

A Primer on Logic
Part 2:
Statements, Truth Functions, and Argument Schemas
Jason Zarri

1. *Introduction*

In this Part our goal is to learn how to determine the validity of *argument schemas*, abstract “forms” or patterns that multiple arguments have in common. Along the way it will be necessary to examine statements in more detail, in particular the ways in which simpler ones can be combined to make more complex ones, for in doing so we will find ways to tell which argument schemas are valid and which are not. We shall thus need to dabble in *formal logic*, the philosophical and mathematical¹ field which studies deductive reasoning.

2. *A Sample Argument Schema*

Let us begin by returning to the test to which you were subjected in Part 1. You were asked to determine whether the following claim was true: If a card has a vowel on one side, then it has an odd number on the other side. This claim is general, in the sense that it is not about any card in particular. It just says that, *whatever* card you consider, if *that* card has a vowel on one side then that same card has an odd number on the other side. For our purposes it will be easier to consider a particular instance of this claim:

(1) If card (i) has a vowel on one side, then card (i) has an odd number on the other side.

In logic, statements of this form are called *conditionals*. A conditional is a statement that says *if* such-and-such is the case, *then* so-and-so is the case. Conditionals are compound statements. The conditional (1) is composed of two statements, namely

¹ It used to be thought by *logicians*, such as Gottlob Frege and Bertrand Russell, that all of mathematics was reducible to formal logic. In that case formal logic wouldn't be just another branch of mathematics, but rather a foundation for the entire discipline. Their hopes were dashed in 1931 when Kurt Gödel published a famous paper (see Bibliography) in which he proved that any system of formal logic which meets certain basic conditions inevitably contains true statements that the system, if consistent, can neither prove nor disprove. So there is at any rate no *single* system of formal logic which is both consistent and capable of proving all the truths of mathematics. Formal logic remains an important branch of mathematics, yet it is nevertheless but one branch among many.

(1*a*) Card (i) has a vowel on one side

and

(1*b*) Card (i) has an odd number on the other side.

The first statement of which a conditional is composed, the one following the ‘if’, is called the *antecedent*. The other statement, which follows the ‘then’, is called the *consequent*. In the conditional (1), (1*a*) is the antecedent and (1*b*) is the consequent.

Now it’s time for another test of your reasoning skills. Suppose (1) is true. Suppose also that card (i) does indeed have a vowel on one side, as you could easily verify by turning back to the beginning of Part 1. What can you conclude?

The conclusion you have drawn is, hopefully, that card (i) has an odd number on its other side. Abstracting from some relatively irrelevant details, your train of reasoning could be represented like so:

If card (i) has a vowel on one side, then card (i) has an odd number on the other side.

Card (i) has a vowel on one side.

Card (i) has an odd number on the other side.

This argument is clearly valid. If you doubt it, imagine that someone told you both that if card (i) has a vowel on one side, then card (i) has an odd number on the other side, and that card (i) *does* have a vowel on one side, but that card (i) does *not* have an odd number on its other side. Would you believe them? And if not, why not?

But on what does the validity of this argument depend? It would seem not to depend on any specific feature of the statements involved. Consider the following argument:

If Socrates is human, then Socrates is mortal.

Socrates is human.

Socrates is mortal.

This argument is also valid, and for the same reason as the first. Yet the meanings of its premises and conclusion are completely unrelated to those of the first argument. The only thing that unites them is that they are both instances of a common pattern or “form”. This form consists of two premises, a conditional statement and its antecedent, followed by a conclusion, which is the conditional’s consequent. If we let the letters ‘p’ and ‘q’ stand in for arbitrary statements, we could represent this form schematically:

If p, then q

p_____

q

This schema is known as *Modus Ponens*. We can substitute any statements we like for ‘p’ and ‘q’, and the resulting instances of the schema will always be valid. The schema itself can also be called valid, in the sense that all of its instances are valid. The instances of Modus Ponens are valid because of their *form* and the meaning of the ‘if...then’ construction, not because of the specific meanings of their constituent statements. The same is true of argument schemas generally. Hence the ‘formal’ in ‘formal logic’.

3. *Kinds of Letters*

It is useful to consider formal logic as being analogous to algebra. In algebra one uses letters to stand in for numbers. Similarly, in formal logic one uses letters to stand in for statements. There are different kinds of letters which have somewhat different uses. The first kind are *atomic statement letters*, or *atomic statements* for short, which are analogous to the constants used in algebra. Such constants always represent a definite number, and atomic statements always represent a definite statement. An atomic statement letter is a single letter—‘A’, for example—which is used to stand in for a statement which is *simple* in a sense which will be explained below. We will use capital letters from the beginning to the end of the alphabet as atomic statements. In the highly unlikely event that we run out of these letters, we could attach numerical subscripts to them to generate more, such as ‘E₁₂.’ Only one atomic statement letter may be assigned to a given statement. Because of that, different atomic statement letters must correspond to different statements.

