

1.7 Proof by Cases

Breaking a complicated proof into cases and proving each case separately is a common, useful proof strategy. Here's an amusing example.

Let's agree that given any two people, either they have met or not. If every pair of people in a group has met, we'll call the group a *club*. If every pair of people in a group has not met, we'll call it a group of *strangers*.

Theorem. *Every collection of 6 people includes a club of 3 people or a group of 3 strangers.*

Proof. The proof is by case analysis⁵. Let x denote one of the six people. There are two cases:

1. Among 5 other people besides x , at least 3 have met x .
2. Among the 5 other people, at least 3 have not met x .

Now, we have to be sure that at least one of these two cases must hold⁶, but that's easy: we've split the 5 people into two groups, those who have shaken hands with x and those who have not, so one of the groups must have at least half the people.

Case 1: Suppose that at least 3 people did meet x .

This case splits into two subcases:

⁵Describing your approach at the outset helps orient the reader.

⁶Part of a case analysis argument is showing that you've covered all the cases. This is often obvious, because the two cases are of the form " P " and "not P ." However, the situation above is not stated quite so simply.

Case 1.1: No pair among those people met each other. Then these people are a group of at least 3 strangers. The theorem holds in this subcase.

Case 1.2: Some pair among those people have met each other. Then that pair, together with x , form a club of 3 people. So the theorem holds in this subcase.

This implies that the theorem holds in Case 1.

Case 2: Suppose that at least 3 people did not meet x .

This case also splits into two subcases:

Case 2.1: Every pair among those people met each other. Then these people are a club of at least 3 people. So the theorem holds in this subcase.

Case 2.2: Some pair among those people have not met each other. Then that pair, together with x , form a group of at least 3 strangers. So the theorem holds in this subcase.

This implies that the theorem also holds in Case 2, and therefore holds in all cases. ■

notice the difference in each pair of subcases ...

1.8 Proof by Contradiction

In a *proof by contradiction*, or *indirect proof*, you show that if a proposition were false, then some false fact would be true. Since a false fact by definition can't be true, the proposition must be true.

Proof by contradiction is always a viable approach. However, as the name suggests, indirect proofs can be a little convoluted, so direct proofs are generally preferable when they are available.

Method: In order to prove a proposition P by contradiction:

1. Write, “We use proof by contradiction.”
2. Write, “Suppose P is false.”
3. Deduce something known to be false (a logical contradiction).
4. Write, “This is a contradiction. Therefore, P must be true.”

Example

We’ll prove by contradiction that $\sqrt{2}$ is irrational. Remember that a number is rational if it is equal to a ratio of integers—for example, $3.5 = 7/2$ and $0.1111\ldots = 1/9$ are rational numbers.

Theorem 1.8.1. $\sqrt{2}$ is irrational.

Proof. We use proof by contradiction. Suppose the claim is false, and $\sqrt{2}$ is rational. Then we can write $\sqrt{2}$ as a fraction n/d in lowest terms.

Squaring both sides gives $2 = n^2/d^2$ and so $2d^2 = n^2$. This implies that n is a multiple of 2 (see Problems 1.10 and 1.11). Therefore n^2 must be a multiple of 4. But since $2d^2 = n^2$, we know $2d^2$ is a multiple of 4 and so d^2 is a multiple of 2. This implies that d is a multiple of 2.

So, the numerator and denominator have 2 as a common factor, which contradicts the fact that n/d is in lowest terms. Thus, $\sqrt{2}$ must be irrational. ■

1.9 Good Proofs in Practice

One purpose of a proof is to establish the truth of an assertion with absolute certainty, and mechanically checkable proofs of enormous length or complexity can accomplish this. But humanly intelligible proofs are the only ones that help someone understand the subject. Mathematicians generally agree that important mathematical results can’t be fully understood until their proofs are understood. That is why proofs are an important part of the curriculum.

