# I

# Basics of Logic

**I.1. INTRODUCTION.** Deductive logic is the logic of "and," "or," and "not." It is useful for classifying, sorting, or searching and can be used for searching a library, using an Internet search engine, or searching any sort of database. The help section for one Internet search engine explains that searching for "Mary AND NOT lamb" finds documents containing "Mary" but not containing "lamb." A library database can also be searched for "Aztec OR Toltec" for a history report. The logic of "and," "or," and "not" gives us a taste of deductive logic, with which we can compare inductive logic. Deductive logic will also be useful in the analysis of Mill's methods of experimental inquiry in Chapter V, and in the treatment of probability in Chapter VI.

**I.2. THE STRUCTURE OF SIMPLE STATEMENTS.** A *statement* is a sentence that makes a definite claim. A straightforward way of making a claim is to (1) identify what you are talking about, and (2) make a claim about it. Thus, in the simple statement "Socrates is bald," the proper name "Socrates" identifies who we are talking about and the predicate "is bald" makes our claim about him. In general, expressions that identify what we are talking about are called *referring expressions* and the expressions used to make factual claims about the things we are talking about are called *characterizing expressions*. Thus, the name "Socrates" refers to a certain individual, and the predicate "is bald" characterizes that individual.

Although proper names are an important type of referring expression, there are others. Pronouns such as "I," "you," "he," and "it" are referring expressions often used in ordinary speech, where context is relied upon to make clear what is being talked about. Sometimes whole phrases are used as referring expressions. In the statement "The first President of the United States had wooden false teeth," the phrase "The first President of the United States" is used to refer to George Washington. He is then characterized as having wooden false teeth (as in fact he did).

Although statements are often constructed out of one referring expression, as in the examples above, sometimes they are constructed out of more than one referring expression, plus an expression that characterizes the relationship between the things referred to. For instance, the statement "Mercury is hotter than Pluto" contains two referring expressions—"Mercury" and "Pluto"—and one characterizing expression—"is hotter than." Characterizing expressions that characterize an individual thing are called *property expressions* or *one-place predicates*. "Is bald," "is red," and "conducts electricity" are examples of property expressions. Characterizing expressions that characterize two or more individual things in relation to one another are called *relational expressions* or *many-place predicates*. "Is hotter than," "is a brother of," "is to the north of," and "is between" are examples of relational expressions.

The basic way to construct a simple statement is to combine referring and characterizing expressions to make the appropriate factual claim. In the next section it will be seen how these simple statements can be combined with logical connectives to form complex statements.

**Exercises**

Pick out the referring and characterizing expressions in the following statements. State whether each characterizing expression is a property expression or a relational expression.

1. Tony loves Cleo.
2. Dennis is tall.
3. This book is confusing.
4. Arizona is between New Mexico and California.
5. Los Angeles is bigger than Chicago.

**I.3. THE STRUCTURE OF COMPLEX STATEMENTS.** Consider the two simple statements "Socrates is bald" and "Socrates is wise." Each of these statements is composed of one referring expression and one characterizing expression. From these statements, together with the words "not," "and," and "or," we can construct a variety of complex statements:

Socrates is *not* bald.

Socrates is bald *and* Socrates is wise.

Socrates is bald *or* Socrates is wise.

Socrates is *not* bald *or* Socrates is wise.

Socrates is bald *and* Socrates is wise *or* Socrates is *not* bald *and* Socrates is *not* wise.

The words "not," "and," and "or" are neither referring nor characterizing expressions. They are called *logical connectives* and are used together with referring and characterizing expressions to make complex factual claims.

We can see how the logical connectives are used in the making of complex factual claims by investigating how the truth or falsity of a complex statement depends on the truth or falsity of its simple constituent statements. A simple statement is true just when its characterizing expression correctly



characterizes the thing or things it refers to. For instance, the statement "Socrates is bald" is true if and only if Socrates is in fact bald; otherwise it is false. Whether a complex statement is true or not depends on the truth or falsity of its simple constituent statements *and* the way that they are put together with the logical connectives. Let us see how this process works for each of the connectives.

