Bentley College
Fall semester, 2004
PH 101: Problems of Philosophy
Instructor: Miles Rind
November 22, 2004

NOTES ON LOGIC, PART III: TRUTH TABLES

§ 1. Truth values. In logic, the mathematical notion of a *value* is applied to statements: a statement is said to have a *truth value*, which consists either in its being true or in its being false. These two truth values are represented by the letters "T" and "F," respectively.

§ 2. Logical connectives. A *logical connective* is a propositional element that, when combined with a statement or statements in a particular fashion, yields a new statement whose truth value is derived according to a fixed rule from the truth value or truth values of the statement or statements from which it is formed. The principal logical connectives are the words "not," "and," "or," and the constructions "if . . . then" and "if and only if."

§ 3. Simple and complex statements. Any statement that contains a logical connective is called a *complex statement*. A statement that does not contain a logical connective is called a *simple statement*. There are five kinds of complex statement, each named according to the logical connective that is used in forming it: *negations* (formed with "not"), *conjunctions* (formed with "and"), *disjunctions* (formed with "or"), *conditionals* (formed with "if . . . then"), and *biconditionals* (formed with "if and only if").

§ 4. Truth conditions and truth tables. As was said above, the truth value of any complex statement is derived by a fixed rule from the truth values of its component statements. To explain the possible truth values of a complex statement in terms of the possible truth values of its component statements is to state what are called the *truth conditions* of the complex statement. The standard way of specifying truth conditions is in the form of a table called a *truth table*. I will explain, first, how truth tables are formed for each of the basic kinds of complex statement, then how truth tables may be used to prove the validity and invalidity of certain argument forms.

§ 5. Negations. A *negative statement*, or *negation*, is a complex statement formed by inserting the word "not" after the main verb of a statement, or by prefixing the expression "It is not the case that" to the statement as a whole. For example, the negation of "My dog has fleas" is "My dog does not have fleas" or, equivalently, "It is not the case that my dog has fleas." The rule for assigning a truth value to a negation is that it is the opposite of the truth value of the negated statement: T if the negated statement is false, and F if the negated statement is true. In other words, if "P" is true then "Not-P" is false, and vice versa. We can express this by the following table (table 1):

P	Not-P
T	F
F	T

The table is to be read as follows: The first column lists the possible truth values of "P"; the second column lists the corresponding possible truth values of "Not P," row by row.

§ 6. Conjunctions. A *conjunctive statement*, or *conjunction*, is a statement formed by combining two statements with the word "and." The statements so combined are called *conjuncts*. For example, "The roof leaks and the floor creaks" is a conjunction of "The roof leaks" and "The floor creaks," which are its conjuncts. The rule for assigning a truth value to a conjunction is that the conjunction is true if both conjuncts are true, and otherwise is false. Since a conjunction is formed of two conjuncts, each of which can have either of two truth values, we have to consider four (that is, 2 × 2) possible combinations of truth values in the conjuncts. Now we know that "P and Q" is true when both "P" and "Q" are true, and false otherwise . So our truth table for "P and Q" will look like this (table 2):

P	Q	P and Q
T	T	T
T	F	F
F	T	F
F	F	F

The way to read the table is to look at each row for a possible pair of truth values of the two conjuncts in the first two columns, then pass to the third column to find the truth value of the conjunction "P and Q" for that combination. The table says that, if "P" is true and "Q" is true, then "P and Q" is true; that if "P" is true and "Q" is false, then "P and Q" is false; and so on for the remaining rows.

§ 7. Disjunctions. A *disjunctive statement*, or *disjunction*, is a statement formed by combining two statements with the word "or." The statements so combined are called *disjuncts*. For example, "That's a screech owl or I don't know owls" is the disjunction of "That's a screech owl" and "I don't know owls," which are its component disjuncts. Now in ordinary speech, statements of the form "P or Q" are usually understood to mean "Either P or Q, but not both." In such cases, the "or" is said to be interpreted *exclusively*. In logic, however, "P or Q" is understood to mean "Either P or Q or both," or "P and / or Q"; that is, the "or" is interpreted *inclusively*. The rule for assigning a truth value to a disjunction is therefore that if at least one of the disjuncts is true, then the disjunction is true; otherwise (namely if both disjunctions are false), the disjunction is false. A truth table for the disjunction "P or Q" therefore looks like this (table 3):

P	Q	P or Q
T	T	T
T	F	T
F	T	T
F	F	F

§ 8. Conditionals. A *conditional statement*, or *conditional*, is a statement formed by combining two statements with the word "if" in front of the first of them, and possibly also the word "then" in front of the second of them. The statement in a conditional that is marked by the "if" is called the *antecedent*; the other is called the *consequent*. For example, "If it's noon, we need to leave" is a conditional in which "It's noon" is the antecedent and "We need to leave" is the consequent.

