CHAPTER 5

Proofs

Introduction

Whenever you are asked, in a problem, to show something, or demonstrate something, or verify something, (or, of course, prove something), you are being asked to do a proof. This chapter deals with answering the following questions from "The Principal Classes of Homework and Exam Problems" in chapter 1:

• "How can I break the proof of theorem *t* down into pieces I can understand?"

• "What theorem or lemma or axiom enables the author to go from this assertion to that one in this proof?"

• "How do I prove the statement *s*?"

Learn How Proofs Work!

Many students of engineering, and even some physics students, even some math students!, don't like doing proofs. They feel that proofs are somehow mere formalities, an attitude summed up by the student who, after a professor had spent the better part of a class hour carefully explaining a proof, stuck his hand up and asked, "Professor C — , is any of this important or is it just mathematics?"

Can you get through a technical education — in particular, one involving a lot of mathematics — without knowing how to do proofs? I suppose so, but it will mean you will have to rely on your intuitive understanding of logic, which almost certainly will be inadequate for all but the simplest proof problems, or else you will have to re-invent the basic rules on your own. (Do you know the difference between "for all *x* there exists a *y* such that...," and, "there exists a *y* such that, for all *x*, ..."?) So why not take the easy way out and learn the rules early in the game? I won't say you need to understand formal logic in order to get through everyday life — although, to comprehend the deception of much advertising, it certainly helps to understand the logical fallacy known as *post hoc ergo propter hoc* ("after this, therefore because of this") — but somehow I feel that to be truly civilized, you need to know, or at least know where to find out if you have to, what a logically valid argument is.

One reason why many students are uncomfortable with proofs is that the students have never been given a clear, concise presentation of what a proof is and how to go about constructing one. The best book I know of for this purpose is Marvin L. Bittinger's *Logic and Proof* (Addison-Wesley, Menlo Park, Calif., 1972; the latest edition is titled, I believe, *Logic, Proof, and Sets*, from the same publisher). It is as brief and clear as I believe the subject can be made. Once in your life you should go through it, or a book like it, and work all the exercises. You may not be able to prove everything you want to after that, but you will at least always know how to set up the proof task, you will understand the meaning of various terms associated with proofs, e.g., *if and only if* (i.e., *equivalence*, often abbreviated *iff*), *contrapositive*, *proof by contradiction* (also known as *indirect proof*), *necessary* vs. *sufficient*, and you will know some basic techniques to try, and know if each step is logically valid. You will also know the relationship between mathematical logic and set theory. Do yourself a favor: once and for all, learn how proofs work!

Proofs and Programs

There is a similarity between mathematical proofs and computer programs that you should keep in mind. A long mathematical proof typically begins with certain definitions, then a number of lemmas (i.e., theorems to be referred to in the main proof) are proved, then the main proof is given, referring to the lemmas as it proceeds. This is precisely the way a well-designed computer program is written: first come various declarations (analogous to definitions), then a series of procedures — in some programming languages they are called "functions" and in Assembly language they are called "subroutines"— (analogous to lemmas) and finally the main program (analogous to the main proof), which is often no more than a page long, and consists primarily of invocations of the procedures.

The following table summarizes the similarities:

Table 1: Similarities Between Programs and Proofs

Computers	Mathematics
subsystem ("large program")	branch of mathematics
function	theorem
algorithm	idea of proof of theorem
implementation of the algorithm in an actual computer program	actual written proof
declaration	definition
procedure, function, subroutine	lemma

Table 1: Similarities Between Programs and Proofs

Computers	Mathematics
main program	main part of proof of theorem
structured program	structured proof

How to Write Up Proofs in an Environment

In other books this section would be called *How to Study Proofs*. When you make an Environment, however, studying the proof and entering it into the Environment are the same thing.

Why Are Proofs So Difficult to Understand?

To many students, it often seems that, somehow, by sheer genius, the author of a proof has managed to string together a sequence of statements which, by God, once you follow them all, do seem to prove the lemma or theorem asserted. But nowhere in sight is the overriding idea that governs the proof and its subordinate proofs. The presentation of the proof wouldn't be much different if in fact the author had come up with the proof by pure trial and error, pure random selection of statements.

The main reason proofs are so difficult to understand is that the way they are presented requires that you understand *all of the argument* as it proceeds. Proofs are written as a succession of paragraphs. Even if, as is often the case, the proofs invoke previously proved lemmas, the proofs require that, as you proceed, you somehow hold in mind the entire argument which has so far been developed. A very difficult task indeed, in many cases, especially when the proof runs to more than one page

Another reason why proofs are so difficult is that textbook authors and professors in classrooms do not consistently give the justifications for each statement in a proof. Often there is a tacit assumption — a knowing wink to future members of the Club, namely, the future mathematicians in the class — that the student will know what the justification is. The effect is what I suspect all too many textbook authors and professors hope that it will be, namely, the intimidation of the student. "If I can't see immediately what the justification of that statement is, then clearly I have no business studying mathematics. In fact, I have no business staying alive!" If pressed as to why they do this, authors and professors will sometimes reply that they wanted to "keep the proof as short as possible". If asked why this is desirable, they will reply that it makes understanding easier. (How do they know?)

