute the totality of possible Logical-Value combinations for Disjunction, as well as for *any two-component proposition* in general (that is, for all propositions that use one Binary Operator).

Next in our example for Disjunction, the actual Disjunction of p and q, which is symbolized now by “$p \lor q$”, is listed in the top row of the table, to the right of the Guide Column, and for each row of the Guide Column we must determine (or in this case, stipulate) a Logical Value for the Disjunction ($p \lor q$) itself. Thus, in the first row of this column of the matrix (just beneath the ‘$\lor$’ symbol itself) we need to provide either a T or an F, thereby defining the Logical Value of the Disjunction according to the first row of the Guide Column (which has two T’s). As we see, the resulting Logical value is a T (for True), which makes sense because when both of two Propositions p and q are True, saying that either $p \lor q$ is True seems intuitive. In addition, it also seems intuitive that either $p \lor q$ is True when either p or q but not both is True, while it seems counterintuitive that $p \lor q$ would be True when both p and q are False. Accordingly, we fill in the remaining rows in this column, and see that any Disjunction whatsoever is T (or True) for the first three rows in the Guide Column, and is F (or False) for the last row only. The reasons for defining Disjunction this way are numerous, some of them being purely Logical. At the same time, it is easy enough to see that a Disjunction should be True when either one or both of it’s component Propositions are True. For instance, if p
represents the simple Proposition “I went to the mall” and \( q \) represents the simple Proposition “I went to the park”, then the Disjunction “\( p \lor q \)”, which of course represents the compound Proposition “I went to the mall or I went to the park”, should be True when either \( p \) or \( q \) or both are True, and False only when both \( p \) and \( q \) are False. Accordingly, as can be seen from Figure 4-1, we have here defined Disjunction as being True whenever either or both of its components is True, and False otherwise.¹

Just as we have constructed this Logical Matrix in order to define Disjunction, we construct Logical Matrices for Negation and Conjunction, although since Conjunction is, like Disjunction, a Binary Operator, while Negation is merely a Monadic or Unary Operator, there will be two Component Propositions for Conjunction, but only one for Negation. In the latter case, then, there will be only two possibilities (and thus only two rows) in the Guide Column for Negation (because \( r = 2^1 \) says that \( r = 2 \)). This also can be see in Figure 4-1, where the Logical Matrix for Negation involves one Proposition only, and this Proposition is displayed in the table as \( \neg p \). Accordingly, there is only one Proposition—\( p \) itself—in the Guide Column, and the first value under \( p \) is T for True, while the second value F for False. These two Logical possibilities are all of the possibilities for one Proposition, because a given simple Proposition can have only these two possible Logical Values—T or F. To the right of \( p \) in the Guide Column, then, we see the negated Proposition \( \neg p \), and under this (and to the right of the symbols T and F in the Guide Column), we see the Logical Values of the compound Proposition \( \neg p \) itself. In the first case, when \( p \) has the value T, \( \neg p \) has the opposite value, F; and when \( p \) has the value F, \( \neg p \) has the opposite value, T. In contrast, for Conjunction (whose Logical Matrix is given to the right of that for Disjunction), which is symbolized in the table as “\( p \land q \)”, the table starts with the very same Guide Column as Disjunction (because—like Disjunction—Conjunction is Binary and requires two component Propositions). Of course, since Conjunction is similar to Disjunction while being actually a different operation, we find different assigned values listed under the compound Proposition itself: ‘\( p \land q \)’. This is because, naturally enough, the Proposition “I went to the mall and I went to the park” (which is represented in the table by “\( p \land q \)”) seems naturally False when either one or both of \( p \) and \( q \) are False, and True only when both \( p \) and \( q \) (that is, when both “I went to the mall” and “I went to the park”) are True.

These three Truth Tables—the Logical Matrices for Negation, Disjunction, and Conjunction, which we call the Basic Operators—comprise the whole of the Semantics of the three Logical Operators. The tables thus tell us everything we need to know about the meaning of these three Operator symbols; and although much more will be unfolded from these humble beginnings, we have, at this point, all that we need in order to start developing our Deductive System. As a first step in this direction, let us introduce two new Logical Operators, in addition to those of Negation, Disjunction, and Conjunction. These new Operators—which are known as Material Implication and Material Equivalence (as in Figure 4-1), and are symbolized by \( > \) and \( <> \), respectively—are not in
themselves necessary for our Deductive System; and in fact we could do without these particular symbols. But doing without them (or, more specifically, without the Operations the symbols represent) would make our Deductive System much more complex, for these new Operators are actually abbreviations for certain compound Propositions constructed from the three Basic Operators—Not, Or, and And. Accordingly, these Operators are dispensable, and are not basic but rather are what we will call Derived Operators, since they are derived from the Basic Operators. However, although these abbreviations are not an absolute requirement for our Logic, the Logical relations they represent are actually required for deduction, and in fact they are the Deductive System, itself (and so these abbreviations will help us save time in our standard deductions). Accordingly, Material Implication and Material Equivalence are of great value in modern Logic; and we shall proceed to the definition of these derived Logical Operators.

The first of these Operators represents the relation of Material Implication, and is itself represented by the symbol >, introduced above. Of course, as you may recall, we are fortunate enough to have inherited a perfectly good Logical definition of (Material) Implication from Philo of Megara (given in Chapter Three), who defined Implication as being False when the Antecedent is False and the Consequent is True, and True in all other cases. And this definition is precisely what we find illustrated in Figure 4-1, under the Logical Matrix labeled Implication. There we see, just as Philo had stipulated, that an Implication is F only when its Antecedent is T while its Consequent is F (as seen in the Guide Column), while it (the Implication) is True otherwise. What we might not be able to see, however, is that the Logical Matrix based on Philo’s definition of Implication is identical to the Logical Matrix for a certain compound Proposition constructed from the basic operations of Negation and Disjunction, in the form of ‘¬p ∨ q’, and is also identical to a certain compound Proposition constructed from Negation and Conjunction, in the form of ‘¬(p ∧ ¬q)’. The identity of these three different Logical Matrices is seen by comparing the table for Implication (given anew below) with that for the Compound Proposition ‘¬p ∨ q’ as well as that of ‘¬(p ∧ ¬q)’ (all three of which are given below), where we find exactly the same entries in all three matrices. Apparently, then, Philo’s brand of Implication is nothing different from, and thus nothing other than, the Disjunction of the-Negation-of-the-Antecedent with the Consequent (not to mention the Negation of the Conjunction of the Antecedent and the-Negation-of-the-Consequent).
As was pointed out earlier, the structure of Implication, as meager as this structure is when we use the symbol ‘>’, is reminiscent of the structure of Argumentation, as this was laid out at the beginning of Chapter Three. And we can make this similarity even more acute by juxtaposing these two structures themselves, that of a) an Argument in general:

Premise ▶ Conclusion

and that of b) Material Implication:

Antecedent > Consequent.

This comparison is illuminating; but as illuminating as it might be, it does not actually reveal the structure of Implication, which is critical for our philosophical understanding of Logic. In order to illustrate the structure of Implication, what we need is an instance of the Inclusion relation among Classes or Sets, of the type that we made use of in our investigation of Aristotle’s Categorical Logic. Accordingly, an example of just such a relation is provided at the bottom of the page, below, where we see that the inner circle—in this case labeled ‘p’—is included entirely within the outer circle—labeled ‘q’, for reasons to be made clear directly. In this particular situation, when a marker is in the ‘p’ circle it is also necessarily in the ‘q’ circle, due to the inclusive relationship of the two circles; and this Set-theoretic situation is equivalent to the Logic-theoretical situation in the Logical Matrix for Implication, with a first row in which p is T and q is T, and the Implication p > q is T, as well. Alternatively, the marker could be outside of the ‘p’ circle but inside the ‘q’ circle, which similar to the singular case in which the second row of the Matrix assigns T to p and F to q, making the Implication p > q turn out to be F. And of course, a marker could be outside of the ‘p’ circle but inside the ‘q’ circle, which is equivalent to the third row of the Matrix, with p being F and q being T, and for which the Implication is again T. And finally, a marker could also be outside of both the ‘p’ and ‘q’ circles, which is equivalent to both p and q being F, although the Implication itself is still T, as in the final row of the Matrix.

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>q</th>
<th>p &gt; q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td></td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>q</th>
<th>~ p ∨ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td></td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>q</th>
<th>~ (p ∧ ~ q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td></td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
<td>T</td>
</tr>
</tbody>
</table>
1) \((p > q)\): If \(p\) is \(T\) then \(q\) is \(T\) — Ex.: If you’re in \(p\), then you’re in \(q\)

2) \((\neg p \lor q)\): Either \(p\) is \(F\) or \(q\) is \(T\) — Ex.: Either you’re \textit{not} in \(p\), or you are \textit{also} in \(q\)

3) \((\neg(p \land \neg q))\): It is \(F\) that \(p\) is \(T\) and \(q\) is \(F\) — Ex.: You \textit{cannot} be in \(p\) and \textit{not} be in \(q\).