The second kind of letters are called *schematic letters*, and are analogous to the variables used in algebra, which can represent any number you please. We have already made use of them in displaying the schema Modus Ponens. Schematic letters do not stand in for particular statements. Instead they are basically placeholders for which particular statements of arbitrary complexity—including atomic statements—can be substituted. Schemas of arbitrary complexity can be substituted for them as well. We will use lowercase letters from ‘p’ to the end of the alphabet as schematic letters. As above, we can append numerical subscripts to them to make sure we don’t run out. In contrast to atomic statement letters, there is no analogous requirement that one must substitute different statements or different schemas for different schematic letters.

However, there *is* a requirement that if a schematic letter occurs more than once in a schema one must substitute the same statement or schema for all occurrences of that letter.

4. Truth Functions and Truth Tables

In what follows we will proceed on the assumption of *bivalence*—that every statement is true or false, and no statement is both. Or, as a logician would put it, there are exactly two *truth values*, **true** and **false**, and each statement has one and only one of them. So instead of saying “It’s true that water is H_2O ”, one could say “‘Water is H_2O ’ has the truth value **true**”. Similarly, one could say “‘The Earth is the center of the solar system’ has the truth value **false**” instead of the more usual “It’s false that the Earth is the center of the solar system”.

In formal logic, as in mathematics generally, there are various operators which take expressions as input and produce other expressions as output. Some of the more common operators are ‘+’, ‘−’, ‘×’, ‘÷’, and ‘=’. For example, one can take the terms ‘2’ and ‘5’ and use the plus sign, ‘+’, to form the complex term ‘2 + 5’. One can then take the term ‘7’ and the equals sign, ‘=’, and form the statement “2 + 5 = 7”. The operators that will concern us in this Part are called *truth functions*. An operator must satisfy two conditions to count as a truth function. First, it must take statements as input and yield more complex statements as output. Second, the truth values of the more complex statements must be completely determined by the truth values of the less complex statements. Most of the truth functions we will encounter take two statements as input and yield a single statement as output. Negation, the sole exception, operates on a single statement to yield a single statement.

The meaning of a truth function can be displayed by means of a *truth table*. A truth table shows how the truth value of a complex statement is determined by the truth values of its constituent statements and the truth function that operates on them. Ultimately, the truth value of a complex statement is determined by the truth values of its atomic statements.

The simplest truth table is the one for *negation*. Let ‘C’ be an atomic statement which stands in for the statement “Cathy is running”. The truth table for the negation of this statement is then:

C	$\sim C$
T	F
F	T

This truth table is interpreted as follows. ‘ \sim ’ is the negation sign, and corresponds to the English phrase ‘it is not the case that’. “ $\sim C$ ”², in full, is read as “It is not the case that Cathy is running”. The rows of the truth table are read from left to right, starting from the top row and working towards the bottom one. In the first row of the table, the ‘T’ in the C column means that C, i.e. “Cathy is running” has the truth value **true**, while the ‘F’ under the “ $\sim C$ ” column means that, in that same situation, “ $\sim C$ ”, “It is not the case that Cathy is running”, has the truth value **false**. In the second row of the table, the ‘F’ in the C column means that C has the truth value **false**, while the ‘T’ under the “ $\sim C$ ” column means that, in that same situation, “ $\sim C$ ” has the truth value **true**.

Using schematic letters, we can obtain the truth table for negation in general, which looks like this:

p	$\sim p$
T	F
F	T

From this truth table we can see that if any given statement is true then its negation is false, and vice versa.

A little more complex, but still easy to grasp, is the truth table for *conjunction*. The components of a conjunction are referred to as its *conjuncts*. Conjunction is usually expressed in English by the word ‘and’, but here we will use the symbol ‘&’ instead:

² I will adopt the convention of enclosing statements and schemas in quotation marks, except when they contain no truth functions.

p	q	$p \& q$
T	T	T
T	F	F
F	T	F
F	F	F

I will spare the reader from having to read a cell-by-cell interpretation of this table. I think it will suffice to say that whatever statements one substitutes for 'p' and 'q', the conjunction of those statements is true if both are true, and false if either or both of them are false.

The next truth table is the one for *disjunction*, also sometimes called *alternation*. The components of a disjunction are called its *disjuncts*. Disjunction is expressed by the English word 'or', and is represented here by the symbol ' \vee ':

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

As you can see, the disjunction of any two statements p and q is true if either one or both of them are true, and is false only if both of them are false. Since it is true when both of them are true, it corresponds to the colloquial expression 'and / or', and can also be called *inclusive disjunction*. If you like, you can come up with a different symbol to express *exclusive disjunction*, which has the same truth table except that the exclusive disjunction of two statements is false when both of them are true.

