

# Propositional calculus

From Wikipedia, the free encyclopedia

**Propositional calculus** (also called **propositional logic**, **sentential calculus**, **sentential logic**, or sometimes **zeroth-order logic**) is the branch of logic concerned with the study of propositions (whether they are true or false) that are formed by other propositions with the use of logical connectives, and how their value depends on the truth value of their components.

## Contents

- 1 Explanation
- 2 History
- 3 Terminology
- 4 Basic concepts
  - 4.1 Closure under operations
  - 4.2 Argument
- 5 Generic description of a propositional calculus
- 6 Example 1. Simple axiom system
- 7 Example 2. Natural deduction system
- 8 Basic and derived argument forms
- 9 Proofs in propositional calculus
  - 9.1 Example of a proof
- 10 Soundness and completeness of the rules
  - 10.1 Sketch of a soundness proof
  - 10.2 Sketch of completeness proof
  - 10.3 Another outline for a completeness proof
- 11 Interpretation of a truth-functional propositional calculus
  - 11.1 Interpretation of a sentence of truth-functional propositional logic
- 12 Alternative calculus
  - 12.1 Axioms
  - 12.2 Inference rule
  - 12.3 Meta-inference rule
  - 12.4 Example of a proof
- 13 Equivalence to equational logics
- 14 Graphical calculi
- 15 Other logical calculi
- 16 Solvers
- 17 See also
  - 17.1 Higher logical levels
  - 17.2 Related topics
- 18 References
- 19 Further reading
  - 19.1 Related works
- 20 External links

## Explanation

Logical connectives are found in natural languages. In English for example, some examples are "and" (conjunction), "or" (disjunction), "not" (negation) and "if" (but only when used to denote material conditional).

The following is an example of a very simple inference within the scope of propositional logic:

Premise 1: If it's raining then it's cloudy.

Premise 2: It's raining.  
Conclusion: It's cloudy.

Both premises and the conclusion are propositions. The premises are taken for granted and then with the application of modus ponens (an inference rule) the conclusion follows.

As propositional logic is not concerned with the structure of propositions beyond the point where they can't be decomposed anymore by logical connectives, this inference can be restated replacing those *atomic* statements with statement letters, which are interpreted as variables representing statements:

Premise 1:  $P \rightarrow Q$   
Premise 2:  $P$   
Conclusion:  $Q$

The same can be stated succinctly in the following way:

$P \rightarrow Q, P \vdash Q$

When  $P$  is interpreted as “It's raining” and  $Q$  as “it's cloudy” the above symbolic expressions can be seen to exactly correspond with the original expression in natural language. Not only that, but they will also correspond with any other inference of this *form*, which will be valid on the same basis that this inference is.

Propositional logic may be studied through a formal system in which formulas of a formal language may be interpreted to represent propositions. A system of inference rules and axioms allows certain formulas to be derived. These derived formulas are called theorems and may be interpreted to be true propositions. A constructed sequence of such formulas is known as a *derivation* or *proof* and the last formula of the sequence is the theorem. The derivation may be interpreted as proof of the proposition represented by the theorem.

When a formal system is used to represent formal logic, only statement letters are represented directly. The natural language propositions that arise when they're interpreted are outside the scope of the system, and the relation between the formal system and its interpretation is likewise outside the formal system itself.

Usually in **truth-functional propositional logic**, formulas are interpreted as having either a truth value of *true* or a truth value of *false*. Truth-functional propositional logic and systems isomorphic to it, are considered to be **zeroth-order logic**.

## History

Although propositional logic (which is interchangeable with propositional calculus) had been hinted by earlier philosophers, it was developed into a formal logic by Chrysippus in the 3rd century BC<sup>[1]</sup> and expanded by his successor Stoics. The logic was focused on propositions. This advancement was different from the traditional syllogistic logic which was focused on terms. However, later in antiquity, the propositional logic developed by the Stoics was no longer understood. Consequently, the system was essentially reinvented by Peter Abelard in the 12th century.<sup>[2]</sup>

Propositional logic was eventually refined using symbolic logic. The 17th/18th-century mathematician Gottfried Leibniz has been credited with being the founder of symbolic logic for his work with the calculus ratiocinator. Although his work was the first of its kind, it was unknown to the larger logical community. Consequently, many of the advances achieved by Leibniz were recreated by logicians like George Boole and Augustus De Morgan completely independent of Leibniz.<sup>[3]</sup>

Just as propositional logic can be considered an advancement from the earlier syllogistic logic, Gottlob Frege's predicate logic was an advancement from the earlier propositional logic. One author describes predicate logic as combining "the distinctive features of syllogistic logic and propositional logic."<sup>[4]</sup> Consequently, predicate logic ushered in a new era in logic's history; however, advances in propositional logic were still made after

Frege, including Natural Deduction, Truth-Trees and Truth-Tables. Natural deduction was invented by Gerhard Gentzen and Jan Łukasiewicz. Truth-Trees were invented by Evert Willem Beth.<sup>[5]</sup> The invention of truth-tables, however, is of uncertain attribution.

Within works by Frege<sup>[6]</sup> and Bertrand Russell,<sup>[7]</sup> are ideas influential to the invention of truth tables. The actual tabular structure (being formatted as a table), itself, is generally credited to either Ludwig Wittgenstein or Emil Post (or both, independently).<sup>[6]</sup> Besides Frege and Russell, others credited with having ideas preceding truth-tables include Philo, Boole, Charles Sanders Peirce<sup>[8]</sup>, and Ernst Schröder. Others credited with the tabular structure include Jan Łukasiewicz, Ernst Schröder, Alfred North Whitehead, William Stanley Jevons, John Venn, and Clarence Irving Lewis.<sup>[7]</sup> Ultimately, some have concluded, like John Shosky, that "It is far from clear that any one person should be given the title of 'inventor' of truth-tables."<sup>[7]</sup>

## Terminology

In general terms, a calculus is a formal system that consists of a set of syntactic expressions (*well-formed formulas*), a distinguished subset of these expressions (axioms), plus a set of formal rules that define a specific binary relation, intended to be interpreted as logical equivalence, on the space of expressions.

When the formal system is intended to be a logical system, the expressions are meant to be interpreted as statements, and the rules, known to be *inference rules*, are typically intended to be truth-preserving. In this setting, the rules (which may include axioms) can then be used to derive ("infer") formulas representing true statements from given formulas representing true statements.

The set of axioms may be empty, a nonempty finite set, a countably infinite set, or be given by axiom schemata. A formal grammar recursively defines the expressions and well-formed formulas of the language. In addition a semantics may be given which defines truth and valuations (or interpretations).

The language of a propositional calculus consists of

1. a set of primitive symbols, variously referred to as *atomic formulas*, *placeholders*, *proposition letters*, or *variables*, and
2. a set of operator symbols, variously interpreted as *logical operators* or *logical connectives*.

A *well-formed formula* is any atomic formula, or any formula that can be built up from atomic formulas by means of operator symbols according to the rules of the grammar.

Mathematicians sometimes distinguish between propositional constants, propositional variables, and schemata. Propositional constants represent some particular proposition, while propositional variables range over the set of all atomic propositions. Schemata, however, range over all propositions. It is common to represent propositional constants by  $A$ ,  $B$ , and  $C$ , propositional variables by  $P$ ,  $Q$ , and  $R$ , and schematic letters are often Greek letters, most often  $\varphi$ ,  $\psi$ , and  $\chi$ .

## Basic concepts

The following outlines a standard propositional calculus. Many different formulations exist which are all more or less equivalent but differ in the details of:

1. their language, that is, the particular collection of primitive symbols and operator symbols,
2. the set of axioms, or distinguished formulas, and
3. the set of inference rules.

Any given proposition may be represented with a letter called a 'propositional constant', analogous to representing a number by a letter in mathematics, for instance,  $a = 5$ . All propositions require exactly one of two truth-values: true or false. For example, let  $P$  be the proposition that it is raining outside. This will be true ( $P$ ) if it is raining outside and false otherwise ( $\neg P$ ).