To be understandable and helpful, more is required of a proof than just logical correctness: a good proof must also be clear. Correctness and clarity usually go together; a well-written proof is more likely to be a correct proof, since mistakes are harder to hide.

In practice, the notion of proof is a moving target. Proofs in a professional research journal are generally unintelligible to all but a few experts who know all the terminology and prior results used in the proof. Conversely, proofs in the first weeks of a beginning course like 6.042 would be regarded as tediously long-winded by a professional mathematician. In fact, what we accept as a good proof later in the term will be different from what we consider good proofs in the first couple of weeks of 6.042. But even so, we can offer some general tips on writing good proofs:

State your game plan. A good proof begins by explaining the general line of reasoning, for example, “We use case analysis” or “We argue by contradiction.”

Keep a linear flow. Sometimes proofs are written like mathematical mosaics, with juicy tidbits of independent reasoning sprinkled throughout. This is not good. The steps of an argument should follow one another in an intelligible order.

A proof is an essay, not a calculation. Many students initially write proofs the way they compute integrals. The result is a long sequence of expressions without explanation, making it very hard to follow. This is bad. A good proof usually looks like an essay with some equations thrown in. Use complete sentences.

Avoid excessive symbolism. Your reader is probably good at understanding words, but much less skilled at reading arcane mathematical symbols. Use words where you reasonably can.

Revise and simplify. Your readers will be grateful.

Introduce notation thoughtfully. Sometimes an argument can be greatly simplified by introducing a variable, devising a special notation, or defining a new term. But do this sparingly, since you’re requiring the reader to remember all that new stuff. And remember to actually *define* the meanings of new variables, terms, or notations; don’t just start using them!

Structure long proofs. Long programs are usually broken into a hierarchy of smaller procedures. Long proofs are much the same. When your proof needed facts that are easily stated, but not readily proved, those facts are best pulled out as preliminary lemmas. Also, if you are repeating essentially the same argument over and over, try to capture that argument in a general lemma, which you can cite repeatedly instead.

Be wary of the “obvious.” When familiar or truly obvious facts are needed in a proof, it’s OK to label them as such and to not prove them. But remember that what’s obvious to you may not be—and typically is not—obvious to your reader.

Most especially, don’t use phrases like “clearly” or “obviously” in an attempt to bully the reader into accepting something you’re having trouble proving. Also, go on the alert whenever you see one of these phrases in someone else’s proof.

Finish. At some point in a proof, you’ll have established all the essential facts you need. Resist the temptation to quit and leave the reader to draw the “obvious” conclusion. Instead, tie everything together yourself and explain why the original claim follows.

Creating a good proof is a lot like creating a beautiful work of art. In fact, mathematicians often refer to really good proofs as being “elegant” or “beautiful.” It takes a practice and experience to write proofs that merit such praises, but to get you started in the right direction, we will provide templates for the most useful proof techniques.

Throughout the text there are also examples of *bogus proofs*—arguments that look like proofs but aren’t. Sometimes a bogus proof can reach false conclusions because of missteps or mistaken assumptions. More subtle bogus proofs reach correct conclusions, but do so in improper ways such as circular reasoning, leaping to unjustified conclusions, or saying that the hard part of the proof is “left to the reader.” Learning to spot the flaws in improper proofs will hone your skills at seeing how each proof step follows logically from prior steps. It will also enable you to spot flaws in your own proofs.

The analogy between good proofs and good programs extends beyond structure. The same rigorous thinking needed for proofs is essential in the design of critical computer systems. When algorithms and protocols only “mostly work” due to reliance on hand-waving arguments, the results can range from problematic to catastrophic. An early example was the [Therac 25](#), a machine that provided radiation therapy to cancer victims, but occasionally killed them with massive overdoses due to a software race condition. A more recent (August 2004) example involved a single faulty command to a computer system used by United and American Airlines that grounded the entire fleet of both companies—and all their passengers!

It is a certainty that we’ll all one day be at the mercy of critical computer systems designed by you and your classmates. So we really hope that you’ll develop the ability to formulate rock-solid logical arguments that a system actually does what you think it does!