Now we come to the truth table for what logicians call the *material conditional* or *material implication*, which is expressed by the symbol ' \supset ':

p	q	$p \supset q$
T	T	T
T	F	F
F	T	T
F	F	T

In my opinion this truth function is the one a beginner will find most puzzling, so it will be given more attention than the others. The truth table for the material conditional is not itself hard to interpret, but of all the truth functions it seems to correspond least well to its English counterpart, which in this case is the 'if...then' construction. Since the properties of the material conditional are something of a mixed bag, I will divide its evaluation into three subsections.

1. **The Good:** The material conditional satisfies Modus Ponens. You can tell by inspecting its truth table that if p and " $p \supset q$ " are both true then so is q.³ In the second and fourth rows, the only cases where q is false, either p is false or " $p \supset q$ " is false. Since Modus Ponens is a valid schema, inferring q from p and " $p \supset q$ " will never take one from truth to falsity.

2. **The Bad:** Because the material conditional is a truth functional operator, one cannot tell whether it is true without knowing the truth values of p and / or q.⁴ Among other things, this means you cannot use Modus Ponens without knowing whether q is true or

³ Strictly speaking, one cannot assign truth values to 'p' and 'q' because they are schematic letters. To be accurate one would have to say something like, "No matter what statement one substitutes for 'p', and no matter what statement one substitutes for 'q', if the statement substituted for 'p' is true ... (etc.)". So, to avoid being needlessly long-winded, from this point on I will write as though assigning truth values directly to schemas and schematic letters.

⁴ If you know that p is false then you know that " $p \supset q$ " is true whatever the truth value of q may be. And if you know that q is true then you know that " $p \supset q$ " is true whatever the truth value of p may be. If all you know is that p is true, or if all you know is that q is false, you cannot determine the truth value of " $p \supset q$ ".

false.⁵ If you already know that q is true, you don't need to infer its truth from anything. And if you already know that q is false, you know that you *cannot* infer its truth from anything.⁶ So while the material conditional satisfies Modus Ponens, the fact that it does is not enough to make reasoning with it useful in practice.

3. The Ugly: Ordinarily, when one asserts a conditional one implies that there is some sort of connection between its antecedent and its consequent. As an example, take the statement "If that ball is red, then that ball has a color". (We shall suppose this statement is uttered in a context where it is clear which ball is meant.) I'd wager that most people would say that the truth of the antecedent guarantees the truth of the consequent; if the ball under discussion is red, it just *has* to be true that it has a color. However, a material conditional can be true when its antecedent and consequent have nothing to do with each other. For instance, from the first row of the truth table it follows that every true statement materially implies every other true statement. Thus, "Electrons are negatively charged" materially implies "Neil Armstrong was the first man to walk on the Moon". Yet no matter how true it may be, "If electrons are negatively charged then Neil Armstrong was the first man to walk on the Moon" would be an odd thing to say, to say the least. Also, it follows from the third and fourth rows of the truth table that a false statement materially implies every true statement as well as every false statement. If we interpret the English 'if...then' construction as a material conditional, the statements "If bananas are vegetables then the sky is blue" and "If George W. Bush is articulate then the Pope is not Catholic"⁷ are both true because their antecedents are both false. The fact that the consequent of the first statement is true and the consequent of the second statement is false makes no difference.

⁵ You can only apply Modus Ponens if you know that p and " $p \supset q$ " are both true, and if you know that p is true you can't know whether " $p \supset q$ " is true without knowing whether q is true.

⁶ If you had a valid argument with q as its conclusion, you would know that it must be unsound.

⁷ In the author's opinion the antecedent of this particular material conditional is pretty clearly false. But however that may be, I'll ask the reader to be kind enough to suppose that it is false for the sake of the ensuing discussion.

For our purposes there's nothing wrong with any of this, but the strangeness of these and similar statements gives intuitive support to the view that ' \supset ' and the English 'if...then' construction do not quite mean the same.⁸

At last we arrive at the final truth table, the one for *material equivalence*, which is expressed by the symbol ' \equiv ':

p	q	$p \equiv q$
T	T	T
T	F	F
F	T	F
F	F	T

According to this truth table, two statements are materially equivalent if they are either both true or both false. If one is true and the other is false, they are not materially equivalent. When two statements p and q are materially equivalent, we can say that p is true *if and only if* q is true.