- We then define truth-functional operators, beginning with negation.  $\neg P$  represents the negation of  $P$ , which can be thought of as the denial of  $P$ . In the example above,  $\neg P$  expresses that it is not raining outside, or by a more standard reading: "It is not the case that it is raining outside." When  $P$  is true,  $\neg P$  is false; and when  $P$  is false,  $\neg P$  is true.  $\neg\neg P$  always has the same truth-value as  $P$ .
- Conjunction is a truth-functional connective which forms a proposition out of two simpler propositions, for example,  $P$  and  $Q$ . The conjunction of  $P$  and  $Q$  is written  $P \wedge Q$ , and expresses that each are true. We read  $P \wedge Q$  for " $P$  and  $Q$ ". For any two propositions, there are four possible assignments of truth values:
  1.  $P$  is true and  $Q$  is true
  2.  $P$  is true and  $Q$  is false
  3.  $P$  is false and  $Q$  is true
  4.  $P$  is false and  $Q$  is false

The conjunction of  $P$  and  $Q$  is true in case 1 and is false otherwise. Where  $P$  is the proposition that it is raining outside and  $Q$  is the proposition that a cold-front is over Kansas,  $P \wedge Q$  is true when it is raining outside and there is a cold-front over Kansas. If it is not raining outside, then  $P \wedge Q$  is false; and if there is no cold-front over Kansas, then  $P \wedge Q$  is false.

- Disjunction resembles conjunction in that it forms a proposition out of two simpler propositions. We write it  $P \vee Q$ , and it is read " $P$  or  $Q$ ". It expresses that either  $P$  or  $Q$  is true. Thus, in the cases listed above, the disjunction of  $P$  with  $Q$  is true in all cases except case 4. Using the example above, the disjunction expresses that it is either raining outside or there is a cold front over Kansas. (Note, this use of disjunction is supposed to resemble the use of the English word "or". However, it is most like the English inclusive "or", which can be used to express the truth of at least one of two propositions. It is not like the English exclusive "or", which expresses the truth of exactly one of two propositions. That is to say, the exclusive "or" is false when both  $P$  and  $Q$  are true (case 1). An example of the exclusive or is: You may have a bagel or a pastry, but not both. Often in natural language, given the appropriate context, the addendum "but not both" is omitted but implied. In mathematics, however, "or" is always inclusive or; if exclusive or is meant it will be specified, possibly by "xor".)
- Material conditional also joins two simpler propositions, and we write  $P \rightarrow Q$ , which is read "if  $P$  then  $Q$ ". The proposition to the left of the arrow is called the antecedent and the proposition to the right is called the consequent. (There is no such designation for conjunction or disjunction, since they are commutative operations.) It expresses that  $Q$  is true whenever  $P$  is true. Thus it is true in every case above except case 2, because this is the only case when  $P$  is true but  $Q$  is not. Using the example, if  $P$  then  $Q$  expresses that if it is raining outside then there is a cold-front over Kansas. The material conditional is often confused with physical causation. The material conditional, however, only relates two propositions by their truth-values—which is not the relation of cause and effect. It is contentious in the literature whether the material implication represents logical causation.
- Biconditional joins two simpler propositions, and we write  $P \leftrightarrow Q$ , which is read " $P$  if and only if  $Q$ ". It expresses that  $P$  and  $Q$  have the same truth-value, and so, in cases 1 and 4,  $P$  is true if and only if  $Q$  is true, and false otherwise.

It is extremely helpful to look at the truth tables for these different operators, as well as the method of analytic tableaux.

## Closure under operations

Propositional logic is closed under truth-functional connectives. That is to say, for any proposition  $\phi$ ,  $\neg\phi$  is also a proposition. Likewise, for any propositions  $\phi$  and  $\psi$ ,  $\phi \wedge \psi$  is a proposition, and similarly for disjunction, conditional, and biconditional. This implies that, for instance,  $\phi \wedge \psi$  is a proposition, and so it can be conjoined with another proposition. In order to represent this, we need to use parentheses to indicate which proposition is conjoined with which. For instance,  $P \wedge Q \wedge R$  is not a well-formed formula, because we do not know if we are conjoining  $P \wedge Q$  with  $R$  or if we are conjoining  $P$  with  $Q \wedge R$ . Thus we must write either  $(P \wedge Q) \wedge R$  to represent the former, or  $P \wedge (Q \wedge R)$  to represent the latter. By evaluating the truth conditions, we see that both expressions have the same truth conditions (will be true in the same cases), and moreover that any proposition formed by arbitrary conjunctions will have the same truth conditions, regardless

of the location of the parentheses. This means that conjunction is associative, however, one should not assume that parentheses never serve a purpose. For instance, the sentence  $P \wedge (Q \vee R)$  does not have the same truth conditions of  $(P \wedge Q) \vee R$ , so they are different sentences distinguished only by the parentheses. One can verify this by the truth-table method referenced above.

Note: For any arbitrary number of propositional constants, we can form a finite number of cases which list their possible truth-values. A simple way to generate this is by truth-tables, in which one writes  $P, Q, \dots, Z$ , for any list of  $k$  propositional constants—that is to say, any list of propositional constants with  $k$  entries. Below this list, one writes  $2^k$  rows, and below  $P$  one fills in the first half of the rows with true (or T) and the second half with false (or F). Below  $Q$  one fills in one-quarter of the rows with T, then one-quarter with F, then one-quarter with T and the last quarter with F. The next column alternates between true and false for each eighth of the rows, then sixteenths, and so on, until the last propositional constant varies between T and F for each row. This will give a complete listing of cases or truth-value assignments possible for those propositional constants.

## Argument

The propositional calculus then defines an *argument* to be a list of propositions. A valid argument is a list of propositions, the last of which follows from—or is implied by—the rest. All other arguments are invalid. The simplest valid argument is modus ponens, one instance of which is the following list of propositions:

$$\begin{array}{l} 1. \quad P \rightarrow Q \\ 2. \quad P \\ \hline \therefore Q \end{array}$$

This is a list of three propositions, each line is a proposition, and the last follows from the rest. The first two lines are called premises, and the last line the conclusion. We say that any proposition  $C$  follows from any set of propositions  $(P_1, \dots, P_n)$ , if  $C$  must be true whenever every member of the set  $(P_1, \dots, P_n)$  is true. In the argument above, for any  $P$  and  $Q$ , whenever  $P \rightarrow Q$  and  $P$  are true, necessarily  $Q$  is true. Notice that, when  $P$  is true, we cannot consider cases 3 and 4 (from the truth table). When  $P \rightarrow Q$  is true, we cannot consider case 2. This leaves only case 1, in which  $Q$  is also true. Thus  $Q$  is implied by the premises.

This generalizes schematically. Thus, where  $\varphi$  and  $\psi$  may be any propositions at all,

$$\begin{array}{l} 1. \quad \varphi \rightarrow \psi \\ 2. \quad \varphi \\ \hline \therefore \psi \end{array}$$

Other argument forms are convenient, but not necessary. Given a complete set of axioms (see below for one such set), modus ponens is sufficient to prove all other argument forms in propositional logic, thus they may be considered to be a derivative. Note, this is not true of the extension of propositional logic to other logics like first-order logic. First-order logic requires at least one additional rule of inference in order to obtain completeness.

The significance of argument in formal logic is that one may obtain new truths from established truths. In the first example above, given the two premises, the truth of  $Q$  is not yet known or stated. After the argument is made,  $Q$  is deduced. In this way, we define a deduction system to be a set of all propositions that may be deduced from another set of propositions. For instance, given the set of propositions

$A = \{P \vee Q, \neg Q \wedge R, (P \vee Q) \rightarrow R\}$ , we can define a deduction system,  $\Gamma$ , which is the set of all propositions which follow from  $A$ . Reiteration is always assumed, so  $P \vee Q, \neg Q \wedge R, (P \vee Q) \rightarrow R \in \Gamma$ . Also, from the first element of  $A$ , last element, as well as modus ponens,  $R$  is a consequence, and so  $R \in \Gamma$ . Because we have not included sufficiently complete axioms, though, nothing else may be deduced. Thus, even though most deduction systems studied in propositional logic are able to deduce  $(P \vee Q) \leftrightarrow (\neg P \rightarrow Q)$ , this one is too weak to prove such a proposition.

