Logical conjunction

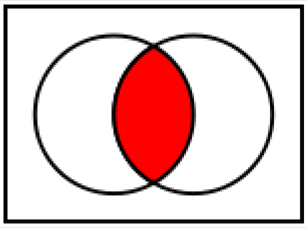
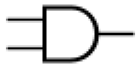
In logic, mathematics and linguistics, And ($\wedge$) is the truth-functional operator of **logical conjunction**; the *and* of a set of operands is true if and only if *all* of its operands are true. The logical connective that represents this operator is typically written as $\wedge$ or $\cdot$.[1][2][3]

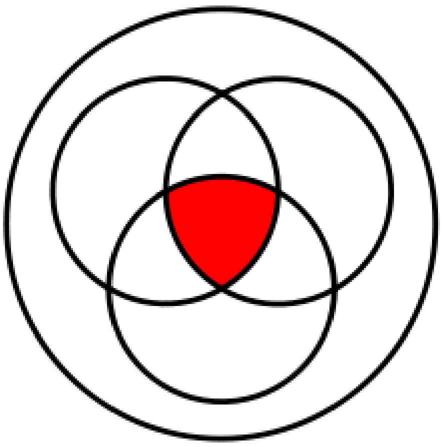
$A \land B$ is true if and only if A is true and B is true.

An operand of a conjunction is a **conjunct**.

Beyond logic, the term "conjunction" also refers to similar concepts in other fields:

- In natural language, the denotation of expressions such as English "and".
- In programming languages, the short-circuit and control structure.
- In set theory, intersection.
- In lattice theory, logical conjunction (greatest lower bound).
- In predicate logic, universal quantification.

Logical conjunction	
AND	
	
Definition	xy
Truth table	(0001)
Logic gate	
Normal forms	
Disjunctive	xy
Conjunctive	xy
Zhegalkin polynomial	xy
Post's lattices	
0-preserving	yes
1-preserving	yes
Monotone	no
Affine	no



Venn diagram of $A \land B \land C$

Contents

Notation

And is usually denoted by an infix operator: in mathematics and logic, it is denoted by $\wedge$,[1][3] **&** or $\times$; in electronics, $\cdot$; and in programming languages, **&**, **&&**, or **and**. In Jan Łukasiewicz's prefix notation for logic, the operator is **K**, for Polish *koniunkcja*.[4]

Definition

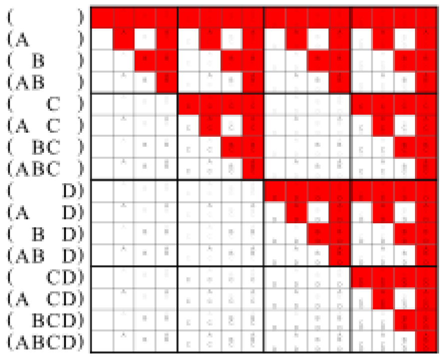
Logical conjunction is an operation on two logical values, typically the values of two propositions, that produces a value of *true* if and only if both of its operands are true.[2][3]

The conjunctive identity is true, which is to say that AND-ing an expression with true will never change the value of the expression. In keeping with the concept of vacuous truth, when conjunction is defined as an operator or function of arbitrary arity, the empty conjunction (AND-ing over an empty set of operands) is often defined as having the result true.

Truth table

The truth table of $A \wedge B$:[2][3]

A	B	$A \wedge B$
T	T	T
T	F	F
F	T	F
F	F	F



Conjunctions of the arguments on the left — The true bits form a Sierpinski triangle.

Defined by other operators

In systems where logical conjunction is not a primitive, it may be defined as[5]

$$A \wedge B = \neg(A \to \neg B)$$

or

$$A \wedge B = \neg(\neg A \vee \neg B).$$

Introduction and elimination rules

As a rule of inference, conjunction introduction is a classically valid, simple argument form. The argument form has two premises, *A* and *B*. Intuitively, it permits the inference of their conjunction.

> *A*,
> *B*.
> Therefore, *A* and *B*.

or in logical operator notation:

A,
B
$\vdash A \land B$

Here is an example of an argument that fits the form *conjunction introduction*:

> Bob likes apples.
> Bob likes oranges.
> Therefore, Bob likes apples and Bob likes oranges.

Conjunction elimination is another classically valid, simple argument form. Intuitively, it permits the inference from any conjunction of either element of that conjunction.

> *A* and *B*.
> Therefore, *A*.

...or alternatively,

> *A* and *B*.
> Therefore, *B*.

In logical operator notation:

$A \land B$
$\vdash A$

...or alternatively,

$A \land B$
$\vdash B$

Negation

Definition

A conjunction $A \land B$ is be proven false by establishing either $\neg A$ or $\neg B$. In terms of the object language, this reads

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

This formula can be seen as a special case of

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

when C is a false proposition.