Assigning truth values to conditional statements faces the difficulty that we often do not think of conditional statements as being true or false. For example, suppose that it is not noon. Is the statement "If it's noon, we need to leave" then true, or is it false? It seems impossible to say (though certainly the conditional statement is not *false* in such a case). The only clear circumstance in which a conditional statement has a truth value is one in which the antecedent is true and the conclusion false: in that case, it is clear that the conditional statement is false. (If it's noon and we do not need to leave, then the statement "If it's noon, we need to leave" is false.) The way we deal with this in logic is to treat all cases in which a conditional is not false as cases in which it is true. The rule for assigning a truth value to a conditional statement is therefore that if the antecedent is true and the consequent false, then the conditional is false; otherwise, it is true. The truth table for a conditional therefore looks like this (table 4):

P	Q	If P then Q
T	T	T
T	F	F
F	T	T
F	F	T

§ 9. Biconditionals. A *biconditional statement,* or *biconditional* is a statements formed by combining two statements with the expression "if and only if" (commonly abbreviated as "iff"). As was explained in part II of these notes (§ 15), a biconditional is the conjunction of two conditional statements that are logical converses of each other, meaning that the statement "P if and only if Q" is an abbreviated way of writing "If P then Q, and if Q then P." To assert a biconditional statement is to assert the equivalence of the two statements combined in it with regard to their truth values. The rule for assigning truth values to a biconditional statement is therefore that the statement is true if both component statements are true or if both component statements are false, and that it is otherwise false (namely if one component statement is true and one is false). So the truth table for a biconditional looks like this (table 5):

P	Q	P ⇔ Q
T	T	T
T	F	F
F	T	F
F	F	T

§ 10. Truth tables and logical equivalence. So far, we have considered truth tables merely as a way of specifying how the logical connectives are to be understood. A more interesting use of them is the use of them to prove *logical equivalences* between complex statements; that is, to prove that any two complex statements having certain logical forms will always have the same truth values, no matter what the truth values of the component statements.

For example, consider the statement "Either the apartment is not taken or the rent is affordable." This has the logical form "Not-P or Q." We construct a truth table for that statement as follows. First, we give the possible combinations of truth values of "P" and "Q" in two columns, as usual. We then add a column with the corresponding values of "Not-P," which will simply be the opposites of the truth values in the "P" column. Finally, we complete a column for "Not-P or Q" by applying the rule for disjunction to the columns for "Not-P" and "Q": in any row in which at least one of those two columns has a "T" in it, the disjunction will be true; otherwise (namely, if both columns have "F" in them), it will be false. The truth table for a statement of this form will therefore look like this (table 6):

P	Q	Not-P	Not-P or Q
T	T	F	T
T	F	F	F
F	T	T	T
F	F	T	T

Now a remarkable thing about the last column of table 6 is that each of its values is correlated with the values of the "P" and "Q" columns in exactly the same way as the last column of the truth table for "If P then Q" (table 4). This can be see by adding to the table a column for "If P then Q," thus (table 7):

P	Q	Not-P	Not-P or Q	If P then Q
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

This table shows that "Not-P or Q" is *logically equivalent* to "If P then Q": for any two statements "P" and "Q," the complex statements "Not-P or Q" and "If P then Q" will necessarily have the same truth values, merely by virtue of their logical form.

§ 11. Truth tables and valid argument forms. The most interesting use of truth tables is in proving the validity or invalidity of argument forms. Consider, for example, the argument form *modus tollens* or denying the consequent (part II, § 9):

(1) If P then Q
(2) Not-Q
(3) Not-P

According to the definition of validity, an argument is valid just in case it is impossible for the premises to be true while the conclusion is false. So what we need to know about this argument form is whether there can be an instance in which the premises are all true and the conclusion false. If there is, then the argument is valid; if not, then it is invalid. Since this argument form is specified in terms of the construction of its statements out of logical connectives and component statements, we can test its validity by constructing a truth table for the three statements of this argument. It will look like this (table 8):

P	Q	If P then Q (1)	Not-Q (2)	Not-P (3)
T	T	T	F	F
T	F	F	T	F
F	T	T	F	T
F	F	**T**	**T**	**T**

The last three columns correspond, in order, to the three steps of the argument. I have added bold type to the last row, as that is the only row in which all (that is, both) of the premises of the argument are true. The table shows that, when all the premises of the argument are true, the conclusion is also true. Thus the table shows that, in *modus tollens*, when both premises are true, the conclusion is also true. It therefore proves that the argument form is valid.

§ 12. Proving invalidity. By contrast, consider the argument form called "denying the antecedent" (part II, § 14):

(1) If P then Q
(2) Not-P
(3) Not-Q

A truth table for this argument form will look like this (table 9):

P	**Q**	**If P then Q (1)**	**Not-P (2)**	**Not-Q (3)**
T	T	T	F	F
T	F	F	F	T
F	T	**T**	**T**	**F**
F	F	**T**	**T**	**T**

Here, again, the last three columns correspond to the three steps of the argument. In this case, however, there are *two* rows in which both premises are true, the third and the fourth (indicated again by bold type). In the fourth row, the conclusion is true too; but in the third row, the conclusion is *false*. This shows that there is a case in which the premises of the argument are all true and the conclusion false. Thus the table shows that the argument form is *invalid*.