It is unfortunate, but nevertheless obvious from the foregoing, that \textit{Material Implication}, as defined by Philo, \textit{cannot guarantee that the Truth of the Consequent follows necessarily from the Truth of the Antecedent}. This can be seen directly from the second row of the Logical Matrix for Implication, in which we see that it is possible to have a True Antecedent and yet a False Consequent; and naturally enough we hold that in this case Implication itself is False. But this contradicts the stipulation we made earlier that actual Logical or Formal Implication should provide us with \textit{absolute certainty} that the Truth of our Conclusions (which are of course just Consequences of deductive Implications, as described in the previous paragraph) follow logically from our the Truth of our Premises (which are themselves just Antecedents of deductive Implications). Consequently, in order to achieve this Logical certainty, we must modify Philo’s Material Implication by adding a that will ensure that it does deliver the absolute certainty of our Logical Conclusions. And here the diagram at the bottom of the previous page can show us the way, for it clearly illustrates the fact that, given an Inclusive relation between the ‘\(p\)’ and the ‘\(q\)’ circles, whenever a marker is present in the ‘\(p\)’ circles it is also in the ‘\(q\)’ circles, and \textit{necessarily} so. That is, if \(p > q\) (which is certainly true of these two circles) and \(p\) is \(T\) (which here means that there is a marker in the ‘\(p\)’ circle, as is in fact the case in the diagram), then necessarily \(q\) is \(T\) (that is, the marker is also in the ‘\(q\)’ circle; again, as in the diagram). In our symbolic notation this geometric situation takes the following “algebraic” form:

\[ [(p > q) \land p] > q \]

This form (which may be rendered verbal as “If ‘\(p\) implies \(q\)’ is True and ‘\(p\)’ is True, then ‘\(q\)’ is True”) is known traditionally as Modus Ponens (for reasons to be made clear directly) and, given the nature of the relation between ‘\(p\)’ and ‘\(q\)’ and the placement of a
marker in ‘p’, it cannot possibly be false. Accordingly, this particular form of Implication, known as Formal Implication (for reasons to be made clear presently, will be used as the backbone of our Deductive System. However, before we turn to the actual development of this, we must first introduce several more concepts that are relevant to our studies.

Tautologies, Contradictions, and the Principle of Duality

To begin with, we introduce the notion of a Tautology—a type of compound Proposition that, like the Formal Implication we just saw, cannot possibly be False. And as peculiar as this notion might sound, Tautologies are actually quite easy to come by; all that we need to do in order to create a Tautology is to disjoin a Proposition with the Negation of that Proposition itself. Thus, for instance, the compound Proposition that says “It is raining or it is not raining” is a Tautology, because it disjoins the simple Proposition “It is raining” with the Negation of itself—“It is not raining.” And this Tautology cannot possibly be False, is thus necessarily True, because given typical weather conditions, either it will be raining or it will not be raining, always. As a result, when we combine these two opposite Propositions into the Disjunction “It is raining or it is not raining” and assert its truth, one or the other of the two component Propositions will have to be True, and this is the same as saying that it is impossible for this particular compound Proposition to be False.

Tautologies, Contradictions, and Duality:

<table>
<thead>
<tr>
<th>Tautologies (T₀):</th>
<th>Contradictions (F₀):</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>(p ∨ ~p)</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>p</td>
<td>~ (p ∧ ~p)</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

Figure 4-2. Logical Matrices for Various Tautologies and Contradictions
To make this perfectly evident, all we need to do is construct a Logical Matrix for this particular disjunctive Proposition. To start this, let us assign the letter \( p \) to the Proposition “It is raining”; the Negation of this Proposition, then—the Proposition “It is not raining”—will be symbolized by \( \sim p \). The Disjunction of these two opposing Propositions will then be \( (p \lor \sim p) \), (which is read as ‘p or not-p’). The table for this Proposition is given at the top left of tables above; and we can see directly from this Logical Matrix that this particular disjunctive Proposition is always True, regardless of the values of \( p \). Propositions that have this form, then—a form that guarantees the Truth of the Proposition—are said to be \textit{formally} True, because the truth of the Proposition follows simply from the \textit{form} of the Proposition itself, regardless of its content. Needless to say, any \textit{formally} True Proposition is known as a \textit{Tautology}. But now, the question naturally arises as to the possibility of Propositions that are always False; that is, is it possible to construct Propositions whose form necessitates the Falsity of the Proposition? And of course, the answer is an unequivocal Yes; and these, too, are quite easy to come be. Consider, for instance, the Logical Matrix given at the top right of the tables above, which contains a conjunctive Proposition, as opposed to the disjunctive Proposition of the Tautology we have been considering. As with the earlier Disjunction, the two component Propositions—each of which is called a Conjunct of the Conjunction—are Logical opposites; that is, the left-most Proposition is \( p \) and the right-most is its opposite, \( \sim p \). Now, however, as is clearly shown in the Matrix, this Conjunction is not always T but rather always F; and this seems intuitively correct. For what this Conjunction says, basically, is the that the Proposition \( p \) is both T and F at the same time. Needless to say, this situation, which is called a Contradiction, is insidious for Logic, because it insists upon the simultaneous truth and falsity of a given Proposition; but if this were allowed in Logic, it would tend to undermine the requirement of necessary truth that is so crucial to our Deductive System. Accordingly, Contradictions, which will here be represented by the symbol \( \text{F}_0 \) (pronounced “eff-sub-oh”), are prohibited in Logic; Tautologies, in contrast, which will be represented by the symbol \( \text{T}_0 \) (pronounced “tee-sub-oh”), are the very foundation of Deductive Logic, and are hence \textit{the} requirement of Deduction.

In the diagram above we have given four matrices: two of which are Tautologies (on the upper left) and two Contradictions (on the upper right); and of these the lower of the two Tautologies is just the negation of the upper of the two Contradictions. And in fact, as it turns out, every negation of a Contradiction is a Tautology, and vice versa. Even more importantly, however, the two Tautologies on the left (the lower of which, recall is the negated Contradiction on the top right) are what we call \textit{Duals} of each other. That is, employing what is called the \textit{Duality Principle}, the upper Tautology can be transformed into the lower Tautology by a 3-step process of \textit{inversion}, in which 1) all the component Propositions involved in a Tautology are negated, 2) the entire Tautology itself is negated, and finally 3) all Conjunction Operators are replaced with Disjunction Operators, and vice versa. To give a specific example, for the upper Tautology, which is
(p ∨ ~p), we \textit{first} negate each of the component Propositions (although here, where p and ~p are involved, negating p makes it ~p and negating ~p makes it p, and so we end up with the same two Propositions with which we started); \textit{next} we negate the entire Tautology (to get ~(~p ∨ p); and \textit{finally} we replace the ∨ Operator with an ∧ Operator, ending up with ~(p ∨ ~p), which therefore is said to be the Dual of (p ∨ ~p). (Please note: for the sake of comparing these Dual Propositions, we have also reversed the order of the two component Propositions, but this is not a necessary step in the Duality process).

As this might suggest, the Duality Principle is an extremely important notion in Formal Logic, because it allows us to reduce the potential number of symbolic expression by one-half. That is, whereas we might assume that in any Formal Logic we will have to use both the expression (p ∨ ~p) and the expression ~(p ∧ ~p), the Duality Principle tells us that these two expressions are Logically Equivalent, and thus we could eliminate one or the other (and thereby reduce the number of Propositions by one-half, as stated). What is more, it turns out that often enough, although the proof of a particular Theorem might be extremely difficult, the proof of the Theorem’s Dual is actually quite easy, a fact that makes the Duality Principle, which in fact extends all the way through Logic and Mathematics, quite useful. For our purposes, however, the most interesting ramification of the Duality Principle can be seen from comparing the Dual Tautologies given in tables above. And interestingly enough, although it may not have been apparent earlier, these two Tautologies are actually two of the traditional Four Laws of Thought, which were assumed, in Classical times, to be the foundation of all Logic. As it turns out, this assumption is not totally off the mark, for in fact the upper of these two Tautologies—given symbolically as (p ∨ ~p)—is the so-called Law of Excluded Middles cast in modern symbolic form. And the lower Tautology, the Dual of the upper—given symbolically as ~(p ∧ ~p)—is the so-called Law of No Contradictions. The first of these so-called Laws of Thought, the Law of Excluded Middles, in Classical form, says that any Proposition must be either True or False, because there is no middle ground in a two-valued Logic (hence, the “Excluded Middles” phrase). In modern symbolism, this becomes (p ∨ ~p), which is of course the upper Tautology. And the second of the Laws of Thought, the Law of No Contradictions, says that no Proposition can be both True and False at the same time (because this would be a Contradiction); and in modern form, of course, this becomes ~(p ∧ ~p), which is the lower Tautology. Now, as it turns out (as we shall see soon enough), both of these traditional Laws of Thought are Axioms—that is, fundamental Propositions—in the Deductive System that we develop here. And this means that at least two of the classical Laws of Thought are in fact Laws of modern Logic. As for the remaining two of the Four Laws of Thought (all of which are included at the in the tables below, for the sake of completeness), they are not fundamental to modern Logic; nevertheless, they can be deduced from the Axioms and are thus
Theorems of our Deductive System, as we are about to see in the next section, in which we actually develop our Deductive System.

