

Rules of Inference and Logic Proofs

A **proof** is an argument from **hypotheses** (assumptions) to a **conclusion**. Each step of the argument follows the laws of logic. In mathematics, a statement is not accepted as valid or correct unless it is accompanied by a proof. This insistence on proof is one of the things that sets mathematics apart from other subjects.

Writing proofs is difficult; there are no procedures which you can follow which will guarantee success. *The patterns which proofs follow are complicated, and there are a lot of them.* You can't expect to do proofs by following rules, memorizing formulas, or looking at a few examples in a book.

For this reason, I'll start by discussing **logic proofs**. Since they are more highly patterned than most proofs, they are a good place to start. They'll be written in **column format**, with each step justified by a **rule of inference**. Most of the rules of inference will come from tautologies. Since a tautology is a statement which is "always true", it makes sense to use them in drawing conclusions.

Like most proofs, logic proofs usually begin with **premises** — statements that you're allowed to assume. The **conclusion** is the statement that you need to prove. The idea is to operate on the premises using rules of inference until you arrive at the conclusion.

Rule of Premises. You may write down a **premise** at any point in a proof.

The second rule of inference is one that you'll use in most logic proofs. It is sometimes called **modus ponendo ponens**, but I'll use a shorter name.

Modus Ponens. If you know P and $P \rightarrow Q$, you may write down Q .

In the rules of inference, it's understood that symbols like " P " and " Q " may be replaced by *any* statements, including compound statements. I'll say more about this later.

Here is a simple proof using modus ponens:

- | | | |
|----|-------------------|---------------------|
| 1. | P | Premise |
| 2. | $P \rightarrow Q$ | Premise |
| 3. | Q | Modus ponens (1, 2) |

I'll write logic proofs in 3 columns. The statements in logic proofs are numbered so that you can refer to them, and the numbers go in the first column. The actual statements go in the second column. The third column contains your justification for writing down the statement.

Thus, statements 1 (P) and 2 ($P \rightarrow Q$) are premises, so the rule of premises allows me to write them down. Modus ponens says that if I've already written down P and $P \rightarrow Q$ — on *any* earlier lines, in either order — then I may write down Q . I did that in line 3, citing the rule ("Modus ponens") and the lines (1 and 2) which contained the statements I needed to apply modus ponens.

As I noted, the " P " and " Q " in the modus ponens rule can actually stand for compound statements — they don't have to be "single letters". For example:

- | | | |
|----|--------------------------------------|---------------------|
| 1. | $(A \vee \neg B) \rightarrow \neg C$ | Premise |
| 2. | $\neg D$ | Premise |
| 3. | $A \vee \neg B$ | Premise |
| 4. | $\neg C$ | Modus ponens (1, 3) |

There are several things to notice here. First, $A \vee \neg B$ is taking the place of P in the modus ponens rule, and $\neg C$ is taking the place of Q . That is, $A \vee \neg B$ and $\neg C$ are *compound statements* which are substituted for " P " and " Q " in modus ponens.

Notice also that the if-then statement $(A \vee \neg B) \rightarrow \neg C$ is listed first and the "if"-part $\neg C$ is listed second. It doesn't matter which one has been written down first, and long as both pieces have already been written down, you may apply modus ponens.

Finally, the statement $\neg D$ didn't take part in the modus ponens step. Perhaps this is part of a bigger proof, and $\neg D$ will be used later. The fact that it came between the two modus ponens pieces doesn't make a difference.

As usual in math, you have to be sure to apply rules *exactly*. For example, this is *not* a valid use of modus ponens:

1. P Premise
2. $(P \wedge Q) \rightarrow R$ Premise
3. R INVALID modus ponens!

Do you see why? To use modus ponens on the if-then statement $(P \wedge Q) \rightarrow R$, you need the “if”-part, which is $P \wedge Q$. You only have P , which is just *part of* the “if”-part. That's not good enough.

Double Negation. In any statement, you may substitute P for $\neg\neg P$ or $\neg\neg P$ for P (and write down the new statement).

