Logical Equivalence and Negation

When two propositions say the same thing, and how to say the opposite

Last modified: 2026-11-06 16:28:53 (PST)

Source

Adapted from Kameron Decker Harris's Fall 2026 lecture notes "Logic 3", which are typed transcriptions of Scott Wehrwein's Spring 2025 handwritten lecture notes. The truth table for the distributive law of "and" over "or" comes from those notes; the exercises are new.

Logical equivalence

Exercise 1 (A rule with a redundant part)**.** A server grants access to a user who is an administrator, or who is an administrator and is on the office network. Let P be "the user is an administrator" and Q be "the user is on the office network".
Make a truth table for $P \vee (P \wedge Q)$. In which rows does it have the same truth value as P?

Solution

Solution 1.

P	Q	$P \wedge Q$	$P \vee (P \wedge Q)$
T	T	T	T
T	F	F	T
F	T	F	F
F	F	F	F

The last column matches the P column in every row, so the rule grants access exactly to the administrators, and the part about the office network changes nothing.

Definition 1 (Logically equivalent)**.** Two propositions A and B, each built from some or all of the propositions $P_1, \ldots, P_n$ with logical connectives, are **logically equivalent** (Wikipedia contributors 2026b), written $A \equiv B$, when they have the same truth value in every row of a truth table on $P_1, \ldots, P_n$.

Example 1 (An equivalence)**.** By Exercise 1, $P \vee (P \wedge Q)$ is logically equivalent to P: the truth table of $P \vee (P \wedge Q)$ has the same column as P, in a table with columns for both P and Q.

Remark 1 (Notation for logical equivalence). $A \equiv B$, read "A is logically equivalent to B", is a claim about A and B, not a new proposition built from them with a logical connective (Definition: Logical connective). Some books write $A \Leftrightarrow B$ or $A \iff B$ for it instead, which is easy to confuse with the biconditional (Definition: If and only if); these notes use $\equiv$.
Logical equivalence can be chained: if $A \equiv B$ and $B \equiv C$, then $A \equiv C$, since in each row A has the value of B, which has the value of C.

Exercise 2 (Not both)**.** A continuous integration server runs two test suites. Let P be "the unit tests pass" and Q be "the integration tests pass".
Is "it is not the case that both suites pass", $\neg(P \wedge Q)$, logically equivalent to "neither suite passes", $\neg P \wedge \neg Q$?

Solution

Solution 2.

P	Q	$P \wedge Q$	$\neg(P \wedge Q)$	$\neg P$	$\neg Q$	$\neg P \wedge \neg Q$
T	T	T	F	F	F	F
T	F	F	T	F	T	F
F	T	F	T	T	F	F
F	F	F	T	T	T	T

No. The two columns differ in the second and third rows, where exactly one suite passes: then it is true that not both pass, but false that neither passes.

Definition 2 (Tautology)**.** A **tautology** is a proposition that is true in every row of its truth table.

Example 2 (A tautology)**.** $P \vee \neg P$ is a tautology (Definition 2):

P	$\neg P$	$P \vee \neg P$
T	F	T
F	T	T

Definition 3 (Contradiction)**.** A **contradiction** is a proposition that is false in every row of its truth table.

Example 3 (A contradiction)**.** $P \wedge \neg P$ is a contradiction (Definition 3):

P	$\neg P$	$P \wedge \neg P$
T	F	F
F	T	F

Remark 2 (Contradictions in proofs). A proof by contradiction is named for this kind of proposition: it assumes the negation of what it proves, and derives a contradiction (Definition 3), such as $Q \wedge \neg Q$.

Theorem 1 (Equivalence is a tautological biconditional)**.** *For propositions A and B, A and B are logically equivalent exactly when $A \Leftrightarrow B$ is a tautology.*

Proof

Proof. By the definition of if and only if, $A \Leftrightarrow B$ is true in a row of the truth table exactly when A and B have the same truth value in that row. So $A \Leftrightarrow B$ is true in every row exactly when A and B have the same truth value in every row, which is the definition of $A \equiv B$. □

Example 4 (An equivalence as a tautology)**.** Since $P \vee (P \wedge Q) \equiv P$ (Example 1), Theorem 1 says that $(P \vee (P \wedge Q)) \Leftrightarrow P$ is a tautology.