*The Four “Laws of Thought”:*

<table>
<thead>
<tr>
<th>The Law of Excluded Middles:</th>
<th>The Law of No Contradictions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>~p</td>
<td>~ (p ∧ ~p)</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>f</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>f</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The Law of Identity:</th>
<th>The Law of Double Negation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>p &lt;=&gt; p</td>
<td>p &lt;=&gt; ~ ~p</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

*The Deductive Apparatus*

We begin the explication of our Deductive System by returning to our example of the type of Implication we need, the primary form of which was given above as [(p > q) ∧ p] > q. This Implication, which will become the backbone of our Deductive System, is known traditionally as Modus Ponens (the Mode of Putting), because it assumes the truth of a Material Implication (in this case, p > q), and then “puts forth” (in Latin “ponens”) the Antecedent of this Implication: p. And as we saw above, this particular “putting forth”—the assertion that p is True—is the step that guarantees the Truth of Modus Ponens. But now what we want to know is this: Is Modus Ponens always True; is it a Tautology? And in order to answer this, of course, all we need to do is construct the Logical Matrix for this compound Proposition; and if the second Implication symbol (the one to the right of the Proposition within the square brackets) has T’s in every row underneath it, then Modus Ponens is tautologous. As we see from the Logical Matrix for Modus Ponens, which is given in the left-hand table of the pair of tables below, under the heading of The Rule of Deduction, this is precisely the case, which makes the second Implication is in fact tautologous (represented in the matrix by the replacement of the > symbol by the symbol =>). This compound Proposition then, which in our Deductive System will not bear the traditional name (Modus Ponens), a name that is opaque to most students and thus of little value, but will instead bear the name given in the diagram (the Rule of Deduction), is always True, and is thus a Tautology. This means, of course, that
this Rule of Deduction is a type of Formal Implication, and it is for this reason that it 
guarantees the Truth of its Conclusion whenever its Premises are True. In fact, it is 
actually impossible for the Conclusion of the Rule of Deduction (or any Formal 
Implication, for that matter) to be False when the Premises (however many there are) are 
True (and this, of course, as we saw earlier, is the definition of the old-fashioned notion 
of Logical Validity). Consequently, whenever we use the Rule of Deduction (or any 
Formal Implication) as a step in any deduction, we can be sure that the step will always 
preserve the Truth of the deduction. This means that if we start a deduction with True 
Premises, whatever other Propositions we can deduce from these Premises using this 
Formal Implication—this Rule of Deduction—could not possibly be False. Accordingly, 
we introduce this new symbol for Formal Implication—to distinguish Formal 
Implication, which cannot be False, from Material Implication, which can be False—by 
inserting an Equals Symbol, ‘=’ in the Material Implication Symbol, ‘>’, to get the 

**Rules of Inference — Formal Implication (=>) from ‘>’ and ‘T₀’**

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>[(p &gt; q) ∧ p] =&gt; q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>t t T t</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>f f T f</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>t f T t</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>t f T f</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>(p ∧ q) =&gt; p</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>t t T t</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>f T f</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>f T t</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>f T f</td>
</tr>
</tbody>
</table>

Implications of this type are called Universally Valid Implications, and as it turns 
out—quite fortunately for our Deductive System—Universally Valid Implications are 
themselves easy enough to come by. As a result we can easily build a deductive system 
from any number of such Tautologies. Accordingly, second example of such rules, this 
time called *The Rule of Reduction* (that’s Re-duction, as opposed to De-duction, as in the 
first rule), is also given above, in the right-hand table. For this rule, rather than starting 
with a Material Implication (as in the Rule of Deduction), we start with a Conjunction; 
and this particular rule tells us that if this Conjunction is True (which we do assume is the 
case), then (since, according to the Logical condition for a True Conjunction, both of its 
component-Conjuncts must be True as well) we may deduce the truth of either 
component Conjunct of the Conjunction. And accordingly, in the table above, we have 
deduced the truth of the Conjunct p from the truth of the Conjunction “p ∧ q”.

19
As these examples of tautological Implication suggest, this is powerful stuff, for we can actually guarantee the validity of our deductions. At the same time, the one-way nature of Formal Implication makes its role in deduction somewhat limited, and it would be nice if we could also have a tautologous Operator that works in both ways; that is, we would like a type of inference that moves not only from Antecedent to Consequent, but—at the same time, and this is important—also works the other way, from Consequent to Antecedent. Fortunately, just such a two-way relation has been defined in Logic, and it is known as Logical Equivalence (for which a matrix was provided in Figure 4-1). What is more, as with our construction of Implication—which started out as Material Implication, but was “upgraded” by Tautology to Formal Implication—so too, Logical Equivalence begins as mere Material Equivalence (symbolized in the matrix by < >), but it becomes Formal Equivalence (symbolized by <=>) when we make it tautologous (which means that it, too will preserve the Truth of our deductions). Starting, then, with Material Equivalence, consider its Logical Matrix, given at the bottom right of Figure 4-1. There we see that if any two Propositions—here symbolized by p and q—share the same Logical Value, regardless of whether this value is T (as in the top row of the Matrix), or F (as in the bottom row) then the statement of their Logical equivalence (as symbolized at the very top of the Logical Matrix: p < > q) is said to be True. However, if the two Propositions have different values (as in the two middle rows of the Matrix), the statement of their equivalence is False. Thus, as with Material Implication, which is sometimes T and sometimes F, Material Equivalence is similarly sometime T and at other times F. What we want, however, is a form of Equivalence that will always be True, and this Formal Equivalence can, again like Formal Implication, be constructed in such a way as to yield Formal Equivalence.

Like Formal Implications, Formal Equivalences are themselves quite easy to come by, and Logical Matrices for two different instances of Formal Equivalence are given on the following page, each with a very specific motive. For the left-most Matrix, the specific motive is to illustrate the definition, as it were, of Material Implication as an abbreviation of a form of (first) Disjunction and Negation, (~p  q), and (second) of Conjunction and Negation, ~(p  ~q). And the motive for the right-most Equivalence is to demonstrate that Equivalence is in fact, as demanded above, a type of Logical Implication that works in both directions: from the Antecedent to the Consequent, and from the Consequent to the Antecedent. Of these two Matrices, the former shows that the two Propositions just given—(~p  q) and ~(p  ~q)—as definitions of Material Implication, are in fact Logically equivalent, and accordingly these two definitions are stipulated (in the middle and last sections of the Proposition in the Matrix) as definiens of the definiendum,2 (p > q), which is given in the first section. And the latter Matrix simply stipulates that a Conjunction of any two Implications that are transpositions of each are (as are p > q and q > p) is Formally Equivalent to a Material Equivalence.
**Rules of Replacement** — Formal Equivalence (\(\iff\)) from ‘\(\supset\)’ and ‘\(T_0\)’

**Definition of Material Implication:**

\[
\begin{array}{c|cccc}
 p & q & (p \supset q) \iff (\neg p \lor q) & \iff & (p \land \neg q) \\
\hline
 T & T & t & T & t & f \\
 T & F & f & T & f & t \\
 F & T & t & T & t & f \\
 F & F & t & T & t & f \\
\end{array}
\]

**Definition of Material Equivalence:**

\[
\begin{array}{c|cccc}
 p & q & (p \lnot \iff q) \iff [(p \supset q) \land (q \supset p)] \\
\hline
 T & T & t & T & t & t \\
 T & F & f & T & f & f \\
 F & T & f & T & t & f \\
 F & F & t & T & t & t \\
\end{array}
\]

With the development of these two Formal Equivalences, we have completed the first stage in the development of our Deductive System. For, combining these two Definitions with the two Formal Implications developed earlier—the Rule of Deduction and the Rule of Reduction—we now have everything we need to actually perform Logical Deductions. This is accomplished, of course, by the Rule of Deduction itself, whose Antecedent consists of two Proposition that state unequivocally that if an inference (that is, a *deduction*, which in the case is the inference \(p \supset q\)) is valid and its Antecedent is True, then we may assert the Truth of the Consequent. In terms of an actual Deduction, this amounts to saying that if we can show that a new line in a Deduction follows necessarily from a previous line and this previous line is True, then the new line will be True as well. This, it need not be pointed out, is Formal Deduction. In addition, the Rule of Reduction says that if we have two lines in a Deduction then we can conjoin these two lines to form a new line, because all the lines in our Deductions are assumed to be True, and thus conjoining any two lines produces a Conjunction that is necessarily True itself. And this too is a type of Formal Deduction. In addition, the definitions introduced above of the two Material relations—the Definition of Material Implication and the Definition of Material Equivalence—can also be used to produce new lines in a Deduction, for whenever we have a line of Deduction that has the form of any component in these Definitions we can substitute, on a new line, any of the alternative forms in the Definitions. This too is a type of Formal Deduction, and these four Propositions, which collectively we will call the four Rules of Transformation for our Deductive System, are given with the respective Logical Matrices in the four tables above. As we can see, the first two of the four Rules of Transformation are called Rules of Inference, for reasons that should by now be obvious; the last two rules, the Definitions, are actually Rules of Replacement, because when any component in these Definitions appears on a line in a Deduction it can be replaced by any of the other component(s) in the Definition.