For example, in this case I'm applying double negation with P replaced by $A \rightarrow \neg C$:

1. $\neg[\neg(A \rightarrow \neg C)]$ Premise
2. $A \rightarrow \neg C$ Double negation (1)

You can also apply double negation “inside” another statement:

1. $A \rightarrow \neg\neg B$ Premise
2. $A \rightarrow B$ Double negation (1)

Double negation comes up often enough that, we'll bend the rules and allow it to be used without doing so as a separate step or mentioning it explicitly. I'll demonstrate this in the examples for some of the other rules of inference.

Modus Tollens. If you know $\neg Q$ and $P \rightarrow Q$, you may write down $\neg P$.

This is a simple example of modus tollens:

1. $\neg Q$ Premise
2. $P \rightarrow Q$ Premise
3. $\neg P$ Modus tollens (1, 2)

In the next example, I'm applying modus tollens with P replaced by C and Q replaced by $A \rightarrow B$:

1. $\neg(A \rightarrow B)$ Premise
2. $C \rightarrow (A \rightarrow B)$ Premise
3. $\neg C$ Modus tollens (1, 2)

The last example shows how you're allowed to “suppress” double negation steps. Do you see how this was done? If I wrote the double negation step explicitly, it would look like this:

1. $\neg(A \rightarrow B)$ Premise
2. $C \rightarrow (A \rightarrow B)$ Premise
3. $C \rightarrow \neg\neg(A \rightarrow B)$ Double negation (2)
4. $\neg C$ Modus tollens (1, 3)

When you apply modus tollens to an if-then statement, be sure that you have the *negation* of the “then”-part. The following derivation is incorrect:

1. Q Premise
2. $P \rightarrow Q$ Premise
3. $\neg P$ INVALID modus tollens!

To use modus tollens, you need $\neg Q$, not Q .

This is also incorrect:

1. Q Premise
2. $P \rightarrow Q$ Premise
3. P INVALID modus tollens!

This looks like modus ponens, but backwards. There is no rule that allows you to do this: The deduction is invalid.

Disjunctive Syllogism. If you know $\neg P$ and $P \vee Q$, you may write down Q .

Here's a simple example of disjunctive syllogism:

1. $\neg P$ Premise
2. $P \vee Q$ Premise
3. Q Disjunctive syllogism (1, 2)

In the next example, I'm applying disjunctive syllogism with $A \vee B$ replacing P and D replacing Q in the rule:

1. $\neg(A \vee B)$ Premise
2. $(A \vee B) \vee D$ Premise
3. D Disjunctive syllogism (1, 2)

In the next example, notice that P is the same as $\neg\neg P$, so it's the negation of $\neg P$.

1. P Premise
2. $\neg P \vee (Q \rightarrow R)$ Premise
3. $Q \rightarrow R$ Disjunctive syllogism (1, 2)

This is another case where I'm skipping a double negation step. Without skipping the step, the proof would look like this:

1. P Premise
2. $\neg P \vee (Q \rightarrow R)$ Premise
3. $\neg\neg P$ Double negation (1)
4. $Q \rightarrow R$ Disjunctive syllogism (2, 3)

DeMorgan's Law. In any statement, you may substitute:

1. $\neg(P \vee Q)$ for $\neg P \wedge \neg Q$.
2. $\neg P \wedge \neg Q$ for $\neg(P \vee Q)$.
3. $\neg(P \wedge Q)$ for $\neg P \vee \neg Q$.
4. $\neg P \vee \neg Q$ for $\neg(P \wedge Q)$.

As usual, after you've substituted, you write down the new statement.

DeMorgan's Law tells you how to distribute \neg across \wedge or \vee , or how to factor \neg out of \wedge or \vee . To distribute, you attach \neg to each term, then change \wedge to \vee or \vee to \wedge . To factor, you factor \neg out of each term, then change \wedge to \vee or \vee to \wedge .

Note that it only applies (directly) to "or" and "and". We'll see how to negate an "if-then" later.

Here's DeMorgan applied to an "or" statement:

1. $\neg(\neg P \vee \neg Q)$ Premise
2. $P \wedge Q$ DeMorgan (1)

Notice that a literal application of DeMorgan would have given $\neg\neg P \wedge \neg\neg Q$. I changed this to $P \wedge Q$, once again suppressing the double negation step.