A third reason why proofs are so difficult to understand is that textbook authors and professors in the classroom do not always explain what the immediate goal of a particular sequence of steps is. Of course, the ultimate goal is to prove the lemma or theorem. But in a long proof, there are typically several interim goals which can be explained, *beforehand*, in language like, "Our immediate goal is prove ... To do this, we will first show ..., then we will show ..., and these two facts will then imply ..."

A Way to Overcome These Difficulties — Use Structured Proofs!

Background: The Control of Complexity in Computer Programming Using Structured Programs

In the early days of computer programming, programmers confronted problems similar to those confronting students of mathematics (and professional mathematicians!). When programmers attempted to understand another person's program (or even one of their own if they hadn't worked on it recently), they had no choice but to understand how each successive statement led toward the value which was to be computed. For programs of more than a page or two, especially programs which had few or no explanatory comments (because the person who wrote the program thought his reasoning was "obvious") this was often a hopeless task insofar as understanding meant *being able to say with some assurance that the program was correct or incorrect*.

A giant step toward the solution of this problem was structured programming, as developed by Dijkstra and others (see "Fundamental Concept 2: 'Structure, or Breaking Complex Thing into Simpler Things", in chapter 3), and which was already magnificently implicit in the very structure of LISP programs. (In pure LISP it is, for all practical purposes, impossible to write an unstructured program.)

The basic idea — as summarized above in the sub-section "Proofs and Programs" on page 99 — was this: begin with a main program made up of a "few" statements or commands, say, half a dozen or so, and show that *if* the function computed by each statement or command was what it was supposed to be, then the function computed by the entire main program would be what it was supposed to be. Then recursively apply this technique to each of the statements or commands in the main program, since each statement or command must be implemented by its own program. Eventually the programmer would get to statements or commands whose implementations were provided by the programming language itself, and could be presumed correct, and the process would stop, having produced a correct program.

The Control of Complexity in Mathematics Through Use of Structured Proof

There is no reason why the basic idea of structured programming cannot be applied to proofs, and that is precisely what we do in Environments.

In a structured proof — see, e.g., "A Much Better Approach" on page 118 — we proceed exactly as in a structured program. We break the proof down into a few major steps — say, less than seven. (The number seven is chosen here because studies by psychologists suggest that this is the maximum number of steps the average programmer can comprehend "at a time".) The top level of the proof is analogous to the main program of a computer program.

The steps must constitute a complete logical argument for the validity of the theorem or lemma **if** *each of the steps is valid.*

The validity of each step is then established by separate proofs referred to at the end of each step. "(See...)" tells the reader where the proof of that step is to be found. Of course, if the step consists of a statement that the course assumes is valid, then you don't need a reference to a proof. You only need to refer to the statement. Sometimes, the proof of the step can be done in a few lines, in which case, you can give the proof immediately below the step.

Each proof of a step, sub-step, etc., is composed of sub-steps, sub-sub-steps, etc., which, *if* each is valid, constitute a valid logical argument for the validity of the step, sub-step, etc. Where the process stops depends on your level of knowledge, or what you are allowed, in the course you are taking, to assert without proof.

Let me emphasize that the argument in each proof and sub-proof must make sense in itself — must be a convincing, valid, logical argument as it stands, the only condition being that you or the textbook author has somewhere proved the validity of each step (indicated by "(See...)"). Thus, for example, a sub-proof might run something like:

n.1 *x* is a *y* having the property *z*.
(See ...)

n.2 But all *y* having the property *z* also have the property *w*.
(See ...)

n.3 Therefore *x* has the property *w* (by a theorem of basic set theory), which is what we were to prove."

Here, n is the number of the step in the next higher level proof which these steps prove the validity of.

Each step that is not proven by a sequence of sub-steps, but instead follows from a known lemma or theorem, must reference that lemma or theorem in a way that will allow you to quickly look it up days, weeks, months, years from now. Without question, the failure by authors of standard textbooks to systemmatically and completely provide these references has probably cost students more time over the years than any other single failing of textbooks. (I have even heard *professional mathematicians* complain about the time they had to spend trying to figure out what lemma or lemmas justified a statement in a proof.)

Think how much time you would have saved over the course of your math studies if you *always* had been given the lemma or theorem reference, with page number, that justified statements that weren't justified in the proof itself. If the lemma or theorem used is not contained in the textbook, then a reference to the lemma or theorem in a well-known text should be given, or else its name, if it has one, or a brief description of it, so that you could quickly find it in a standard text.

You can see, I think, that structured proof is a giant step toward removing the three difficulties explained above under "Why Are Proofs So Difficult to Understand?" on page 100.

