DEPARTMENT OF MATHEMATICS
TECHNICAL REPORT

THE ROLE OF LOGIC IN THE VALIDATION OF MATHEMATICAL PROOFS

Annie Selden
John Selden

February 1999

No. 1999-1

TENNESSEE TECHNOLOGICAL UNIVERSITY
Cookeville, TN 38505

THE ROLE OF LOGIC IN THE VALIDATION OF MATHEMATICAL PROOFS[*]

Annie Selden
John Selden

February 2, 1999

Mathematics departments rarely require students to study very much logic before working with proofs. Normally, the most they will offer is contained in a small portion of a "bridge" course designed to help students move from more procedurally-based lower-division courses (e.g., calculus and differential equations) to more proof-based upper division courses (e.g., abstract algebra and real analysis). What accounts for this seeming neglect of an essential ingredient of deductive reasoning?[1] We will suggest a partial answer by comparing the contents of traditional logic courses with the kinds of reasoning used in proof validation, our name for the process by which proofs are read and checked.

First, we will discuss the style in which mathematical proofs are traditionally written and its apparent utility for reducing validation errors. We will then examine the relationship between the need for logic in validating proofs and the contents of traditional logic courses. Some topics emphasized in logic courses do not seem to be called upon very often during proof validation, whereas other kinds of reasoning, not often emphasized in such courses, are frequently used. In addition, the rather automatic way in which logic, such as modus ponens, needs to be used during proof validation does not appear to be improved by traditional teaching, which often emphasizes truth tables, valid arguments, and decontextualized exercises. Finally, we will illustrate these ideas with a proof validation, in which we explicitly point out the uses of logic. We will not discuss proof construction, a much more complex process than validation. However, constructing a proof includes validating it, and hence, during the validation phase, calls on the same kinds of reasoning.

Throughout this paper we will refer to a number of ideas from both cognitive psychology and mathematics education research. We will find it useful to discuss short-term, long-term, and working memory, cognitive load, internalized speech and vision, and schemas, as well as reflection, unpacking the meaning of statements, and the distinction between procedural and conceptual knowledge.

Proofs and Validations

In the mathematics education research literature, "proof" has been used in a variety of ways. The proofs we will discuss here will be the kind that undergraduate students normally see in their mathematics texts and classes. These are specialized natural language arguments which deductively establish the correctness of theorems and are similar to, although often shorter and more detailed than, the proofs found in mathematics research journals. Proofs have been discussed from various points of views – structure [12], explanatory power [9], students' conceptions [13], students' reasoning errors [20], and specialized formats [6, 11]. However, all

[*] This paper was originally presented at the DIMACS Symposium on Teaching Logic and Reasoning, Rutgers University, 25 - 26 July 1996.

[1] It appears to be widely believed among mathematicians that logic courses, per se, are of little benefit in helping students learn to understand and create proofs. Research in mathematics education has not directly examined the accuracy of this belief, but it is consistent with Nisbett's, et al's work [15] on the effects of logic courses.

of these are beyond the scope of this paper, which focuses on proofs as establishers of the correctness of theorems and the role of logic in validating proofs.

When a mathematician reads a proof in order to check its correctness, he or she does something quite different from reading a newspaper or book. Between the actual reading of the sentences, a mathematician may assent to claims, draw inferences, ask and answer questions, bring in outside knowledge, construct visual images, or develop subproofs. The entire proof may even be reconstructed and supplemented using one's personal domain specific knowledge, which may itself be altered during this construction process. Such a personal version of a proof may be reviewed from beginning to end several times, and some mathematicians may even produce an abbreviated version which can be encompassed within a single train of thought, i.e., in a monologue of internalized speech and vision, uninterrupted by significant attention to other things.

We call this mental process validation. It is a form of reflection that can be as short as a few minutes or stretch into days or more, but in general, it is much more complex and detailed than the corresponding written proof. Although its ostensible purpose is to establish the correctness of a theorem, it is often used when there is considerable reason to believe a theorem is correct. In such cases, mathematicians act as if the theorem were in question. Indeed, validation appears to be instrumental in mathematicians' learning of new mathematics. Skill at validation is an implicit part of the mathematics curriculum, but is rarely explicitly taught. Beginning undergraduate mathematics students may well be unaware of its existence and importance.

