Chapter 4

Statement Logic

In this chapter, we treat sentential or statement logic, logical systems built on sentential connectives like and, or, not, if, and if and only if. Our goals here are threefold:

1. To lay the foundation for a later treatment of full predicate logic.
2. To begin to understand how we might formalize linguistic theory in logical terms.
3. To begin to think about how logic relates to natural language syntax and semantics (if at all!).

4.1 The intuition

The basic idea is that we will define a formal language with a syntax and a semantics. That is, we have a set of rules for how statements can be constructed and then a separate set of rules for how those statements can be interpreted. We then develop—in precise terms—how we might prove various things about sets of those statements.

4.2 Basic Syntax

We start with a finite set of letters: \{p, q, r, s, \ldots\}. These become the infinite set of atomic statements when we add in primes, e.g. \(p', p'', p''', \ldots\). This infinite set can be defined recursively.
CHAPTER 4. STATEMENT LOGIC

Definition 4 (Atomic statements) Recursive definition:

1. Any letter \{p, q, r, s, \ldots\} is an atomic statement.

2. Any atomic statement followed by a prime, e.g. \(p', q'', \ldots\) is an atomic statement.

3. There are no other atomic statements.

The first clause provides for a finite number of atomic statements: 26. The second clause is the recursive one, allowing for an infinite number of atomic statements built on the finite set provided by the first clause. The third clause is the limiting case.

Using the sentential connectives, these can be combined into the set of well-formed formulas: WFFs. We also define WFF recursively.

Definition 5 (WFF) Recursive definition:

1. Any atomic statement is a WFF.

2. Any WFF preceded by \(\neg\) is a WFF.

3. Any two WFFs can be made into another WFF by writing one of these symbols between them, ‘\(\land\)’, ‘\(\lor\)’, ‘\(\rightarrow\)’, or ‘\(\leftrightarrow\)’, and enclosing the result in parentheses.

4. Nothing else is a WFF.

The first clause is the base case. Notice that it already provides for an infinite number of WFFs since there are an infinite number of atomic statements. The second clause is recursive and provides for an infinite number of WFFs directly, e.g. \(\neg p\), \(\neg\neg p\), \(\neg\neg\neg p\), etc. The third clause is also recursive and thus also provides for an infinite number of WFFs. Given two WFFs \(p\) and \(q\), it provides for \((p \land q)\), \((p \lor q)\), \((p \rightarrow q)\), and \((p \leftrightarrow q)\). The fourth clause is the limiting case. The second and third clauses can be combined with each other to produce infinitely large expressions. For example:

\[(4.1)\]  
\[
(p \rightarrow \neg\neg(\neg p \lor (p \lor p)))
\]
\[
((p \rightarrow p) \leftrightarrow (p \land q))
\]
\[
(p \land (q \lor (r \land (s \lor t))))
\]
\[
\ldots
\]
Be careful with this notation. The point of it is precision, so we must be precise in how it is used. For example, parentheses are required for the elements of the third clause and disallowed for the elements of the second clause. Thus, the following are not WFFs: \( \neg(p) \), \((\neg p)\), \(p \lor q\), etc.\(^1\) The upshot is that the definition of a WFF is a primitive ‘syntax’ for our logical language.

This syntax is interesting from a number of perspectives. First, it is extremely simple. Second, it is unambiguous; there is one and only one ‘parse’ for any WFF. Third, it is, in some either really important or really trivial sense, “artificial”. Let’s look at each of these points a little more closely.

### 4.2.1 Simplicity

If we want to compare the structures we have developed with those of natural language, we can see that the structures proposed here are far more limited. The best analogy is that atomic statements are like simple clauses and WFFs are combinations of clauses. Thus, we might see \( p \) as analogous to *Ernie likes logic* and \( q \) as analogous to *Apples grow on trees*. We might then take \( (p \land q) \) as analogous to *Ernie likes logic and apples grow on trees*.

