

Truth function

In logic, a **truth function**^[1] is a function that accepts truth values as input and produces a unique truth value as output. In other words: The input and output of a truth function are all truth values; a truth function will always output exactly one truth value; and inputting the same truth value(s) will always output the same truth value. The typical example is in propositional logic, wherein a compound statement is constructed using individual statements connected by logical connectives; if the truth value of the compound statement is entirely determined by the truth value(s) of the constituent statement(s), the compound statement is called a **truth function**, and any logical connectives used are said to be **truth functional**.^[2]

Classical propositional logic is a truth-functional logic,^[3] in that every statement has exactly one truth value which is either true or false, and every logical connective is truth functional (with a correspondent truth table), thus every compound statement is a truth function.^[4] On the other hand, modal logic is non-truth-functional.

Contents

Overview

Table of binary truth functions

Functional completeness

Algebraic properties

Arity

Principle of compositionality

Computer science

See also

Notes

References

Further reading

Overview

A logical connective is truth-functional if the truth-value of a compound sentence is a function of the truth-value of its sub-sentences. A class of connectives is truth-functional if each of its members is. For example, the connective "and" is truth-functional since a sentence like "Apples are fruits and carrots are vegetables" is true *if, and only if* each of its sub-sentences "apples are fruits" and "carrots are vegetables" is true, and it is false otherwise. Some connectives of a natural language, such as English, are not truth-functional.

Connectives of the form "x believes that ..." are typical examples of connectives that are not truth-functional. If e.g. Mary mistakenly believes that Al Gore was President of the USA on April 20, 2000, but she does not believe that the moon is made of green cheese, then the sentence

"Mary believes that Al Gore was President of the USA on April 20, 2000"

is true while

"Mary believes that the moon is made of green cheese"

is false. In both cases, each component sentence (i.e. "Al Gore was president of the USA on April 20, 2000" and "the moon is made of green cheese") is false, but each compound sentence formed by prefixing the phrase "Mary believes that" differs in truth-value. That is, the truth-value of a sentence of the form "Mary believes that..." is not determined solely by the truth-value of its component sentence, and hence the (unary) connective (or simply *operator* since it is unary) is non-truth-functional.



The class of classical logic connectives (e.g. &, →) used in the construction of formulas is truth-functional. Their values for various truth-values as argument are usually given by truth tables. Truth-functional propositional calculus is a formal system whose formulae may be interpreted as either true or false.



Table of binary truth functions

In two-valued logic, there are sixteen possible truth functions, also called Boolean functions, of two inputs *P* and *Q*. Any of these functions corresponds to a truth table of a certain logical connective in classical logic, including several degenerate cases such as a function not depending on one or both of its arguments. Truth and falsehood are denoted as 1 and 0, respectively, in the following truth tables for sake of

brevity.

Contradiction/False				Tautology/True																																	
Notation	Equivalent formulas	Truth table	Venn diagram	Notation	Equivalent formulas	Truth table	Venn diagram																														
\perp "bottom"	$P \wedge \neg P$ Opq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>P</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr></table>		Q			0	1	0	0	0	P	1	0	1	0	0		\top "top"	$P \vee \neg P$ $\vee pq$	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>P</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>		Q			0	1	0	1	1	P	1	1	1	1	1	
	Q																																				
	0	1																																			
0	0	0																																			
P	1	0																																			
1	0	0																																			
	Q																																				
	0	1																																			
0	1	1																																			
P	1	1																																			
1	1	1																																			

Proposition P				Negation of P																											
Notation	Equivalent formulas	Truth table	Venn diagram	Notation	Equivalent formulas	Truth table	Venn diagram																								
P	p lpq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>P</td><td>1</td><td>1</td></tr></table>		Q			0	1	0	0	0	P	1	1		$\neg P$ $\neg P$	Np Fpq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>P</td><td>0</td><td>0</td></tr></table>		Q			0	1	0	1	1	P	0	0	
	Q																														
	0	1																													
0	0	0																													
P	1	1																													
	Q																														
	0	1																													
0	1	1																													
P	0	0																													