Memory and Validation

The relationship between validation, especially validation errors, and the way mathematical proofs are normally written can perhaps best be viewed by considering the demands on the capacity of short-term memory, so we begin our analysis with a brief discussion of memory.

Except for processes associated with perception, there are three main kinds of memory systems – short-term, long-term, and working [1]. The half-life of short-term memory is approximately fifteen seconds and its capacity is approximately seven chunks of information [14]. These chunks can consist of any information that can be thought of as a unit, e.g., words, familiar patterns of chess pieces, or the Pythagorean theorem. One is aware of the contents of short-term memory and it is readily available for reasoning. The rest of memory is called long-term and has a very large capacity. However, one is not aware of its contents and it is not directly available for reasoning. Information in long-term memory can be activated, i.e., represented in short-term memory. Finally, working memory consists of short-term memory, including information activated from long-term memory, and the reasoning and control mechanisms that swap information between short-term and long-term memory.

In analyzing validations, which are trains of thought rather than simple incidences of recall or inference, we will also discuss two subcategories of long-term memory. We will call memory that lasts a number of years, one's knowledge base. In the case of validation, this is one's knowledge of mathematics and its conventions, as well as of logic and reasoning. On the other hand, the more ephemeral, partially activated memory arising from a validation itself, we call local memory. It might include that a proof started with "Let x be a real number" or that one has used the triangle inequality in justifying some part of the proof. One is not continually aware of this local memory. It is not part of working memory, but is very easily accessible to it. Such local memory grows and persists during a validation, but may well be forgotten when it is no longer useful.

The Style in which Proofs are Written

The style in which proofs are written greatly influences validation, increasing the reliability of proofs, which are

mathematicians' form of natural language argument, and are, as in other disciplines, the ultimate arbitrator for rationally establishing (public) knowledge. Alternative, simplified, "clearer" styles, such at the two-column proofs of high school geometry have been devised, and it seems plausible that such approaches might produce more student successes and generally improve reasoning ability. However, using such an alternative style, students often come to view the form of proofs, rather than their substance, as what counts [19].

In practice, even geometry students who are successful in making two-column proofs may not connect proofs with reasoning in other settings. A hint of this problem came from a colleague who, in teaching a noncredit college geometry course, reported than one of his students incredulously asked, "You mean proofs can contain words?" More detailed analyses have revealed successful geometry students who fail to see the utility of deductive arguments in making geometric constructions, even when a relevant proof, from which one could extract the proper construction, remained in full view on the blackboard [18, pp. 355 - 357]. Another "clearer" style with more reasons, consistent symbols, and semi-fixed format has been developed for use in discrete mathematics by Gries and Schneider [6, 7].

Although the style in which proofs are written has evolved within the mathematics community mostly without plan or design, it is very robust and has a number of identifiable features. Perhaps the clearest of these is that proofs are not reports of the proving process. No one publishes in a proof that he or she originally thought, say, that commutativity would play a key role, but subsequently discovered it led nowhere, or that he or she had an inspiration regarding a new concept, such as quaternions, while alighting from a bus. Such observations, while interesting, are left to historical accounts and works like Hadamard's on the psychology of mathematical invention [8].

Indeed, we will argue that, in contrast to reporting the proving process, the main distinguishing features of proofs all tend to minimize the probability of validation errors (by experts). As an inescapable consequence, the validation process may take longer, the domain knowledge required may be increased, and novice validators may be perplexed.

Another global feature of proofs is their terseness. There is little of the redundancy that enhances understanding in everyday verbal communication. One does not say essentially the same thing in several ways. For example, while a local restaurant might post a sign saying "Our soups are prepared fresh each and every day," in a proof involving a homomorphism, f, that has just been shown to be an isomorphism, one does not normally remind readers that this means that f is one-to-one and has zero kernel. Such terseness is not a matter of oversight or ignorance of the needs of one's readers. Proofs are written for an idealized reader – mathematicians know from their own experience that differing validators have differing knowledge with which to validate a given proof.

Few people would write in such terse prose were their primary intention to sway as many readers as possible. This concise style has the effect of placing much of the locus of control with the validator, instead of the writer of a proof. While validators know they can supplement proofs, they may well resist omitting or altering significant parts of them (concerned they might overlook something important). Thus, a terse style allows a validator's supplemental proofs to "fit" closely with his or her own knowledge base and avoids mentally cycling through unnecessary or distracting material – all of which helps minimize demands on the validator's working memory and the probability of validator error.