This analogy is fine, but it’s easy to see that the structures allowed by statement logic are quite primitive in comparison with human language.

First, natural language provides for many many mechanisms to build simple sentences. Statement logic only allows the “prime”. Second, while we have five ways to build on atomic statements in logic, natural language allows many more ways to combine sentences.

\(^1\)There are some variations in symbology that you’ll find when you look at other texts. The negation symbol in something like \( \neg a \) can also be written \( \sim a \). Likewise, the ‘and’ symbol in something like \( a \land b \) can also be written \( a \& b \). There are lots of other notational possibilities and parentheses are treated in different ways.
4.2.2 Lack of Ambiguity

On the other hand, this impoverished syntax has a very nice property: it is unambiguous. Recall that sentences in natural language can often be parsed in multiple ways. For example, we saw that a sentence like \textit{Ernie saw the man with the binoculars} has two different meanings associated with two different structures.

This cannot happen with WFFs; structures will always have a single parse. Consider, for example, a WFF like \((p \land (q \land r))\). Here, the fact that \(q\) and \(r\) are grouped together before \(p\) is included is apparent from the parentheses. The other parse would be given by \(((p \land q) \land r)\). An ambiguous structure like \((p \land q \land r)\) is not legal.

4.2.3 Artificiality

Both of these properties limit the use of statement logic in encoding expressions of human language. We will see below, however, that the semantics of statement logic are impoverished as well. It turns out that the syntax is demonstrably just powerful enough to express a particular set of meanings.

4.3 Basic Semantics

The set of truth values or meanings that our sentences can have is very impoverished: \(T\) or \(F\). We often interpret these as ‘true’ and ‘false’ respectively,
but this does not have to be the case. For example, we might interpret them as ‘blue’ and ‘red’, 0 and 1, ‘apples’ and ‘oranges’, etc. An atomic statement like $p$ can exhibit either of these values.

Let’s go through each of the connectives and see how they affect the meaning of the larger WFF. Negation reverses the truth value of its WFF. If $p$ is true, then $\neg p$ is false; if $p$ is false, then $\neg p$ is true.

\[(4.3) \begin{array}{c|c} p & \neg p \\ \hline T & F \\ F & T \end{array}\]

Conjunction, logical ‘and’, combines two WFFs. If both are true, the combination is true. In all other cases, the combination is false.

\[(4.4) \begin{array}{c|c|c} p & q & (p \land q) \\ \hline T & T & T \\ T & F & F \\ F & T & F \\ F & F & F \end{array}\]

Disjunction, logical ‘or’, combines two WFFS. If both are false, the combination is false. In all other cases, the combination is true.

\[(4.5) \begin{array}{c|c|c} p & q & (p \lor q) \\ \hline T & T & T \\ T & F & T \\ F & T & T \\ F & F & F \end{array}\]

The conditional, or logical ‘if’, is false just in case the left side is true and the right side is false. In all other cases, it is true.
Finally, the biconditional, or ‘if and only if’, is true when both sides are true or when both sides are false. If the values of the conjuncts do not agree, the biconditional is false.

\[
\begin{array}{c|c|c|c}
  p & q & (p \rightarrow q) \\
  \hline
  T & T & T \\
  T & F & F \\
  F & T & T \\
  F & F & T \\
\end{array}
\]

These truth values should be seen as a primitive semantics. However, as with the syntax, the semantics differs from natural language semantics in a number of ways. For example, the system is obviously a lot simpler than natural language. Second, it is deterministic. If you know the meanings—truth values—of the parts, then you know the meaning—truth value—of the whole.

Let’s do some examples.

\begin{itemize}
  \item \((p \rightarrow q) \lor r\)
  \item \neg(p \leftrightarrow p)
  \item \neg

\begin{itemize}
  \item \((p \lor (q \land (r \lor \neg s)))\)
  \item \((p \lor \neg p)\)
\end{itemize}
Let’s consider the first case above: \((p \rightarrow q) \lor r\). How do we build a truth table for this? First, we collect the atomic statements: \(p\), \(q\), and \(r\). To compute the possible truth values of the full WFF, we must consider every combination of truth values for the component atomic statements. Since each statement can take on one of two values, there are \(2^n\) combinations for \(n\) statements. In the present case, there are three atomic statements, so there must be \(2^3 = 2 \times 2 \times 2 = 8\) combinations.