Other proof strategies

If A implies $\neg B$, then both $\neg A$ as well as A prove the conjunction false:

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

In other words, a conjunction can actually be proven false just by knowing about the relation of its conjuncts, and not necessary about their truth values.

This formula can be seen as a special case of

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

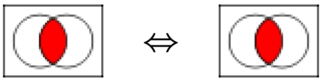
when C is a false proposition.

Either of the above are constructively valid proofs by contradiction.

Properties

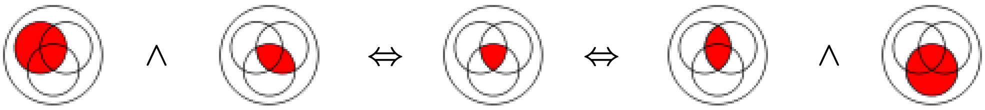
commutativity: yes

$A \land B \quad \Leftrightarrow \quad B \land A$



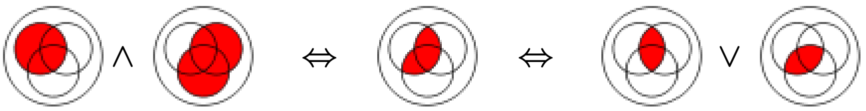
associativity: yes

$A \quad \land \quad (B \land C) \quad \Leftrightarrow \qquad (A \land B) \quad \land \quad C$



distributivity: with various operations, especially with *or*

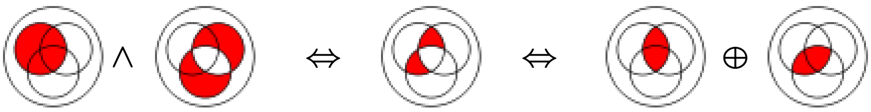
$A \quad \land (B \lor C) \quad \Leftrightarrow \qquad (A \land B) \lor (A \land C)$



others

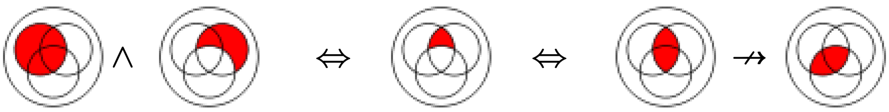
with exclusive or:

$A \quad \land (B \oplus C) \quad \Leftrightarrow \qquad (A \land B) \oplus (A \land C)$



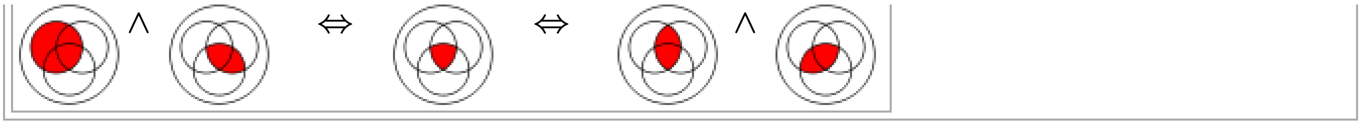
with material nonimplication:

$A \quad \land (B \nrightarrow C) \quad \Leftrightarrow \qquad (A \land B) \nrightarrow (A \land C)$

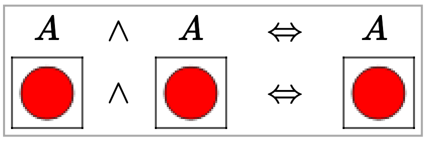


with itself:

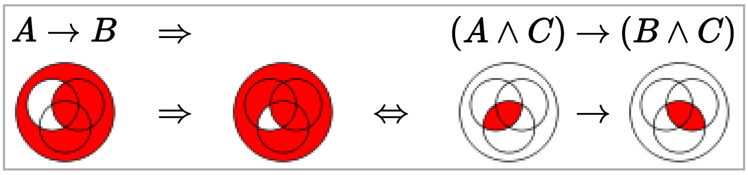
$A \quad \land (B \land C) \quad \Leftrightarrow \qquad (A \land B) \land (A \land C)$



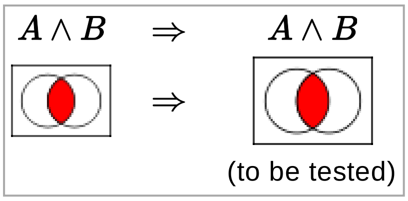
idempotency: yes



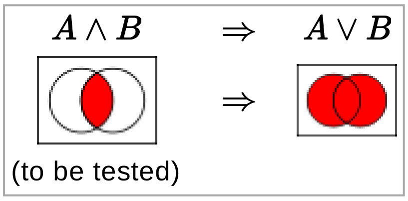
monotonicity: yes



truth-preserving: yes
When all inputs are true, the output is true.



falsehood-preserving: yes
When all inputs are false, the output is false.