Exercise 3 (Does "or" distribute over "and"?)**.** In arithmetic, multiplication distributes over addition: $a(b+c) = ab + ac$. Use a truth table to decide whether "or" distributes over "and": is $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$?

Solution

Solution 3.

A	B	C	$B \wedge C$	$A \vee (B \wedge C)$	$A \vee B$	$A \vee C$	$(A \vee B) \wedge (A \vee C)$
T	T	T	T	T	T	T	T
T	T	F	F	T	T	T	T
T	F	T	F	T	T	T	T
T	F	F	F	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	F	T	F	F
F	F	T	F	F	F	T	F
F	F	F	F	F	F	F	F

Yes: the fifth and last columns match in every row. Addition does not distribute over multiplication in arithmetic: $1 + 2 \cdot 3 = 7$, but $(1 + 2)(1 + 3) = 12$. But "or" distributes over "and".

Theorem 2 (Distributive laws for propositions)**.** *For propositions A, B and C:*

$$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$$
$$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$$

Proof

Proof. By the definitions of and and or, the truth table for the first law is:

A	B	C	$A \wedge (B \vee C)$	$(A \wedge B) \vee (A \wedge C)$
T	T	T	T	T
T	T	F	T	T
T	F	T	T	T
T	F	F	F	F
F	T	T	F	F
F	T	F	F	F
F	F	T	F	F

F	F	F	F	F

and for the second law:

A	B	C	$A \vee (B \wedge C)$	$(A \vee B) \wedge (A \vee C)$
T	T	T	T	T
T	T	F	T	T
T	F	T	T	T
T	F	F	T	T
F	T	T	T	T
F	T	F	F	F
F	F	T	F	F
F	F	F	F	F

In each table the last two columns match in every row, so each law is a logical equivalence. □

Example 5 (Distributing "and")**.** For an integer x, "x is even, and $x > 0$ or $x < 0$" is logically equivalent, by law 1 of Theorem 2, to "x is even and $x > 0$, or x is even and $x < 0$".

Remark 3 (Checking an equivalence in Racket). A truth table is a loop over every combination of truth values, so a program can check an equivalence. In Racket, `for*/and` runs its body for every combination of `p` and `q`, and returns `#f` as soon as the body returns `#f`, and `#t` if it never does:

```
(for*/and ([p '(#t #f)]
           [q '(#t #f)])
  (equal? (or p (and p q)) p))
```

This returns `#t`, as Exercise 1 found. Replacing the body with `(equal? (not (and p q)) (and (not p) (not q)))` returns `#f`, as Exercise 2 found.

Negation

Exercise 4 (When is the door locked?)**.** A door opens when the badge is valid, the PIN is correct, and it is a weekday. Let B be "the badge is valid", N be "the PIN is correct" and W be "it is a weekday", so the door stays locked when $\neg(B \wedge N \wedge W)$ is true. Describe when the door stays locked with a sentence that puts "not" only in front of B, N or W, never in front of a compound proposition.

Solution

Solution 4. The door stays locked when the badge is not valid, the PIN is not correct, or it is not a weekday: $\neg B \vee \neg N \vee \neg W$. $B \wedge N \wedge W$ is false exactly when at least one of B, N and W is false.

Theorem 3 (Double negation)**.** *For a proposition P,*

$$\neg\neg P \equiv P.$$

Proof

Proof. By the definition of not, the truth table is:

P	$\neg P$	$\neg\neg P$
T	F	T
F	T	F

The first and last columns match in both rows, so the two sides are logically equivalent. □

Example 6 (Not unreadable)**.** "It is not the case that the file is not readable" means "the file is readable" (Theorem 3).

Theorem 4 (Replacing a part by an equivalent part)**.** *Let A and B be logically equivalent propositions. Let F be a proposition built from A and other propositions with logical connectives, and let F' be F with A replaced by B. Then $F \equiv F'$.*

Proof

Proof. The truth value of F in a row of its truth table is computed from the truth values of its parts in that row, using the truth table of each connective. F' is computed the same way, except that B takes the place of A. A and B have the same truth value in every row, so each step of the computation gives the same value for F' as for F. So F and F' have the same truth value in every row. □

Example 7 (Replacing a part)**.** Since $P \vee (P \wedge Q) \equiv P$ (Example 1), Theorem 4 gives $R \wedge (P \vee (P \wedge Q)) \equiv R \wedge P$ and $\neg(P \vee (P \wedge Q)) \equiv \neg P$.