To recapitulate, then, at this point in the development of our Deductive System we have our Lexicon, which consists of Symbols to represent the Propositions we wish to manipulate; we have our Syntactical Rules or Rules of Formation, which tell us how to
create Well-Formed Formulas from Symbols in our Lexicon; we have our Logical Matrices, which give us the Logical Values assigned to the Basic and Derived Operators given in the Lexicon and the Rules of Formation; and we have our four Transformation Rules, which allow us to transform the Well-Formed Formulas into Theorems, created according to the Rules of Formation.

This all but completes the development of our Deductive System, and we have but one step left. The last step, then, will be to introduce five more Definitions, to be used as Axioms, that will complete the list of Axioms needed to do our deductions. These five Definitions, however, are not strictly speaking a necessary part of a deductive system, and they could be excluded from our development. Accordingly, we introduce what are typically known as the Five Formal Properties (or Axioms) of Algebraic Structures.

1) The Axioms of Identities:

\[ I_\lor: \quad (p \lor F_0) \iff p \]
\[ I_\land: \quad (p \land T_0) \iff p \]

2) The Axioms of Opposites:

\[ O_\lor: \quad (p \lor \neg p) \iff T_0 \]
\[ O_\land: \quad (p \land \neg p) \iff F_0 \]

3) The Axioms of Commutativity:

\[ C_\lor: \quad (p \lor q) \iff (q \lor p) \]
\[ C_\land: \quad (p \land q) \iff (q \land p) \]

4) The Axioms of Associativity:

\[ A_\lor: \quad [(p \lor (q \lor r))] \iff [(p \lor q) \lor r] \]
\[ A_\land: \quad [(p \land (q \land r))] \iff [(p \land q) \land r] \]

5) The Axioms of Distributivity:

\[ D_\lor: \quad [(p \lor (q \land r))] \iff [(p \lor q) \land (p \lor r)] \]
\[ D_\land: \quad [(p \land (q \lor r))] \iff [(p \land q) \lor (p \land r)] \]

As they will, however, provide us with certain structures to which we can apply our Deductive System, they are certainly useful to our System. However, as each of these five properties will be applied both to the Disjunction Operator (\lor) and to the
Conjunction Operator \((\wedge)\), the final result will have ten forms, which themselves are typically called the *ten* Axioms of Algebraic Structures. The complete list of these Axioms is given above, for which even the slightest experience in Mathematics will have made the last three pairs of Axioms familiar to our readers, as we shall see directly. As for the first two Axioms, they too may be familiar from Mathematics, although their use there typically goes unmentioned (and thus often unnoticed) and they will consequently require a bit more explanation than the other three Axioms. For the sake of clarity, however, we will begin with the more familiar last three Axioms: those of Commutativity, Associativity, and Distributivity. The first of these, the Axiom of Commutativity, should be familiar enough from elementary Arithmetic, for it says simply that, as with the Addition and Multiplication of Numbers, the order of any Disjunction or Conjunction can be reversed or *commuted*. Using this Axiom, then, the left Disjunct or Conjunct (located to the left of the Operator symbol) commutes to the right of the Operator, and the right Disjunct or Conjunct commutes to the left of the Operator. In Arithmetic, of course, this is exemplified, for instance, by the expressions \(3 + 4\) (for Addition) and \(3 \times 4\) (for Multiplication) being commuted to \(4 + 3\) and \(4 \times 3\), respectively. And similarly, the presence of this commutative rule in our Deductive System allows us to commute the Propositions \((p \lor q)\) and \((p \land q)\) into \((q \land p)\) and \((q \lor p)\), respectively. Next, the Axiom of Associativity, which should also be familiar from Arithmetic, says that when any three Propositions are either Disjoined or Conjoined, it does not matter what order we pair them up for operation. In Arithmetic this means that \(3 + 4 + 5\) can be operated on as \((3 + 4) + 5\), in which we first pair up the two numbers on the left (3 and 4), add them together, and then add their sum to the number 5; or we can first pair the last two numbers (4 and 5), add them together, and then add this sum to the number 3; either way, we get the same final sum: 12. For our Deductive System, this means that we can associate the first two of any three Disjuncts of a Disjunction and then disjoin these with the third and last Disjunct; or we can associate the last two Disjuncts and then disjoin these with the first Disjunct. And of course, the same may be said, *mutatis mutandis*, of Multiplication in Arithmetic and Conjunction in our Deductive System. Turning finally (for the last three Axioms) to the Distributive Axiom, here we must proceed with care, for the role of the Distributive Axiom as we are familiar with it from Arithmetic is slightly different from its role in Logic. In Arithmetic, that is, we know that if we have an expression that combines Multiplication with Addition in a particular way—such as in the expression \(3 \times (4 + 5)\)—we can *distribute* the Multiplication factor (that is, the “3 x”) over the terms of the addition (that is, over the “4 + 5”) so that we end up with the distributed expression \((3 \times 4) + (3 \times 5)\). We also know, however, that if the operations are reversed, as in \(3 + (4 \times 5)\), we *cannot* distribute the *Addition* over the *Multiplication*, because that is not allowed in Arithmetic. In Logic, by contrast, we actually *can* distribute either *Operator* over the *other*; and accordingly, as we can see in Axiom list
above, we have two Distribution Axioms, one in which we distribute Disjunction over Conjunction and one in which we distribute Conjunction over Disjunction.

On a philosophical note, we must point out that this difference in the Distribution Axioms for Mathematics and Logic marks the first major difference between these two great deductive systems, which in many other ways are fairly isomorphic. Other differences do exist, however, and we meet the next set of differences when we consider the first two Axioms in our list: those of Identities and Opposites. The first of these, the Axiom of Identities, introduces into our Deductive System the notion that one unique element in our Lexicon works in such a way that, when it is one of the terms of either a Disjunction or a Conjunction, the result of the operation is just the other component in the operation. In Arithmetic, where we have certainly seen this but may not have been aware of it, we have the unique elements 0 (for Addition) and 1 (for Multiplication); accordingly, for any Addition, \( n + 0 = n \) (regardless of what number \( n \) is); and for any Multiplication, \( n \times 1 = n \) (regardless of what number \( n \) is). And similarly in Logic, as we can see in the list of Axioms, we have the unique element \( F_0 \) (which represent, as you will recall, any Contradiction) for the operation of Disjunction, and we have the unique element \( T_0 \) (which represent any Tautology) for Conjunction. Accordingly, for any Disjunction, \((p \lor F_0) \iff p\) (regardless of what Proposition \( p \) stands for); and for any Conjunction, \((p \land T_0) \iff p\) (regardless of what \( p \) represents). As should be fairly evident, these two Logical forms are perfectly isomorphic to their arithmetic counterparts. Turning to the second pair of Axioms, the Axioms of Opposites, what we find here is that for every Proposition whatsoever, say \( p \), there exists an Opposite Proposition, called “the Negation of \( p \)”, such that \((p \lor \neg p) \iff T_0\) and \((p \land \neg p) \iff F_0\).

And of course we should be familiar with this from Arithmetic, for it is well known that there every expression of the form \((n + -n) = 0\) and any expression of the form \(1 \times 1^{-1} = 1\). In addition, for these Arithmetic forms, the results of these operations with Opposites results in the Identity Element for each of the Operations (that is, \( n + -n = 0 \), where 0 is the Identity for +; and \( n \times n^{-1} = 1 \), where 1 is the Identity for \( x \)). And a similar situation does exists in Logic, although here, again, caution is needed, because these situations are not exactly the same (that is, with respect to this point Logic and Arithmetic are not perfectly isomorphic). In fact, whenever we operate on Opposites in Logic the results we get are indeed Identity Elements, but rather than being the Identity Element of the given Operator, we get the Identity Element for the other Operator! That is, when we work out the results of the Proposition \((p \lor \neg p)\) we get an Identity Element—in this case, \( F_0 \)—but this is not the Identity Element for \( \lor \), the Disjunction Operator used here, but rather for \( \land \), Conjunction. And vice versa when we conjoin Opposites; we get the Identity Element \( T_0 \), which is not the Identity for \( \land \), but rather for \( \lor \). This, then, is the second major difference between the deductive systems of Logic and Mathematics.