Once you start doing proofs in this way — I mean, doing difficult proofs this way, since, obviously, you will not want to invoke all this machinery for proofs that are only a few lines long — you will see how negligent most textbook authors are. For, by rights, they should *always* make explicit exactly what level of knowledge they are assuming on the part of the student, *and then* they should provide detail in their proofs *down to that level of knowledge*. Period. Everything else should be explicitly referenced to theorems, lemmas, definitions in the textbook.

And you should now see how devious the argument is that by omitting justifications of statements, a proof is made shorter and therefore easier to understand: it is true that it is easier to understand a small number of steps at a time than a large number of steps at a time. That is why you should have no more than about seven steps at each level of your proof. It is *not* true that by keeping the total number of steps small, the entire proof will be more easily understandable, because usually that means that the author is simply leaving out groups of steps and/or explicit references to other theorems and lemmas. He is giving you an "encoding" of the proof. But the process of "decoding" takes time! You either have to have previously learned and retained in memory the omitted steps, or you have to spend time figuring out what the omitted steps are, and then why they are valid. To do the latter, you must either do the proofs yourself or find them elsewhere in the textbook or in another textbook. The technique of structured proof trades pages of paper (or computer screens) for time: a structured proof is normally longer than the typical textbook proof, but it can be understood and memorized in much less time.

Structured Proofs Make All Proofs Look "the Same"!

We quote from an email we sent to a person doing research on ways of aiding undergraduates' comprehension of proofs in the traditional format.

> All structured proofs can be represented by finite trees, in which:
>
> The statement to be proved (e.g., lemma, theorem) is associated with the root node;
>
> Descending from that node is a small number of branches; the nodes on the ends of the branches, reading, say, from left to right, being the statements that, taken in succession, constitute a correct proof of the node statement;
>
> Descending from each of these nodes is a small number of branches, the nodes on the ends of these branches, again, reading, say, from left to right, being the statements that, taken in succession, constitute a correct proof of the node statement immediately above.
>
> Etc.
>
> Of course, some proofs of statements can be simply a reference to a known true statement from another subject, or to a statement proved earlier in the course. Eventually, levels must be reached in which this is true for all the statements descending from each node.
>
> That's it! End of story! The tree format I have just described works for simple textbook examples, or for more difficult exercises at the end of a chapter, or for proofs in PhD theses, or for Wiles's proof of Fermat's Last Theorem, which originally at least ran to well over 100 pages, or for ...

Why Isn't Structured Proof the Normal Way We Do Proofs?

The idea of structuring proofs in the same way that programmers structure programs goes back to the seventies at least, although the earliest use of the term "structured proof" I have seen in print is the title of an example given by Leslie Lamport in

Knuth, Donald E., et al, *Mathematical Writing*, MAA Notes No. 14, The Mathematical Association of America, Washington, D.C., 1989, p. 72. Later papers on the subject include: Lamport, Leslie, "How to Write a Proof", *American Mathematical Monthly*, Vol. 102, No. 7, Aug.-Sept., 1995, pp. 600-608; and Back, Ralph; Grundy, Jim; von Wright, Joakim; "Structured Calculational Proof", *Formal Aspects of Computing*, 9(5-6), 1997, 469-483.

So why hasn't structured proof become the norm? Why do mathematicians continue to use a format for presenting proofs that makes the proofs more difficult to understand than necessary? I think there are at least two reasons:

(1) Mathematicians are naturally conservative about the way they present their results because the present way, which goes back to Euclid, some 300 years before Christ, has enabled them to produce such a stunning body of knowledge. It is a tried and true method.

(2) The present way enables them to keep the Club exclusive: the kind of subtlety of language, the kind of — let's call what it is: pedantry — that is currently required for an acceptable mathematical writing style, can normally only be learned by those who have attended certain schools. If it is virtually impossible to write in the style I require without your having studied under me, or under the (very few) people I consider my equals, then I am able to severely limit membership in my Club.

(It would make a very interesting master's, or perhaps even PhD, thesis to investigate the subtleties of language (not of content, but of language) that distinguish papers published in the most prestigious mathematics journals from those published in the least prestigous journals. It would be the equivalent of a study of what is considered good manners among the most elite social stratum in this country, vs. what is considered good manners in the middle class.)

In passing, let me try to dispel the myth that journal papers are how new results in mathematics are communicated. This wasn't true before the development of the Internet, when colleagues circulated drafts of papers among each other, and it is even less true now, when it is becoming more and more common to be able to access mathematical papers on the web. Publication in journals is for the purpose of registering what those in the specialty already know. It is for building careers, not for communicating knowledge.