In Class Problems Week 1, Fri

Problem 1. Albert & Ronitt announce that they plan a surprise 6.042 quiz next week. Their students wonder if the quiz could be next Friday. The students realize that it obviously cannot, because if it hadn't been given before Friday, everyone would know that there was only Friday left on which to give it, so it wouldn't be a surprise any more.

So the students ask whether Albert & Ronitt could give the surprise quiz Thursday? They observe that if the quiz wasn't given *before* Thursday, it would have to be given *on* the Thursday, since they already know it can't be given on Friday. But having figured that out, it wouldn't be a surprise if the quiz was on Thursday either. Similarly, the students reason that the quiz can't be on Wednesday, Tuesday, or Monday. Namely, it's impossible for Albert & Ronitt to give a surprise quiz next week. All the students now relax having concluded that Albert & Ronitt must have been bluffing.

And since no one expects the quiz, that's why, when Albert & Ronitt give it on Tuesday next week, it really is a surprise!

What do you think is wrong with the students' reasoning?

Answer:

↳ reasoning for the surprise day not being Friday is true since in that scenario there are no other days available. Meaning, this case is conditioned on $M \rightarrow Th$ already having occurred.

↳ when applying the same reasoning to $M \rightarrow Th$, we cannot isolate the day because Friday must still be available since the condition above is not yet met.

Solution

↳ "surprise" is not a mathematical concept and the "proof" assumed some axioms w/o defining it.

Problem 2. Identify the antecedents and conclusions of each of the following deductions and

translate them into propositional logic notation using logical operators:

$\wedge ::=$ AND,
 $\vee ::=$ OR,
 $\neg ::=$ NOT,
 $\rightarrow ::=$ IMPLIES,
 $\leftrightarrow ::=$ IFF (if and only if)

This may require that you "pin down" a statement that could be interpreted in more than one way.

Identify which of the deductions are sound ones.

(a) Jane and Pete won't both win the math prize. Pete will win either the math prize or the chemistry prize. Jane will win the math prize. Thus, Pete will win the chemistry prize.

Let $J(w=m)$ = Jane winning math

$P(w=m)$ = Pete winning math

$P(w=c)$ = Pete winning chem

✓ Statement:

$\{\neg [J(w=m) \wedge P(w=m)] \wedge [P(w=m) \vee P(w=c)] \wedge [J(w=m)]\} \rightarrow P(w=c)$

✓ Soundness: sound...

✓ correct

(b) The main course will be beef or fish. The vegetable will be peas or corn. We will not have both fish as a main course and corn as a vegetable. Therefore, we will not have both beef as a main course and peas as a vegetable.

Let: M_b = main course is beef
 M_f = main course is fish
 V_p = vegie is peas
 V_c = vegie is corn

✓ Statement: $\frac{(M_b \vee M_f), (V_p \vee V_c), \neg(M_f \wedge V_c)}{\neg(M_b \wedge V_p)}$

✓ Soundness: Sound since $\neg(M_f \wedge V_c) \rightarrow \neg M_f \vee \neg V_c$

Not sound...

There need not be only one M or V.

$\rightarrow M_b \vee V_c$ (via $A \& B$)
 $\rightarrow \neg M_b \wedge \neg V_p$
 $\rightarrow \neg(M_b \wedge V_p)$

nope!
 that is, $M_b \vee M_f$
 is true for 3 cases:
 $M_b, M_b / M_f \wedge M_b$
 meaning, not exclusive or.

→ note that this is diff w/ (a) since conclusion of (a) does not rely on MVC being an exclusive or.

nope should imply
 $\neg(M_b \vee V_p)$
 hence still not sound.

$$\begin{aligned} \neg(P \vee Q) &\Leftrightarrow (\neg P \wedge \neg Q) \\ \neg(P \wedge Q) &\Leftrightarrow (\neg P \vee \neg Q) \end{aligned}$$

De Morgan's Law

(c) Either John or Bill is telling the truth. Either Sam or Bill is lying. Thus, either John is telling the truth or Sam is lying.