Conditional Disjunction. If you know $P \rightarrow Q$, you may write down $\neg P \vee Q$.

If you know $\neg P \vee Q$, you may write down $P \rightarrow Q$.

Here's the first direction:

1. $\neg P \vee Q$ Premise
2. $P \rightarrow Q$ Conditional disjunction (1)

And here's the second:

1. $P \rightarrow Q$ Premise
2. $\neg P \vee Q$ Conditional disjunction (1)

The first direction is key: Conditional disjunction allows you to convert "if-then" statements into "or" statements.

We'll see below that biconditional statements can be converted into pairs of conditional statements. Together with conditional disjunction, this allows us in principle to reduce the five logical connectives to three (negation, conjunction, disjunction). But DeMorgan allows us to change conjunctions to disjunctions (or vice versa), so in principle we could do everything with just "or" and "not". The reason we don't is that it would make our statements much longer: The use of the other connectives is like shorthand that saves us writing.

In additional, we can solve the problem of negating a conditional that we mentioned earlier.

1. $\neg(P \rightarrow Q)$ Premise
2. $\neg(\neg P \vee Q)$ Conditional disjunction (1)
3. $P \wedge \neg Q$ DeMorgan (2)

We've derived a new rule! Let's write it down.

Negating a Conditional. If you know $\neg(P \rightarrow Q)$, you may write down $P \wedge \neg Q$.

If you know $P \wedge \neg Q$, you may write down $\neg(P \rightarrow Q)$.

The first direction is more useful than the second. Personally, I tend to forget this rule and just apply conditional disjunction and DeMorgan when I need to negate a conditional. But you may use this if you wish.

Constructing a Conjunction. If you know P and Q , you may write down $P \wedge Q$.

Think about this to ensure that it makes sense to you. If $P \wedge Q$ is true, you're saying that P is true and that Q is true. So on the other hand, you need both P true and Q true in order to say that $P \wedge Q$ is true.

Here's an example. Notice that I put the pieces in parentheses to group them after constructing the conjunction.

1. $P \leftrightarrow Q$ Premise
2. $P \vee Q$ Premise
3. $(P \leftrightarrow Q) \wedge (P \vee Q)$ Constructing a conjunction (1, 2)

Rule of Syllogism. If you know $P \rightarrow Q$ and $Q \rightarrow R$, then you may write down $P \rightarrow R$.

The Rule of Syllogism says that you can "chain" syllogisms together. For example:

1. $P \rightarrow (Q \vee R)$ Premise
2. $(Q \vee R) \rightarrow \neg S$ Premise
3. $P \rightarrow \neg S$ Rule of syllogism (1, 2)

Definition of Biconditional. If you know $P \leftrightarrow Q$, you may write down $P \rightarrow Q$ and you may write down $Q \rightarrow P$. If you know $P \rightarrow Q$ and $Q \rightarrow P$, you may write down $P \leftrightarrow Q$.

First, a simple example:

1. $P \leftrightarrow Q$ Premise
2. $Q \rightarrow P$ Definition of biconditional (1)

By the way, a standard mistake is to apply modus ponens to a biconditional (" \leftrightarrow "). Modus ponens applies to conditionals (" \rightarrow "). So this isn't valid:

1. $P \leftrightarrow Q$ Premise
2. Q Premise
3. P INVALID - Not modus ponens!

With the same premises, here's what you need to do:

1. $P \leftrightarrow Q$ Premise
2. Q Premise
3. $Q \rightarrow P$ Definition of biconditional (1)
4. P Modus ponens (2, 3) \square

Decomposing a Conjunction. If you know $P \wedge Q$, you may write down P and you may write down Q .

This rule says that you can decompose a conjunction to get the individual pieces:

1. $P \wedge (Q \leftrightarrow \neg R)$ Premise
2. P Decomposing a conjunction (1)
2. $Q \leftrightarrow \neg R$ Decomposing a conjunction (1)

Note that you *can't* decompose a disjunction!

1. $P \vee Q$ Premise
2. P INVALID!