# Generic description of a propositional calculus

A **propositional calculus** is a formal system  $\mathcal{L} = \mathcal{L}(\mathbf{A}, \Omega, \mathbf{Z}, \mathbf{I})$ , where:

- The *alpha set*  $\mathbf{A}$  is a finite set of elements called *proposition symbols* or *propositional variables*. Syntactically speaking, these are the most basic elements of the formal language  $\mathcal{L}$ , otherwise referred to as *atomic formulas* or *terminal elements*. In the examples to follow, the elements of  $\mathbf{A}$  are typically the letters  $p, q, r$ , and so on.
- The *omega set*  $\Omega$  is a finite set of elements called *operator symbols* or *logical connectives*. The set  $\Omega$  is partitioned into disjoint subsets as follows:

$$\Omega = \Omega_0 \cup \Omega_1 \cup \dots \cup \Omega_j \cup \dots \cup \Omega_m.$$

In this partition,  $\Omega_j$  is the set of operator symbols of *arity*  $j$ .

In the more familiar propositional calculi,  $\Omega$  is typically partitioned as follows:

$$\begin{aligned}\Omega_1 &= \{\neg\}, \\ \Omega_2 &\subseteq \{\wedge, \vee, \rightarrow, \leftrightarrow\}.\end{aligned}$$

A frequently adopted convention treats the constant logical values as operators of arity zero, thus:

$$\Omega_0 = \{0, 1\}.$$

Some writers use the tilde ( $\sim$ ), or  $N$ , instead of  $\neg$ ; and some use the ampersand ( $\&$ ), the prefixed  $K$ , or  $\cdot$  instead of  $\wedge$ . Notation varies even more for the set of logical values, with symbols like  $\{\text{false}, \text{true}\}$ ,  $\{F, T\}$ , or  $\{\perp, \top\}$  all being seen in various contexts instead of  $\{0, 1\}$ .

- The *zeta set*  $\mathbf{Z}$  is a finite set of *transformation rules* that are called *inference rules* when they acquire logical applications.
- The *iota set*  $\mathbf{I}$  is a finite set of *initial points* that are called *axioms* when they receive logical interpretations.

The *language* of  $\mathcal{L}$ , also known as its set of *formulas*, *well-formed formulas*, is inductively defined by the following rules:

1. Base: Any element of the alpha set  $\mathbf{A}$  is a formula of  $\mathcal{L}$ .
2. If  $p_1, p_2, \dots, p_j$  are formulas and  $f$  is in  $\Omega_j$ , then  $(f(p_1, p_2, \dots, p_j))$  is a formula.
3. Closed: Nothing else is a formula of  $\mathcal{L}$ .

Repeated applications of these rules permits the construction of complex formulas. For example:

1. By rule 1,  $p$  is a formula.
2. By rule 2,  $\neg p$  is a formula.
3. By rule 1,  $q$  is a formula.
4. By rule 2,  $(\neg p \vee q)$  is a formula.

## Example 1. Simple axiom system

Let  $\mathcal{L}_1 = \mathcal{L}(\mathbf{A}, \Omega, \mathbf{Z}, \mathbf{I})$ , where  $\mathbf{A}, \Omega, \mathbf{Z}, \mathbf{I}$  are defined as follows:

- The alpha set  $\mathbf{A}$ , is a finite set of symbols that is large enough to supply the needs of a given discussion, for example:

$$\mathbf{A} = \{p, q, r, s, t, u\}.$$

- Of the three connectives for conjunction, disjunction, and implication ( $\wedge$ ,  $\vee$ , and  $\rightarrow$ ), one can be taken as primitive and the other two can be defined in terms of it and negation ( $\neg$ ).<sup>[9]</sup> Indeed, all of the logical connectives can be defined in terms of a sole sufficient operator. The biconditional ( $\leftrightarrow$ ) can of course be defined in terms of conjunction and implication, with  $a \leftrightarrow b$  defined as  $(a \rightarrow b) \wedge (b \rightarrow a)$ .

Adopting negation and implication as the two primitive operations of a propositional calculus is tantamount to having the omega set  $\Omega = \Omega_1 \cup \Omega_2$  partition as follows:

$$\begin{aligned}\Omega_1 &= \{\neg\}, \\ \Omega_2 &= \{\rightarrow\}.\end{aligned}$$

- An axiom system discovered by Jan Łukasiewicz formulates a propositional calculus in this language as follows. The axioms are all substitution instances of:
  - $(p \rightarrow (q \rightarrow p))$
  - $((p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r)))$
  - $((\neg p \rightarrow \neg q) \rightarrow (q \rightarrow p))$
- The rule of inference is modus ponens (i.e., from  $p$  and  $(p \rightarrow q)$ , infer  $q$ ). Then  $a \vee b$  is defined as  $\neg a \rightarrow b$ , and  $a \wedge b$  is defined as  $\neg(a \rightarrow \neg b)$ . This system is used in Metamath set.mm formal proof database.

## Example 2. Natural deduction system

Let  $\mathcal{L}_2 = \mathcal{L}(\mathbf{A}, \Omega, \mathbf{Z}, \mathbf{I})$ , where  $\mathbf{A}$ ,  $\Omega$ ,  $\mathbf{Z}$ ,  $\mathbf{I}$  are defined as follows:

- The alpha set  $\mathbf{A}$ , is a finite set of symbols that is large enough to supply the needs of a given discussion, for example:

$$\mathbf{A} = \{p, q, r, s, t, u\}.$$

- The omega set  $\Omega = \Omega_1 \cup \Omega_2$  partitions as follows:

$$\begin{aligned}\Omega_1 &= \{\neg\}, \\ \Omega_2 &= \{\wedge, \vee, \rightarrow, \leftrightarrow\}.\end{aligned}$$

In the following example of a propositional calculus, the transformation rules are intended to be interpreted as the inference rules of a so-called *natural deduction system*. The particular system presented here has no initial points, which means that its interpretation for logical applications derives its theorems from an empty axiom set.

- The set of initial points is empty, that is,  $\mathbf{I} = \emptyset$ .
- The set of transformation rules,  $\mathbf{Z}$ , is described as follows:

Our propositional calculus has ten inference rules. These rules allow us to derive other true formulas given a set of formulas that are assumed to be true. The first nine simply state that we can infer certain well-formed formulas from other well-formed formulas. The last rule however uses hypothetical reasoning in the sense that in the premise of the rule we temporarily assume an (unproven) hypothesis to be part of the set of inferred formulas to see if we can infer a certain other formula. Since the first nine rules don't do this they are usually described as *non-hypothetical* rules, and the last one as a *hypothetical* rule.

In describing the transformation rules, we may introduce a metalanguage symbol  $\vdash$ . It is basically a convenient shorthand for saying "infer that". The format is  $\Gamma \vdash \psi$ , in which  $\Gamma$  is a (possibly empty) set of formulas called premises, and  $\psi$  is a formula called conclusion. The transformation rule  $\Gamma \vdash \psi$  means that if every proposition in  $\Gamma$  is a theorem (or has the same truth value as the axioms), then  $\psi$  is also a theorem. Note that considering the following rule Conjunction introduction, we will know whenever  $\Gamma$  has more than one formula, we can always safely reduce it into one formula using conjunction. So for short, from that time on we may represent  $\Gamma$  as one formula instead of a set. Another omission for convenience is when  $\Gamma$  is an empty set, in which case  $\Gamma$  may not appear.

#### **Negation introduction**

From  $(p \rightarrow q)$  and  $(p \rightarrow \neg q)$ , infer  $\neg p$ .

That is,  $\{(p \rightarrow q), (p \rightarrow \neg q)\} \vdash \neg p$ .

#### **Negation elimination**

From  $\neg p$ , infer  $(p \rightarrow r)$ .

That is,  $\{\neg p\} \vdash (p \rightarrow r)$ .

#### **Double negative elimination**

From  $\neg\neg p$ , infer  $p$ .