Proposition Q				Negation of Q																											
Notation	Equivalent formulas	Truth table	Venn diagram	Notation	Equivalent formulas	Truth table	Venn diagram																								
Q	q Hpq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>P 0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr></table>		Q			0	1	P 0	0	1	1	0	1		$\neg Q$ $\neg Q$	Nq Gpq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>P 0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>		Q			0	1	P 0	1	0	1	1	0	
	Q																														
	0	1																													
P 0	0	1																													
1	0	1																													
	Q																														
	0	1																													
P 0	1	0																													
1	1	0																													

Conjunction				Alternative denial																											
Notation	Equivalent formulas	Truth table	Venn diagram	Notation	Equivalent formulas	Truth table	Venn diagram																								
$P \wedge Q$ $P \& Q$ $P \cdot Q$ $P \text{ AND } Q$	$P \leftrightarrow \neg Q$ $\neg P \leftrightarrow Q$ $\neg P \downarrow \neg Q$ Kpq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>P</td><td>0</td><td>1</td></tr></table>		Q			0	1	0	0	0	P	0	1		$P \uparrow Q$ $P \mid Q$ $P \text{ NAND } Q$	$P \rightarrow \neg Q$ $\neg P \leftrightarrow Q$ $\neg P \vee \neg Q$ Dpq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>P</td><td>1</td><td>0</td></tr></table>		Q			0	1	0	1	1	P	1	0	
	Q																														
	0	1																													
0	0	0																													
P	0	1																													
	Q																														
	0	1																													
0	1	1																													
P	1	0																													

Disjunction				Joint denial																											
Notation	Equivalent formulas	Truth table	Venn diagram	Notation	Equivalent formulas	Truth table	Venn diagram																								
$P \vee Q$ $P \text{ OR } Q$	$P \leftrightarrow \neg Q$ $\neg P \leftrightarrow Q$ $\neg P \uparrow \neg Q$ $\neg(\neg P \wedge \neg Q)$ Apq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>P</td><td>1</td><td>1</td></tr></table>		Q			0	1	0	0	1	P	1	1		$P \downarrow Q$ $P \text{ NOR } Q$	$P \leftrightarrow \neg Q$ $\neg P \leftrightarrow Q$ $\neg P \wedge \neg Q$ Xpq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>P</td><td>0</td><td>0</td></tr></table>		Q			0	1	0	1	0	P	0	0	
	Q																														
	0	1																													
0	0	1																													
P	1	1																													
	Q																														
	0	1																													
0	1	0																													
P	0	0																													

Material nonimplication				Material implication																											
Notation	Equivalent formulas	Truth table	Venn diagram	Notation	Equivalent formulas	Truth table	Venn diagram																								
$P \nrightarrow Q$ $P \not\supset Q$ $P > Q$	$P \wedge \neg Q$ $\neg P \downarrow Q$ $\neg P \leftrightarrow \neg Q$ Lpq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>P</td><td>1</td><td>0</td></tr></table>		Q			0	1	0	0	0	P	1	0		$P \rightarrow Q$ $P \supset Q$ $P \leq Q$	$P \uparrow \neg Q$ $\neg P \vee Q$ $\neg P \leftrightarrow \neg Q$ Cpq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>P</td><td>0</td><td>1</td></tr></table>		Q			0	1	0	1	1	P	0	1	
	Q																														
	0	1																													
0	0	0																													
P	1	0																													
	Q																														
	0	1																													
0	1	1																													
P	0	1																													

Converse nonimplication				Converse implication																											
Notation	Equivalent formulas	Truth table	Venn diagram	Notation	Equivalent formulas	Truth table	Venn diagram																								
$P \nleftarrow Q$ $P \not\leftarrow Q$ $P < Q$	$P \downarrow \neg Q$ $\neg P \wedge Q$ $\neg P \nrightarrow \neg Q$ Mpq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>P</td><td>0</td><td>0</td></tr></table>		Q			0	1	0	0	1	P	0	0		$P \leftarrow Q$ $P \subset Q$ $P \geq Q$	$P \vee \neg Q$ $\neg P \uparrow Q$ $\neg P \rightarrow \neg Q$ Bpq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>P</td><td>1</td><td>1</td></tr></table>		Q			0	1	0	1	0	P	1	1	
	Q																														
	0	1																													
0	0	1																													
P	0	0																													
	Q																														
	0	1																													
0	1	0																													
P	1	1																													