Example 8 (Simplifying part of a proposition)**.**
By double negation, $\neg\neg Q \equiv Q$. Take F to be $\neg P \vee \neg\neg Q$, with $A = \neg\neg Q$ and $B = Q$. By replacing a part by an equivalent part,

$$\neg P \vee \neg\neg Q \equiv \neg P \vee Q.$$

Theorem 5 (De Morgan's laws for propositions)**.** *For any propositions P and Q (Wikipedia contributors 2026a):*

$$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$$
$$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$$

Proof

Proof. Each side of each law is built from P and Q, so by Definition 1 it is enough to compare the two sides in each of the four rows of a truth table:

P	Q	$\neg(P \wedge Q)$	$\neg P \vee \neg Q$	$\neg(P \vee Q)$	$\neg P \wedge \neg Q$
T	T	F	F	F	F
T	F	T	T	F	F
F	T	T	T	F	F
F	F	T	T	T	T

The third and fourth columns agree in every row, which proves the first law, and the fifth and sixth columns agree in every row, which proves the second. □

Example 9 (Neither coordinate positive)**.** For real numbers x and y, by the second of De Morgan's laws, "not ($x > 0$ or $y > 0$)" is logically equivalent to "$x \leq 0$ and $y \leq 0$".

Exercise 5 (A broken deployment rule)**.** A team's rule is "if the tests pass, then the build is deployed". Let P be "the tests pass" and Q be "the build is deployed". The rule is broken on the days when $\neg(P \Rightarrow Q)$ is true.
Write $\neg(P \Rightarrow Q)$ without $\Rightarrow$, and in English.

Solution

Solution 5. The rule is broken on a day when the tests pass and the build is not deployed: $P \wedge \neg Q$. On a day when the tests fail, the rule is not broken, whether or not the build is deployed.
"If the tests pass, then the build is not deployed", $P \Rightarrow \neg Q$, is not the negation: on a day when the tests fail, it is true, and so is the rule.

Theorem 6 (Negating a conditional)**.** *For propositions P and Q,*

$$\neg(P \Rightarrow Q) \equiv P \wedge \neg Q.$$

Proof

Proof. By the definition of implies, $P \Rightarrow Q$ is false exactly when P is true and Q is false. So $\neg(P \Rightarrow Q)$ is true exactly in that row of the truth table. By the definition of and, $P \wedge \neg Q$ is true exactly when P is true and $\neg Q$ is true. That is the same row. □

Example 10 (When a search claim fails)**.** The negation of "if the list is sorted, then binary search finds the key" is "the list is sorted and binary search does not find the key" (Theorem 6).

Corollary 1 (A conditional without the arrow)**.** *For propositions P and Q,*

$$(P \Rightarrow Q) \equiv (\neg P \vee Q).$$

Proof

Proof.

1. By double negation, $P \Rightarrow Q \equiv \neg\neg(P \Rightarrow Q)$.

2. By negating a conditional and replacing a part by an equivalent part, $\neg\neg(P \Rightarrow Q) \equiv \neg(P \wedge \neg Q)$.
3. By the first of De Morgan's laws, $\neg(P \wedge \neg Q) \equiv \neg P \vee \neg\neg Q$.
4. By double negation and replacing a part by an equivalent part again, $\neg P \vee \neg\neg Q \equiv \neg P \vee Q$.

□

Example 11 (A rule as an "or")**.** By Corollary 1, "if the tests pass, then the build is deployed" is logically equivalent to "the tests fail or the build is deployed".

Negating quantified propositions

Exercise 6 (Not every test passes)**.** Let T be the set of tests in a test suite, and let $P(t)$ be "test t passes". The report "every test passes" is $\forall t \in T, P(t)$.
Write the negation of this report in English and in symbols, without putting $\neg$ in front of a quantifier.

Solution

Solution 6. "Some test fails": $\exists t \in T, \neg P(t)$. "Every test passes" is false exactly when at least one test does not pass.
"Every test fails", $\forall t \in T, \neg P(t)$, is not the negation: when some tests pass and some fail, both it and "every test passes" are false.