*Not*. We *deny* or *negate* a simple statement by placing the word "not" at the appropriate place within it. For instance, the denial or negation of the simple statement "Socrates is bald" is the complex statement "Socrates is not bald." Often we abbreviate a statement by using a single letter; for example, we may let the letter "*s*" to stand for "Socrates is bald." We may deny a statement by placing a sign for negation, "~," in front of the letter that abbreviates that statement. Thus, "~*s*" stands for "Socrates is not bald." Now it is obvious that when a statement is true its denial is false, and when a statement is false its denial is true. Using the shorthand introduced above, we can symbolize this information in the following *truth table*, where T stands for true and F for false:

| $p$ | $\sim p$ |
|---|---|
| T | F |
| F | T |

What this table tells us is that if the statement "*p*" is true, then its denial, "~*p*," is false. If the statement "*p*" is false, then its denial, "~*p*," is true. The truth table is a summary of the way in which the truth or falsity of the complex statement depends on the truth or falsity of its constituent statements.

*And*. We form the *conjunction* of two statements by putting the word "and" between them. Each of the original statements is then called a *conjunct*. A conjunction is true just when both of the conjuncts are true. Using the symbol "&" to abbreviate the word "and" we can represent this in the following truth table:

| $p$ | $q$ | $p\&q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

Here we have four possible combinations of truth and falsity that the constituent statements "*p*" and "*q*" might have, and corresponding to each combination we have an entry telling us whether the complex statement "*p*&*q*" is true or false for that combination. Thus, in the case where "*p*" is true and "*q*" is true, "p&q" is also true. Where "p" is true and "q" is false, "p&q" is false. Where "p" is false and "q" is true, "p&q" is again false. And where both "p" and "q" are false, "p&q" remains false.

*Or*. The word "or" has two distinct uses in English. Sometimes "p or q" means "either p or q, *but not both*," as in "I will go to the movies or I will stay home and study." This is called the *exclusive* sense of "or." Sometimes "p or q" means "p or q *or both*," as in "Club members or their spouses may attend." This is called the *inclusive* sense of "or." We are especially interested in the inclusive sense of "or," which we shall represent by the symbol "v." "pvq" is called a *disjunction* (or alternation), with "p" and "q" being the *disjuncts*. The truth table for disjunction is:

| *p* | *q* | *pvq* |
|---|---|---|
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

By reference to the truth tables for "~," "&," and "v" we can construct a truth table for any complex statement. Consider the complex statement "Socrates is not bald or Socrates is wise." This complex statement contains two simple constituent statements: "Socrates is bald" and "Socrates is wise." We may abbreviate the first statement as "s" and the second as "w." We can then symbolize the complex statement as "~svw." We may use the following procedure to construct a truth table for this complex statement:

> Step 1: List all the possible combinations of truth and falsity for the simple constituent statements, "s," "w."
>
> Step 2: For each of these combinations, find whether "~s" is true or false from the truth table for negation.
>
> Step 3: For each of the combinations, find whether "~svw" is true or false from step 2 and the truth table for disjunction.

The result is the following truth table for "~svw":

| | Step 1 | | Step 2 | Step 3 |
|---|---|---|---|---|
| | *s* | *w* | *~s* | *~svw* |
| Case 1: | T | T | F | T |
| Case 2: | T | F | F | F |
| Case 3: | F | T | T | T |
| Case 4: | F | F | T | T |



This truth table tells us exactly what factual claim the complex statement makes, for it shows us in which cases that statement is true and in which it is false.

Because a truth table tells us what factual claim is made by a complex statement, it can tell us when two statements make the same factual claim. Let us examine the truth table for "(s&w)v(~s&w)":

| | *s* | *w* | *~s* | *s&w* | *~s&w* | *(s&w)v(~s&w)* |
|---|---|---|---|---|---|---|
| Case 1: | T | T | F | T | F | T |
| Case 2: | T | F | F | F | F | F |
| Case 3: | F | T | T | F | T | T |
| Case 4: | F | F | T | F | F | F |

Note that in reading across the truth table we start with the simple constituent statements, proceed to the next largest complex statements, until we finally arrive at the complex statement that is the goal. The truth table shows that the final complex statement is true in cases 1 and 3 and false in cases 2 and 4. But notice that the simple statement "w" is also true in cases 1 and 3 and false in cases 2 and 4. This shows that the simple statement "w" and the complex statement "(s&w)v(~s&w)" make the same factual claim. To claim that Socrates is either bald and wise or not bald and wise is just a complicated way of claiming that Socrates is wise. When two statements make the same factual claim, they are *logically equivalent*.