5. Complex Statements and Complex Schemas

We will now learn how to form and evaluate complex statements and schemas. (From this point on, I will sometimes refer to both statements and schemas as *expressions*.)

First, it will be necessary to say something about parentheses. As in other branches of mathematics, in formal logic parentheses provide an easy way to avoid ambiguity. Take the arithmetical expression ' $5 - 3 + 7$ '. Without relying on an order of operations, you wouldn't know whether to start by adding or subtracting. Parentheses tell you which you should do first.

⁸ In English there is also a different sort of conditional called a *counterfactual conditional*. One example would be the statement "If the American colonies had lost the Revolutionary War the French Revolution would not have happened." I don't pretend to know whether this conditional is true or false, but it may be either, and even if it is true its truth is clearly not entailed by the fact that its antecedent is false, unlike a material conditional.

The above expression is ambiguous between ' $(5 - 3) + 7$ ' and ' $5 - (3 + 7)$ '. Parentheses can make all the difference between 9 and -5!

There are several rules about how to use parentheses. They help determine what counts as a proper logical expression, much as the rules of English grammar determine what counts as a proper English sentence. Since the same rules apply statements and schemas, I will only mention schemas in what follows.

The first rule is that parentheses come in pairs. To every left parenthesis there must correspond a right parenthesis, and conversely. This may seem obvious enough, but I've stated it anyway for the sake of completeness. So " $(p \ \& \ q)$ " is improper, as are " $(r \equiv s)$ " and " $((p \supset \sim t))$ ". The second rule specifies that we may only put parentheses around schemas, not schematic letters. Thus " $\sim(r \ \& \ s)$ " is proper while " $\sim((r) \ \& \ (s))$ " is not. The third rule requires only one set of parentheses for any given component of a schema, so the outer set of parentheses in " $((p \ \& \ q)) \vee r$ " is unnecessary and the schema is improper. The fourth rule demands that there must be parentheses around any schema formed when two schematic letters or schemas are joined by a truth function. Thus " $((p \supset q) \equiv w)$ " and " $(r \ \& \ (s \ \& \ t))$ " are proper, while " $(p \supset q \equiv w)$ " and " $(r \ \& \ s \ \& \ t)$ " are not. The would-be schema " $(p \supset q \equiv w)$ " is ambiguous between two readings, " $((p \supset q) \equiv w)$ " and " $(p \supset (q \equiv w))$ ", and those readings may or may not have the same truth value.⁹

⁹The following truth table verifies this:

p	q	w	$p \supset q$	$(q \equiv w)$	$((p \supset q) \equiv w)$	$(p \supset (q \equiv w))$
T	T	T	T	T	T	T
T	T	F	T	F	F	F
T	F	T	F	F	F	F
T	F	F	F	T	T	T
F	T	T	T	T	T	T
F	T	F	T	F	F	T
F	F	T	T	F	T	T
F	F	F	T	T	F	T

By contrast, “(r & s & t)” is not ambiguous, for the schemas “(r & (s & t))” and “((r & s) & t)” will always have the same truth value, as one could check by constructing the appropriate truth tables. In spite of that, we will require the extra parentheses within conjunctions to keep things uniform. We will also require extra parentheses within disjunctions in those cases where the rules require them, although disjunctions are unambiguous in the same way that conjunctions are.

There are three exceptions to the fourth rule. The first is that the outermost parentheses of a schema are optional, so instead of “(p & q)” one may write “p & q” and instead of “(¬(r ≡ t))” one may write “¬(r ≡ t)”. The second is that when forming the negation of a negation (or the negation of a negation of a negation, or the negation of a negation of a negation of a negation...) one need not use any parentheses. The third is that when forming a complex schema of which one or both of the components being joined is a negation, one need not put parentheses around the negation(s). So instead of “p ⊃ (¬q)” and “(¬r) ∨ (¬s)” one may write “p ⊃ ¬q” and “¬r ∨ ¬s”.

Finally, the last rule concerning parentheses states that one may use square brackets or curly brackets wherever one may use parentheses. Doing so can make longer, more complex schemas easier to read. Thus

$$(((p \vee q) \& \sim q) \equiv ((p \supset p) \supset p)) \equiv (q \equiv q)$$

appears daunting, while

$$\{[(p \vee q) \& \sim q] \equiv [(p \supset p) \supset p]\} \equiv (q \equiv q)$$

appears (slightly) less so.

There is one more rule we will observe when putting statements or schemas together, which is that we shall never combine the two. That is, we will never have any hybrid expression like “(A ⊃ q) & (r ∨ (C ≡ D))”. While it is not impossible to find a use for them,¹⁰ they would needlessly complicate things and are not allowed.