Here we are suggesting only an effect of a terse style, not that this effect is a conscious goal of authors, who may think more about avoiding publication of "obvious" material and journal page limitations. In addition, terseness is a matter of balance – no proof at all would shift all control (and the burden of constructing the proof) to the validator, which is hardly desirable.

However abstract a proof's subject, the style of its writing tends to be "concrete", that is, wherever possible quantified statements are avoided through use of universal instantiation. For example, one would avoid including "For every positive number x, $f(x) < 0$" in a proof. Instead, one might first write "Let x be a positive number" and later observe that "$f(x) < 0$."

The quantified version is precise and may well represent the information the author of a proof has in mind. However, from the perspective of a validator, it is likely to (momentarily) require more of the limited resources of working memory than either of its replacements. While a portion of propositional calculus (e.g., modus ponens) appears to be immediately accessible to everyone's working memory, this appears to be less so for predicate calculus. For example, beginning university students often lack an understanding of the relationship between negation and quantifiers, thinking that "All x are not y" is the negation of "All x are y." [2]

It has been noted that the ability to draw correct inferences in predicate calculus can be improved by the mere suggestion that one examine arguments in terms of single instances of the variables involved [22]. The properties of one instance can readily be held in short-term memory. However, more than one such instance may be required in a given argument and this can place considerable burden on working memory, and hence, increase the probability of error.

Unquantified sentences such as "Let x be a positive number" and "$f(x) < 0$" can be regarded as about a single arbitrary, but fixed object so only one's propositional calculus abilities need be invoked when drawing inferences. Thus, a concrete style of proof reduces demands on a validator's (and a prover's) working memory; thereby, reducing the chance of processing errors.

Another stylistic feature of proofs is the insistence that the symbols correspond in a one-to-one way with the objects under consideration. That is, if one lets x represent a number (even an arbitrary, but fixed one), one does not later let x represent another number, or even let y also represent the first number. This also extends across domains for which differences in context would allow differing meanings of a symbol to be understood. That is, one does not normally let X represent both a topological space and its dimension, even though the differing contexts of spaces and numbers would allow the meaning to be discerned.

This restriction might be thought to apply so broadly to discourse as not to be a distinguishing characteristic of proofs. However, we have observed that novice undergraduate proof writers often violate it, suggesting it is not generally followed outside of mathematics. The benefit of this feature of proofs is not to the prover for whom it amounts to a restriction, but to the validator. If there is a chance that a symbol might refer to two distinct objects, then a validator would require additional information on this at each occurrence of the symbol. To constantly retain this information in short-term memory would be a considerable burden, and thus, increase the probability of error. If, on the other hand, this information were contained within local memory, a validator would not be continually aware of it and would have to recall it when needed. This additional activity would also draw on the resources of working memory and increase the probability of error. In either case, ignoring this stylistic feature would tend to increase validator errors.

In general discourse, there are a number of characteristics of arguments which, while not essential, tend to make them more widely understood or convincing. These include the tendency to provide detailed explanations or supporting evidence and the inclusion of definitions and rewordings (a form of redundancy) for purposes of clarity or completeness. However, such features are usually reduced to a minimum or avoided entirely in proofs.

Assertions in proofs, as in other forms of argument, can be followed by reasons or explanations, but in proofs, these are usually kept very brief, perhaps only the mentioning of a theorem or definition. Even much of the detailed reasoning leading to a (sub-)claim may be omitted in proofs. This is not a matter of oversight nor of attributing unrealistic knowledge or sophistication to validators. Often authors write out the details for themselves and then remove them from the final published proof. Similarly, overviews, which in other forms of argument can support understanding, are reduced to a minimum in proofs. Typically, these consist only in remarking that a proof is by contradiction or involves cases. Also, questions can sometimes be used to focus a reader's attention on important points, thereby making a general argument more convincing or memorable. However, this way of writing (and teaching) is forgone in proofs, even proofs in expository or pedagogical texts.