Structured proof requires a much less elaborate writing style. It reduces the importance of *syntax* in mathematics papers, because it imposes a stricter form on proofs: form does some of the work that otherwise words have to do.The form of a structured proof is very simple, as we have seen. It consists of a few steps, each of which consists of a statement and a reference to the justification of the statement. Period. True, a brief idea of the strategy to be used may often be required at the start of the proof as an aid to understanding, and each statement must be clear. But there is much less need for the kind of fussing over style which is present in a book like Knuth's *Mathematical Writing* (ibid.), with, e.g., its pages of discussion devoted to the corrrect usage of "which" vs. "that". Be honest: if you read in a textbook, "*A* is the set which we have been seeking," and someone asked you, "What is *A*?", would you be unable to answer the question until *which* was replaced by *that*? Or if you read, "In order to rapidly calculate this quantity, we can use the method described on p. 27," would you throw up your hands in utter perplexity as to what the sentence means because you do not understand the meaning of split infinitives like "to rapidly calculate"? One of the experts quoted in Knuth's book says, "A split infinitive should really jar. 'It's got to light up in red!'" (ibid., p. 47). Apparently, the expert is unaware that the book that is considered by many to be the supreme authority on correct English usage, namely, Fowler's *Modern English Usage*, allows split infinitives in many contexts. In fairness, however, I must point out that Knuth and another of his experts do not prohibit split infinitives at all times (ibid., p. 114).

In *Mathematical Writing* (ibid.) we read "Don [Knuth] says that a computer program is a piece of literature. ('I look forward to the day when a Pulitzer Prize will be given for the best computer program of the year.')" — ibid., p. 21. To me, this reveals such a colossal misconception of the nature of computer programs that I cannot help but point it out here, and cannot help but warn students to question the value of what Knuth says about the writing of mathematics. A computer program is emphatically *not* a piece of literature. Ever since the late fifties, complex computer programs have been generated by other computer programs (namely, by programs called compilers and interpreters), and programs, as far as we know, have no literary skills at all. No computer program, to my knowledge, has produced anything that anyone would want to call a piece of literature. A computer program, whether generated by a machine or not, is a sequence of instructions that can be executed by a machine. Making a program easy to understand by humans is *not* primarily accomplished by literary skills, it is primarily accomplished by organization and formatting. When a class of programs aimed at solving a particular type of problem starts to become too complex, the solution is normally to define a higher-level language that hides many of the details that contribute to that complexity. The solution is *not* to find programmers with greater literary talents!

In mathematics, a similar prejudice towards words and style prevails regarding the nature of proof. "A proof is a story," says Ian Stewart in a book published in 2006[1]. No, it isn't. A proof is — or should be — an argument presented in structured format with the goal of enabling qualified readers to understand it to the depth they choose in the shortest possible time. If you want to read stories, become an English major.

It is remarkable how Stewart misunderstands the nature and purpose of structured proof. He says: “Structured proofs make explicit every step in the logic, be it deep or trivial, clever or obvious. ...there can be no doubt that [structrured proofs] can ve very effective in making sure that students really do understand the details.” [1]

But the main purpose of structured proof is not to present details! It is to place details where they belong, namely, at the deeper levels of the proof, and to allow the reader to understand the Big Picture, the structure of the argument of each step and each sub-step. Stewart’s distinguishing between deep and trivial, clever and obvious steps reveals all too clearly the mentality of the professor: “the good students will understand this next statement...”. How does he know? Has he explicitly stated, at the start of his textbook, exactly the minimum set of skills and the minimum knowledge he assumes of all readers? The successive structuring of the proofs of steps ends with those steps the reader can be assumed to accept based on the stated minimum set of skills and minimum knowledge. Whether a step is deep or clever has nothing to do with it!

Why should a proof contain knowing winks and nudges aimed at future members of the Club? Why should a proof be yet another mechanism for separating the winners from the losers? Why can’t a proof be simply what it is, namely, an argument?

Perhaps the belief — which is certainly held by many mathematicians besides Stewart — that “a proof is a story” is part of the reason for those narratives that are so common in textbooks and are so perplexing to many students, namely, the narratives that begin, without explanation of their purpose, with sentences like, “Suppose that ..., and suppose further that ... Then ... This in turn implies ...” and on and on, leading the reader down the walk through the woods until only at the very end does he or she discover that the purpose of this story has been to prove the statement p. Perhaps the textbook author hoped to make the proof of the statement p more interesting in this way. I don’t know. But I do know that, whether textbook authors like it or not, an argument is more easily comprehended if the reader knows in advance what the purpose of the argument is.

An obsession with words permeates contemporary mathematics culture. A minimum of two published papers a year is required for tenure. Thereafter, one of the important measures of a mathematician’s reputation is the number of papers he has published. The paragraph-based, prose-centered format of proofs has placed a pre-

1. *Letters to a Young Mathematician*, Basic Books, N.Y., 2006, p. 89. My criticism of Stewart’s view of the nature of proof is in no way intended to detract from his deservedly high reputation as a mathematician or from his reputation as one of the best popularizers of mathematics of our time.
1. Ibid., p. 91.

mium on fine points of style, much of it the most outrageous pedantry. But what matters is not numbers of papers but originality of ideas. That is what is important.

I don't believe that the Math Dept. should be part of the English Dept.

I urge you not to dismiss structured proof until you have rewritten a few textbook proofs in structured form, and seen how this form makes clear the essence of the argument, and how the form enables you to dispense with a great many concerns about "how shall I say this?" It is a basic tenet of this book that the less important — the less needed — the fine points of prose writing become, the better for both the creators and users of mathematics. "Let the format do your writing for you" is a motto I recommend, always understanding that this is a bit of an exaggeration, since *some* writing will always be necessary.