¹⁰ Because they contain schematic letters, such hybrid expressions are neither true nor false. We can still assess them for validity, though. For example, take the schema “p ⊃ p”. We shall see below that this always holds, no matter what statement one substitutes for ‘p’. Then consider the following, which I will call a *partial schema*:

The next notion you should be familiar with is that of the *main operator* of an expression. The main operator of an expression is the one that determines the truth value of the whole expression. For those who adhere to the above rules governing the use of parentheses, the main operator is easy enough to identify. If you leave out the outermost parentheses, it is the only operator which is not within any set of parentheses.¹¹ Thus in “ $p \& (q \vee r)$ ” the main operator is ‘ $\&$ ’, in “ $(p \supset q) \vee (q \supset p)$ ” the main operator is ‘ \vee ’, in “ $\sim(q \equiv s)$ ” the main operator is ‘ \sim ’, and so on.

We are now in a position to explain in what sense it is that the statements which atomic statement letters stand in for are simple. Consider the statement “Most dogs are friendly”, which we shall represent by the atomic statement letter ‘ D ’. D is equivalent to “ $\sim\sim D$ ”, as the following truth table shows:

D	$\sim D$	$\sim\sim D$
T	F	T
F	T	F

Negation acts on a negated statement in the same way that it acts on a non-negated one, so “ $\sim\sim D$ ” is true when “ $\sim D$ ” is false and “ $\sim\sim D$ ” is false “ $\sim D$ ” is true. So “ $\sim\sim D$ ” is true if and only if D is. D is also equivalent to “ $D \& D$ ” and “ $D \vee D$ ”. For as we have already seen, a conjunction is true just in case both of its conjuncts are true, and a disjunction is true just in case at least one of its disjuncts is true. If D is true then “both” D and D are true, and hence “ $D \& D$ ” is true too. Furthermore, if D is true then “at least one of” D or D is true, so “ $D \vee D$ ” is true as well. On the

$$\frac{C}{p \supset p}$$

Since “ $p \supset p$ ” is true on any substitution, a single-premise argument from C to any of these substitutions is always valid. So we could say that the above partial schema is valid in much the same sense that schemas are.

¹¹ Unless the only operators occurring in the expression are negation signs, or the expression is a complex of which one or both of the components being joined is a negation. In the first case the main operator is the outermost negation sign, in the second it is the truth function which joins the components.

other hand, if D is false, “both” D and D are false, and so is “D & D”. Also, if D is false, then “neither” D nor D is true, so “D \vee D” is false too.

Since D was chosen arbitrarily, we have shown that *any* statement can be expressed by means of truth functional compounds. This is true even if the original does not contain any term that corresponds to a truth function, as with “Most dogs are friendly”.

Not all statements are simple in the way that “Most dogs are friendly” is. Some *require* a truth function for their expression. One example would be “Cathy is running and George W. Bush is articulate”. Letting ‘B’ stand in for “George W. Bush is articulate”, the example should be symbolized as “C & B”, *not* as any single atomic statement letter. In general, any English statement which contains a term that corresponds to a truth function should be represented by a truth functional compound.

6. Tautologies, Contradictions, and Indeterminacies

The truth tables we have encountered so far have all been fairly simple. But things are about to get more complicated, and also (I hope) more interesting.

I have said above that “If George W. Bush is articulate then the Pope is not Catholic” is true if we read it as a material conditional, its false consequent notwithstanding. This is because any material conditional with a false antecedent is true. Now, what if we need to evaluate a more complex statement that has the above conditional as a constituent? First, we must express the conditional formally. We have already symbolized “George W. Bush is articulate” as ‘B’, but what about “The Pope is not Catholic”? Because it is a truth functional compound, it cannot be represented by an atomic statement letter. To symbolize it we must separate out the component which corresponds to negation, which in this case is the word ‘not’, and rewrite the statement as “It is not the case that the Pope is Catholic”. Letting ‘P’ stand in for “The Pope is Catholic”, we can symbolize “It is not the case that the Pope is Catholic” as “ \sim P”. The entire conditional is then symbolized as “B \supset \sim P”.

Given the preceding, what should we make of this more complex statement,

$$(B \supset \sim P) \equiv (\sim B \vee \sim P)$$

?

Since we already know the truth values of B and P, we don't need to list all the ways that truth values can be assigned to them. We can thus begin by constructing a *partial truth table* for this statement, a truth table with only one row:

B	P	$\sim B$	$\sim P$	$B \supset \sim P$	$\sim B \vee \sim P$	$(B \supset \sim P) \equiv (\sim B \vee \sim P)$
F	T	T	F	T	T	T

Because B and “ $\sim P$ ” are both false, “ $\sim B$ ” and P are both true. Since B is false, “ $B \supset \sim P$ ” must be true. And since at least one of “ $\sim B$ ” and “ $\sim P$ ” is true, namely “ $\sim B$ ”, “ $\sim B \vee \sim P$ ” is true as well. Furthermore, as “ $B \supset \sim P$ ” and “ $\sim B \vee \sim P$ ” both have the same truth value, the material equivalence “ $(B \supset \sim P) \equiv (\sim B \vee \sim P)$ ” is also true.