There will then be eight rows in our truth table. The number of columns is governed by the number of atomic statements plus the number of instances of the connectives. In the case at hand, we have three atomic statements and two connectives: ‘\(-\)’ and ‘\(\lor\)’. Thus there will be five columns in our table.

We begin with columns labeled for the three atomic statements plus every possible combination of their values: 8.

\[
\begin{array}{ccc}
\text{p} & \text{q} & \text{r} & \ldots \\
T & T & T & \\
T & T & F & \\
T & F & T & \\
T & F & F & \\
F & T & T & \\
F & T & F & \\
F & F & T & \\
F & F & F & \\
\end{array}
\]

We then construct the additional rows by building up from the atomic statements. In the case at hand, we next construct a column for \((p \rightarrow q)\). We do this from the values in columns one and two, following the pattern outlined in (4.6).
(4.9) \[ \begin{array}{ccc|c} p & q & r & (p \rightarrow q) \\ \hline T & T & T & T \\ T & T & F & T \\ T & F & T & F \\ T & F & F & F \\ F & T & T & T \\ F & T & F & T \\ F & F & T & T \\ F & F & F & T \\ \end{array} \]

Finally, we construct the last column from the values in columns three and four using the pattern outlined in (4.5).

(4.10) \[ \begin{array}{ccc|cc} p & q & r & (p \rightarrow q) & ((p \rightarrow q) \lor r) \\ \hline T & T & T & T & T \\ T & T & F & T & T \\ T & F & T & F & T \\ T & F & F & F & F \\ F & T & T & T & T \\ F & T & F & T & T \\ F & F & T & T & T \\ F & F & F & T & T \\ \end{array} \]

The remaining WFFs are left as an exercise.

4.4 The Meanings of the Connectives

The names of the individual connectives and the corresponding methods for constructing truth tables suggest strong parallels with the meanings of various natural language connectives. There are indeed parallels, but it is essential to keep in mind that the connectives of statement logic have very
precise interpretations that can differ wildly from our intuitive understandings of the corresponding expressions in English. Let’s consider each one of the connectives and give an example for how each differs in interpretation from the corresponding natural language expression.

### 4.4.1 Negation

The negation connective switches the truth value of the WFF it attaches to. Thus \( \neg p \) bears the opposite value from \( p \), whatever that is. There are a number of ways this differs from natural language.

First, as we discussed in chapter 2, some languages use two negative words to express a single negative idea. Thus the Spanish *Ernie no vio nada* ‘Ernie saw nothing’ uses two negative words *no* and *nada* to express a single negative. In the version of statement logic that we have defined, adding a second instance of ‘\( \neg \)’ undoes the effect of the first. Thus \( \neg \neg p \) bears the same truth value as \( p \) and the opposite truth value from \( \neg p \).

Another difference between natural language negation and formal statement logic negation can be exemplified with the following pair of sentences:

\[
(4.11) \quad \begin{align*}
\text{Ernie likes logic.} \\
\text{Ernie doesn’t like logic.}
\end{align*}
\]

In natural language, these sentences do not exhaust the range of possibilities. Ernie could simply not care. That is, in natural language a sentence and its negation do not exhaust the range of possibilities. In statement logic, they do. Thus either \( p \) is true or \( \neg p \) is true. There is no other possibility.

### 4.4.2 Conjunction

Natural language conjunction is also different from statement logic conjunction. Consider, for example, a sentence like the following:

\[
(4.12) \quad \text{Ernie went to the library and Hortence read the book.}
\]

A sentence like this has several implications beyond whether Ernie went to the library and whether Hortence read some book. In particular, the sentence implies that these events are connected. For example, the book was probably borrowed from the library. Another implication is that the events happened
in the order they are given: Ernie first went to the library and then Hortence read the book.