A More Formulaic Approach Can Speed Writing and Understanding

I want to conclude this sub-section by arguing that *all* of mathematical writing, not only proofs, can be made far more formulaic than the current proponents of "good style" believe. You need not read the following if you have no intention of ever doing any writing of formal mathematics.

Semantic Categories in Mathematical Writing

Whenever we write something in mathematics, we have a specific purpose in mind: to define something, to state a lemma or theorem or conjecture, to prove a lemma or theorem or conjecture, to give an informal idea of what we are about to prove, etc. Each of these purposes — each of these tasks, "jobs" — define what I will call a "semantic category". In other words, by a "semantic category" I mean a set of words, phrases, or sentences having a single purpose.

Definitions constitute a semantic category. Here the purpose is to establish the meaning of a term which we introduce.

Lemmas, theorems, corollaries, and conjectures constitute another semantic category. Here the purpose is to state (in the first three cases) something that is true, and, in the last case, something that we believe might be true.

Proofs constitute another semantic category. Here the purpose is to establish the logical validity of a lemma, theorem, corollary, or conjecture.

Informal remarks constitute another semantic category. The purpose here is to aid a reader's understanding of a formal argument, or to call his attention to certain matters. This category is the one and only one where literary skills will continue to be of great value.

References constitute another semantic category. The purpose here is to let the reader know where the proof of a relevant lemma, theorem, or corollary can be found, or where an exposition of a given subject can be found.

Indexes constitute another semantic category. The purpose here is to enable the reader to rapidly find definitions, lemma statements, proofs.

In mathematics papers published in journals, *abstracts* constitute yet another semantic category.

The reason why I have given a special name, "semantic categories", to these familiar parts of mathematical texts is that I want to separate the task to be performed, from the means of performing it. I want to put into its proper place the kind of pedantry that is promoted in books like the above-mentioned one by Prof. Knuth. I want to make clear that there are many different ways to perform each task.

But you might be inclined to say that in principle there is an *infinite* number of words, phrases and sentences, in each category. That is true, but I will argue that there is only a finite number of sub-tasks, sub-"jobs" — sub-categories — in each category, and that each can be covered by a format. This is not the place to go into detail on each of these. I will simply give a few examples:

- Every logical statement can be expressed using the logical terms **if**, **if-and-only-if** (**iff**), **and**, **or**, **not**, **then**, **there exists**, **for all**, and others, in the way that is taught in basic logic and set theory courses. There is nothing sacred about literary variations on **if** such as "let" and "suppose".
- Every definition can be expressed in one of a small number of formats, e.g., "**If** <*statement of condition(s)*> **then** we say that ... is <*new term*>".
- As we have seen, every proof can be expressed in the form of a structured proof.

At this point, you might argue that it is well known that all of mathematics can be formally expressed using symbolic logic, and so why am I going through all the trouble of making that case again. The answer is that most human beings find it very difficult to read mathematics that is expressed in nothing but the symbols of symbolic logic and the symbols of a given branch of mathematics. My purpose here is, instead, to argue that if mathematics aimed at human consumption were written in the more formulaic way described, then (1) the mathematics would be much quicker to write, and (2) it would be much quicker to understand, because recognizing formats would replace much of the labor that currently has to be done in reading prose.

I will conclude with an extremely controversial statement: if the mathematics world continues to insist on doing things the old way, then I do not believe it should be the author's responsibility to fine-tune his paper to the stylistic requirements of this journal, and then that journal. Keeping up with the syntactic nuances that each journal editor has decided are the *sine qua non* of publication is not how a mathematician should be spending his time! He should be able to give his paper to a person expert in the journal's requirements, pay him a reasonable fee, answer his questions about obscurities in his paper, and that should be that! He has better things to do with his time.

Contrary to popular opinion, mathematics is not a language. It is a system of abstractions, held together by formal logic, and representable in an infinity of languages, some of them better for human use than others.

Aim for Understanding Each Proof and Sub-Proof "at a Glance"

In writing up proofs, you use the same techniques which were described in "Aim for 'Understanding at a Glance'" in chapter 4. In addition to these, you can and should use the following techniques, which apply specifically to proofs.

Use Standard Form for all Theorem Statements

A standard form for all theorem statements makes it immediately clear what the logical form of the theorem is. This is a great help in constructing a proof. In my own experience, a simple form incorporating indents seems best, e.g.:

If ...
  and ...
  or ...
Then ....

For all ...
  ...
  iff
  ...

If ...
Then ...
    iff
    ...
        and not ...

etc.

The rules, in brief, are:

• Begin every line with a logical connective — **If**, **and**, **or**, **Then**, etc. — (except when a line is the continuation of a long previous line). (In paper Environments, you can use a double-underline instead of boldface.)

• Indent when the logical connective(s) are within the scope of the next previous logical connective.

