Sentential Calculus

Logic is the science of correct, or as we shall say, *valid*, argument. A valid argument is one in which it is not possible for the premisses to be true and the conclusion false, and, moreover, you can tell just by looking at the structure of the argument that it is not possible for the premisses to be true and the conclusion false. So, to do logic, we have to look at the structure of arguments. An argument is made up of sentences and the sentences are made up of words, and we want to see how the words that appear in a correct argument fit together so as to preclude the possibility that the premisses are true and the conclusion false.

Since the arguments we are interested in are mostly formulated in English, the most natural way to proceed would be to examine the way English sentences are built up out of English words, trying to determine what the structure of an English sentence contributes to fixing the conditions under which the sentence is true. Regrettably, we can't proceed that way, because we understand altogether too little about what makes an English sentence true. Natural languages are just too complicated.

Instead, we proceed indirectly, working not with natural languages but with formal languages that are vastly simpler than English, but that are still rich enough to display a lot of the features we find in English. We can straightforwardly determine which arguments in the formal language are valid. Next, we'll learn how to translate from English into the formal language. The plan is to do the translations is such a way that, if the translated argument is valid in the formal language, then English argument must have been valid as well. Thus we can show an English argument valid by providing a translation into a demonstrably valid argument in the formal language. The formal languages are designed to exhibit the logical structure of English arguments in an especially stark form.

We'll begin with *sentential calculus* (*SC*), which provides a model of the way compound English sentences are built up out of simple English sentences. Later, we'll deepen the analysis by examining the internal structure of simple English sentences.

The sentences of the formal languages of the sentential calculus are built up out of simple sentence, called atomic sentences, by putting sentences together using what grammarians call "conjunctions" and what logicians call "connectives." Thus the formal language symbol "$\wedge$" corresponds to the English word "and" or "but." If the formal-language sentence "A" means the same as the English sentence "Jack went up the hill" and "B" means the same as "Jill went up the hill," then "Jack and Jill went up the hill" will be translated "$(A \wedge B)$." The sentence will be true if Jack and Jill both went up the hill, and it will be false if neither went up the hill or if one went up the hill but the other didn't. We can sum the situation up with a truth table:

A	B	(A ∧ B)
T	T	T
T	F	F
F	T	F
F	F	F

"∨" translates the English "or." Thus "(A ∨ B)" means "Jack or Jill went up the hill." It's not entirely clear whether we would ordinarily count the English sentence as true if both Jack and Jill climbed the hill. Mathematicians, at least, always use "or" to mean the inclusive "or"; "A or B" means "A or B or both." This is the usage we'll follow here. Thus we'll use "∨" to mean what lawyers mean when they write "and/or." Thus

A	B	(A ∨ B)
T	T	T
T	F	T
F	T	T
F	F	F

"¬" corresponds to the English "not." "^¬B" is "Jack didn't go up the hill" and "¬B" is "Jill didn't go up the hill."

A	B	¬A	¬B
T	T	F	F
T	F	F	T
F	T	T	F
F	F	T	T

We can combine connectives to make more and more complicated sentences. For example, "Jack went up the hill but Jill did not" is "(A ∧ ¬B)," "Jack and Jill didn't both go up the hill" is "^¬(A ∧ B)," and "Neither Jack nor Jill went up the hill" is "¬(A ∨ B)." Their truth tables are

A	B	(A ∧ ¬B)	¬(A ∧ B)	¬(A ∨ B)
T	T	F	F	F
T	F	T	T	F
F	T	F	T	F
F	F	F	T	T

The connective "→" corresponds roughly to the English "if... then" construction. "If Jack went up the hill then so did Jill" is translated "(A → B)." Its truth table is

A	B	(A → B)
T	T	T
T	F	F
F	T	T

F	F	T

The correlation between the formal language connective "$\rightarrow$" and the English "if...then" is not very good. If I say "If the Giants win the pennant, they'll win the World Series," then if the Giants win both the pennant and the Series, I will have said something true, and if the Giants win the series without winning the Series I will have said something false. But if the Giants don't win the pennant, we don't normally think of what I said as either true or false. A bet that "If the Giants win the pennant they'll win the World Series" would be

won if the Giants win both the pennant and the Series;
lost if the Giants win the pennant but not the Series; and
called off if the Giants don't win the pennant.

A conditional statement like "If the Giants win the pennant, they'll win the World Series" is similar to a conditional promise "If I win the lottery I'll share it with you." Such a promise is kept if I win the lottery and share and broken if I win the lottery without sharing, but if I don't win the lottery, the promise has no effect, and it is neither kept nor broken. Thus the truth table for the English "if...then" looks like this

A	B	if A, then B
T	T	T
T	F	F
F	T	___
F	F	___

In making up the artificial language, logicians filled in the blanks pretty much arbitrarily. When you translate the English "if...then" by the formal language "$\rightarrow$", results you get aren't very reliable as a guide to what's going on in English.

"$\rightarrow$" is also read "only if." "Jack went up the hill only if Jill did" is "$(A \rightarrow B)$."