These sorts of connections and implications do not apply to a WFF like \((p \land q)\). All we know is the relationship between the truth value of the whole and the truth values of the parts: \((p \land q)\) is true just in case \(p\) is true and \(q\) is true.

### 4.4.3 Disjunction

Disjunction in natural language is also interpreted differently from logical disjunction. Consider the following example.

\[(4.13)\] Ernie went to the library or Hortence read the book.

A sentence like this has the interpretation that one of the two events holds, but not both. Thus one might interpret this sentence as being true just in case the first part is true and the second false or the second part is true and the first is false, but not if both parts are true.

This is in contrast to a WFF like \((p \lor q)\), which is true if either or both disjuncts are true.

### 4.4.4 Conditional

A natural language conditional is also subject to a different interpretation from the statement logic conditional. Consider a sentence like the following:

\[(4.14)\] If pigs can fly, Ernie will go to the library.

A sentence like this would normally be interpreted as indicating that Ernie will not be going to the library. That is, if the antecedent is obviously false, the consequent—the second statement—must also be false.

This is not true of the statement logic conditional. A statement logic conditional like \((p \rightarrow q)\) is false just in case \(p\) is true and \(q\) is false; in all other cases, it is true. Thus, if \(p\) is false, the whole expression is true regardless of the truth value of \(q\).
4.4.5 Biconditional

The biconditional is probably the least natural of the logical connectives. The expression \textit{if and only if} does not appear to be used in colloquial English. Rather, to communicate the content of a biconditional in spoken English, one frequently hears rather convoluted expressions like the following:

\[(4.15) \text{ Hortence will read the book if Ernie goes to the library and won’t if he doesn’t.} \]

4.5 How Many Connectives?

Notice that each of the connectives, except \(\neg\), can be defined in terms of the others. The expressions in the second column below are true just when the expressions in the third column below are true.

\[
(4.16) \begin{array}{|c|c|c|}
\hline
\text{if} & (p \to q) & (\neg p \lor q) \\
\text{iff} & (p \leftrightarrow q) & ((p \to q) \land (q \to p)) \\
\text{and} & (p \land q) & \neg (\neg p \lor \neg q) \\
\text{or} & (p \lor q) & \neg (\neg p \land \neg q) \\
\hline
\end{array}
\]

Consider, for example, the truth table for the first row.

\[
(4.17) \begin{array}{|c|c|c|c|c|}
\hline
p & q & (p \to q) & \neg p & (\neg p \lor q) \\
\hline
T & T & T & F & T \\
T & F & F & F & F \\
F & T & T & T & T \\
F & F & T & T & T \\
\hline
\end{array}
\]

Notice that the values for the third and fifth columns are identical, indicating that these expressions have the same truth values under the same circumstances. What this means is that any expression with a conditional can be replaced with a negation plus disjunction, and vice versa. The same is true for the other rows in the table in (4.16).

We can, therefore, have a logical system that has the same expressive potential as the one we’ve defined with five connectives, but that has only
two connectives: \( \neg \) plus and one of \( \rightarrow \), \( \land \), or \( \lor \). For example, let’s choose \( \neg \) and \( \land \). The following chart shows how all four connectives can be expressed with only combinations of those two connectives.

\[
\begin{array}{l|l}
(4.18) & \text{if} \quad (p \rightarrow q) & \neg(\neg p \land \neg q) \\

& \text{iff} \quad (p \leftrightarrow q) & (\neg(\neg p \land \neg q) \land \neg(\neg q \land \neg p)) \\

& \text{and} \quad (p \land q) & \text{No translation necessary} \\

& \text{or} \quad (p \lor q) & \neg(\neg p \land \neg q)
\end{array}
\]