That is,  $\neg\neg p \vdash p$ .

#### **Conjunction introduction**

From  $p$  and  $q$ , infer  $(p \wedge q)$ .

That is,  $\{p, q\} \vdash (p \wedge q)$ .

#### **Conjunction elimination**

From  $(p \wedge q)$ , infer  $p$ .

From  $(p \wedge q)$ , infer  $q$ .

That is,  $(p \wedge q) \vdash p$  and  $(p \wedge q) \vdash q$ .

#### **Disjunction introduction**

From  $p$ , infer  $(p \vee q)$ .

From  $q$ , infer  $(p \vee q)$ .

That is,  $p \vdash (p \vee q)$  and  $q \vdash (p \vee q)$ .

#### **Disjunction elimination**

From  $(p \vee q)$  and  $(p \rightarrow r)$  and  $(q \rightarrow r)$ , infer  $r$ .

That is,  $\{p \vee q, p \rightarrow r, q \rightarrow r\} \vdash r$ .

#### **Biconditional introduction**

From  $(p \rightarrow q)$  and  $(q \rightarrow p)$ , infer  $(p \leftrightarrow q)$ .

That is,  $\{p \rightarrow q, q \rightarrow p\} \vdash (p \leftrightarrow q)$ .

#### **Biconditional elimination**

From  $(p \leftrightarrow q)$ , infer  $(p \rightarrow q)$ .

From  $(p \leftrightarrow q)$ , infer  $(q \rightarrow p)$ .

That is,  $(p \leftrightarrow q) \vdash (p \rightarrow q)$  and  $(p \leftrightarrow q) \vdash (q \rightarrow p)$ .

#### **Modus ponens (conditional elimination)**

From  $p$  and  $(p \rightarrow q)$ , infer  $q$ .

That is,  $\{p, p \rightarrow q\} \vdash q$ .

#### **Conditional proof (conditional introduction)**

From [accepting  $p$  allows a proof of  $q$ ], infer  $(p \rightarrow q)$ .

That is,  $(p \vdash q) \vdash (p \rightarrow q)$ .

## **Basic and derived argument forms**



# Basic and Derived Argument Forms

Name	Sequent	Description
Modus Ponens	$((p \rightarrow q) \wedge p) \vdash q$	If $p$ then $q$ ; $p$ ; therefore $q$
Modus Tollens	$((p \rightarrow q) \wedge \neg q) \vdash \neg p$	If $p$ then $q$ ; not $q$ ; therefore not $p$
Hypothetical Syllogism	$((p \rightarrow q) \wedge (q \rightarrow r)) \vdash (p \rightarrow r)$	If $p$ then $q$ ; if $q$ then $r$ ; therefore, if $p$ then $r$
Disjunctive Syllogism	$((p \vee q) \wedge \neg p) \vdash q$	Either $p$ or $q$ , or both; not $p$ ; therefore, $q$
Constructive Dilemma	$((p \rightarrow q) \wedge (r \rightarrow s) \wedge (p \vee r)) \vdash (q \vee s)$	If $p$ then $q$ ; and if $r$ then $s$ ; but $p$ or $r$ ; therefore $q$ or $s$
Destructive Dilemma	$((p \rightarrow q) \wedge (r \rightarrow s) \wedge (\neg q \vee \neg s)) \vdash (\neg p \vee \neg r)$	If $p$ then $q$ ; and if $r$ then $s$ ; but not $q$ or not $s$ ; therefore not $p$ or not $r$
Bidirectional Dilemma	$((p \rightarrow q) \wedge (r \rightarrow s) \wedge (p \vee \neg s)) \vdash (q \vee \neg r)$	If $p$ then $q$ ; and if $r$ then $s$ ; but $p$ or not $s$ ; therefore $q$ or not $r$
Simplification	$(p \wedge q) \vdash p$	$p$ and $q$ are true; therefore $p$ is true
Conjunction	$p, q \vdash (p \wedge q)$	$p$ and $q$ are true separately; therefore they are true conjointly
Addition	$p \vdash (p \vee q)$	$p$ is true; therefore the disjunction $p$ or $q$ is true
Composition	$((p \rightarrow q) \wedge (p \rightarrow r)) \vdash (p \rightarrow (q \wedge r))$	If $p$ then $q$ ; and if $p$ then $r$ ; therefore if $p$ is true then $q$ and $r$ are true
De Morgan's Theorem(1)	$\neg(p \wedge q) \vdash (\neg p \vee \neg q)$	The negation of ( $p$ and $q$ ) is equiv. to (not $p$ or not $q$ )
De Morgan's Theorem(2)	$\neg(p \vee q) \vdash (\neg p \wedge \neg q)$	The negation of ( $p$ or $q$ ) is equiv. to (not $p$ and not $q$ )
Commutation(1)	$(p \vee q) \vdash (q \vee p)$	$(p$ or $q)$ is equiv. to $(q$ or $p)$
Commutation(2)	$(p \wedge q) \vdash (q \wedge p)$	$(p$ and $q)$ is equiv. to $(q$ and $p)$
Commutation(3)	$(p \leftrightarrow q) \vdash (q \leftrightarrow p)$	$(p$ is equiv. to $q)$ is equiv. to $(q$ is equiv. to $p)$
Association(1)	$(p \vee (q \vee r)) \vdash ((p \vee q) \vee r)$	$p$ or $(q$ or $r)$ is equiv. to $(p$ or $q)$ or $r$
Association(2)	$(p \wedge (q \wedge r)) \vdash ((p \wedge q) \wedge r)$	$p$ and $(q$ and $r)$ is equiv. to $(p$ and $q)$ and $r$
Distribution(1)	$(p \wedge (q \vee r)) \vdash ((p \wedge q) \vee (p \wedge r))$	$p$ and $(q$ or $r)$ is equiv. to $(p$ and $q)$ or $(p$ and $r)$
Distribution(2)	$(p \vee (q \wedge r)) \vdash ((p \vee q) \wedge (p \vee r))$	$p$ or $(q$ and $r)$ is equiv. to $(p$ or $q)$ and $(p$ or $r)$
Double Negation	$p \vdash \neg\neg p$	$p$ is equivalent to the negation of not $p$
Transposition	$(p \rightarrow q) \vdash (\neg q \rightarrow \neg p)$	If $p$ then $q$ is equiv. to if not $q$ then not $p$
Material Implication	$(p \rightarrow q) \vdash (\neg p \vee q)$	If $p$ then $q$ is equiv. to not $p$ or $q$
Material Equivalence(1)	$(p \leftrightarrow q) \vdash ((p \rightarrow q) \wedge (q \rightarrow p))$	$(p$ iff $q)$ is equiv. to (if $p$ is true then $q$ is true) and (if $q$ is true then $p$ is true)
Material Equivalence(2)	$(p \leftrightarrow q) \vdash ((p \wedge q) \vee (\neg p \wedge \neg q))$	$(p$ iff $q)$ is equiv. to either $(p$ and $q$ are true) or $(p$ and $q$ are false)
Material Equivalence(3)	$(p \leftrightarrow q) \vdash ((p \vee \neg q) \wedge (\neg p \vee q))$	$(p$ iff $q)$ is equiv to., both $(p$ or not $q$ is true) and $(p$ or not $q$ is true)
Exportation <sup>[10]</sup>	$((p \wedge q) \rightarrow r) \vdash (p \rightarrow (q \rightarrow r))$	from (if $p$ and $q$ are true then $r$ is true) we

		can prove (if $q$ is true then $r$ is true, if $p$ is true)
Importation	$(p \rightarrow (q \rightarrow r)) \vdash ((p \wedge q) \rightarrow r)$	If $p$ then (if $q$ then $r$ ) is equivalent to if $p$ and $q$ then $r$
Tautology (1)	$p \vdash (p \vee p)$	$p$ is true is equiv. to $p$ is true or $p$ is true
Tautology (2)	$p \vdash (p \wedge p)$	$p$ is true is equiv. to $p$ is true and $p$ is true
Tertium non datur (Law of Excluded Middle)	$\vdash (p \vee \neg p)$	$p$ or not $p$ is true
Law of Non-Contradiction	$\vdash \neg(p \wedge \neg p)$	$p$ and not $p$ is false, is a true statement

## Proofs in propositional calculus

One of the main uses of a propositional calculus, when interpreted for logical applications, is to determine relations of logical equivalence between propositional formulas. These relationships are determined by means of the available transformation rules, sequences of which are called *derivations* or *proofs*.