Exclusive disjunction				Biconditional																											
Notation	Equivalent formulas	Truth table	Venn diagram	Notation	Equivalent formulas	Truth table	Venn diagram																								
$P \nleftrightarrow Q$ $P \neq Q$ $P \oplus Q$ $P \text{ XOR } Q$	$P \leftrightarrow \neg Q$ $\neg P \leftrightarrow Q$ $\neg P \leftrightarrow \neg Q$ Jpq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>P</td><td>1</td><td>0</td></tr></table>		Q			0	1	0	0	1	P	1	0		$P \leftrightarrow Q$ $P \equiv Q$ $P \text{ XNOR } Q$ $P \text{ IFF } Q$	$P \leftrightarrow \neg Q$ $\neg P \leftrightarrow Q$ $\neg P \leftrightarrow \neg Q$ Epq	<table><tr><td></td><td colspan="2">Q</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>P</td><td>0</td><td>1</td></tr></table>		Q			0	1	0	1	0	P	0	1	
	Q																														
	0	1																													
0	0	1																													
P	1	0																													
	Q																														
	0	1																													
0	1	0																													
P	0	1																													

Functional completeness

Because a function may be expressed as a composition, a truth-functional logical calculus does not need to have dedicated symbols for all of the above-mentioned functions to be functionally complete. This is expressed in a propositional calculus as logical equivalence of certain compound statements. For example, classical logic has $\neg P \vee Q$ equivalent to $P \rightarrow Q$. The conditional operator " \rightarrow " is therefore not necessary for a classical-based logical system if " \neg " (not) and " \vee " (or) are already in use.

A minimal set of operators that can express every statement expressible in the propositional calculus is called a *minimal functionally complete set*. A minimally complete set of operators is achieved by NAND alone $\{\uparrow\}$ and NOR alone $\{\downarrow\}$.

The following are the minimal functionally complete sets of operators whose arities do not exceed 2:^[5]

One element

$\{\uparrow\}, \{\downarrow\}$.

Two elements

$\{\vee, \neg\}, \{\wedge, \neg\}, \{\rightarrow, \neg\}, \{\leftarrow, \neg\}, \{\rightarrow, \perp\}, \{\leftarrow, \perp\}, \{\rightarrow, \leftrightarrow\}, \{\leftarrow, \leftrightarrow\}, \{\rightarrow, \leftrightarrow\}, \{\leftarrow, \leftrightarrow\}, \{\leftrightarrow, \neg\}, \{\leftrightarrow, \neg\}, \{\leftrightarrow, \top\}, \{\leftrightarrow, \top\}, \{\leftrightarrow, \leftrightarrow\}, \{\leftrightarrow, \leftrightarrow\}$.

Three elements

$\{\vee, \leftrightarrow, \perp\}, \{\vee, \leftrightarrow, \leftrightarrow\}, \{\vee, \leftrightarrow, \top\}, \{\wedge, \leftrightarrow, \perp\}, \{\wedge, \leftrightarrow, \leftrightarrow\}, \{\wedge, \leftrightarrow, \top\}$.

Algebraic properties

Some truth functions possess properties which may be expressed in the theorems containing the corresponding connective. Some of those properties that a binary truth function (or a corresponding logical connective) may have are:

- **associativity**: Within an expression containing two or more of the same associative connectives in a row, the order of the operations does not matter as long as the sequence of the operands is not changed.
- **commutativity**: The operands of the connective may be swapped without affecting the truth-value of the expression.
- **distributivity**: A connective denoted by \cdot distributes over another connective denoted by $+$, if $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ for all operands a, b, c .
- **idempotence**: Whenever the operands of the operation are the same, the connective gives the operand as the result. In other words, the operation is both truth-preserving and falsehood-preserving (see below).
- **absorption**: A pair of connectives \wedge, \vee satisfies the absorption law if $a \wedge (a \vee b) = a$ for all operands a, b .