The other connective in common use is "$\leftrightarrow$", which corresponds to the English phrase "if and only if." "Jack went up the hill if and only if Jill went up the hill," translated "$(A \leftrightarrow B)$," will be true if Jack and Jill both went up the hill or if neither of them went up the hill; it will be false if only one of them went up the hill.

A	B	$(A \leftrightarrow B)$
T	T	T
T	F	F
F	T	F
F	F	T

Some terminology: A sentence of the form $(\varphi \vee \psi)$ is called a *disjunction*, and φ and ψ are its *disjuncts*. A sentence of the form $(\varphi \wedge \psi)$ is a *conjunction*, and φ and ψ are its *conjuncts*. $\neg\varphi$ is a *negation*, and φ is its *negatum*. $(\varphi \rightarrow \psi)$ is a *conditional,* with φ as its *antecedent* and ψ as its *consequent*. $(\varphi \leftrightarrow \psi)$ is a *biconditional* and φ and ψ are its *components*.

The Greek letters aren't a part of the formal language. They're something I use in English to talk about the formal language. We'll employ the Greek letters as variables ranging over the sentences of the formal language. When I say that a sentence of the form $(\varphi \rightarrow \psi)$ is a conditional, with φ as its antecedent and ψ as its consequent, what I mean is that a sentence of the formal language obtained by taking two sentences, putting an arrow between them, and enclosing the result in parentheses is a conditional, with the sentence that appears just before the arrow as its antecedent and the sentence that appears just afterward as its consequent.

A language for the sentential calculus is determined by specifying a nonempty set of simple expressions to serve as the *atomic sentences*. Usually we'll take uppercase Roman letters, sometimes with Arabic numeral subscripts, to serve as our atomic sentences. The sentences of the language, which we refer to as SC sentences, constitute the smallest class of expression that:

> **contains the atomic sentences;**
> **contains $\neg\varphi$ whenever it contains φ; and**
> **contains $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$, $(\varphi \rightarrow \psi)$, and $(\varphi \leftrightarrow \psi)$ whenever it contains both φ and ψ.**

In other words, if _ is the set of all sets of expressions that satisfy the three conditions, then an expression is an SC sentence if and only if it is a member of every member of _.

Another way to give the same definition, without the Greek letters, is this:

> **Every atomic sentence is an SC sentence.**
> **The result of writing "$\neg$" in front of an SC sentence is always an SC sentence.**
> **Whenever you take two SC sentences, write one of the symbols "$\wedge$," "$\vee$," "$\rightarrow$," and "$\leftrightarrow$" between them, and enclose the result parentheses, what you get is an SC sentence.**
> **Nothing is an SC sentence unless it's required to be by the three clauses above.**

Notice the role that parentheses play in preventing ambiguities. If you combine "A," "B," and "C" by first taking the disjunction of "A" and "B," then taking the conjunction with "C," you get something different from what you'd get by taking the disjunction of "A" with the conjunction of "B" and "C." The first procedure gives you "$((A \lor B) \land C$," whereas the second yields "$(A \lor (B \land C))$." Indeed, we have the following:

> **Unique Readability. A sentence of the language for the sentential calculus is built up from sentential letters in a unique way.**

Unique readability is an important respect in which the formal language is an improvement on English. The English sentence "Jack went up the hill or Jill went up the hill and someone fetched a pail of water" is ambiguous.

The proof of unique readability, which involves a meticulous investigation of the mating habits of parentheses [in brief, they are monogamous and heterosexual], will not be given here, but I will give the idea of the proof. The idea of the proof is that we can represent the structure of any given sentence by a finite tree in which each node of the tree is labeled by a sentence. The given sentence is the label of the trunk of tree. Whenever a node is labeled by a conjunction, there will be two other nodes directly beneath the node, each labeled with one of the conjuncts. Similarly for disjunctions, conditionals, and biconditionals. A node labeled by a negation has just one node directly beneath it, labeled by the sentence negated. The leaves of the tree are labeled by atomic sentences. For example, here is the tree associated with the sentence "$\neg(A \land (B \land \neg(\neg A \land C)))$":

$$\neg(A \land (B \land \neg(\neg A \land C)))$$

$$(A \land (B \land \neg(\neg A \land C)))$$

$$A \qquad\qquad\qquad (B \land \neg(\neg A \land C))$$

$$B \qquad\qquad \neg(\neg A \land C)$$

$$(\neg A \land C)$$

$$\neg A \qquad\quad C$$

$$A$$

Unique readability is proved by showing that each sentence is associated with one and only one labeled tree.

Because of unique readability, we know that every SC sentence falls into exactly one of the following six categories:

atomic sentence
conjunction
disjunction
conditional
biconditional
negation

Now we have our formal language, but we can't yet use it to say anything, because the sentences of the language don't mean anything. So far, the sentences of the language are just strings of symbols, no more meaningful than "!Q#W$%&)(*[<>?@[]+@♠_xZ^?_♥." Before a sentence of the language can be used to make a statement that is either true or false, we have to interpret the formal language. This is the task to which we now turn.