In the discussion to follow, a proof is presented as a sequence of numbered lines, with each line consisting of a single formula followed by a *reason* or *justification* for introducing that formula. Each premise of the argument, that is, an assumption introduced as an hypothesis of the argument, is listed at the beginning of the sequence and is marked as a "premise" in lieu of other justification. The conclusion is listed on the last line. A proof is complete if every line follows from the previous ones by the correct application of a transformation rule. (For a contrasting approach, see proof-trees).

### Example of a proof

- To be shown that  $A \rightarrow A$ .
- One possible proof of this (which, though valid, happens to contain more steps than are necessary) may be arranged as follows:

Example of a Proof

Number	Formula	Reason
1	$A$	premise
2	$A \vee A$	From (1) by disjunction introduction
3	$(A \vee A) \wedge A$	From (1) and (2) by conjunction introduction
4	$A$	From (3) by conjunction elimination
5	$A \vdash A$	Summary of (1) through (4)
6	$\vdash A \rightarrow A$	From (5) by conditional proof

Interpret  $A \vdash A$  as "Assuming  $A$ , infer  $A$ ". Read  $\vdash A \rightarrow A$  as "Assuming nothing, infer that  $A$  implies  $A$ ", or "It is a tautology that  $A$  implies  $A$ ", or "It is always true that  $A$  implies  $A$ ".

## Soundness and completeness of the rules

The crucial properties of this set of rules are that they are *sound* and *complete*. Informally this means that the rules are correct and that no other rules are required. These claims can be made more formal as follows.

We define a *truth assignment* as a function that maps propositional variables to **true** or **false**. Informally such a truth assignment can be understood as the description of a possible state of affairs (or possible world) where certain statements are true and others are not. The semantics of formulas can then be formalized by defining for

which "state of affairs" they are considered to be true, which is what is done by the following definition.

We define when such a truth assignment  $A$  satisfies a certain well-formed formula with the following rules:

- $A$  satisfies the propositional variable  $P$  if and only if  $A(P) = \text{true}$
- $A$  satisfies  $\neg\varphi$  if and only if  $A$  does not satisfy  $\varphi$
- $A$  satisfies  $(\varphi \wedge \psi)$  if and only if  $A$  satisfies both  $\varphi$  and  $\psi$
- $A$  satisfies  $(\varphi \vee \psi)$  if and only if  $A$  satisfies at least one of either  $\varphi$  or  $\psi$
- $A$  satisfies  $(\varphi \rightarrow \psi)$  if and only if it is not the case that  $A$  satisfies  $\varphi$  but not  $\psi$
- $A$  satisfies  $(\varphi \leftrightarrow \psi)$  if and only if  $A$  satisfies both  $\varphi$  and  $\psi$  or satisfies neither one of them

With this definition we can now formalize what it means for a formula  $\varphi$  to be implied by a certain set  $S$  of formulas. Informally this is true if in all worlds that are possible given the set of formulas  $S$  the formula  $\varphi$  also holds. This leads to the following formal definition: We say that a set  $S$  of well-formed formulas *semantically entails* (or *implies*) a certain well-formed formula  $\varphi$  if all truth assignments that satisfy all the formulas in  $S$  also satisfy  $\varphi$ .

Finally we define *syntactical entailment* such that  $\varphi$  is syntactically entailed by  $S$  if and only if we can derive it with the inference rules that were presented above in a finite number of steps. This allows us to formulate exactly what it means for the set of inference rules to be sound and complete:

**Soundness:** If the set of well-formed formulas  $S$  syntactically entails the well-formed formula  $\varphi$  then  $S$  semantically entails  $\varphi$ .

**Completeness:** If the set of well-formed formulas  $S$  semantically entails the well-formed formula  $\varphi$  then  $S$  syntactically entails  $\varphi$ .

For the above set of rules this is indeed the case.

## Sketch of a soundness proof

(For most logical systems, this is the comparatively "simple" direction of proof)

Notational conventions: Let  $G$  be a variable ranging over sets of sentences. Let  $A$ ,  $B$  and  $C$  range over sentences. For " $G$  syntactically entails  $A$ " we write " $G$  proves  $A$ ". For " $G$  semantically entails  $A$ " we write " $G$  implies  $A$ ".

We want to show:  $(A)(G)$  (if  $G$  proves  $A$ , then  $G$  implies  $A$ ).

We note that " $G$  proves  $A$ " has an inductive definition, and that gives us the immediate resources for demonstrating claims of the form "If  $G$  proves  $A$ , then ...". So our proof proceeds by induction.

- I. Basis. Show: If  $A$  is a member of  $G$ , then  $G$  implies  $A$ .
- II. Basis. Show: If  $A$  is an axiom, then  $G$  implies  $A$ .
- III. Inductive step (induction on  $n$ , the length of the proof):
  - a. Assume for arbitrary  $G$  and  $A$  that if  $G$  proves  $A$  in  $n$  or fewer steps, then  $G$  implies  $A$ .
  - b. For each possible application of a rule of inference at step  $n + 1$ , leading to a new theorem  $B$ , show that  $G$  implies  $B$ .

Notice that Basis Step II can be omitted for natural deduction systems because they have no axioms. When used, Step II involves showing that each of the axioms is a (semantic) logical truth.

The Basis steps demonstrate that the simplest provable sentences from  $G$  are also implied by  $G$ , for any  $G$ . (The proof is simple, since the semantic fact that a set implies any of its members, is also trivial.) The Inductive step will systematically cover all the further sentences that might be provable—by considering each case where we might reach a logical conclusion using an inference rule—and shows that if a new sentence is provable, it is

also logically implied. (For example, we might have a rule telling us that from " $A$ " we can derive " $A$  or  $B$ ". In III.a We assume that if  $A$  is provable it is implied. We also know that if  $A$  is provable then " $A$  or  $B$ " is provable. We have to show that then " $A$  or  $B$ " too is implied. We do so by appeal to the semantic definition and the assumption we just made.  $A$  is provable from  $G$ , we assume. So it is also implied by  $G$ . So any semantic valuation making all of  $G$  true makes  $A$  true. But any valuation making  $A$  true makes " $A$  or  $B$ " true, by the defined semantics for "or". So any valuation which makes all of  $G$  true makes " $A$  or  $B$ " true. So " $A$  or  $B$ " is implied.) Generally, the Inductive step will consist of a lengthy but simple case-by-case analysis of all the rules of inference, showing that each "preserves" semantic implication.

By the definition of provability, there are no sentences provable other than by being a member of  $G$ , an axiom, or following by a rule; so if all of those are semantically implied, the deduction calculus is sound.

## Sketch of completeness pr oof

(This is usually the much harder direction of proof.)

We adopt the same notational conventions as above.

We want to show: If  $G$  implies  $A$ , then  $G$  proves  $A$ . We proceed by contraposition: We show instead that if  $G$  does **not** prove  $A$  then  $G$  does **not** imply  $A$ .