Theorem 7 (De Morgan's laws for quantifiers)**.** *For a set S and a predicate $P(x)$ whose domain of discourse is S:*

$$\neg(\forall x \in S, P(x)) \equiv \exists x \in S, \neg P(x)$$
$$\neg(\exists x \in S, P(x)) \equiv \forall x \in S, \neg P(x)$$

Proof

Proof.

1. By the definition of the universal quantifier, $\forall x \in S, P(x)$ is false exactly when $P(x)$ is false for at least one x in S. By the definition of the existential quantifier, that is exactly when $\exists x \in S, \neg P(x)$ is true.

2. By the definition of the existential quantifier, $\exists x \in S, P(x)$ is false exactly when $P(x)$ is false for every x in S. By the definition of the universal quantifier, that is exactly when $\forall x \in S, \neg P(x)$ is true.

□

Example 12 (No natural square root of 2)**.** By part 2 of Theorem 7, $\neg(\exists n \in \mathbb{N}, n^2 = 2)$ has the same truth value as $\forall n \in \mathbb{N}, n^2 \neq 2$.

Remark 4 (De Morgan's laws for quantifiers). Theorem 7 is often written $\neg\forall x, P(x) \equiv \exists x, \neg P(x)$ and $\neg\exists x, P(x) \equiv \forall x, \neg P(x)$. These notes say "has the same truth value" instead, because $P(x)$ is a predicate, not a proposition, so Definition 1, which compares truth tables, does not apply.
On a finite set $S = \{a, b\}$, $\forall x \in S, P(x)$ has the same truth value as $P(a) \wedge P(b)$, and $\exists x \in S, P(x)$ has the same truth value as $P(a) \vee P(b)$, so Theorem 7 is De Morgan's laws (Theorem 5) applied to $P(a)$ and $P(b)$.
Racket shows the same pattern on a list: `(not (andmap even? (list 2 4 5)))` and `(ormap odd? (list 2 4 5))` both return `#t`.

Remark 5 (Pushing a negation inward). To negate a proposition built with $\neg$, $\wedge$, $\vee$, $\Rightarrow$ and quantifiers, move the $\neg$ inward one step at a time:

- across a quantifier, with Theorem 7, which swaps $\forall$ and $\exists$;
- across $\wedge$ or $\vee$, with De Morgan's laws (Theorem 5);
- across $\Rightarrow$, with Theorem 6;
- across another $\neg$, with double negation (Theorem 3).

Stop when every $\neg$ sits in front of a proposition with no connective or quantifier inside it. The result is easier to read and to check than a $\neg$ in front of the whole proposition.

Exercise 7 (Completing a palindrome)**.** Let W be the set of words made of the letters a and b, including the empty word, and write st for word s followed by word t. A palindrome is a word that reads the same forwards and backwards, such as abba.
Write the negation of

$$\forall s \in W,\ \exists t \in W,\ st \text{ is a palindrome}$$

without putting $\neg$ in front of a quantifier.

Solution

Solution 7. Apply Theorem 7 twice, from the outside in. First, the negation has the truth value of

$$\exists s \in W,\ \neg(\exists t \in W,\ st \text{ is a palindrome}).$$

For each word s, Theorem 7 says that $\neg(\exists t \in W,\ st$ is a palindrome) has the truth value of $\forall t \in W,\ st$ is not a palindrome. These have the same truth value for every s, so the $\exists s$ propositions built from them do too (Definition: Existential quantifier), and the negation has the truth value of

$$\exists s \in W,\ \forall t \in W,\ st \text{ is not a palindrome.}$$

In English: some word cannot be made into a palindrome by adding letters at its end.

Exercise 8 (Which one is true?)**.** Which is true: $\forall s \in W,\ \exists t \in W,\ st$ is a palindrome, from Exercise 7, or its negation?

Solution

Solution 8. The original is true, so its negation is false. For any word s, let t be s written backwards; then st reads the same backwards. For example, for $s = \text{aab}$, $t = \text{baa}$ and $st = \text{aabbaa}$.

References

Wikipedia contributors. 2026a. *De Morgan's Laws — Wikipedia, the Free Encyclopedia.* https://en.wikipedia.org/wiki/De_Morgan%27s_laws.

Wikipedia contributors. 2026b. *Logical Equivalence — Wikipedia, the Free Encyclopedia.* https://en.wikipedia.org/wiki/Logical_equivalence.