Number the Steps in Every Proof and Sub-Proof

In mathematics journals, steps are almost never numbered; each proof is written out as a succession of paragraphs. Since textbooks are written by mathematicians, the same practice is carried over, to the great disadvantage of the student. Numbering of steps is *essential* for structured proofs, because the numbers are the way you identify proofs and sub-proofs of each step. Also, the numbers save you the all-too-familiar time and labor which textbooks force upon you of searching for exactly what the author was referring to when he said “as shown above”.

In “Theorem (NZ 2.25b), step 1.3, proof”, in the Appendix, “A Number Theory Environment (partial)”, the decimal system employed is, I think, obvious. For example, the steps in the proof of step 1 of the main theorem are numbered 1.1, 1.2, 1.3, etc.; the steps in the proof of step 1.3 are numbered 1.3.1, 1.3.2, 1.3.3, etc.

Get Rid of the Equals Sign!

The title of this section is an exaggeration, of course. The equals sign was a wonderful invention for mathematics[1], but it is by no means the only way to represent equality. A much better symbol in many cases is the horizontal curly brace or equivalent, as shown in the following example:

$$\underbrace{\left((8x^2)/3 \;+\; (16x)/4 \right) \;-\; \underbrace{(x^2/3 + x/4)}_{-\;x^2/3 \;-\; x/4}}_{7x^2/3 + 15x/4}$$

More examples, taken from the working of calculus problems, will be included in the next edition of this book. If you have a long equation or other logical statement, and you want to show, in that equation or statement, what various terms or groups of terms are equal to, the horizontal curly brace does this much more directly than a succession of equations, each of which requires that you look back to the original long equation or logical statement to find the term or group of terms on one side of the equation. Too much eye-work and brain-work! Use the curly brace instead. (In my experience, professors and graders are perfectly willing to accept the use of this symbol, provided you make clear at the start of your paper what it means.)

The point is, you should use whatever symbol, equals sign or curly brace, makes understanding more rapid, more immediate.

Have a Section, “Proofs, Tricks for Doing”

A mathematician to whom I explained the Environment idea said it was too limited, and among the reasons he gave was that an important proof in a branch of mathematics usually provides a new method for doing other proofs. “Fine,” I said. “I’ll just add a section that gives references to examples of various proof techniques. That's one of the beauties of the Environment idea!” He wasn’t convinced, but on the basis of

1. “The equals sign was invented by Robert Recorde in his *Whetstone of Witte*, 1557, but it did not come into general use until more than a hundred years later.... Fermat [1601-1665] never used it, preferring always to write ‘æq’ or ‘adæq’ or fuller Latin words like ‘adæquibantur’ that these terms abbreviate...Descartes used ‘=’ to mean something completely different...” — Knuth, Donald E., et al., *Mathematical Writing*, MAA Notes No. 14, The Mathematical Association of America, Washington, D.C., 1989, p. 91.

long experience, I am. You can always add a new section whose title is some subject that you find is important. Of course, it takes a few seconds' thought to determine what the title should be. I hope it is clear to you that "proofs, tricks for doing" will probably result in faster look-up time days, weeks, months from now, than, e.g., "tricks for doing proofs". It may take a few additional seconds' thought to check for reasonable synonyms you should add to your Environment, with a simple one-line reference, "See ...".

Use Drawings!

In every mathematics class, the professor makes free use of drawings, diagrams, arrows, circling of terms, in order to explain proofs. But when the same professor writes a textbook, particularly an advanced textbook, he feels compelled to use as few drawings as possible, presumably because he accepts the mathematicians' Party line that pictures are not "rigorous". But consider what one of the 20th century's great mathematicians said on this subject:

"A heavy warning used to be given that pictures are not rigorous; this has never had its bluff called and has permanently frightened its victims into playing for safety. Some pictures, of course, are not rigorous, but I should say most are (and I use them whenever possible myself). An obviously legitimate case is to use a graph to define an awkward function (e.g. behaving differently in successive stretches): recently I had to plough through a definition quite comparable with the 'bad' one above [i.e., in a preceding example not quoted here], where a graph would have told the story in a matter of seconds. *This* sort of pictoriality does not differ in status from a convention like 'SW corner', now fully acclimatized. But pictorial *arguments*, while not so purely conventional, can be quite legitimate." —Littlewood, J. E., *Littlewood's Miscellany*, ed. Bela Bollobas, Cambridge University Press, N.Y., 1990, p. 54.

Littlewood then gives an example of a proof that relies on a picture. He continues, "This [proof] is rigorous (and printable), in the sense that in translating into symbols no step occurs that is not both unequivocal and trivial. For myself I *think* like this whenever the subject matter permits." — ibid., p. 55.

And yet, as Littlewood points out, the myth dies hard. A young mathematician once asked me to criticize a paper he had just completed. It was only five or six pages long, but after several cursory readings, I found I couldn't understand it, so I did what I always do in such circumstances: I started at the beginning and began making a succession of drawings to show the development of his argument. Eventually I was able to understand it, and found it quite simple and elegant.