What's wrong with this? If you know that $P \vee Q$ is true, you know that *one* of P or Q must be true. The problem is that you don't know *which one* is true, so you can't *assume* that either one *in particular* is true.

On the other hand, it is easy to *construct* disjunctions.

Constructing a Disjunction. If you know P , and Q is *any* statement, you may write down $P \vee Q$.

This says that if you know a statement, you can "or" it with *any other statement* to construct a disjunction.

1. P Premise
2. $P \vee$ "Calvin sleeps with a night light." Constructing a disjunction (1)

Notice that it doesn't matter what the other statement is! Once you know that P is true, any "or" statement with P must be true: An "or" statement is true if at least one of the pieces is true.

The next two rules are stated for completeness. They are easy enough that, as with double negation, we'll allow you to use them without a separate step or explicit mention.

Commutativity of Conjunctions. In any statement, you may substitute $P \wedge Q$ for $Q \wedge P$ (and write down the new statement).

Commutativity of Disjunctions. In any statement, you may substitute $P \vee Q$ for $Q \vee P$ (and write down the new statement).

Here is commutativity for a conjunction:

1. $P \wedge \neg Q$ Premise
2. $\neg Q \wedge P$ Commutativity (1)

Here is commutativity for a disjunction:

1. $(Q \rightarrow P) \vee R$ Premise
2. $R \vee (Q \rightarrow P)$ Commutativity (1) \square

Before I give some examples of logic proofs, I'll explain where the rules of inference come from. You've probably noticed that the rules of inference correspond to tautologies. In fact, you can start with tautologies and use a small number of **simple inference rules** to derive all the other inference rules.

Three of the simple rules were stated above: The Rule of Premises, Modus Ponens, and Constructing a Conjunction. Here are two others. We've been using them without mention in some of our examples if you look closely.

Equivalence You may replace a statement by another that is logically equivalent. (Recall that P and Q are **logically equivalent** if and only if $P \leftrightarrow Q$ is a tautology.)

For instance, since P and $\neg\neg P$ are logically equivalent, you can replace P with $\neg\neg P$ or $\neg\neg P$ with P . This is Double Negation. As I mentioned, we're saving time by not writing out this step.

Substitution. You may take a known tautology and substitute for the simple statements.

This amounts to my remark at the start: In the statement of a rule of inference, the simple statements ("P", "Q", and so on) may stand for compound statements. "May stand for" is the same as saying "may be substituted with". We've been doing this without explicit mention.

Here's an example. The Disjunctive Syllogism tautology says

$$(\neg P \wedge (P \vee Q)) \rightarrow Q.$$

Suppose you have $\neg(A \wedge D)$ and $(A \wedge D) \vee C$ as premises. Here's how you'd apply the simple inference rules and the Disjunctive Syllogism tautology:

- | | | |
|----|---|--------------------------------------|
| 1. | $\neg(A \wedge D)$ | Premise |
| 2. | $(A \wedge D) \vee C$ | Premise |
| 3. | $\neg(A \wedge D) \wedge ((A \wedge D) \vee C)$ | Constructing a conjunction (1, 3) |
| 4. | $\neg(A \wedge D) \wedge ((A \wedge D) \vee C) \rightarrow C$ | Substitution (Disjunctive Syllogism) |
| 5. | C | Modus ponens (3, 4) |

Notice that I used four of the five simple inference rules: the Rule of Premises, Modus Ponens, Constructing a Conjunction, and Substitution. In line 4, I used the Disjunctive Syllogism tautology $(\neg P \wedge (P \vee Q)) \rightarrow Q$ by substituting

$$A \wedge D \text{ for } P \text{ and } C \text{ for } Q.$$

(Some people use the word "instantiation" for this kind of substitution.)

The advantage of this approach is that you have only five simple rules of inference. The disadvantage is that the proofs tend to be longer. With the approach I'll use, Disjunctive Syllogism is a rule of inference, and the proof is:

1. $\neg(A \wedge D)$ Premise
2. $(A \wedge D) \vee C$ Premise
3. C Disjunctive syllogism (1,2)

The approach I'm using turns the tautologies into rules of inference beforehand, and for that reason you won't need to use the Equivalence and Substitution rules that often. But you *are* allowed to use them, and here's where they might be useful. Suppose you're writing a proof and you'd like to use a rule of inference —

but it wasn't mentioned above. Write down the corresponding logical statement, then construct the truth table to prove it's a tautology (if it isn't on the tautology list). Then use Substitution to use your new tautology.