This marks the completion of the development of our Deductive System; and so now we are ready to actually perform some deductions. In this, our express purpose will be to
deduce various Theorems that we can then use to do further deductions; and ultimately we would continue on, extending our deduction of Theorems into the realms of Arithmetic, Algebra, etc. Here, however, we have neither the space nor the time, not to mention the need, for such advanced deductions; and accordingly we will limit ourselves to only the number of Logical deductions we need to familiarize ourselves with this process. For the sake of clarity, the complete list of Axioms is given below, while a complete list of the Theorems we will be using is given at the bottom of the following page. Happily, of course, we will not need to develop proofs for every one of these Theorems, although a complete collection of the proofs is provided in Appendix 1. As for the Axioms, we must not fail to point out that these rules make use of both the one-way deduction of Formal Implication and the two-way deduction of Formal Equivalence. This is significant, philosophically, for it demonstrates that, as suggested earlier in our studies, the essence of Logical Deductions lies in the one-way relation of Contiguity (exemplified here in the Formal Implications) and the two-way relation of Isomorphism (exemplified here in the Formal Equivalences).

### Rules of Formal Implication

<table>
<thead>
<tr>
<th>Rule</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>FI</td>
<td>[(p &gt; q) \land p ] =&gt; q</td>
</tr>
<tr>
<td>SD</td>
<td>(p \land q) =&gt; p</td>
</tr>
</tbody>
</table>

### Rules of Formal Equivalence

<table>
<thead>
<tr>
<th>Rule</th>
<th>Expression</th>
</tr>
</thead>
</table>
| MI   | (p > q) <=> (-p v q)  
|      | <=> -(p \land -q) |
| ME   | (p < > q) <=> [(p > q) \land (q > p)]  
|      | <=> [(p \land q) \lor (-p \land -q)] |

<table>
<thead>
<tr>
<th>Rule</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>I\lor</td>
<td>(p \lor F_0) &lt;=&gt; p</td>
</tr>
<tr>
<td>I\land</td>
<td>(p \land T_0) &lt;=&gt; p</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>O\lor</td>
<td>(p \lor \neg p) &lt;=&gt; T_0</td>
</tr>
<tr>
<td>O\land</td>
<td>(p \land \neg p) &lt;=&gt; F_0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>C\lor</td>
<td>(p \lor q) &lt;=&gt; (q \lor p)</td>
</tr>
<tr>
<td>C\land</td>
<td>(p \land q) &lt;=&gt; (q \land p)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>A\lor</td>
<td>[p \lor (q \lor r)] &lt;=&gt; [(p \lor q) \lor r]</td>
</tr>
<tr>
<td>A\land</td>
<td>[p \land (q \land r)] &lt;=&gt; [(p \land q) \land r]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>D\lor</td>
<td>[p \lor (q \land r)] &lt;=&gt; [(p \lor q) \land (p \lor r)]</td>
</tr>
<tr>
<td>D\land</td>
<td>[p \land (q \lor r)] &lt;=&gt; [(p \land q) \lor (p \land r)]</td>
</tr>
</tbody>
</table>
Turning, then, to the deduction of the Theorems, consider the first Theorem, entitled Disjunctive Augmentation. This Theorem is so named because in it we start with just a single Proposition—here, \( r \)—as shown to the left of the Formal Implication symbol, and to this we add or “augment”, by means of Disjunction, any other Proposition whatsoever (although, of course, usually when we do add a Proposition, we added one that we think we might need at some later point in our proof). Disjunctive Augmentation—or, as it is often called, Addition—is thus a powerful Logical tool, allowing as it does for the introduction of formerly unavailable Propositions. And of course, the deduction of this Theorem here demonstrates that it is a legitimate Logical move, and in fact is a Tautology. And what this Theorem says is that if \( r \) (or in fact any Proposition) is assumed to be True, then \( r \) necessarily implies the truth of \( (r \lor s) \), where \( s \) too represents any Proposition (other than \( r \), of course).

**Theorem 2 — Disjunctive Augmentation (DA; Addition):** \( r \Rightarrow (r \lor s) \)

1) \( r \) \hspace{1cm} Hyp
2) \( r \lor F_0 \) \hspace{1cm} I\lor(1)
3) \( r \lor (s \land \lnot s) \) \hspace{1cm} O\land(2)
4) \( (r \lor s) \land (r \lor \lnot s) \) \hspace{1cm} D\lor(3)
5) \( (r \lor s) \) \hspace{1cm} SD(4)

Analyzing the deduction, line 1) is simply the left-hand side of the candidate Theorem, while the remaining lines actually carry out the deduction, using various Axioms to legitimate the construction of each new line. In line 2), then, we employ the Axiom of Identities for \( \lor \), which tells us that if we have any Proposition that is True (which we assume of \( r \) in line 1)) then we can transform this into \( r \lor F_0 \), because this latter expression is Logically Equivalent to \( r \). Next, in line 3), we use the Axiom of Opposites for \( \land \), which tells us that \( F_0 \) is Logically Equivalent to any Contradiction, and so we replace the \( F_0 \) of line 2) with the explicit Contradiction \( (s \land \lnot s) \). This, however, gives us in line 3) a combination of Disjunction (outside of the parentheses) and Conjunction, (inside the parentheses), in which situation we can legitimately distribute the Disjunction over the Conjunction. Accordingly, in line 4) we do just that, using the Distribution of Disjunction over Conjunction to get the new, distributed Proposition given in line 4). And finally, in line 5) we employ the Rule of Symbolic Deduction, that says whenever we have a Conjunction that is assumed to be True (as we have in line 4)), then we can assert the truth of either Conjunct, and this is precisely what we do to get the final line of the deduction, line 5). When we put the whole thing together, as above, it tells us that the truth of \( r \) necessarily implies the truth of \( r \lor s \), as stated in this Theorem.

At this point, of course, we do not expect the reader to appreciate the utility of proving this Theorem, since we have not provided the requisite experience to understand
what is really going on here. But the utility of this particular Theorem can be made readily apparent by considering the following deduction, which is a proof of one of the four so-called Laws of Thought introduced earlier, the Law of Identity. This law basically says that a thing is what it is, although for our purposes we take it to mean that a given Proposition is Logically Equivalent to itself.

Theorem 5 — The Law of Identity (LI): \( r \leftrightarrow r \) [Reflexivity of \( \leftrightarrow \)]

1) \( r \)  
2) \( r \lor \sim r \)  
3) \( \sim r \lor r \)  
4) \( r \Rightarrow r \)  
5) \( r \)  

Hyp  
DA,T02(1)  
Cv(2)  
MI(3)  
FI(1,4)

Analyzing this deduction, line 1) merely hypothesizes the truth of the left-hand side of the proposed Theorem, \( r \), as in the former proof; the rest of the proof then deduces the right-hand side (which here just happens to be the same as the left-hand side) from line 1). More specifically, Line 2) of the proof contains an instance of Theorem 2), which we proved above and whose utility we are illustrating here; and thus by means of Disjunctive Augmentation, or Theorem 2, we can add \( \sim r \) to the original \( r \) and thereby create the given Disjunction: \( r \lor \sim r \). In line 3) we introduce yet another Axiom, but this one is straightforward enough, since it is just the Axiom of Commutativity, which allows us to reverse the order of the Propositions in line 2). The reason that we do this reversal is displayed in line 4): it allows us to use the Definition of Material Implication (abbreviated here as MI), which shows that line 3), \( \sim r \lor r \), is Logically Equivalent to line 4), \( r \Rightarrow r \). And this is precisely what we need to complete the proof, for (first) line 4) gives us a True Implication (as assumed in the proof), and (second) line 1) says that \( r \) itself, which just happens to be the Antecedent of line 4), is also True; accordingly, we can thus apply the Rule of Implication (symbolized by FI in the proof) to lines 4) and 1) to get the truth of the Consequent of line 4): \( r \). In effect, this completes the proof. However, since the Antecedent in line 4) is the same as the Consequent in line 4) we can actually reverse the Implication itself; and although this seems rather fruitless (since both the Antecedent and the Consequent in line 4) are \( r \)), what it actually accomplishes is a demonstration that this particular Implication is Logically necessary in both direction and is thus a Logical Equivalence. And this move, finally, completes the proof in toto.

Although we could continue proving the remaining Theorems, this would unnecessarily take up a considerable amount of time in our studies, and it is thus left to the instructors discretion. Rather than multiplying proofs ad nauseam, our time would be better served by a philosophical discussion of the more relevant Theorems, as well as the implications of such deductive systems for human understanding in general. Of course,
the student cannot be expected to be familiar with the standard list of Theorems (as these are typically found in Logic texts), and so to facilitate our discussion we provide a complete list of the Theorems, below. It should be noted that the Theorems listed on the left-hand side are Formal Implications, while those on the right-hand side are Formal Equivalences. As such, the list illustrates the fact that, as mentioned several times, Formal Deduction is based upon Contiguity (giving the Implications) and Isomorphism (giving the Equivalences).