Definitions, especially unfamiliar ones, are sometimes included in general arguments, and students who are new to proving theorems tend to insert them in their proofs. However, general mathematical definitions (i.e., those useful beyond the confines of a single proof) are almost never quoted within proofs. For example, if one has just proved that f is a homomorphism, one would not also ask readers to recall that a function g is a homomorphism if and only if for every x and y, $g(x+y) = g(x) + g(y)$. This is also true for definitions introduced for the first time, and hence, not familiar to readers. Furthermore, general arguments sometimes

[2] This difficulty with negation of universal quantifiers seems common in our culture. A recent ad on television proclaimed, "All people are not created equal," presumably intending the negation of "All people are created equal."

include the entirety of a logical argument, such as one employing modus ponens, perhaps to make them more convincing (or inescapable). In a mathematical setting, for example, this would amount to saying, "We know differentiable functions are continuous and $f(x)$ is differentiable, so $f(x)$ must be continuous," rather than simply asserting "$f(x)$ is continuous" and leaving affirmation to the validator.

Finally, it appears that most people depend heavily on redundancy for comprehension of everyday text. However, in proofs this is avoided. Even the symbols in proofs may not explicitly represent the complexity of the structures to which they refer. For example, in dealing with parametric equations, one can write "x, y" rather than "$x(t), y(t)$," and in referring to a group, one normally writes "G," rather than "$(G, +)$."

The above features, while seeming to enhance general arguments, are avoided or minimized in proofs – whose main purpose is not explanation or enhancement of readers' understanding[3]. Such a minimalist style does, however, tend to reduce the moment-to-moment load on working memory. One effect of avoiding everything inessential to a proof is that an expert validator need not devote working memory to constantly deciding what is or is not important to check – everything is important. Another effect is to enhance a validator's control over his or her chunking. Were inessential material included in proofs, a validator might feel compelled to read it, even on multiple passes through a proof, making it more difficult to grasp the essence of the proof. But if the same information were chunked in a validator's local memory, it would be readily accessible when needed. While a validator may require considerable effort to get such information into local memory initially, this would occur prior to, and not interfere with working memory while, reviewing the entire proof.

There appear to be three main impediments to an (expert) validator's reliably determining the correctness of a proof: lack of domain knowledge (including logic), domain misconceptions, and working memory overload. Lack of specific domain knowledge, e.g., not knowing or not thinking of using the triangle inequality, is easily detectable during validation, which can be suspended while the validator locates or derives such information. This causes no threat to the integrity of proofs and does not influence the style in which proofs are written. Misconceptions, e.g., believing all continuous functions are differentiable, are not easily detectable by a single validator, although two validators who disagree on the correctness of a proof could bring these to light. The most grievous misconceptions are those shared by the prover and a validator, but nothing about the style in which proofs are written would enable these to be discovered. Thus, the only effect upon the integrity of proofs grounded in the style of their writing comes from working memory overload, and we have argued above that proofs are written so as to minimize this. This is done, even at the expense of making validations longer and proofs less persuasive or less widely understood (to naive readers).

Beginning Logic Courses and Validation

Perhaps the first thing to notice about beginning logic courses is that they are not essential to mathematics. Many mathematicians have started their work with proofs without ever having taken a logic course and without any familiarity with symbolic logic. Such mathematicians must have first validated proofs using only their natural, everyday reasoning ability, which was then perfected through critical use. This natural reasoning or logical ability is, in some degree, universal, i.e., everyone seems to correctly use modus ponens, universal instantiation, and a few other aspects of propositional and predicate calculus. However, this pervasive natural ability by no means includes all, or even most of, propositional and predicate calculus. For example, many people do not consider modus tollens arguments as valid (48% in a study by Rips, [16, p. 177]), confuse the conditional with the biconditional, and incorrectly negate multiply quantified statements. Calculus teachers and textbook writers routinely warn students that the theorem, "If a series converges, then the limit of the n^{th} term is zero," provides a test for divergence, not for convergence [23, p. 581], usually to no avail.

This untrained, natural ability has been studied extensively by cognitive psychologists [16, 5], but not much seems to be known about the way it is perfected through critical use or formal courses. It would be

[3] Some mathematics education researchers have noted that the style in which many proofs are written is ill-suited for pedagogical purposes. They prefer "proofs that explain" over "proofs that merely prove" [9].

informative to see the research techniques of mathematics education applied to the learning of logic[4].