When I went over the paper with him, I told him that he could present the underlying idea in a couple of drawings occupying a single page and thus save the reader the trouble I had had to go through. He replied, *"If you make it too easy to understand, people don't read the paper.*"

In the ensuing discussion, he repeated the standard argument that a good drawing makes it too easy to overlook logical errors. I countered that it is also easy to overlook logical errors in a long, written, argument. In fact, about the same time as I read the paper, I heard a computer scientist remark in a lecture that good mathematicians don't memorize the details of proofs, but instead memorize a picture from which they can easily reconstruct the proof.

Here is Littlewood again:

"In presenting a mathematical argument the great thing is to give the educated reader the chance to catch on at once to the momentary point and take details for granted; his successive mouthfuls should be such as can be swallowed at sight; in case of accidents, or in case he wishes for once to check in detail, he should have only a clearly circumscribed little problem to solve (e.g. to check an identity: *two* trivialities omitted can add up to an *impasse*). The unpractised writer, even after the dawn of conscience, gives him no chance; before he can spot the point he has to tease his way through a maze of symbols of which not the tiniest suffix can be skipped." — ibid., p. 49.

Littlewood then gives an example of a three-page, truly intimidating proof of a famous theorem by Weierstrass that a function $f(x_1, x_2)$ continuous in a rectangle R, can be uniformly approximated by a sequence of polynomials in x_1, x_2. He remarks that the example "is unduly favorable to the criminal since the main point is hard to smother." He follows this proof with a "civilized" one consisting of a single drawing and about a page of text.

As proofs start running into hundreds of pages— Andrew Wiles' first proof of Fermat's Last Theorem was over a hundred pages long — mathematicians will have to start asking themselves just what it means to understand a proof and what it means to agree to a proof's validity. Perhaps computer verification will become more important, but then we confront the problem of verifying the verifying program. Personally, I would much rather bet on the correctness of a structured, illustrated proof that several mathematicians had found to be correct, than on the correctness of an unstructured version of the same proof that had been checked by machine.

How to Use These Techniques to Construct Your Own Proofs

You can use all the techniques described above to develop proofs of your own. Keep in mind that, just as you can use a computer program without having any idea of how the program works (consider the programs in your pocket calculator), so you can use a theorem without having any idea of its proof(s). Many students, particularly first and second year college math students, think that, if you're *really* good — if you were really *meant* to study mathematics — somehow you always know how the proofs go of the theorems you use. Somehow you quickly understand the proofs. Somehow it is *indecent* to use a theorem whose proof you don't understand, or even know. Not true! Throughout your study of mathematics you have been using theorems you weren't even aware of, some elementary examples being the theorems guaranteeing the correctness of the familiar grade school algorithms for doing addition, subtraction, multiplication and division, not to mention those guaranteeing the correctness of the rules of factoring.

Here is a general procedure for doing proofs:

1. Be sure you know what you are trying to prove, i.e., be sure you write down what you are trying to prove, in structured form, as described above in "Use Standard Form for all Theorem Statements" on page 110.

2. Look up the meanings of all terms you don't know. (Easy if you have an Environment!)

3. Set up a tentative main proof whose steps you believe you can prove, even though you may not know how to prove them at the moment.

It is amazing how much mileage I have gotten out of this practice in homework and exam problems. For example, with time running short, I would write something like,

"We are being asked to prove ...", "We would have a proof if we could prove:

"1. ...

"2. ...

"3. ...

"The proof of 1. seems straightforward, given theorems ... and ...

"The proof of 2. is more difficult. I will outline the steps that seem likely to yield a proof..."

etc.

I think that one reason I have gotten such a large proportion of partial credit from this approach is that it says to the professor or grader: "Here is a person who knows how to think. He may not have at his fingertips all the knowledge we wish he did, but he certainly knows how to deal with a mathematical problem."

"The test of intelligence [is] not how much we know how to do, but how we behave when we don't know what to do." — Holt, John, *How Children Learn*, Dell Publishing Co., Inc., N.Y., 1967, p. 116.

4. Scan all theorems that seem related to the theorem you are trying to prove. (Again, much easier with an Environment!) Work on the steps you feel you can make progress in. Forget about proceeding in a "logical" order. You may be able to prove step 5 long before you prove step 1. Remember that what you are trying to do is build a chain of steps from something that has been proved, to the theorem you are trying to prove. Remember, too, that just as you don't need to have any idea of how a computer program works in order to use it, you *don't* need to understand the proof of a theorem in order to use the theorem itself!

There is something else you should keep in mind when you are working on proofs. Fundamental intuitions — the basic ideas on which proofs are built — are often pictorial or accompanied by almost childlike expressions: "It must repeat!", "The numbers grow farther apart!" Such pictures and expressions seem to come from a "mathematical subconscious" in the sense that they have a peculiar simplicity and that they are not, in this form, in any sense "logical". I believe that this subconscious exists in all students who are attracted to, or are at least curious about, mathematics, regardless of their grades in school. Of course, it is by no means equally developed in all students, or even in all mathematicians. It is important to be aware of the difference between the way this subconscious expresses ideas, and the way they are supported in proofs. Any suggestion that the two are the same, or that intuition is less important than the long, impressive-looking chains of argument on paper, is simply wrong.