A set of truth functions is functionally complete if and only if for each of the following five properties it contains at least one member lacking it:

- **monotonic**: If $f(a_1, \dots, a_n) \leq f(b_1, \dots, b_n)$ for all $a_1, \dots, a_n, b_1, \dots, b_n \in \{0,1\}$ such that $a_1 \leq b_1, a_2 \leq b_2, \dots, a_n \leq b_n$. E.g., $\vee, \wedge, \top, \perp$.
- **affine**: For each variable, changing its value either always or never changes the truth-value of the operation, for all fixed values of all other variables. E.g., $\neg, \leftrightarrow, \leftrightarrow, \top, \perp$.
- **self dual**: To read the truth-value assignments for the operation from top to bottom on its truth table is the same as taking the complement of reading it from bottom to top; in other words, $f(\neg a_1, \dots, \neg a_n) = \neg f(a_1, \dots, a_n)$. E.g., \neg .
- **truth-preserving**: The interpretation under which all variables are assigned a truth value of 'true' produces a truth value of 'true' as a result of these operations. E.g., $\vee, \wedge, \top, \rightarrow, \leftrightarrow, \subset$. (see validity)
- **falsehood-preserving**: The interpretation under which all variables are assigned a truth value of 'false' produces a truth value of 'false' as a result of these operations. E.g., $\vee, \wedge, \leftrightarrow, \perp, \not\subset, \not\supset$. (see validity)

Arity

A concrete function may be also referred to as an *operator*. In two-valued logic there are 2 nullary operators (constants), 4 unary operators, 16 binary operators, 256 ternary operators, and 2^{2^n} n -ary operators. In three-valued logic there are 3 nullary operators (constants), 27 unary operators, 19683 binary operators, 7625597484987 ternary operators, and 3^{3^n} n -ary operators. In k -valued logic, there are k nullary operators, k^k unary operators, k^{k^2} binary operators, k^{k^3} ternary operators, and k^{k^n} n -ary operators. An n -ary operator in k -valued logic is a function from $\mathbb{Z}_k^n \rightarrow \mathbb{Z}_k$. Therefore, the number of such operators is $|\mathbb{Z}_k|^{|\mathbb{Z}_k^n|} = k^{k^n}$, which is how the above numbers were derived.

However, some of the operators of a particular arity are actually degenerate forms that perform a lower-arity operation on some of the inputs and ignores the rest of the inputs. Out of the 256 ternary boolean operators cited above, $\binom{3}{2} \cdot 16 - \binom{3}{1} \cdot 4 + \binom{3}{0} \cdot 2$ of them are such degenerate forms of binary or lower-arity operators, using the inclusion-exclusion principle. The ternary operator $f(x, y, z) = \neg x$ is one such operator which is actually a unary operator applied to one input, and ignoring the other two inputs.

"Not" is a unary operator, it takes a single term ($\neg P$). The rest are binary operators, taking two terms to make a compound statement ($P \wedge Q, P \vee Q, P \rightarrow Q, P \leftrightarrow Q$).

The set of logical operators Ω may be 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, Ω_j is the set of operator symbols of arity j .

In the more familiar propositional calculi, Ω is typically partitioned as follows:

$$\text{nullary operators: } \Omega_0 = \{\perp, \top\}$$

$$\text{unary operators: } \Omega_1 = \{\neg\}$$

$$\text{binary operators: } \Omega_2 \subseteq \{\wedge, \vee, \rightarrow, \leftrightarrow\}$$

Principle of compositionality

Instead of using truth tables, logical connective symbols can be interpreted by means of an interpretation function and a functionally complete set of truth-functions (Gamut 1991), as detailed by the principle of compositionality of meaning. Let I be an interpretation function, let Φ, Ψ be any two sentences and let the truth function f_{nand} be defined as:

- $f_{\text{nand}}(\top, \top) = \text{F}; f_{\text{nand}}(\top, \text{F}) = f_{\text{nand}}(\text{F}, \top) = f_{\text{nand}}(\text{F}, \text{F}) = \text{T}$

Then, for convenience, f_{not} , f_{or} f_{and} and so on are defined by means of f_{nand} :

- $f_{\text{not}}(x) = f_{\text{nand}}(x, x)$
- $f_{\text{or}}(x, y) = f_{\text{nand}}(f_{\text{not}}(x), f_{\text{not}}(y))$
- $f_{\text{and}}(x, y) = f_{\text{not}}(f_{\text{nand}}(x, y))$

or, alternatively f_{not} , f_{or} f_{and} and so on are defined directly:

- $f_{\text{not}}(\text{T}) = \text{F}; f_{\text{not}}(\text{F}) = \text{T};$
- $f_{\text{or}}(\text{T}, \text{T}) = f_{\text{or}}(\text{T}, \text{F}) = f_{\text{or}}(\text{F}, \text{T}) = \text{T}; f_{\text{or}}(\text{F}, \text{F}) = \text{F}$
- $f_{\text{and}}(\text{T}, \text{T}) = \text{T}; f_{\text{and}}(\text{T}, \text{F}) = f_{\text{and}}(\text{F}, \text{T}) = f_{\text{and}}(\text{F}, \text{F}) = \text{F}$

Then

- $I(\sim) = I(\neg) = f_{\text{not}}$
- $I(\&) = I(\wedge) = f_{\text{and}}$
- $I(\vee) = I(\vee) = f_{\text{or}}$
- $I(\sim\Phi) = I(\neg\Phi) = I(\neg)(I(\Phi)) = f_{\text{not}}(I(\Phi))$
- $I(\Phi \wedge \Psi) = I(\wedge)(I(\Phi), I(\Psi)) = f_{\text{and}}(I(\Phi), I(\Psi))$

etc.

Thus if S is a sentence that is a string of symbols consisting of logical symbols $v_1 \dots v_n$ representing logical connectives, and non-logical symbols $c_1 \dots c_n$, then if and only if $I(v_1) \dots I(v_n)$ have been provided interpreting v_1 to v_n by means of f_{nand} (or any other set of functional complete truth-functions) then the truth-value of $I(\mathbf{s})$ is determined entirely by the truth-values of $c_1 \dots c_n$, i.e. of $I(c_1) \dots I(c_n)$. In other words, as expected and required, S is true or false only under an interpretation of all its non-logical symbols.

Computer science

Logical operators are implemented as logic gates in digital circuits. Practically all digital circuits (the major exception is DRAM) are built up from NAND, NOR, NOT, and transmission gates. NAND and NOR gates with 3 or more inputs rather than the usual 2 inputs are fairly common, although they are logically equivalent to a cascade of 2-input gates. All other operators are implemented by breaking them down into a logically equivalent combination of 2 or more of the above logic gates.

The "logical equivalence" of "NAND alone", "NOR alone", and "NOT and AND" is similar to Turing equivalence.

The fact that all truth functions can be expressed with NOR alone is demonstrated by the Apollo guidance computer.

See also

- Bertrand Russell and Alfred North Whitehead, *Principia Mathematica*, 2nd edition
- Ludwig Wittgenstein, *Tractatus Logico-Philosophicus*, Proposition 5.101
- Bitwise operation
- Binary function
- Boolean domain
- Boolean function
- Boolean logic
- Boolean-valued function
- List of Boolean algebra topics
- Logical connective
- Logical constant
- Modal operator
- Propositional calculus
- Truth-functional propositional logic
- Truth table
- Truth values

Notes

1. Roy T. Cook (2009). *A Dictionary of Philosophical Logic*, p. 294: Truth Function. Edinburgh University Press.
2. Roy T. Cook (2009). *A Dictionary of Philosophical Logic*, p. 295: Truth Functional. Edinburgh University Press.
3. Internet Encyclopedia of Philosophy: Propositional Logic (<http://www.iep.utm.edu/prop-log/>), by Kevin C. Klement
4. Roy T. Cook (2009). *A Dictionary of Philosophical Logic*, p. 47: Classical Logic. Edinburgh University Press.
5. Wernick, William (1942) "Complete Sets of Logical Functions," *Transactions of the American Mathematical Society* 51: 117–32. In his list on the last page of the article, Wernick does not distinguish between \leftarrow and \rightarrow , or between \nwarrow and \nearrow .

References

- *This article incorporates material from TruthFunction on PlanetMath, which is licensed under the Creative Commons Attribution/Share-Alike License.*

Further reading

- Józef Maria Bocheński (1959), *A Précis of Mathematical Logic*, translated from the French and German versions by Otto Bird, Dordrecht, South Holland: D. Reidel.
 - Alonzo Church (1944), *Introduction to Mathematical Logic*, Princeton, NJ: Princeton University Press. See the Introduction for a history of the truth function concept.
-

Retrieved from "https://en.wikipedia.org/w/index.php?title=Truth_function&oldid=1024970605"

This page was last edited on 25 May 2021, at 00:57 (UTC).

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.