What we’ve done here is take each equivalence in the chart in (4.16) through successive equivalences until all that remains are instances of negation and conjunction. For example, starting with \( (p \rightarrow q) \), we first replace the conditional with negation and disjunction, i.e. \( (\neg p \lor q) \). We then use the equivalence of disjunction and conjunction in row three of (4.16) to replace the disjunction with an instance of conjunction, adding more negations in the process. The following truth table for \( (p \rightarrow q) \) and \( \neg(\neg p \land \neg q) \) shows that they are equivalent.

\[
\begin{array}{l|l|l|l|l|l|l|l|l}
(4.19) & p & q & (p \rightarrow q) & \neg p & \neg q & (\neg p \land \neg q) & \neg(\neg p \land \neg q) \\
T & T & T & T & F & T & F & T \\
T & F & F & F & T & T & T & F \\
F & T & T & T & F & F & F & T \\
F & F & T & T & F & T & T & T \\
\end{array}
\]

The third and eighth columns have identical truth values.

In fact, it is possible to have the same expressive potential with only one connective: the Sheffer stroke, e.g. in \( (p \mid q) \). A formula built on the Sheffer stroke is false just in case both components are true; else it is true.

\[
\begin{array}{l|l|l}
(4.20) & p & q \mid (p \mid q) \\
T & T & F \\
T & F & T \\
F & T & T \\
F & F & T \\
\end{array}
\]
Consider, for example, how we might use the Sheffer stroke to get negation. The following truth table shows the equivalence of \( \neg p \) and \((p \mid p)\).

\[(4.21)\]

\[
\begin{array}{c|c|c}
    p & \neg p & (p \mid p) \\
    \hline
    T & F & F \\
    F & T & T \\
\end{array}
\]

The following truth table shows how we can use the Sheffer stroke to get conjunction.

\[(4.22)\]

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

We leave the remaining equivalences as exercises.

### 4.6 Tautology, Contradiction, Contingency

We’ve seen that the truth value of a complex formula can be built up from the truth values of its component atomic statements. Sometimes, this process produces a formula that can be either true or false. Such a formula is referred to as a *contingency*. All the formulas we have given so far have been of this sort. Sometimes, however, the formula can only be true or can only be false. The former is a *tautology* and the latter a *contradiction*.

For example, \( p \rightarrow q \) is a contingency, since it can be true or false.

\[(4.23)\]

\[
\begin{array}{c|c|c}
    p & q & (p \rightarrow q) \\
    \hline
    T & T & T \\
    T & F & F \\
    F & T & T \\
    F & F & T \\
\end{array}
\]
The rightmost column of this truth table contains instances of \( T \) and instances of \( F \). Notice that there are no “degrees” of contingency. If both values are possible, the formula is contingent.

We also have formulas that can only be true: tautologies. For example, \( p \rightarrow p \) is a tautology.

\[
\begin{array}{c|c|c}
   p & (p \rightarrow p) \\
   \hline
   T & T \\
   F & T \\
\end{array}
\]

Even though \( p \) can be either true or false, the combination can only bear the value \( T \). A similar tautology can be built using the biconditional.

\[
\begin{array}{c|c|c}
   p & (p \leftrightarrow p) \\
   \hline
   T & T \\
   F & T \\
\end{array}
\]

Likewise, we have contradictions: formulas that can only be false, e.g. \( (p \land \neg p) \).

\[
\begin{array}{c|c|c|c}
   p & \neg p & (p \land \neg p) \\
   \hline
   T & F & F \\
   F & T & F \\
\end{array}
\]

Again, even though \( p \) can have either value, this combination of connectives can only be false.

## 4.7 Proof

We can now consider proof and argument. Notice first that tautologies built on the biconditional entail that the two sides of the biconditional are \textit{logical equivalents}. That is, each side is true just in case the other side is true and each side is false just in case the other side is false. For example, \( ((p \land q) \leftrightarrow \neg(\neg p \lor \neg q)) \) has this property. \( (p \land q) \) is logically equivalent to \( \neg(\neg p \lor \neg q) \).