*The Theorems of the Deductive Systems*

<table>
<thead>
<tr>
<th>Formal Implications</th>
<th>Formal Equivalences</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DA:</strong></td>
<td><strong>LI:</strong></td>
</tr>
<tr>
<td>( r \Rightarrow (r \lor s) )</td>
<td>( r \Leftrightarrow r )</td>
</tr>
<tr>
<td><strong>DR:</strong></td>
<td><strong>UN:</strong></td>
</tr>
</tbody>
</table>
| \[ (p \lor q) \land \neg p \] \Rightarrow q | \{ \[( r \lor s) < > T_{0} \land [ (r \land s) < > F_{0}] \} \Leftrightarrow \\
| & \left( -r < > s \right) |
| **MT:** | **DN:** |
| \[ (p > q) \land \neg q \] \Rightarrow \neg p | \neg (\neg s) \Leftrightarrow s |
| **TI:** | **AB:** |
| \[ ((p > q) \land (q > r)) \] \Rightarrow (p > r) | \( (p > q) \Leftrightarrow [p > (p \land q)] \) |
| **CD:** | **EX:** |
| \[ ((p > q) \land (r > s) \land (p \lor r)) \] \Rightarrow (q \lor s) | \( [p > q] \Leftrightarrow [p > (q > r)] \) |
| **CP:** | **ID:** |
| \( (p > q) \Leftrightarrow (\neg q > \neg p) \) | \( (r \lor r) \Leftrightarrow r \) |
| & \( (r \land r) \Leftrightarrow r \) |
| **AN:** | **DM:** |
| \( (r \lor T_{0}) \Leftrightarrow T_{0} \) | \( (r \lor T_{0}) \Leftrightarrow T_{0} \) |
| \( (r \land F_{0}) \Leftrightarrow F_{0} \) | \( (r \land F_{0}) \Leftrightarrow F_{0} \) |
| **DI:** | **DM:** |
| \( (\neg (p \lor q) \Leftrightarrow (\neg p \land \neg q) \) | \( (\neg (p \lor q) \Leftrightarrow (\neg p \land \neg q) \) |
| \( \neg (p \lor q) \Leftrightarrow (\neg p \lor \neg q) \) | \( \neg (p \land q) \Leftrightarrow (\neg p \lor \neg q) \) |
| \( \neg F_{0} \Leftrightarrow T_{0} \) | \( \neg T_{0} \Leftrightarrow F_{0} \) |
| \( \neg T_{0} \Leftrightarrow F_{0} \) |
Proof Theory — Direct Proofs, Indirect Proofs, and Inconsistency

In the Deductive System that we have developed here we have, as we noted at the outset, used a mixed approach, including within our necessarily Syntactical system a modicum of Semantics. For us, this Semantics took the form of the Logical Matrices; not only the five primary Matrices—those of the Operators $\lor, \land, \neg, >$, and $< >$—but other matrices as well, such as those of the four Laws of Thought. At this point, however, it is time to present our Deductive System in its purest form, a form that foregoes all Semantics in favor of a purely Syntactical approach. For, as it turns out, the structure and functions provided by the Semantics of the Logical Matrices can be delivered in a fully serviceable form (and actually was so delivered in our Deductive System as it stands, although we took no notice of it) by appropriately structured Rules of Syntactics.

Accordingly, a purely Syntactical system is given below in three steps. In step 1) we give the Lexicon, which is identical in this case to the Lexicon given earlier, and hence we need not concern ourselves further with it. In step 2), however, we take a slightly different turn than previously, for rather than provide a complete Grammar we give only the Rules of Syntactics, as stipulated above, foregoing all Semantics. Here, these syntactical rules are called Rules of Formation or Production, and they comprise two sets: 1) Rules of Expression, which dictate the allowable forms that our Logical expressions may take when entered upon a line of proof; and 2) Rules of Replacement, which provide allowable substitutions for various forms of expression. This latter set of rules, the Rules of Replacement, are more or less exactly what we met with before in the ten Axioms of Algebraic Structure, although here we have added the Definition of Material Implication and the Definition of Material Equivalence, which are of course the same type of expression as the Axioms of Algebraic Structures—Equivalences. As for the Rules of Expression themselves, they are not unlike what we had before in the Formation Rules above, although here we have included rules for Implication ($>$) and Equivalence ($< >$), since we need these forms to be among the allowable forms of initial expressions. Note, however, that neither these two inferential operators nor the counterparts here of the Basic Operators used earlier ($, \neg, \lor, \land$) have any meaning whatsoever; in this purely syntactical system we have forsaken Semantics, meaning, altogether.

1) The Lexicon, $(P,R)$, of the Deductive System, $DS$:

Operands: a Class $P = (p, q, r, s, t, \ldots, T_0, F_0)$ of Primitive Variables
Operators: a Class $R = (\approx, \neg, \lor, \land, >, < >)$ of Primitive Constants
Metalogical Symbols: a Class $M = ((), [], \{, \}, \ldots)$
2) The Syntactics of DS — Rules of Formation — Expression and Replacement:

Rules of Expression:

Rules for $$\approx$$ and $$\sim$$:

Position: $$E\approx: \approx p \ / \ p$$
Negation: $$E\sim: \sim p$$

Rules for $$\lor$$ and $$\land$$:

Disjunction: $$E\lor: p \lor q$$
Conjunction: $$E\land: p \land q$$

Rules for $$\rightarrow$$ and $$\leftrightarrow$$:

Implication: $$EI: p \rightarrow q$$
Equivalence: $$EE: p \leftrightarrow q$$

Rules of Replacement:

Identities: $$I\lor: p \leftrightarrow (p \lor F_0)$$
$$I\land: p \leftrightarrow (p \land T_0)$$

Opposites: $$O\lor: T_0 \leftrightarrow (p \lor \sim p)$$
$$O\land: F_0 \leftrightarrow (p \land \sim p)$$

Commutation: $$C\lor: (p \lor q) \leftrightarrow (q \lor p)$$
$$C\land: (p \land q) \leftrightarrow (q \land p)$$

Association: $$A\lor: [p \lor (q \lor r)] \leftrightarrow [(p \lor q) \lor r]$$
$$A\land: [p \land (q \land r)] \leftrightarrow [(p \land q) \land r]$$

Distribution: $$D\lor: [p \lor (q \land r)] \leftrightarrow [(p \lor q) \land (p \lor r)]$$
$$D\land: [p \land (q \lor r)] \leftrightarrow [(p \land q) \lor (p \land r)]$$

Material Implication: $$MI: (p \rightarrow q) \leftrightarrow (\sim p \lor q)$$
$$MI: (p \rightarrow q) \leftrightarrow \sim(p \land \sim q)$$

Material Equivalence: $$ME: (p \leftrightarrow q) \leftrightarrow [(p \lor q) \land (q \lor p)]$$
$$ME: (p \leftrightarrow q) \leftrightarrow [(p \land q) \lor (\sim p \land \sim q)]$$

3) The Theorization of DS — Rules of Derivation (Proof Theory):

The Rule of Induction: $$SI: [p], [q] \Rightarrow [(p \land q)]$$
The Rule of Deduction: $$SD: [(p \land q)] \Rightarrow [p] \text{ and } [q]$$
The Rule of Implication: $$FI: \{[p], [(p \rightarrow q)]\} \Rightarrow [q]$$
Finally, in step 3) of our development (as shown above) we provide the Rules of Derivation, and these too are not unlike the their earlier counterparts. For we have here as before the Rule of Formal Implication and the Rule of Symbolic Deduction, and these two rules work precisely as they did above. In addition, however, one new rule has been included, the Rule of Symbolic Induction, which says that any two lines of a Proof may be conjoined on a new line. This addition is necessary here because, whereas in our earlier system we could justify conjoining any two lines of a deduction by means of the Logical Matrix for Conjunction, here we do not have recourse to Logical Matrices. Nevertheless, we will still occasionally have the need to produce new lines of a Proof by conjoining previously existing lines, and the Rule of Abduction gives us the ability to do just that. For this Syntactical Deductive System, then, we have three Rules of Derivation, as opposed to two in the previous system.

Turning, now, to the notion of Proof itself, we offer the following definition of a Proof:

\[ \text{The Definition of 'Proof': A 'proof' is 'a finite sequence of Expressions in which the first Expression is justified by any Rule of Expression; and every other Expression is justified by: 1) any Rule of Replacement; 2) any Rule of Derivation, or 3) any Theorem.'} \]

A Proof, then, is a series of Logical Expression, the initial line of which bears one of the forms allowable under the Rules of Expression listed above, and which is of course assumed to be True; while the forms and the truth of the subsequent lines are either derived from the initial line or are further assumptions necessary for the Proof. As before, the structure of a Proof is based upon the Axiom of Deduction (known traditionally, we remind the reader, as Modus Ponens); and accordingly, the form of a Proof is nothing but the form of the Axiom of Formal Implication. This form is of course that of an Implication (or an Equivalence, which amounts to the same thing) in which, as before, all of the lines of the Proof from the initial line to the last-but-one form a Conjunction that serves as the Antecedent of this Implication, while the last line itself constitutes the assertion of the Consequent of this Implication. As Implications, then (either explicitly, or implicitly as Equivalences), Proofs constitute a form of inferential Argument, and the last Expression in a Proof, accordingly, is called the Conclusion of the Argument or Proof (or merely “the Conclusion of the Proof”), while each of the other Expressions are called Premises of the Proof.