If you go to the market for pizza, one approach is to buy the ingredients — the crust, the sauce, the cheese, the toppings — take everything home, assemble the pizza, and put it in the oven. Using tautologies together with the five simple inference rules is like making the pizza from scratch. But you could also go to the market and buy a frozen pizza, take it home, and put it in the oven. Using lots of rules of inference that come from tautologies — the approach I'll use — is like getting the frozen pizza.

Here are some proofs which use the rules of inference. In each case, some **premises** — statements that are assumed to be true — are given, as well as a statement to prove. A proof consists of using the rules of inference to produce the statement to prove from the premises.

Example. Premises: $\begin{cases} \neg A \rightarrow (C \wedge D) \\ A \rightarrow B \\ \neg B \end{cases}$.

Prove: C .

- | | | |
|----|-----------------------------------|-------------------------------|
| 1. | $A \rightarrow B$ | Premise |
| 2. | $\neg B$ | Premise |
| 3. | $\neg A$ | Modus tollens (1,2) |
| 4. | $\neg A \rightarrow (C \wedge D)$ | Premise |
| 5. | $C \wedge D$ | Modus ponens (3,4) |
| 6. | C | Decomposing a conjunction (5) |

It is one thing to see that the steps are correct; it's another thing to see how you would think of making them. I used my **experience with logical forms** combined with **working backward**.

I'm trying to prove C , so I looked for statements containing C . Only the first premise contains C . I saw that C was contained in the consequent of an if-then; by modus ponens, the consequent follows if you know the antecedent.

The "if"-part of the first premise is $\neg A$. Hence, I looked for another premise containing A or $\neg A$. The only other premise containing A is the second one. In this case, A appears as the "if"-part of an if-then. By modus tollens, $\neg A$ follows from the negation of the "then"-part B . But I noticed that I had $\neg B$ as a premise, so all that remained was to run all those steps forward and write everything up.

It's common in logic proofs (and in math proofs in general) to work backwards from what you want on scratch paper, then write the real proof forward. The second part is important!

In order to do this, I needed to have a hands-on familiarity with the basic rules of inference: Modus ponens, modus tollens, and so forth. You'll acquire this familiarity by writing logic proofs.

You also have to concentrate in order to remember where you are as you work backwards. You may need to scribble stuff on scratch paper to avoid getting confused. Keep practicing, and you'll find that this gets easier with time. \square

Example. Premises: $\begin{cases} P \wedge Q \\ P \rightarrow \neg(Q \wedge R) \\ S \rightarrow R \end{cases}$.

Prove: $\neg S$.

- | | | |
|----|----------------------------------|-------------------------------|
| 1. | $P \wedge Q$ | Premise |
| 2. | P | Decomposing a conjunction (1) |
| 3. | Q | Decomposing a conjunction (1) |
| 4. | $P \rightarrow \neg(Q \wedge R)$ | Premise |
| 5. | $\neg(Q \wedge R)$ | Modus ponens (3,4) |
| 6. | $\neg Q \vee \neg R$ | DeMorgan (5) |
| 7. | $\neg R$ | Disjunctive syllogism (3,6) |
| 8. | $S \rightarrow R$ | Premise |
| 9. | $\neg S$ | Modus tollens (7,8) \square |

Example. Premises: $\begin{cases} \neg(A \vee B) \rightarrow C \\ \neg A \\ \neg C \end{cases}$.

Prove: B .

- | | | |
|----|--------------------------------|-----------------------------|
| 1. | $\neg(A \vee B) \rightarrow C$ | Premise |
| 2. | $\neg C$ | Premise |
| 3. | $A \vee B$ | Modus tollens (1,2) |
| 4. | $\neg A$ | Premise |
| 5. | B | Disjunctive syllogism (3,4) |

Notice that in step 3, I would have gotten $\neg\neg(A \vee B)$. I omitted the double negation step, as I have in other examples. \square