Truth tables may also be used to show that two complex statements make conflicting factual claims. For example, the claim made by the statement "~s&~w" obviously conflicts with the claim made by the statement "s&w." Socrates cannot both be bald and wise and be not bald and not wise. This conflict is reflected in a truth table for both statements:

| | *s* | *w* | *~s* | *~w* | *s&w* | *~s&~w* |
|---|---|---|---|---|---|---|
| Case 1: | T | T | F | F | T | F |
| Case 2: | T | F | F | T | F | F |
| Case 3: | F | T | T | F | F | F |
| Case 4: | F | F | T | T | F | T |

The statement "s&w" is true only in case 1, while the statement "~s&~w" is true only in case 4. There is no case in which both statements are true. Thus, the two statements make conflicting factual claims. When two statements make conflicting factual claims, they are *inconsistent* with each other, or *mutually exclusive*.

There are some peculiar complex statements that make no factual claim whatsoever. If we say "Either Socrates is bald or Socrates is not bald" we have really not said anything at all about Socrates. Let us see how this situation is reflected in the truth table for "*sv~s*":

| | *s* | *~s* | *sv~s* |
|---|---|---|---|
| Case 1: | T | F | T |
| Case 2: | F | T | T |

The reason why the statement "*sv~s*" makes no factual claim is that it is true no matter what the facts are. This is illustrated in the truth table by the statement being true in all cases. When a complex statement is true, no matter what the truth values of its constituent statements are, that statement is called a *tautology*.

At the opposite end of the scale from a tautology is the type of statement that makes an impossible claim. For instance, the statement "Socrates is bald and Socrates is not bald" must be false no matter what the state of Socrates' head. This is reflected in the truth table by the statement being false in all cases:

| | *s* | *~s* | *s&~s* |
|---|---|---|---|
| Case 1: | T | F | F |
| Case 2: | F | T | F |

Such a statement is called a *self-contradiction*. Self-contradictions are false no matter what the facts are, in contrast to tautologies, which are true no matter what the facts are. Statements that are neither tautologies nor self-contradictions are called *contingent statements* because whether they are true or not is contingent on what the facts are. A contingent statement is true in some cases and false in others.

The purpose of this section has been to convey an understanding of the basic ideas behind truth tables and the logical connectives. We shall apply these ideas in our discussion of Mill's methods and the theory of probability.

The main points of this section are:

1. Complex statements are constructed from simple statements and the logical connectives "~," "&," and "v."

2. The truth tables for "~," "&," and "v" show how the truth or falsity of complex statements depends on the truth or falsity of their simple constituent statements.



3. With the aid of the truth tables for "~," "&," and "v," a truth table may be constructed for any complex statement.

4. The truth table for a complex statement will have a case for each possible combination of truth or falsity of its simple constituent statements. It will show in each case whether the complex statement is true or false.

5. The factual claim made by a complex statement can be discovered by examining the cases in which it is true and those in which it is false.

6. If two statements are true in exactly the same cases, they make the same factual claim and are said to be logically equivalent.

7. If two statements are such that there is no case in which they are both true, they make conflicting factual claims and are said to be inconsistent with each other, or mutually exclusive.

8. If a statement is true in all cases, it is a tautology; if it is false in all cases, it is a self-contradiction; otherwise it is a contingent statement.

### Exercises

1. Using truth tables, find which of the following pairs of statements are logically equivalent, which are mutually exclusive, and which are neither:
   a. *p*, *~~p*.
   b. *~pv~q*, *p&q*.
   c. *p&~q*, *~(p&q)*.
   d. *~pvq*, *p&~q*.
   e. *(pvp)&q*, *p&(qvq)*.
   f. *~(~pvq)*, *p&~q*.
2. Using truth tables, find which of the following statements are tautologies, which are self-contradictions, and which are contingent statements:
   a. *~~pv~p*.
   b. *pvqvr*.
   c. *(pvp)&~(pvp)*.
   d. *(pv~q)v~(pv~q)*.
   e. *p&q&r*.
   f. *~~(pv~p)*.
   g. *~pvpvq*.

**I.4. SIMPLE AND COMPLEX PROPERTIES.** Just as complex statements can be constructed out of simple ones using the logical connectives, so complex properties (or property expressions) can be constructed out of simple ones using "and," "or," and "not." These complex properties are the categories used in "Boolean search" of databases. For example, from "Persian Gulf country," "Iraq," and "Iran" you can form the complex property "Persian Gulf country AND NOT (Iraq OR Iran)." We will use capital letters to abbreviate property expressions.