Not surprisingly, Proofs come in various types, which include (but are not limited to) Direct Proofs and Indirect Proofs. Direct Proofs, as the name suggest, address themselves in a straightforward manner to the truth of the proposed Theorem, which means that they show that the proposed Theorem is a Tautology. Indirect Proofs, in
contrast, follow the more roundabout path of showing that the Negation of the proposed Theorem is a Contradiction, which of course implies that the proposed Theorem itself is a Tautology (according to the Duality of Tautologies and Contradictions). We begin with the Direct Proof, of which we give three different forms.

1) Direct Proof of Tautology:

Show that $T_7 \iff T_0/A_m$ (for some $n,m \in \mathbb{N}$); i.e., show $T_7 \iff T_0$

In this first form of the Direct Proof (in which $T_7$ is the Theorem to be proved, $T_n$ is any Theorem that has been proved, and $A_m$ is any Axiom) we attempt to demonstrate that the proposed Theorem $T_7$ is Equivalent to a Tautology. A specific example of this form of Direct Proof is given below:

**Theorem 8 — Double Negation (DN):**

$\sim (\sim s) \iff s$

1) $(s \lor \sim s) \iff T_0$ \quad O \quad $(s \land \sim s) \iff F_0$
2) $(\sim s \lor s) \iff T_0$ \quad C \quad $(\sim s \land s) \iff F_0$
3) \quad $\sim (\sim s) \iff s$ \quad UN [T07]

In this Theorem, entitled Double Negation, we wish to show that the Negation of the Negation of a Proposition is Logically Equivalent to the Proposition itself, which in the notation used above becomes $\sim (\sim s) \iff s$. To begin, we merely list, as our initial line, line 1), the two forms of the Axiom of Opposites; and although we would typically be required to provide the Rule of Expression (which in this case would be EE) that justifies the particular Logical form used here, we can actually forego this because all of the Rules of Replacement (of which EE is an instance) constitute legitimate forms. Next, we apply the Axiom of Commutativity to line 1) to get line 2), whose two parts just so happen to be tantamount to the two the two Premises of Theorem 7 (on the Uniqueness of Negatives, which we have not proved but whose proof is provided in the Appendix). What this Theorem tells us is that, if the Disjunction of two Propositions is Equivalent to a Tautology (as is given on the left of line 2)) and the Conjunction of the same two Propositions is Equivalent to a Contradiction (as given on the right of line 2)), then the Negation of the first Proposition (here, $\sim s$) is Logically Equivalent to the second Proposition. Accordingly, the Conclusion, line 3) states just that, and the proof that the Negation of the Negation of a Proposition is Logically Equivalent to the original Proposition is thus complete.

For the next two forms of the Direct Proof the tactic is to show that the Antecedent of an Implication or an Equivalence formally implies the Consequent (and vice versa for the Equivalence). Before proceeding to the proofs, however, the student should note that
in forms 2) and 3) of the Direct Proof the symbol $T_A$ represents the Antecedent, while $T_C$ represents the Consequent, of the proposed Theorem; keep in mind also that the Formal Equivalence ($\iff$) in 3) is a Conjunction of two Formal Implications, as defined, and thus all Proofs involving Formal Equivalence are tantamount to two Formal Implications.

2) Direct Proof of Formal Implication:

Show $T_A \Rightarrow T_C$ (when the $T$ uses $\Rightarrow$)

This Proof concerns itself with a form of Inference that is traditionally known as Modus Tollens, literally the "Mode of Tacking", and which, like Modus Ponens, is one of the earliest recognized forms of valid reasoning. In our proof of Modus Tollens, the tactic is to demonstrate that the Antecedent of the Implication Formally Implies the Consequent, where the Antecedent involves, as earlier with Modus Ponens, a Material Implication in Conjunction with a single Proposition. Unlike Modus Ponens, however, in Modus Tollens, although its second Conjunct, a single Proposition, involves a form of one of the Propositions in the Material Implication—here, $q$, the Consequent, whereas with Modus Ponens it was $p$, the Antecedent—the Proposition $q$ used here is given in negated form, thus: $\neg q$. But (recalling the inclusive geometric relation of $p > q$ from our earlier explanation, in which ‘$p$’ is entirely included within ‘$q$’) it now becomes fairly apparent that, since $\neg q$ (as the second Conjunct) basically means that no marker is anywhere in the ‘$q$’ circle, it also means that no marker can possibly be in the ‘$p$’ circle either. Consequently, we may deduce $\neg p$ from $(p > q) \land \neg q)$, as we do below in the actual Proof. This Proof, then, is a fitting algebraic counterpart to the intuitive inferences we have just derived from the geometric representation of this Theorem.

Theorem 4 — Modus Tollens (MT, ‘The Mode of Taking’): $[(p > q) \land \neg q] \Rightarrow \neg p$

1) $p > q$ Hyp1, EI
2) $\neg p \lor q$ MI(1)
3) $q \lor \neg p$ C\lor(2)
4) $\neg q$ Hyp2, E\neg
5) $\neg p$ DR,T03(3, 4)

In the proof, however, as opposed to our geometric intuitions, we are using the algebraic approach (that is, we are doing a Proof), and the tactic is to assume the truth of each of the Conjuncts of the Conjunctive Antecedent in the proposed Theorem, and then logically deduce the truth of the Consequent. Accordingly, in step 1) we assume the truth of $p > q$, which of course gives us our required Inclusion relation; step 2) then modifies the form of 1) by applying the Definition of Material Implication. Step 3) reverses the order of the
Conjuncts of line 2) by means of Commutativity, so that we can get q in plane sight, as it were; and then step 4) introduces the second Conjunct of the Antecedent of the proposed Theorem, which again is assumed to be True. But now we notice that, with lines 3) and 4), we have the makings of Theorem 2, Disjunctive Reduction (traditionally known as the Disjunctive Syllogism), and this allows us to deduce line 5), which of course is the desired denial of the Antecedent of \( p > q \), and the Proof is thus complete.

3) Direct Proof of Formal Equivalence:

Show \( T_A \iff T_C \) (if \( T_7 \) uses \( \iff \))

This next Proof, number 3), uses a tactic similar to that used in proving the previous Theorem, in that it demonstrates a Formal logical relation between the two components of the proposed Theorem. Here, however, in contrast to the previous Proof, in which at least one step in the Proof involves a Formal Implication, we need to ensure that each and every step entails a Formal Equivalence. This is requirement, of course, because the Theorem itself is intended to demonstrate an Equivalence; consequently, if we have even one step in the Proof that is tantamount a Formal Implication (as in line 5) of Theorem 3, above), the Proof will not posses the required two-way flow of Equivalence, and thus will not work both ways (from the top, down; and from the bottom, up). And if this is the case we will not have demonstrated Formal Equivalence. Accordingly, each step in the Proof of Theorem 13, below, involves some type of Formal Equivalence, and with just a slight adjustment of the application of these steps we can read the Proof from the top down or from the bottom up (as we shall see). In the former direction, line 1) gives the left-hand side of the proposed Equivalence as an assumption, which coincidently involves the same Material Implication, \( p > q \), as in the previous Proof; and accordingly, line 2) applies the same justification as in the previous Proof, to produce the disjunctive form. In addition, line 3) also makes the same move as above, which is merely to commute the two sides of the Conjunction of line 2). Next, however, we require a new step, which this time is the application of the Theorem of Double Negation; and this sets us up for the move in step 5), where we call upon the Definition of Material Implication. This gives us the Transposition of the Implication of line 1), and the first half of the proof is complete.

Theorem 13 — Transposition (TP): \((p > q) \iff (\neg q > \neg p)\)

<table>
<thead>
<tr>
<th>Step</th>
<th>Statement</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>( p &gt; q )</td>
<td>Hyp, EI</td>
</tr>
<tr>
<td>2)</td>
<td>( \neg p \lor q )</td>
<td>MI</td>
</tr>
<tr>
<td>3)</td>
<td>( q \lor \neg p )</td>
<td>C_\lor</td>
</tr>
<tr>
<td>4)</td>
<td>( \neg (\neg q) \lor \neg p )</td>
<td>DN [T08]</td>
</tr>
<tr>
<td>5)</td>
<td>( \neg q &gt; \neg p )</td>
<td>MI</td>
</tr>
</tbody>
</table>
For the second half of the proof, which is required to show the Equivalence, we move the “Hyp, EI” justification used in line 1) to line 5); then, by shifting all of the other justifying rules up one line (so that line 5)’s justification moves up to line 4), line 4)’s up to line 3), etc.), we read the proof from the bottom up, and the proof is thus complete, in toto.

4) Indirect Proof: Show that \( \neg T \iff F_0 \)

Finally (for the different forms of proof demonstrated here), we introduce what is called the Indirect Proof, in which, as symbolized above, we attempt to demonstrate that the Theorem in question (\( \neg T \)) is Formally Equivalent to a Contradiction (\( F_0 \)). And this indirect move does indeed accomplish our task—which is to show that \( T \) is tautologous—because we are using a two-valued Logic. In a two-valued Logic, as we know, if a Proposition is a Contradiction (even if the Proposition starts out negative) then its negation must be a Tautology. More precisely (as symbolically represented above), if—as this form of proof insists—the Negation of a proposed Theorem is a Contradiction (whose values are always F) then the Negation of this negated Theorem (\( \neg \neg T \))—not to mention its equivalent, the negated Contradiction (\( \neg F_0 \))—can only have values that are always T. But of course, if this is case, the un-negated Proposition itself is a Tautology, and the task is accomplished.