- I.  $G$  does not prove  $A$ . (Assumption)
- II. If  $G$  does not prove  $A$ , then we can construct an (infinite) **Maximal Set**,  $G^*$ , which is a superset of  $G$  and which also does not prove  $A$ .
  - a. Place an ordering on all the sentences in the language (e.g., shortest first, and equally long ones in extended alphabetical ordering), and number them ( $E_1, E_2, \dots$ )
  - b. Define a series  $G_n$  of sets ( $G_0, G_1, \dots$ ) inductively:
    - i.  $G_0 = G$
    - ii. If  $G_k \cup \{E_{k+1}\}$  proves  $A$ , then  $G_{k+1} = G_k$
    - iii. If  $G_k \cup \{E_{k+1}\}$  does **not** prove  $A$ , then  $G_{k+1} = G_k \cup \{E_{k+1}\}$
  - c. Define  $G^*$  as the union of all the  $G_n$ . (That is,  $G^*$  is the set of all the sentences that are in any  $G_n$ .)
  - d. It can be easily shown that
    - i.  $G^*$  contains (is a superset of)  $G$  (by (b.i));
    - ii.  $G^*$  does not prove  $A$  (because if it proves  $A$  then some sentence was added to some  $G_n$  which caused it to prove  $A$ ; but this was ruled out by definition); and
    - iii.  $G^*$  is a Maximal Set with respect to  $A$ : If any more sentences whatever were added to  $G^*$ , it would prove  $A$ . (Because if it were possible to add any more sentences, they should have been added when they were encountered during the construction of the  $G_n$ , again by definition)
- III. If  $G^*$  is a Maximal Set with respect to  $A$ , then it is **truth-like**. This means that it contains  $C$  only if it does **not** contain  $\neg C$ ; If it contains  $C$  and contains "If  $C$  then  $B$ " then it also contains  $B$ ; and so forth.
- IV. If  $G^*$  is truth-like there is a  $G^*$ -Canonical valuation of the language: one that makes every sentence in  $G^*$  true and everything outside  $G^*$  false while still obeying the laws of semantic composition in the language.
- V. A  $G^*$ -canonical valuation will make our original set  $G$  all true, and make  $A$  false.
- VI. If there is a valuation on which  $G$  are true and  $A$  is false, then  $G$  does not (semantically) imply  $A$ .

QED

## Another outline for a completeness pr oof

If a formula is a tautology, then there is a truth table for it which shows that each valuation yields the value true for the formula. Consider such a valuation. By mathematical induction on the length of the subformulas, show that the truth or falsity of the subformula follows from the truth or falsity (as appropriate for the valuation) of each propositional variable in the subformula. Then combine the lines of the truth table together two at a time by using " $(P \text{ is true implies } S) \text{ implies } ((P \text{ is false implies } S) \text{ implies } S)$ ". Keep repeating this until all dependencies on propositional variables have been eliminated. The result is that we have proved the given tautology. Since every tautology is provable, the logic is complete.

## Interpretation of a truth-functional propositional calculus

An **interpretation of a truth-functional propositional calculus**  $\mathcal{P}$  is an assignment to each propositional symbol of  $\mathcal{P}$  of one or the other (but not both) of the truth values truth (**T**) and falsity (**F**), and an assignment to the connective symbols of  $\mathcal{P}$  of their usual truth-functional meanings. An interpretation of a truth-functional propositional calculus may also be expressed in terms of truth tables.<sup>[11]</sup>

For  $n$  distinct propositional symbols there are  $2^n$  distinct possible interpretations. For any particular symbol  $a$ , for example, there are  $2^1 = 2$  possible interpretations:

1.  $a$  is assigned **T**, or
2.  $a$  is assigned **F**.

For the pair  $a, b$  there are  $2^2 = 4$  possible interpretations:

1. both are assigned **T**,
2. both are assigned **F**,
3.  $a$  is assigned **T** and  $b$  is assigned **F**, or
4.  $a$  is assigned **F** and  $b$  is assigned **T**.<sup>[11]</sup>

Since  $\mathcal{P}$  has  $\aleph_0$ , that is, denumerably many propositional symbols, there are  $2^{\aleph_0} = \mathfrak{c}$ , and therefore uncountably many distinct possible interpretations of  $\mathcal{P}$ .<sup>[11]</sup>

## Interpretation of a sentence of truth-functional propositional logic

If  $\phi$  and  $\psi$  are formulas of  $\mathcal{P}$  and  $\mathcal{I}$  is an interpretation of  $\mathcal{P}$  then:

- A sentence of propositional logic is *true under an interpretation*  $\mathcal{I}$  iff  $\mathcal{I}$  assigns the truth value **T** to that sentence. If a sentence is true under an interpretation, then that interpretation is called a *model* of that sentence.
- $\phi$  is *false under an interpretation*  $\mathcal{I}$  iff  $\phi$  is not true under  $\mathcal{I}$ .<sup>[11]</sup>
- A sentence of propositional logic is *logically valid* if it is true under every interpretation

$\models \phi$  means that  $\phi$  is logically valid

- A sentence  $\psi$  of propositional logic is a *semantic consequence* of a sentence  $\phi$  iff there is no interpretation under which  $\phi$  is true and  $\psi$  is false.
- A sentence of propositional logic is *consistent* iff it is true under at least one interpretation. It is inconsistent if it is not consistent.

Some consequences of these definitions:

- For any given interpretation a given formula is either true or false.<sup>[11]</sup>
- No formula is both true and false under the same interpretation.<sup>[11]</sup>
- $\phi$  is false for a given interpretation iff  $\neg\phi$  is true for that interpretation; and  $\phi$  is true under an interpretation iff  $\neg\phi$  is false under that interpretation.<sup>[11]</sup>
- If  $\phi$  and  $(\phi \rightarrow \psi)$  are both true under a given interpretation, then  $\psi$  is true under that interpretation.<sup>[11]</sup>

- If  $\models_P \phi$  and  $\models_P (\phi \rightarrow \psi)$ , then  $\models_P \psi$ .<sup>[11]</sup>
- $\neg\phi$  is true under  $\mathcal{I}$  iff  $\phi$  is not true under  $\mathcal{I}$ .
- $(\phi \rightarrow \psi)$  is true under  $\mathcal{I}$  iff either  $\phi$  is not true under  $\mathcal{I}$  or  $\psi$  is true under  $\mathcal{I}$ .<sup>[11]</sup>
- A sentence  $\psi$  of propositional logic is a semantic consequence of a sentence  $\phi$  iff  $(\phi \rightarrow \psi)$  is logically valid, that is,  $\phi \models_P \psi$  iff  $\models_P (\phi \rightarrow \psi)$ .<sup>[11]</sup>

## Alternative calculus

It is possible to define another version of propositional calculus, which defines most of the syntax of the logical operators by means of axioms, and which uses only one inference rule.

### Axioms

Let  $\phi, \chi$ , and  $\psi$  stand for well-formed formulas. (The well-formed formulas themselves would not contain any Greek letters, but only capital Roman letters, connective operators, and parentheses.) Then the axioms are as follows:

Axioms		
Name	Axiom Schema	Description
THEN-1	$\phi \rightarrow (\chi \rightarrow \phi)$	Add hypothesis $\chi$ , implication introduction
THEN-2	$(\phi \rightarrow (\chi \rightarrow \psi)) \rightarrow ((\phi \rightarrow \chi) \rightarrow (\phi \rightarrow \psi))$	Distribute hypothesis $\phi$ over implication
AND-1	$\phi \wedge \chi \rightarrow \phi$	Eliminate conjunction
AND-2	$\phi \wedge \chi \rightarrow \chi$	
AND-3	$\phi \rightarrow (\chi \rightarrow (\phi \wedge \chi))$	Introduce conjunction
OR-1	$\phi \rightarrow \phi \vee \chi$	Introduce disjunction
OR-2	$\chi \rightarrow \phi \vee \chi$	
OR-3	$(\phi \rightarrow \psi) \rightarrow ((\chi \rightarrow \psi) \rightarrow (\phi \vee \chi \rightarrow \psi))$	Eliminate disjunction
NOT-1	$(\phi \rightarrow \chi) \rightarrow ((\phi \rightarrow \neg\chi) \rightarrow \neg\phi)$	Introduce negation
NOT-2	$\phi \rightarrow (\neg\phi \rightarrow \chi)$	Eliminate negation
NOT-3	$\phi \vee \neg\phi$	Excluded middle, classical logic
IFF-1	$(\phi \leftrightarrow \chi) \rightarrow (\phi \rightarrow \chi)$	Eliminate equivalence
IFF-2	$(\phi \leftrightarrow \chi) \rightarrow (\chi \rightarrow \phi)$	
IFF-3	$(\phi \rightarrow \chi) \rightarrow ((\chi \rightarrow \phi) \rightarrow (\phi \leftrightarrow \chi))$	Introduce equivalence

- Axiom **THEN-2** may be considered to be a "distributive property of implication with respect to implication."
- Axioms **AND-1** and **AND-2** correspond to "conjunction elimination". The relation between **AND-1** and **AND-2** reflects the commutativity of the conjunction operator.
- Axiom **AND-3** corresponds to "conjunction introduction."
- Axioms **OR-1** and **OR-2** correspond to "disjunction introduction." The relation between **OR-1** and **OR-2** reflects the commutativity of the disjunction operator.
- Axiom **NOT-1** corresponds to "reductio ad absurdum."
- Axiom **NOT-2** says that "anything can be deduced from a contradiction."
- Axiom **NOT-3** is called "tertium non datur" (Latin: "a third is not given") and reflects the semantic valuation of propositional formulas: a formula can have a truth-value of either true or false. There is no third truth-value, at least not in classical logic. Intuitionistic logicians do not accept the axiom **NOT-3**.