We can use a method to examine complex properties which is quite similar to the method of truth tables used to examine complex statements. Whether a complex property is present or absent in a given thing or event depends on whether its constituent simple properties are present or absent, just as the truth or falsity of a complex statement depends on the truth or falsity of its simple constituent statements. When the logical connectives are used to construct complex properties, we can refer to the following presence tables, where "*F*" and "*G*" stand for simple properties and where "P" stands for "present" and "A" for "absent":

| Table I | | Table II | | | Table III | | |
|---|---|---|---|---|---|---|---|
| *F* | ~*F* | *F* | *G* | *F&G* | *F* | *G* | *FvG* |
| P | A | P | P | P | P | P | P |
| A | P | P | A | A | P | A | P |
| | | A | P | A | A | P | P |
| | | A | A | A | A | A | A |

Note that these tables are exactly the same as the truth tables for the logical connectives except that "present" is substituted for "true" and "absent" is substituted for "false." With the aid of these presence tables for the logical connectives, we can construct a presence table for any complex property in exactly the same way as we constructed truth tables for complex statements. The presence table for a complex property will have a case for each possible combination of presence or absence of its simple constituent properties. For each case, it will tell whether the complex property is present or absent. As an illustration, we may construct a presence table for "~*FvG*":

| | *F* | *G* | ~*F* | ~*FvG* |
|---|---|---|---|---|
| Case 1: | P | P | A | P |
| Case 2: | P | A | A | A |
| Case 3: | A | P | P | P |
| Case 4: | A | A | P | P |



There are other parallels between the treatment of complex statements and the treatment of complex properties. Two complex properties are *logically equivalent* if they are present in exactly the same cases; two properties are *mutually exclusive* if there is no case in which they are both present. When a property is present in all cases (such as "*Fv~F*") it is called a *universal property*. A universal property is analogous to a tautology. When a property is absent in all cases, it is called a *null property*. A null property is analogous to a self-contradiction. The properties in which we are most interested in inductive logic are those which are neither universal nor null. These are called *contingent* properties.

**Exercises**

1. Using presence tables, find which of the following pairs of properties are logically equivalent, which are mutually exclusive, and which are neither:
   a. *~FvG, ~~Gv~F.*
   b. *~Fv~G, ~(F&G).*
   c. *~FvG, F&~G.*
   d. *Fv~(F&G), ~(F&G)&F.*
   e. *~F&~G, ~(FvG).*
   f. *~(FvGvH), FvGvH.*
   g. *F&~G, ~(F&G).*
2. Using presence tables, find out which of the following properties are universal, which are null, and which are contingent:
   a. *~FvGvF.*
   b. *(FvF)&~(FvF).*
   c. *~(Fv~F).*
   d. *(Fv~G)&(Gv~F).*
   e. *FvGvH.*
   f. *~(F&~G)v~(Gv~F).*

**I.5. VALIDITY.** We can use the truth tables of section I.3 to investigate whether one statement (the conclusion) follows logically from some others (the premises). If it does, we have a valid argument; otherwise we don't. An argument is *valid* if the conclusion is true in every case in which the premises are all true. The argument:

$p$
therefore, $p\&q$

is not valid because there is a case in which the premise, "*p*," is true and the conclusion, "*p&q*," is false. It is case 2 in the following truth table:

| | *p* | *q* | *p&q* |
|---|---|---|---|
| Case 1: | T | T | T |
| Case 2: | T | F | F |
| Case 3: | F | T | F |
| Case 4: | F | F | F |

But the argument:

*p*
therefore, *pvq*

is valid because every case in which the premise, "*p*," is true (cases 1 and 2 in the following truth table) is a case in which the conclusion, "*pvq*," is true.

| | *p* | *q* | *pvq* |
|---|---|---|---|
| Case 1: | T | T | T |
| Case 2: | T | F | T |
| Case 3: | F | T | T |
| Case 4: | F | F | F |

Here is an example of a valid argument with two premises:

~*p*
*pvq*
therefore, *q*

We can establish its validity by looking at the following truth table:

| | *p* | *q* | ~*p* | *pvq* |
|---|---|---|---|---|
| Case 1: | T | T | F | T |
| Case 2: | T | F | F | T |
| Case 3: | F | T | T | T |
| Case 4: | F | F | T | F |

First we find the cases where both premises, "~*p*" and "*pvq*," are true. Only case 3 qualifies. Then we can check that the conclusion, "*q*," is true in case 3.

Here you have a little taste of deductive logic. In the next chapter we consider a larger picture that includes both inductive and deductive logic.



**Exercises**

Check the following arguments for validity using truth tables:

1. *p*
   therefore, *pvp*
2. *p*
   therefore, *p&p*
3. ~(*pvq*)
   therefore, ~*q*
4. ~(*p&q*)
   therefore, ~*p*
5. ~*p*
   ~*q*
   therefore, ~(*pvq*)
6. ~*p*
   ~*q*
   *pvqvr*
   therefore, *r*