To exemplify this form of reasoning, we give an example of the Indirect Proof whose entailed Proposition has long been recognized as an extremely important form of valid reasoning, traditionally known as the Hypothetical Syllogism. The reason for this traditional designation, of course, is the fact that the Antecedent of the Theorem involves two component Implications (which are also known as Hypotheticals) that can be viewed as being the two Premises of an Aristotelian Syllogism, while the a third Implication (in the Consequent of the Hypothetical) can be viewed as the Conclusion of a Syllogism. This syllogism thus contains only Hypotheticals, and its syllogistic form, which is the traditional form of Argumentation, is as follows:

1) \( p \to q \)
2) \( q \to r \)
\[ \therefore \ p \to r \]

Now, since what the syllogistic form represents according to Aristotle is expressed by the statement “If Premise 1) is True and Premise 2) is True, then the Conclusion is True”, we can see the justification for recasting this traditional Argument into the modern form given in the above. Accordingly, we shall analyze this Argument in its modern form, which basically says that from the Conjunction of two Implications in which the
Consequent of the first Implication appears as the Antecedent of the second Implication, a third Implication can be deduced whose Antecedent is the Antecedent of the first Implication and whose Consequent is the Consequent of the second Implication. The modern form, then, which is known as the Transitivity of Implication, demonstrates that the “chain together” of two appropriate Implications—such as are given by \((p > q)\) and \((q > r)\)—allows us to move from the first Antecedent, *transiting* over the two identical \(q\) in the middle, to the second Consequent. This transiting of the Implication is illustrated in the proposed Theorem below. Unfortunately, however, this particular proof is a bit too complicated for us to go into all of its details, and we shall accordingly only touch upon its more significant features (although we do provide the complete proof, as follows):

Theorem 14 — Transitivity of Implication (TI, HS): \([p > q) \land (q > r)] \Rightarrow (p > r)\)

1) \(~[[p > q) \land (q > r)] \Rightarrow (p > r)]\) Ass, E~
2) \(~[[p > q) \land (q > r)] \land ~(p > r)]\) MI(1)
3) \((p > q) \land (q > r) \land ~(p > r)\) DN(2) [T08]
4) \(p > q\) SD(3)
5) \(q > r\) SD(3)
6) \(~(p > r)\) SD(3)
7) \(~(p \lor r)\) MI(6)
8) \((p \land ~r)\) DM(7)[T10]
9) \(~p\land q\) MI(4)
10) \(~p\) P(1)(9) 10) \(q\) P(2)(9)
11) \(p\) SD(8) 11) \(~q\) DN(10) [T08]
12) \(p \land ~p\) SI(10,11) = \(F_0\) 12) \(~q \lor r\) MI(5)
13) \(r\) DR(12,11) [T03]
14) \(~r\) SD(8)
15) \(r \land ~r\) SI(13,14) = \(F_0\)

The first feature we should note is that this proof is in fact an Indirect Proof since, as we see in line 1), the proposed Theorem is itself negated and, as we see in lines 12) and 15), the two conclusions are equivalent to Contradictions. And this peculiar feature—having two Conclusions instead of just one, as we typically do—is the second feature we should note, for it is one we have not seen before. This form of proof, however, which is known as a branching proof, is a legitimate form of proof; nor is it restricted to the Indirect Proof. In fact, this form of proof can be used whenever we have deduced the truth of a Disjunction whose two Disjuncts we can also demonstrate to be False. And this is possible because, as we know all too well by now, if a Disjunction is True then at least one of its Disjuncts must be True; but if the Disjunction’s two Disjuncts are themselves equivalent to Contradictions, then the Disjunction itself cannot be True. Note, however,
that in the actual example proof we have formally deduced the Disjunction of line 9) (the Disjunction that, here, leads to the two Contradictions) from the Negation of the proposed Theorem. Consequently, the negated Theorem itself is a Contradiction, and thus by the proof given previously (or more significantly, from its formal counterpart, Theorem 9) the proposed Theorem itself, Theorem 14, is a Tautology, and the proof is thus completed.

Inconsistency

This is about as much of Proof Theory as we need to concern ourselves at the present time (although of course the instructor may wish to develop this subject in greater detail). Before closing the present section, however, we must say a few words about Inconsistency, a topic in Logic that is of fundamental importance. As used here, the word ‘inconsistency’ is synonymous with ‘contradiction’; and thus the opposite of Inconsistency, Consistency, is synonymous with Tautology, and this is an absolute requirement of deductive systems. Above, this requirement was provided for us by prohibiting Contradictions in Logic; now, however, we will learn why Contradictions are so pernicious, and why they must be avoided in Logic. Accordingly, this short proof begins, in step 1), with a Contradiction, \( p \land \lnot p \), which also happens to be a Conjunction, and from the assumed truth of this Conjunction we can infer the truth of either Conjunct, according to the Rule of Reduction. Accordingly, line 2) asserts the truth of \( p \); and line 3) then uses the Theorem of Disjunctive Augmentation on line 2) to introduce a new Proposition, which in this case is \( q \), giving line 3) the form \( p \lor q \). But from line 1) we can also assert the truth of the second Conjunct, \( \lnot p \), which is then submitted on line 4) by means of the same rule as in line 2). Finally, combining lines 3), a Disjunction, and 4), the denial of the left-most Disjunct, we can use the Theorem of Disjunction Reduction (also known as Theorem 2) to deduce the desired Consequent, \( q \), and this is given on line 5).

The Perniciousness of Inconsistency: \( (p \land \lnot p) \Rightarrow q \)

\[
\begin{align*}
1) & \quad p \land \lnot p & \text{Hyp, E} \land \\
2) & \quad p & \text{SD}(1) \\
3) & \quad p \lor q & \text{DA}(2) \ [T02] \\
4) & \quad \lnot p & \text{SD}(1) \\
5) & \quad q & \text{DR}(3,4) \ [T03]
\end{align*}
\]

This proof is quite simple, and is, I think, easy enough to follow; but its simplicity may in fact conceal its significance. The key here is to remember that the Propositional
Variable q represents any Proposition whatsoever, which means that from any Contradiction we can deduce the truth of anything and everything. But that is not at all what we want from our Logic; rather, we wish to prove only what is actually or possibly True, and conversely we do not want any falsehoods to crop up in our Conclusions. In Inductive Logic, of course, this is a patently impossible goal; because of Induction’s reliance upon perception, not to mention the Inductive leap, Inductive Logic is always susceptible to error. In Deductive Logic, however, we can and must ensure Consistency, and there are ways of testing the proving the Consistency of any legitimate deductive system. For us, however, whose purpose is merely an introductory level understanding of Logic, Consistency begins with the rejection of all Contradictions from our Logic systems.

Higher Order Logics — Quantification and Predication

The final topic for discussion in the development of our Deductive System is that of will concern itself with certain aspects of what are called Higher Order Logics. Specifically, we shall briefly consider Quantification Theory and Predication Theory. The Logic that we have developed thus far is known as First Order Logic; this Logic is concerned only with the inter-relations between and among Propositions, with no concern for the contents of the Proposition. Second Order Logic, however, does look “inside” Propositions, and discovers there concepts that include notions of quantity or number, albeit in the simplified form of “All”, “Some”, or “None”. And Third Order Logic, which considers the Logic of Predicates assigned to Subjects, takes the Logical analysis a step further.

Quantification Theory, which, as the name suggests, introduces into our current system the two quantifying symbols that we met with in our exposition of Aristotle’s Categorical Logic: \( \forall \) for “All” and \( \exists \) for “Some” (or, strictly speaking, “There exists at least one”). As the student will recall, our use of these symbols in Aristotle’s Logic was in effect illegitimate, since the great Peripatetic philosopher did not himself employ symbols for these concepts. As our use of them was merely utilitarian and for the sake of simplicity, and since they did not in fact alter the nature and characteristics of Aristotle’s Logic, this illegitimacy were excusable. Here, however, the use of these two symbols is not only legitimate but is absolutely necessary, in order to meet the needs of the higher orders of Logic.
References

1 Needless to say, this “or both” phrase is critical, because without this phrase we would have the type of Disjunction we need, which is known as Inclusive Disjunction. Rather, we would have a different form of Disjunction, known as Exclusive Disjunction. Ironically enough, the latter type of Disjunction is used in computer design, but it is not preferred in the form of Deductive Logic we are developing here, which requires the Inclusive Disjunction defined in Figure 4-6.

2 As we shall learn in Chapter Five, in a definition the symbol that is being defined is called the definiendum, while the definition itself is called the definiens. Accordingly, we adopt this terminology here, albeit somewhat prematurely.