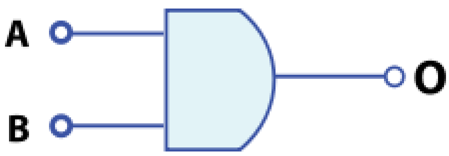


Walsh spectrum: (1,-1,-1,1)

Nonlinearity: 1 (the function is bent)

If using binary values for true (1) and false (0), then *logical conjunction* works exactly like normal arithmetic multiplication.

Applications in computer engineering



AND logic gate

In high-level computer programming and digital electronics, logical conjunction is commonly represented by an infix operator, usually as a keyword such as "`AND`", an algebraic multiplication, or the ampersand symbol & (sometimes doubled as in &&). Many languages also provide short-circuit control structures corresponding to logical conjunction.

Logical conjunction is often used for bitwise operations, where `0` corresponds to false and `1` to true:

- `0 AND 0 = 0`,
- `0 AND 1 = 0`,
- `1 AND 0 = 0`,
- `1 AND 1 = 1`.

The operation can also be applied to two binary words viewed as bitstrings of equal length, by taking the bitwise AND of each pair of bits at corresponding positions. For example:

- `11000110 AND 10100011 = 10000010`.

This can be used to select part of a bitstring using a bit mask. For example, `10011101 AND 00001000 = 00001000` extracts the fifth bit of an 8-bit bitstring.

In computer networking, bit masks are used to derive the network address of a subnet within an existing network from a given IP address, by ANDing the IP address and the subnet mask.

Logical conjunction "`AND`" is also used in SQL operations to form database queries.

The Curry–Howard correspondence relates logical conjunction to product types.

Set-theoretic correspondence

The membership of an element of an intersection set in set theory is defined in terms of a logical conjunction: $x \in A \cap B$ if and only if $(x \in A) \wedge (x \in B)$. Through this correspondence, set-theoretic intersection shares several properties with logical conjunction, such as associativity, commutativity and idempotence.

Natural language

As with other notions formalized in mathematical logic, the logical conjunction *and* is related to, but not the same as, the grammatical conjunction *and* in natural languages.

English "and" has properties not captured by logical conjunction. For example, "and" sometimes implies order having the sense of "then". For example, "They got married and had a child" in common discourse means that the marriage came before the child.

The word "and" can also imply a partition of a thing into parts, as "The American flag is red, white, and blue." Here, it is not meant that the flag is *at once* red, white, and blue, but rather that it has a part of each color.

See also

- And-inverter graph
- AND gate
- Bitwise AND
- Boolean algebra (logic)
- Boolean algebra topics
- Boolean conjunctive query
- Boolean domain
- Boolean function
- Boolean-valued function
- Conjunction elimination
- De Morgan's laws
- First-order logic
- Fréchet inequalities
- Grammatical conjunction

- Logical disjunction
- Logical negation
- Logical graph
- Operation
- Peano–Russell notation
- Propositional calculus

References

1. "Comprehensive List of Logic Symbols" (https://mathvault.ca/hub/higher-math/math-symbols/logic-symbols/). *Math Vault*. 2020-04-06. Retrieved 2020-09-02.
2. "Conjunction, Negation, and Disjunction" (https://philosophy.lander.edu/logic/conjunct.html). *philosophy.lander.edu*. Retrieved 2020-09-02.
3. "2.2: Conjunctions and Disjunctions" (https://math.libretexts.org/Courses/Monroe_Community_College/MTH_220_Discrete_Math/2%3A_Logic/2.2%3A_Conjunctions_and_Disjunctions). *Mathematics LibreTexts*. 2019-08-13. Retrieved 2020-09-02.
4. Józef Maria Bocheński (1959), *A Précis of Mathematical Logic*, translated by Otto Bird from the French and German editions, Dordrecht, South Holland: D. Reidel, passim.
5. Smith, Peter. "Types of proof system" (http://www.logicmatters.net/resources/pdfs/ProofSystems.pdf) (PDF). p. 4.

External links

- "Conjunction" (https://www.encyclopediaofmath.org/index.php?title=Conjunction), *Encyclopedia of Mathematics*, EMS Press, 2001 [1994]
- Wolfram MathWorld: Conjunction (http://mathworld.wolfram.com/Conjunction.html)
- "Property and truth table of AND propositions" (https://web.archive.org/web/20170506173821/http://www.math.hawaii.edu/~ramsey/Logic/And.html). Archived from the original (http://www.math.hawaii.edu/~ramsey/Logic/And.html) on May 6, 2017.