Let: J = John truth
 B = Bill truth
 S = Sam truth

✓ Statement: $\frac{J \vee B, \neg S \vee \neg B}{J \vee \neg S}$

Soundness:

$J \vee B \rightarrow \begin{cases} J \\ B \\ J \wedge B \end{cases}$
 $\neg S \vee \neg B \rightarrow \begin{cases} \neg S \\ \neg B \\ \neg S \wedge \neg B \end{cases}$

only contradicting
 combi are:
 $B, \neg B$
 $B, \neg S \wedge \neg B$
 $J \wedge B, \neg B$
 $J \wedge B, \neg S \wedge \neg B$

$J \vee \neg S$
 is not contradicting

✓ Sound.

(d) Either sales will go up and the boss will be happy, or expenses will go up and the boss won't be happy. Therefore, sales and expenses will not both go up.

Let S = sales up
 B = boss happy
 E = expenses up

statement:
$$\frac{(S \wedge B) \vee (E \wedge \neg B)}{\neg (S \wedge E)}$$
 ✓

Soundness:

Counterexample: antecedent can be true if either Δ or ∇ .

if Δ only \rightarrow then $\neg B$ and $E, \neg E$ doesn't affect ∇

Hence $(S \wedge E)$ is possible

Not sound ✓

\hookrightarrow better put $S \wedge B \wedge E$

Problem 3. Boolean logic comes up in digital circuit design using the convention that **T** corresponds to 1 and **F** to 0. For example, suppose we want to describe a circuit with $n + 1$ inputs $a_n, a_{n-1}, \dots, a_1, a_0$ which are the $n + 1$ bits of the binary representation of an integer, k , between 0 and $2^{n+1} - 1$. We want outputs $o_{n+1}, o_n, \dots, o_1, o_0$ to be the bits of $k + b$ where b is a single bit.

For example, for $n = 1$, the formulas

1 if both 0 or both 1 \rightarrow
 $o_0 ::= a_0 \oplus b$
 $c_1 ::= a_0 \wedge b$ the carry into column 1
 $o_1 ::= c_1 \oplus a_1$
 $c_2 ::= c_1 \wedge a_1$ the carry into column 2
 $o_2 ::= c_2$

do the job. Here \oplus is the "mod 2 sum" operator: $a \oplus b$ is 1 iff $a + b$ is even.

no more time!

shouldn't \oplus be exclusive or?
 i.e. 1 is $a \neq b$?

(a) Generalize the example above for any $n \geq 0$. That is, give simple formulas for o_i and c_i for $0 \leq i \leq n + 1$.

Solution. Define

$o_0 ::= a_0 \oplus b,$
 $c_{i+1} ::= a_i \wedge c_i$ for $0 \leq i \leq n,$
 $o_{i+1} ::= c_{i+1} \oplus a_{i+1}$ for $0 \leq i < n,$
 $o_{n+1} ::= c_{n+1}.$

(b) Write similar definitions for the $n + 1$ bits of the sum of two binary numbers $a_n, a_{n-1}, \dots, a_1, a_0$ and $b_n, b_{n-1}, \dots, b_1, b_0$.

Solution. Define

$o_0 ::= a_0 \oplus b_0,$
 $c_{i+1} ::= (a_i \wedge b_i) \vee (a_i \wedge c_i) \vee (b_i \wedge c_i)$ for $0 \leq i \leq n,$
 $o_{i+1} ::= c_{i+1} \oplus a_{i+1} \oplus b_{i+1}$ for $0 \leq i < n,$
 $o_{n+1} ::= c_{n+1}.$

(c) How many Boolean operations does your system use to calculate the sum?

Solution. The scheme above uses $3(n + 1)$ AND'S, $2n + 1$ MOD-2-SUMS and $2(n + 1)$ OR's for a total of $7n + 5$ operations.