The above truth table represents only one case. Let us see what happens to the truth value of “ $(B \supset \sim P) \equiv (\sim B \vee \sim P)$ ” as we momentarily ignore what we know about B and P and allow their truth values to vary. We must now construct a complete truth table:

B	P	$\sim B$	$\sim P$	$B \supset \sim P$	$\sim B \vee \sim P$	$(B \supset \sim P) \equiv (\sim B \vee \sim P)$
T	T	F	F	F	F	T
T	F	F	T	T	T	T
F	T	T	F	T	T	T
F	F	T	T	T	T	T

You can see that something interesting is going on here. The last column has only ‘T’s in it. This is due to the fact that $(B \supset \sim P) \equiv (\sim B \vee \sim P)$ is a *tautology*. It is true no matter what truth values are assigned to the atomic statements of which it is ultimately a truth function. Tautologies are *logically true* because their truth is determined by their form. They hold “come what may,” as the philosopher W. V. O. Quine put it in his article “Two Dogmas of Empiricism”¹². Things could have turned out differently in many different ways; dinosaurs might never have become extinct, Immanuel Kant could have spent his entire career in a dogmatic slumber, and George W. Bush could have been ever so slightly more eloquent, but no matter how different the world could have been every tautology would still have been true. The fact that the above statement is a tautology has nothing to do with the meaning assigned to ‘B’ or ‘P’, for it is but one instance of the schema $(r \supset \sim s) \equiv (\sim r \vee \sim s)$, and any instance of that schema is a tautology. Also, this tautology is an example of a *logical equivalence*, which is a material equivalence that happens to be logically true.¹³ Because all tautologies are true as a matter of logic, they are all logically equivalent to each other.

In addition to tautologous statements, *schemas* can also be called tautologies in the sense that every statement which is an instance of a tautologous schema is itself a tautology.

While every tautology is logically equivalent to every other tautology, some are more famous than others. The “big three” are the so-called “Laws of Logic”, “ $p \supset p$ ”, “ $\sim(p \ \& \ \sim p)$ ” and “ $p \vee \sim p$ ”. The first is referred to as the *Law of Identity*, the second as the *Law of Non-Contradiction*, and the last as the *Law of Excluded Middle*. Basically, the first says that if a given statement is true then that very statement is true, the second that no statement is both true and not true, and the last that every statement is either true or not true. This truth table confirms that these Laws have the distinction of being tautologies:

¹² For full reference, see Bibliography.

¹³ On this way of speaking every logical equivalence is a material equivalence, but the converse is not true. Some may reserve the term ‘material equivalence’ for an equivalence which is *not* a logical equivalence. I don’t think either usage is “more correct” than the other, but I do think mine is reasonable in light of the fact that two statements cannot be logically equivalent without being either both true or both false, and according to the truth table for material equivalence that does make them materially equivalent. I grant that the other usage is reasonable too, for if one adopts it it will always be clear whether one means that two statements are logically equivalent or *merely* materially equivalent when one calls them materially equivalent. Nevertheless, I think that on my usage it will always be clear from the context which is meant.

p	$\sim p$	$p \& \sim p$	$p \supset p$	$\sim(p \& \sim p)$	$p \vee \sim p$
T	F	F	T	T	T
F	T	F	T	T	T

The opposite of a tautology is a *contradiction*. A contradiction is false no matter what truth values are assigned to the atomic statements of which it is ultimately a truth function, and hence it is not just false but *logically false*. As with tautologies, *schemas* can also be called contradictions in the sense that every statement which is an instance of a contradictory schema is itself a contradiction. It is easy enough to find examples of contradictions; one can simply take a tautology and prefix a negation operator to it. So “ $\sim(p \supset p)$ ”, “ $\sim\sim(p \& \sim p)$ ” (equivalently, “ $p \& \sim p$ ”) and “ $\sim(p \vee \sim p)$ ” are all contradictions. Conversely, if one wants further examples of tautologies, one can simply negate a contradiction. And like tautologies, all contradictions are logically equivalent to each other.