## Inference rule

The inference rule is modus ponens:

$$\phi, \phi \rightarrow \chi \vdash \chi.$$

## Meta-inference rule

Let a demonstration be represented by a sequence, with hypotheses to the left of the turnstile and the conclusion to the right of the turnstile. Then the deduction theorem can be stated as follows:

*If the sequence*

$$\phi_1, \phi_2, \dots, \phi_n, \chi \vdash \psi$$

*has been demonstrated, then it is also possible to demonstrate the sequence*

$$\phi_1, \phi_2, \dots, \phi_n \vdash \chi \rightarrow \psi.$$

This deduction theorem (DT) is not itself formulated with propositional calculus: it is not a theorem of propositional calculus, but a theorem about propositional calculus. In this sense, it is a meta-theorem, comparable to theorems about the soundness or completeness of propositional calculus.

On the other hand, DT is so useful for simplifying the syntactical proof process that it can be considered and used as another inference rule, accompanying modus ponens. In this sense, DT corresponds to the natural conditional proof inference rule which is part of the first version of propositional calculus introduced in this article.

The converse of DT is also valid:

*If the sequence*

$$\phi_1, \phi_2, \dots, \phi_n \vdash \chi \rightarrow \psi$$

*has been demonstrated, then it is also possible to demonstrate the sequence*

$$\phi_1, \phi_2, \dots, \phi_n, \chi \vdash \psi$$

in fact, the validity of the converse of DT is almost trivial compared to that of DT:

*If*

$$\phi_1, \dots, \phi_n \vdash \chi \rightarrow \psi$$

*then*

$$1: \phi_1, \dots, \phi_n, \chi \vdash \chi \rightarrow \psi$$

$$2: \phi_1, \dots, \phi_n, \chi \vdash \chi$$

*and from (1) and (2) can be deduced*

$$3: \phi_1, \dots, \phi_n, \chi \vdash \psi$$

*by means of modus ponens, Q.E.D.*

The converse of DT has powerful implications: it can be used to convert an axiom into an inference rule. For example, the axiom AND-1,

$$\vdash \phi \wedge \chi \rightarrow \phi$$

can be transformed by means of the converse of the deduction theorem into the inference rule

$$\phi \wedge \chi \vdash \phi$$

which is conjunction elimination, one of the ten inference rules used in the first version (in this article) of the propositional calculus.

## Example of a proof

The following is an example of a (syntactical) demonstration, involving only axioms **THEN-1** and **THEN-2**:

**Prove:**  $A \rightarrow A$  (Reflexivity of implication).

**Proof:**

$$1. (A \rightarrow ((B \rightarrow A) \rightarrow A)) \rightarrow ((A \rightarrow (B \rightarrow A)) \rightarrow (A \rightarrow A))$$

Axiom **THEN-2** with  $\phi = A, \chi = B \rightarrow A, \psi = A$

$$2. A \rightarrow ((B \rightarrow A) \rightarrow A)$$

Axiom **THEN-1** with  $\phi = A, \chi = B \rightarrow A$

$$3. (A \rightarrow (B \rightarrow A)) \rightarrow (A \rightarrow A)$$

From (1) and (2) by modus ponens.

$$4. A \rightarrow (B \rightarrow A)$$

Axiom **THEN-1** with  $\phi = A, \chi = B$

$$5. A \rightarrow A$$

From (3) and (4) by modus ponens.

## Equivalence to equational logics

The preceding alternative calculus is an example of a Hilbert-style deduction system. In the case of propositional systems the axioms are terms built with logical connectives and the only inference rule is modus ponens. Equational logic as standardly used informally in high school algebra is a different kind of calculus from Hilbert systems. Its theorems are equations and its inference rules express the properties of equality, namely that it is a congruence on terms that admits substitution.

Classical propositional calculus as described above is equivalent to Boolean algebra, while intuitionistic propositional calculus is equivalent to Heyting algebra. The equivalence is shown by translation in each direction of the theorems of the respective systems. Theorems  $\phi$  of classical or intuitionistic propositional calculus are translated as equations  $\phi = \mathbf{1}$  of Boolean or Heyting algebra respectively. Conversely theorems  $x = y$  of Boolean or Heyting algebra are translated as theorems  $(x \rightarrow y) \wedge (y \rightarrow x)$  of classical or intuitionistic calculus respectively, for which  $x \equiv y$  is a standard abbreviation. In the case of Boolean algebra  $x = y$  can also be translated as  $(x \wedge y) \vee (\neg x \wedge \neg y)$ , but this translation is incorrect intuitionistically.

In both Boolean and Heyting algebra, inequality  $x \leq y$  can be used in place of equality. The equality  $x = y$  is expressible as a pair of inequalities  $x \leq y$  and  $y \leq x$ . Conversely the inequality  $x \leq y$  is expressible as the equality  $x \wedge y = x$ , or as  $x \vee y = y$ . The significance of inequality for Hilbert-style systems is that it



corresponds to the latter's deduction or entailment symbol  $\vdash$ . An entailment

$$\phi_1, \phi_2, \dots, \phi_n \vdash \psi$$

is translated in the inequality version of the algebraic framework as

$$\phi_1 \wedge \phi_2 \wedge \dots \wedge \phi_n \leq \psi$$

Conversely the algebraic inequality  $x \leq y$  is translated as the entailment

$$x \vdash y.$$

The difference between implication  $x \rightarrow y$  and inequality or entailment  $x \leq y$  or  $x \vdash y$  is that the former is internal to the logic while the latter is external. Internal implication between two terms is another term of the same kind. Entailment as external implication between two terms expresses a metatruth outside the language of the logic, and is considered part of the metalanguage. Even when the logic under study is intuitionistic, entailment is ordinarily understood classically as two-valued: either the left side entails, or is less-or-equal to, the right side, or it is not.

Similar but more complex translations to and from algebraic logics are possible for natural deduction systems as described above and for the sequent calculus. The entailments of the latter can be interpreted as two-valued, but a more insightful interpretation is as a set, the elements of which can be understood as abstract proofs organized as the morphisms of a category. In this interpretation the cut rule of the sequent calculus corresponds to composition in the category. Boolean and Heyting algebras enter this picture as special categories having at most one morphism per homset, i.e., one proof per entailment, corresponding to the idea that existence of proofs is all that matters: any proof will do and there is no point in distinguishing them.

## Graphical calculi

It is possible to generalize the definition of a formal language from a set of finite sequences over a finite basis to include many other sets of mathematical structures, so long as they are built up by finitary means from finite materials. What's more, many of these families of formal structures are especially well-suited for use in logic.

For example, there are many families of graphs that are close enough analogues of formal languages that the concept of a calculus is quite easily and naturally extended to them. Indeed, many species of graphs arise as *parse graphs* in the syntactic analysis of the corresponding families of text structures. The exigencies of practical computation on formal languages frequently demand that text strings be converted into pointer structure renditions of parse graphs, simply as a matter of checking whether strings are well-formed formulas or not. Once this is done, there are many advantages to be gained from developing the graphical analogue of the calculus on strings. The mapping from strings to parse graphs is called *parsing* and the inverse mapping from parse graphs to strings is achieved by an operation that is called *traversing* the graph.

