Proofs without Goofs

Andreas Klappenecker

The basic principle of a mathematical proof is quite simple. You assume that the hypothesis of your claim is true, and then you go on to show that the conclusion must hold by just using logical deductions and known facts.

The primary goal is to develop a well-structured, simple, readable proof.

It is often a good idea to revise a proof several times until you get a satisfying clarity and elegance. I recommend you to keep in mind the following remarks.

- Each statement in the theorem or proof should be a *complete* English sentence. The usual punctuation rules apply, even when you use formulas. Basically, if you read it out loud, then it should be a well-formed sentence.

- Do not use formal logic symbols, such as '$\exists$', '$\forall$', '$\Longrightarrow$', '$\therefore$', unless your text is on formal logic; write 'there exists', 'for all', 'implies' and 'therefore' instead.

- Good definitions can increase the readability of your proof. However, you should avoid to introduce you own terms if established terminology is available. You might find it amusing to call an integer *green* if it is a divisible by 2; however, your readers would prefer the term *even* integer.

- You can call your claims Theorem, Proposition, Lemma, Corollary, and Scholium. They all have the form "If *foo* holds, then *bar* must hold". Recall that *foo* is said to be the hypothesis and *bar* the conclusion of your claim. It is simply a matter of taste how you call your claims.

  Generally, the term *Theorem* is used for important results, *Proposition* for results that are perhaps not as important as theorems, and *Lemma* for auxilliary results. A reader will pay more attention to theorems than to lemmas.

  A proposition or theorem can have a direct consequence that requires little or no proof; such a claim is generally called a *Corollary*. If the Corollary is extremely important, then you can elevate it to a *Scholium*, but this term is not widely known, so you better use it rarely.

- Structure you proof! A proof that extends over several pages is hard to follow, unless it is nicely structured. Sometimes it is better to proof a theorem through a sequence of easy to understand lemmas, rather than long winded statement.

  You can also structure a long proof of a theorem by dividing it into smaller (numbered) steps instead of a sequences of lemmas. This is advisable if the lemmas would have awkward formulations.

Proof Techniques. The available proof techniques are quite limited. In a *direct proof*, you assume the hypothesis and derive the conclusion is small steps that use only the definitions, know facts, and logical rules. The next lemma is an example of this proof technique.

Lemma 1. *The sum of two even integers is an even integer.*

Proof. Suppose that x and y are even integers, meaning that there are integers x' and y' such that $x = 2x'$ and $y = 2y'$. It follows that

$$x + y = 2x' + 2y' = 2(x' + y')$$

is again an even number. □

In a *proof by contradiction*, you suppose that the theorem was not true, and the you derive a contradiction. It is generally a good idea to prepare the reader that the proof will be by contradiction, for instance, by starting your proof with the line "Seeking a contradiction, we assume".

Lemma 2. *There are infinitely many prime numbers.*

Proof. Seeking a contradiction, we assume that there exist only a finite number of primes, say $p_1, \ldots, p_n$. However, the integer $q = p_1 p_2 \cdots p_n + 1$ is not divisible by any of the primes p_k for $1 \leq k \leq n$. Therefore, q must be a prime number that is not in our list of prime numbers and this contradicts our assumption. □

Perhaps the most important proof technique for the analysis of algorithms is to proof statements by mathematical induction. The basic principle is again quite simple. Suppose that $P(n)$ is a statement with a positive integer n as a parameter. You have to show that the base case $P(1)$ is true, and that the truth of $P(n)$ implies the truth of $P(n + 1)$.

Theorem 1. *If you make n straight cuts across a pizza, then you divide the pizza into at most $n(n + 1)/2 + 1$ pieces.*

Proof. We prove the statement by induction on the number of cuts. If we make a single cut, then we divide the pizza into two pieces, which proves the case $n = 1$. Suppose that the claim is true for n cuts, meaning that the pizza is cut into at most $n(n + 1)/2 + 1$ pieces. The $(n + 1)$-th cut will create the largest possible number of additional pieces if we cut through all previous n cutting lines and avoid intersection points of previous cutting lines. Our new cut will create an additional piece when it crosses the first line, another one when it crosses the second line, etc., until it crosses the nth line, creating $n + 1$ additional pieces in total. Thus, after making cut number $n + 1$, we have at most

$$n(n + 1)/2 + 1 + n + 1 = \frac{n^2 + n + 2n + 2}{2} + 1 = \frac{(n + 1)(n + 2)}{2} + 1$$

pieces of pizza, as desired. □