A statement which is neither a tautology nor a contradiction is *logically indeterminate*. Unlike tautologies and contradictions, such a statement can be either true or false because its truth value is not determined by its form. A truth table for a logically indeterminate statement will always contain at least one ‘T’ and one ‘F’ in the column for the truth function which is its main operator. Some pairs of logically indeterminate statements are logically equivalent and some are not. We have seen that “ $B \supset \sim P$ ” and “ $\sim B \vee \sim P$ ” are logically equivalent although neither is logically true or logically false. By contrast, the statements “ $C \& D$ ” and “ $C \equiv D$ ” are both logically indeterminate and yet they are not logically equivalent. Also, no logically indeterminate statement is logically equivalent to a tautology or a contradiction, otherwise it would *be* one or the other, yet because bivalence holds it must be *materially* equivalent either to every tautology or every contradiction.

A word of warning: One cannot speak of an indeterminate schema as one can speak of a tautologous or contradictory schema. For example, the one-letter schema p doesn’t appear

contradictory, but any statement you please can be substituted for it, including the blatant contradiction “ $B \ \& \ \sim B$ ”. Another example is the safe-looking conjunction “ $r \ \& \ s$ ”, for which one could substitute “ $\sim(C \supset C) \ \& \ \sim(D \vee \sim D)$ ”, which denies an instance of the first and an instance of the third Laws of Logic. So even if a schema is not contradictory, it can still have contradictory statements as instances. In the same way, a non-tautologous schema can have tautologies as instances. For these reasons we will not say that schemas are logically indeterminate.

7. *Assessing Argument Schemas*

With all the pieces in place, we are finally in a position to learn how to assess argument schemas for validity. The basic procedure is simple enough. To assess an argument schema you must first identify its premises and conclusion. Then you must conjoin the premises to get one big conjunction. With that done, the next step is to form a material conditional with the conjunction as its antecedent and the conclusion as its consequent. Lastly, you have to make a truth table for this conditional and determine whether or not it is a tautology. If it is the argument schema with which you started is valid, and if it’s not the schema is invalid.

There are two ways to construct a truth table for the conditional, one longer and one shorter. The long way is to make a complete truth table, and see if the truth value of the conditional varies as its schematic letters are assigned different truth values. The short way is to make a partial truth table, put an ‘F’ under the column with ‘ \supset ’ as its main operator, and see if you can consistently assign truth values to the other constituents of the schema.

First, an illustration of the long way. We already know that Modus Ponens is a valid schema, but now we can show this more rigorously. Using ‘ \supset ’ as our conditional, the schema is:

$p \supset q$

p _____

q

We begin by conjoining the premises to get “ $(p \supset q) \& p$ ”. Then we take the conclusion, q , and form the conditional “ $[(p \supset q) \& p] \supset q$ ” Next we formulate the appropriate truth table:

p	q	$p \supset q$	$(p \supset q) \& p$	$[(p \supset q) \& p] \supset q$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

You can see that the schema is a tautology and hence that Modus Ponens is a valid argument schema.

Now for an illustration of the short way. The first step is to form the following, partial truth table:

p	q	$p \supset q$	$(p \supset q) \& p$	$[(p \supset q) \& p] \supset q$

Next, we put an ‘F’ under the last (i.e., rightmost) column:

p	q	$p \supset q$	$(p \supset q) \& p$	$[(p \supset q) \& p] \supset q$
				F

Since a material conditional is false only if its antecedent is true and its conclusion is false, we put a ‘T’ under the “ $(p \supset q) \& p$ ” column and an ‘F’ under the ‘q’ column:

p	q	$p \supset q$	$(p \supset q) \& p$	$[(p \supset q) \& p] \supset q$
	F		T	F

Because “ $(p \supset q) \& p$ ” is true, we know that “ $p \supset q$ ” and p are both true:

p	q	$p \supset q$	$(p \supset q) \& p$	$[(p \supset q) \& p] \supset q$
T	F	T	T	F

But wait a moment! There is now a ‘T’ in the ‘p’ column and an ‘F’ in the ‘q’ column, which should mean that “ $p \supset q$ ” is false, and yet there is a ‘T’ under it. If we add the required ‘F’ anyway, we get:

p	q	$p \supset q$	$(p \supset q) \& p$	$[(p \supset q) \& p] \supset q$
T	F	T, F	T	F

Our truth table is now patently contradictory, for it represents “ $p \supset q$ ” as being both true and false. We don’t have to proceed any further. “ $[(p \supset q) \& p] \supset q$ ” cannot be false. It is a tautology, and the argument schema from which we derived it must be valid.

The attentive reader may object that this “short” way looks a lot longer than the “long” way on paper. While it certainly *looks* longer, I think you’ll find that once you master it it will take a lot less time in practice.

