

Statement Logic Basics

1.1. Syntax of Statement Logic	1
1.2. Semantics of Statement Logic	2
1.2.1. Negation	2
1.2.2. Conjunction	2
1.2.3. Disjunction	3
1.2.4. The Conditional	3
1.2.5. The Biconditional	3
1.2.6. Truth value of complex formula	4
1.3. Tautologies, contradictions and contingencies	4
1.4. Logical equivalence and laws of statement logic	5

1.1. Syntax of Statement Logic

One of the main features of formal languages is a well-defined Syntax and Semantics. Syntax deals with the *structure* of expressions (and formulas) of language. Semantics deals with the *meaning* of expressions of language.

The syntax of Statement Logic (= Propositional Logic, Sentential Logic) is very simple. Formulas of Statement Logic are built of elements of two main kinds: *atomic statements* and *logical connectives*, symbols of logical operations.

We assume an infinite set *Atom* of *atomic statements* represented by the symbols p, q, r, s, \dots , with primes or subscripts added as needed, $Atom = \{p, q, r, s, \dots\}$

Several *logical connectives* are used: the unary connective “ \neg ” (*negation*) and the binary connectives “ \wedge ” (*conjunction*), “ \vee ” (*disjunction*), “ \rightarrow ” (*conditional*) and “ \leftrightarrow ” (*biconditional*).

Note. Sometimes some other symbols are used with the same names (and meaning): “ \sim ” for negation, “ $\&$ ” for conjunction, “ \supset ” for conditional (called also *implication*) and “ \equiv ” or “ \approx ” for biconditional (called also *equivalence symbols*).

Syntactic Rules:

1. Any atomic statement is a formula.
2. If ϕ and ψ are formulas then $\neg\phi$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$ and $(\phi \leftrightarrow \psi)$ are also formulas.
3. There are no other formulas (in Statement Logic).

Examples of formulas: $p, q, (p \vee q), \neg(p \rightarrow (p \wedge q))$

Notes. Atomic formulas (atomic statements) are considered as logical representatives of simple declarative sentences of natural language. The connectives $\wedge, \vee, \rightarrow$ and \leftrightarrow are intended as counterparts of natural language connectives, for example, English *and, or, if ... then*, and *if and only if*, when used to conjoin declarative sentences. \neg is a unary operator, its English counterpart is *not* or *it is not the case that*. But the relation between (statement) logic and natural language is not uncontroversial. [On first sight, it would seem that there are major discrepancies between the interpretations of some of the logical connectives and of their natural language counterparts. With careful attention to the distinction between semantics and pragmatics, as urged and exemplified in the work of

Grice, one can make a much more plausible case for a close semantic correspondence between the logical connectives and their natural language counterparts. But it is still controversial; see, for instance, Angelika Kratzer's work on conditionals.]

1.2. Semantics of Statement Logic

The semantics of statement logic is nearly as simple as its syntax. First of all we consider two *truth values*: 1 (*true*) and 0 (*false*). Let us denote the set of truth values by D_t , $D_t = \{1,0\}$. We are thus working with two-valued logic. Systems with more than two values have also been studied. But they will not concern us here.

Each atomic statement is assumed to have assigned to it one of the two truth values. Each complex formula also receives a truth value, which is determined by

- (1) the truth values of its syntactic components, and
- (2) the syntactic structure of the formula, i.e. , its connectives and their arrangements in the formula.

So, semantics of every formula is defined compositionally, with the help of its syntax. As we see, syntactically every complex formula is built from its immediate constituents with the help of some connective. The truth value of the formula is determined by truth values of these constituents by *truth-functional* properties of the connective used in the formula. These truth-functional properties of connectives are usually given in the form of *truth tables*.

Below we give the truth tables for the five connectives used in our formulas. In the following, ϕ and ψ stand for any arbitrary formulas, atomic or complex.

1.2.1. Negation

Negation reverses the truth value of the statement to which it is attached. For any formula ϕ , if ϕ is true, then $\neg\phi$ is false, and, conversely, if ϕ is false, then $\neg\phi$ is true. This is summarized in the truth table below

ϕ	$\neg\phi$
1	0
0	1

1.2.2. Conjunction

The result of logical conjunction is true iff both of its conjuncts are true. It corresponds to the meaning of English *and* conjoining two declarative sentences.