## Other logical calculi

Propositional calculus is about the simplest kind of logical calculus in current use. It can be extended in several ways. (Aristotelian "syllogistic" calculus, which is largely supplanted in modern logic, is in *some* ways simpler – but in other ways more complex – than propositional calculus.) The most immediate way to develop a more complex logical calculus is to introduce rules that are sensitive to more fine-grained details of the sentences being used.

First-order logic (a.k.a. first-order predicate logic) results when the "atomic sentences" of propositional logic are broken up into terms, variables, predicates, and quantifiers, all keeping the rules of propositional logic with some new ones introduced. (For example, from "All dogs are mammals" we may infer "If Rover is a dog then Rover is a mammal".) With the tools of first-order logic it is possible to formulate a number of theories, either with explicit axioms or by rules of inference, that can themselves be treated as logical calculi. Arithmetic is the

best known of these; others include set theory and mereology. Second-order logic and other higher-order logics are formal extensions of first-order logic. Thus, it makes sense to refer to propositional logic as "*zeroth-order logic*", when comparing it with these logics.

Modal logic also offers a variety of inferences that cannot be captured in propositional calculus. For example, from "Necessarily *p*" we may infer that *p*. From *p* we may infer "It is possible that *p*". The translation between modal logics and algebraic logics concerns classical and intuitionistic logics but with the introduction of a unary operator on Boolean or Heyting algebras, different from the Boolean operations, interpreting the possibility modality, and in the case of Heyting algebra a second operator interpreting necessity (for Boolean algebra this is redundant since necessity is the De Morgan dual of possibility). The first operator preserves 0 and disjunction while the second preserves 1 and conjunction.

Many-valued logics are those allowing sentences to have values other than *true* and *false*. (For example, *neither* and *both* are standard "extra values"; "continuum logic" allows each sentence to have any of an infinite number of "degrees of truth" between *true* and *false*.) These logics often require calculational devices quite distinct from propositional calculus. When the values form a Boolean algebra (which may have more than two or even infinitely many values), many-valued logic reduces to classical logic; many-valued logics are therefore only of independent interest when the values form an algebra that is not Boolean.

## Solvers

Finding solutions to propositional logic formulas is an NP-complete problem. However, practical methods exist (e.g., DPLL algorithm, 1962; Chaff algorithm, 2001) that are very fast for many useful cases. Recent work has extended the SAT solver algorithms to work with propositions containing arithmetic expressions; these are the SMT solvers.

## See also

### Higher logical levels

- First-order logic
- Second-order propositional logic
- Second-order logic
- Higher-order logic

### Related topics

- |                               |   |                         |
|-------------------------------|---|-------------------------|
| ■ Boolean algebra (logic)     | ■ Equational logic                      | ■ Operation             |
| ■ Boolean algebra (structure) | ■ Existential graph                     | ■ Paul of Venice        |
| ■ Boolean algebra topics      | ■ Frege's propositional calculus        | ■ Peirce's law          |
| ■ Boolean domain              | ■ Implicational propositional calculus  | ■ Peter of Spain        |
| ■ Boolean function            | ■ Intuitionistic propositional calculus | ■ Propositional formula |
| ■ Boolean-valued function     | ■ Jean Buridan                          | ■ Symmetric difference  |
| ■ Categorical logic           | ■ Laws of Form                          | ■ Truth function        |
| ■ Combinational logic         | ■ Logical graph                         | ■ Truth table           |
| ■ Combinatory logic           | ■ Logical NOR                           | ■ Walter Burley         |
| ■ Conceptual graph            | ■ Logical value                         | ■ William of Sherwood   |
| ■ Disjunctive syllogism       |   |                         |
| ■ Entitative graph            |   |                         |

## References

1. Bobzien, Susanne (1 January 2016). Zalta, Edward N., ed. *The Stanford Encyclopedia of Philosophy* (<http://plato.stanford.edu/archives/spr2016/entries/logic-ancient/>) – via Stanford Encyclopedia of Philosophy.
2. Marenbon, John (2007). *Medieval philosophy: an historical and philosophical introduction*. Routledge. p. 137.
3. Peckhaus, Volker (1 January 2014). Zalta, Edward N., ed. *The Stanford Encyclopedia of Philosophy* (<http://plato.stanford.edu/archives/spr2014/entries/leibniz-logic-influence/>) – via Stanford Encyclopedia of Philosophy.
4. Hurley, Patrick (2007). *A Concise Introduction to Logic 10th edition*. Wadsworth Publishing. p. 392.
5. Beth, Evert W.; "Semantic entailment and formal derivability", series: Mededelingen van de Koninklijke Nederlandse Akademie van Wetenschappen, Afdeling Letterkunde, Nieuwe Reeks, vol. 18, no. 13, Noord-Hollandsche Uitg. Mij., Amsterdam, 1955, pp. 309–42. Reprinted in Jaakko Intikka (ed.) *The Philosophy of Mathematics*, Oxford University Press, 1969
6. Truth in Frege (<http://frege.brown.edu/heck/pdf/unpublished/TruthInFrege.pdf>)
7. "Russell: the Journal of Bertrand Russell Studies" (<http://digitalcommons.mcmaster.ca/cgi/viewcontent.cgi?article=1219&context=russelljournal>).
8. Anellis, Irving H. (2012). "Peirce's Truth-functional Analysis and the Origin of the Truth Table". *History and Philosophy of Logic*. **33**: 87–97. doi:10.1080/01445340.2011.621702 (<https://doi.org/10.1080%2F01445340.2011.621702>).
9. Wernick, William (1942) "Complete Sets of Logical Functions," *Transactions of the American Mathematical Society* **51**, pp. 117–132.
10. Toida, Shunichi (2 August 2009). "Proof of Implications" ([http://www.cs.odu.edu/~toida/nerzic/content/logic/prop\\_logic/implications/implication\\_proof.html](http://www.cs.odu.edu/~toida/nerzic/content/logic/prop_logic/implications/implication_proof.html)). *CS381 Discrete Structures/Discrete Mathematics Web Course Material*. Department Of Computer Science, Old Dominion University. Retrieved 10 March 2010.
11. Hunter, Geoffrey (1971). *Metalogic: An Introduction to the Metatheory of Standard First-Order Logic*. University of California Pres. ISBN 0-520-02356-0.

## Further reading

- Brown, Frank Markham (2003), *Boolean Reasoning: The Logic of Boolean Equations*, 1st edition, Kluwer Academic Publishers, Norwell, MA. 2nd edition, Dover Publications, Mineola, NY.
- Chang, C.C. and Keisler, H.J. (1973), *Model Theory*, North-Holland, Amsterdam, Netherlands.
- Kohavi, Zvi (1978), *Switching and Finite Automata Theory*, 1st edition, McGraw–Hill, 1970. 2nd edition, McGraw–Hill, 1978.
- Korfhage, Robert R. (1974), *Discrete Computational Structures*, Academic Press, New York, NY.
- Lambek, J. and Scott, P.J. (1986), *Introduction to Higher Order Categorical Logic*, Cambridge University Press, Cambridge, UK.
- Mendelson, Elliot (1964), *Introduction to Mathematical Logic*, D. Van Nostrand Company.

## Related works

- Hofstadter, Douglas (1979). *Gödel, Escher, Bach: An Eternal Golden Braid*. Basic Books. ISBN 978-0-465-02656-2.

## External links

- Klement, Kevin C. (2006), "Propositional Logic", in James Fieser and Bradley Dowden (eds.), *Internet Encyclopedia of Philosophy*, Eprint.
- Formal Predicate Calculus, contains a systematic formal development along the lines of Alternative calculus
- *forall x: an introduction to formal logic*, by P.D. Magnus, covers formal semantics and proof theory for sentential logic.
- Chapter 2 / Propositional Logic from Logic In Action
- Propositional sequent calculus prover on Project Nayuki. (*note*: implication can be input in the form  $\neg X \mid Y$ , and a sequent can be a single formula prefixed with  $>$  and having no commas)

- This page was last edited on 22 September 2017, at 16:51.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.