To make sure that you understand how to evaluate argument schemas, we will go over a couple more examples. Consider this schema, which is known as *Disjunctive Syllogism*:

$p \vee r$

$\sim r$

p

Again we start with the long way. We conjoin “ $p \vee r$ ” and “ $\sim r$ ” to obtain “ $(p \vee r) \& \sim r$ ”. Then we take ‘ p ’ and form the conditional “ $[(p \vee r) \& \sim r] \supset p$ ”. Finally we assess it by constructing a complete truth table:

p	r	$\sim r$	$p \vee r$	$(p \vee r) \& \sim r$	$[(p \vee r) \& \sim r] \supset p$
T	T	F	T	F	T
T	F	T	T	T	T
F	T	F	T	F	T
F	F	T	F	F	T

If we use the short way instead, we end up with the following partial truth table:

p	r	$\sim r$	$p \vee r$	$(p \vee r) \& \sim r$	$[(p \vee r) \& \sim r] \supset p$
F	F	T	T, F	T	F

The reason we get this result is that if “ $[(p \vee r) \& \sim r] \supset p$ ” is false “ $(p \vee r) \& \sim r$ ” must be true and p must be false. Since “ $(p \vee r) \& \sim r$ ” is true, both “ $p \vee r$ ” and “ $\sim r$ ” must be true too. And because “ $\sim r$ ” is true r must be false. But if p and r are both false then so is “ $p \vee r$ ”, for it requires at least one of them to be true. Yet we have already said that “ $p \vee r$ ” is true. Again we have a

contradiction. We have to conclude that “ $[(p \vee r) \& \sim r] \supset p$ ” is a tautology and that Disjunctive Syllogism is a valid argument schema.

We have now come to the final example. Take this argument schema, which is called *Hypothetical Syllogism*:

$$p \supset q$$

$$\underline{q \supset r}$$

$$p \supset r$$

Conjoining the premises gives us “ $(p \supset q) \& (q \supset r)$ ”. Taking the conclusion “ $p \supset r$ ” and forming the required conditional gives us “ $[(p \supset q) \& (q \supset r)] \supset (p \supset r)$ ”. The complete truth table for this conditional is:

p	q	r	$p \supset q$	$q \supset r$	$p \supset r$	$(p \supset q) \& (q \supset r)$	$[(p \supset q) \& (q \supset r)] \supset (p \supset r)$
T	T	T	T	T	T	T	T
T	T	F	T	F	F	F	T
T	F	T	F	T	T	F	T
T	F	F	F	T	F	F	T
F	T	T	T	T	T	T	T
F	T	F	T	F	T	F	T
F	F	T	T	T	T	T	T
F	F	F	T	T	T	T	T

A quick comparison of the above monstrosity of a truth table with the following partial truth table might give you some appreciation for the short way of doing things:

p	q	r	$p \supset q$	$q \supset r$	$p \supset r$	$(p \supset q) \& (q \supset r)$	$[(p \supset q) \& (q \supset r)] \supset (p \supset r)$
T	T	F	T	T, F	F	T	F

You can arrive at this partial truth table via the following line of reasoning. First, if the schema “ $[(p \supset q) \& (q \supset r)] \supset (p \supset r)$ ” is false then “ $(p \supset q) \& (q \supset r)$ ” is true and “ $p \supset r$ ” is false. Since “ $(p \supset q) \& (q \supset r)$ ” is true, “ $p \supset q$ ” and “ $q \supset r$ ” are also true, for they are its conjuncts. Now, from the fact that “ $p \supset r$ ” is false it follows that p is true and r is false for the now familiar reason that “ $p \supset r$ ” is a material conditional. Yet if “ $p \supset q$ ” and p are both true then q is true too, by Modus Ponens. But if q is true and r is false then “ $q \supset r$ ” is also false. In spite of that, we have already concluded that “ $q \supset r$ ” is true. Once more we have reasoned ourselves into a contradiction. Thus “ $[(p \supset q) \& (q \supset r)] \supset (p \supset r)$ ” must be a tautology, and Hypothetical Syllogism must be a valid argument schema.

8. Conclusion

You have just learned how to tell whether an argument schema is valid when its validity depends on nothing more than the way its constituent schemas are combined truth-functionally. Any particular argument which is an instance of a valid schema is also valid. In Part 3 we’ll see that the validity of an argument can also depend on the logical structure of the statements that compose it, and not just the way those statements are put together.

Bibliography

Gödel, Kurt. “Über formal unentscheidbare Sätze der *Principia Mathematica* und verwandter Systeme, I”, [“On Formally Undecidable Propositions of *Principia Mathematica* and Related Systems”]. Published in *Monatshefte für Mathematik und Physik*, 38: 173–198, 1931.

Quine, Willard van Orman. “Two Dogmas of Empiricism”, *Philosophical Review*, 60: 20–43, 1951.