The truth table for the logical connective \wedge is given below:

ϕ	ψ	$(\phi \wedge \psi)$
1	1	1
1	0	0
0	1	0
0	0	0

Note that ϕ and ψ are variables denoting any formulas whatsoever and there are four rows in the table corresponding to the four ways of assigning two truth values independently to two statements.

1.2.3. Disjunction.

The logical connective \vee has the following truth table:

ϕ	ψ	$(\phi \vee \psi)$
1	1	1
1	0	1
0	1	1
0	0	0

Thus the disjunction of two statements is true if at least one of the disjuncts is true; it is false only if both are false.

1.2.4. The Conditional.

The truth table for the conditional is shown below:

ϕ	ψ	$(\phi \rightarrow \psi)$
1	1	1
1	0	0
0	1	1
0	0	1

So the formula $(\phi \rightarrow \psi)$ is false only in the case when its *antecedent* (ϕ) is true and the *consequent* (ψ) is false. This table mirrors the use of conditionals in mathematics (in inferences in proofs); its correspondence to natural language *if-then* is controversial.

1.2.5. The Biconditional.

The truth table for the biconditional is shown below:

ϕ	ψ	$(\phi \leftrightarrow \psi)$
1	1	1
1	0	0
0	1	0
0	0	1

The biconditional corresponds to *if and only if*, abbreviated as *iff*, as with the conditional, its correspondence to natural language *if and only if* is controversial (if one considers *if and only if* a part of “natural” natural language at all).

1.2.6. Truth value of complex formula

The truth tables provide a general and systematic method of computing the truth value of any arbitrary complex statement. The number of lines in the truth table (for a given formula) is determined by the requirement that all possible combinations of truth values of atomic statements must be considered. In general, there are 2^n lines when there are n atomic statement in the formula. The order of evaluating the constituent statements is from the most deeply embedded ones to the outermost. So to construct a truth table for $(\neg(p \rightarrow q) \leftrightarrow (p \wedge r))$ one would proceed as follows:

- (i) construct columns for the atomic statements p , q and r ,
- (ii) construct columns for $(p \rightarrow q)$ and for $(p \wedge r)$,
- (iii) construct a column for $\neg(p \rightarrow q)$,
- (iv) construct the truth table for the entire formula

The entire process is laid out in the following table:

p	q	r	$(p \rightarrow q)$	$(p \wedge r)$	$\neg(p \rightarrow q)$	$(\neg(p \rightarrow q) \leftrightarrow (p \wedge r))$
1	1	1	1	1	0	0
1	1	0	1	0	0	1
1	0	1	0	1	1	1
1	0	0	0	0	1	0
0	1	1	1	0	0	1
0	1	0	1	0	0	1
0	0	1	1	0	0	1
0	0	0	1	0	0	1

We can view this table as representing the function which determines the semantic value of a formula from the semantic values of its parts. Arguments of this function are the truth values which are the semantic values of the atomic statements and its result (shown in the last column) is the truth value of the whole formula. (The middle columns show how the function is built up compositionally.) If a formula contains n atomic statements, the corresponding function is of the type

$$D_t^n \rightarrow D_t.$$

1.3. Tautologies, contradictions and contingencies

Statements (formulas) can be classified according to their truth tables (functions they denote). A statement is called a *tautology* if the final column in its truth table contains nothing but 1's, i.e. the statement is always true, whatever the initial assignment of truth values to its atomic statements. Such statements are true simply because of the meaning of the connectives. A statement is called a *contradiction* if the final column in its truth table contains nothing but 0's, i.e. the statement is always false, whatever the initial

assignment of truth values to its atomic statements. All the other statements with both 1 and 0 in their truth table are called *contingencies* or *contingent statements*. Their truth or falsity does depend on the initial truth value assignment to their atomic statements.

Some examples of each type are:

tautologies: $(p \vee \neg p), (p \rightarrow p), (p \leftrightarrow p)$

contradictions: $\neg(p \vee \neg p), (p \wedge \neg p)$

contingencies: $p, (p \vee p), (p \wedge p)$.