Finally, it is important that students as well as experts in mathematical subjects realize that the finished, published proof is not a "thing" in itself but merely the latest stage in a sequence of approximations. It is "clear and concise" only to those who have one way or another passed through the lower stages of understanding. It is not an object, but a stage in a human activity.

An Example of How Bad the Presentation of Proofs in Textbooks Can Be

In the course of writing this chapter, I searched for textbook proofs that I could redo as structured proofs, in order to show the advantages of the latter. It occurred to me that the proof of the Schwarz-Christoffel transformation might be a good example. The transformation is discussed in most elementary courses in complex number theory.

I had no idea just how *bad* the presentation of this proof is, at least in the textbooks at my disposal. I first studied the version in Churchill's textbook. Then, in an attempt to overcome the difficulties I found, I studied Cohn's and Spiegel's texts. (Reference information is given at the end of this chapter.) It was at times hard to believe that each of these texts was discussing the same transformation!

The Example

Let me begin with Churchill's text. The chapter devoted to the transformation begins,

"In this chapter we construct a transformation, known as the Schwarz-Christoffel transformation, which maps the *x* axis and the upper half of the *z* plane onto a given simple closed polygon and its interior in the *w* plane." — Churchill, p. 239.

Well, that is straight-forward enough. The next section is titled, "91. Mapping the Real Axis onto a Polygon" which begins with the sentence, "We represent the unit vector tangent to a smooth directed arc *C* at a point z_0 by the complex number *t*."

If you have understood just the main points of the chapter you are now reading, you should be scratching your head, and perhaps exclaiming to yourself, "*What*?" Exactly what are we proving? Why wasn't that stated *first*? Only a couple of paragraphs later are we told what follows if the arc *C* happens to be a segment of the *x* axis. We are left to write down the exact statement that has just been proved.

Suppose we now look ahead a few pages. We find that on the third page of the chapter, there is a sub-title "The Schwarz-Christoffel Transformation". And that's it! No more sub-titles until the sixth page of the chapter!

The actual transformation is not even stated until the *fifth page*!

It is business as usual — a succession of paragraphs set forth deductions that, at the end, we discover are proofs of facts that appear to have some relevance to the subject at hand!

I found myself thinking of a line from the Bible: "Father, forgive them, for they know not what they do!" But here "them" is the textbook authors.

Let the reader decide if he or she would prefer the above-described deductive *sprawl* (which I am sure exists in any basic textbook on complex number theory the reader owns) to something like the following (on the first page of the chapter):

A Much Better Approach

Theorem. *Given a simple closed polygon of $n \geq 3$ vertices in the w plane, a transformation, F(z), called the Schwarz-Christoffel Transformation, exists that:*

maps the x axis of the z plane onto the boundary of the polygon, and that

maps the upper half of the z plane, excluding the x axis, onto the interior of the polygon.

Proof:

Step 1.

We assert without proof that $F(z)$ must meet the following requirements.

1. $F(z)$ must be continuous.

2. $F(z)$ must map n points on the x axis of the z plane, where $n \geq 3$, onto the n vertices of the polygon.

3. $F(z)$ must map the intervals between the n points on the x axis, onto the sides of the polygon.

4. $F(z)$ must map, one-to-one, the upper half of the z plane, excluding the x axis, onto the interior of the polygon.

Step 2:

The function $F(z) = A\int_{z_0}^{z} (s - x_1)^{-k_1}(s - x_2)^{-k_2}\ldots(s - x_{n-1})^{-k_{n-1}}ds + B$ meets the above requirements, where,

$x_1, x_2, ..., x_{n-1}$ are points on the x-axis, with $x_1 < x_2 < ... < x_{n-1}$;
$x_n = \infty$;

$k_i\,\pi$ = the size of the exterior angle at the vertex $w(i)$ of the polygon;

A, B are complex constants that establish the location and size of the polygon;

[z_0 is undefined in the definition in Churchill]

Proof that $F(z)$ meets requirement 1. See...

Proof that $F(z)$ meets requirement 2. See...

Proof that $F(z)$ meets requirement 3. See...

Proof that $F(z)$ meets requirement 4. See...

□

Now we have the Big Picture before us at the very start. Now we know what needs to be proved. We don't have to plow through a lot of detail in order to find out why we are plowing through the detail!

Textbooks Consulted

Churchill, Ruel V., Brown, James W., Verhey, Roger F., *Complex Variables and Applications,* 3rd ed., McGraw-Hill, Inc., 1976
Cohn, Harvey, *Conformal Mapping on Riemann Surfaces*, Dover Publications, Inc., N.Y., 1967.
Spiegel, Murray R., *Complex Variables*, Schaum's Outline Series, McGraw-Hill Book Company, N.Y., 1964.