1.4. Logical equivalence and laws of statement logic

Two statements are *logically equivalent* if they have the same truth value for any possible assignment of truth values to their atomic parts. To denote logical equivalence between two arbitrary statements ϕ and ψ we write $\phi \leftrightarrow \psi$. Note that “double arrow” is not a new connective for statements, but rather a metalanguage symbol expressing logical equivalence. Read $\phi \leftrightarrow \psi$ as “ ϕ if and only if ψ ” and $\phi \leftrightarrow \psi$ as “the statement ϕ is logically equivalent to the statement ψ ”.

It is a property of statement logic that if a biconditional statement is a tautology, the two constituent statement so connected are logically equivalent. For example, $\neg(p \vee q)$ and $(\neg p \wedge \neg q)$ are logically equivalent, and $(\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q))$ is a tautology. Logically equivalent statements are important because they can freely replace one another in any statement without affecting its truth value.

Below we write down some most frequently used “laws” of equivalency. But first let us add to the set *Atom* of atomic statements two constants: *True* and *False* and let us for convenience use for them the same symbols as for truth values, 1 and 0.

Laws of statement logic

1. Idempotent Laws

(a) $(\phi \vee \phi) \leftrightarrow \phi$

(b) $(\phi \wedge \phi) \leftrightarrow \phi$

2. Commutative Laws

(a) $(\phi \vee \psi) \leftrightarrow (\psi \vee \phi)$

(b) $(\phi \wedge \psi) \leftrightarrow (\psi \wedge \phi)$

3. Associative Laws

(a) $((\phi \vee \psi) \vee \omega) \leftrightarrow (\phi \vee (\psi \vee \omega))$

(b) $((\phi \wedge \psi) \wedge \omega) \leftrightarrow (\phi \wedge (\psi \wedge \omega))$

4. Distributive Laws

(a) $(\phi \vee (\psi \wedge \omega)) \leftrightarrow ((\phi \vee \psi) \wedge (\phi \vee \omega))$

(b) $(\phi \wedge (\psi \vee \omega)) \leftrightarrow ((\phi \wedge \psi) \vee (\phi \wedge \omega))$

5. Identity Laws

(a) $(\phi \vee 0) \leftrightarrow \phi$

(c) $(\phi \wedge 1) \leftrightarrow \phi$

(b) $(\phi \vee 1) \leftrightarrow 1$

(d) $(\phi \wedge 0) \leftrightarrow 0$

6. Complement Laws

(a) $(\varphi \vee \neg\varphi) \Leftrightarrow 1$

(c) $(\varphi \wedge \neg\varphi) \Leftrightarrow 0$

(b) $\neg\neg\varphi = \varphi$ (double negation)

7. DeMorgan's Laws

(a) $\neg(\varphi \vee \psi) \Leftrightarrow (\neg\varphi \wedge \neg\psi)$

(b) $\neg(\varphi \wedge \psi) \Leftrightarrow (\neg\varphi \vee \neg\psi)$

8. Conditional Laws

(a) $(\varphi \rightarrow \psi) \Leftrightarrow (\neg\varphi \vee \psi)$

(b) $(\varphi \rightarrow \psi) \Leftrightarrow (\neg\psi \rightarrow \neg\varphi)$

(c) $(\varphi \rightarrow \psi) \Leftrightarrow \neg(\varphi \wedge \neg\psi)$

9. Biconditional Laws

(a) $(\varphi \leftrightarrow \psi) \Leftrightarrow ((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$ (b) $(\varphi \leftrightarrow \psi) \Leftrightarrow ((\neg\varphi \wedge \neg\psi) \vee (\varphi \wedge \psi))$

The first seven kinds of laws are similar to the first seven kinds of set-theoretical laws (Lecture 1), as well as to the axioms for Boolean algebras. The connections among set theory, logic, and Boolean algebra form an interesting topic and provide the basis for algebraic approaches to logic.

These laws also demonstrate that some connectives could be represented with the help of others. For example, all connectives can be represented in terms of combinations with negation and conjunction, or with negation and disjunction. There exist two binary connectives, “Sheffer stroke” and “Quine’s dagger”, each of which has the property that all the other connectives can be represented with the help of it (see Appendix BI in PTMW, p. 238).