Topics Not Much Used in Validation of Proofs

Much of what is in beginning logic courses is little used during the validation of proofs, e.g., truth tables and Venn diagrams are not directly used. While logical equivalences, implications, and the validity of arguments from propositional calculus are all used in proofs, only the simpler forms occur very often. Complex logical argument can be broken down into simpler, more familiar arguments leading from premises to conclusion, and mathematical proofs tend to do this – consequently, validators rarely confront great logical complexity. Thus, what validators need most is to be very familiar with the simpler aspects of propositional calculus (somewhat beyond what people can normally do without training). While beginning logic students are occasionally asked to analyze Lewis Carroll type "nonsense" arguments involving a half dozen or so premises[5], real expertise at handling this sort of logical complexity is not called for in mathematical proofs, despite their length and domain specific complexity.

While some predicate calculus is needed to validate proofs, only the simpler aspects of it occur – those which tend to avoid quantifiers. For example, if a theorem is about all integers, the proof may begin, "Let n be an integer," after which the argument is about a fixed, but arbitrary n. However, predicate calculus is essential in establishing the link between the statement of a theorem and its proof and in bringing information from (outside) definitions and theorems into a proof.

The probability that a particular argument form, e.g., DeMorgan's laws, will occur in any randomly selected proof appears to be quite small. It would be instructive to know which aspects of logic courses occur most often in proofs that undergraduate students normally encounter.

Topics Underemphasized in Beginning Logic Courses

There are several kinds of logic or logic-related topics, which though useful in validation, are not emphasized in traditional logic courses, perhaps because these are seen as unimportant or transparent. We will discuss substitution, interpreting the logical structure of informally written statements, applying theorems and definitions to situations in proofs, understanding the language of proofs, and recognizing logical structures in the context of mathematics. We suspect all of these are difficult for students just beginning their work with proofs and are amenable to explicit instruction[6].

Most college students have no problem substituting "3" for the "x" in $x + 2 = 5$ and one might hope that more complex substitutions were equally transparent for them. After all, the constellation of concepts consisting of substitution, variables, and quantifiers is used by quite young children when understanding the natural language sentence, "You will all get a piece of candy." If arithmetic teachers regularly took a small amount of time away from concrete calculation to build on these ideas, students might gradually come to see variables as functioning like pronouns in English sentences, with similar substitution properties. For

[4] Mathematics education researchers might find the work of diSessa [3] on p-prims a useful guide here. He and other physics education researchers have often been concerned with altering students' pre-existing conceptions of the world (those originating in everyday experience) and bringing them into alignment with current scientific theory – a situation that seems relevant here. To date, much of mathematics education research has been devoted to studying the construction of concepts, like function, limit, or integral, which have not spontaneously arisen from students' everyday experiences.

[5] An example is the following: If he goes to a party, he does not fail to brush his teeth. To look fascinating it is necessary to be tidy. If he is an opium eater, then he has no self-command. If he brushes his hair, he looks fascinating. He wears white kid gloves only if he goes to a party. Having no self-command is sufficient to make one untidy. Therefore, ...

[6] By "explicit instruction," we mean a variety of instructional techniques including explorational activities and group work, as well as the more traditional lecturing and homework exercises. We are suggesting certain topics, now omitted or underemphasized, should be part of the explicit, rather than the implicit curriculum. Most students need help learning these topics, or even knowing they are important. Unfortunately, many of these have been considered part of "mathematical maturity" in the past.

example, children could be asked, “Which of the numbers 7, 16, 26, 13, is greater than 5? All, none, some?” Apparently, this does not happen. The concept of variable appears to be first introduced in algebra, in a way disconnected from previous experience.

Perhaps partly as a result of this background, beginning calculus students often have difficulty with long or multi-level substitutions such as those required to calculate the derivative of a particular function at a point using the definition. When applying complex theorems to their own research, mathematicians sometimes make explicit substitutions for each occurrence of a variable in order to ensure that they have grasped the exact meaning. We suspect that mathematicians make such substitutions in several steps, keeping track of them in local memory, aided by few written symbols. However, many beginning students may lack such procedural skills or the tendency to use them, and hence, introduce errors by attempting to carry out complex multi-level substitutions in working memory. Whatever the origin of students' difficulties with substitution, they could probably benefit from additional explicit instruction in a variety of multi-level substitution problems[7].

Another logical difficulty students have when validating proofs is interpreting the logical structure of theorems and definitions written in the informal style of most textbooks. Theorems and definitions can be expressed formally in natural language, explicitly naming all variables and employing the usual logical connectives and quantifiers (and, or, not, if-then, if-and-and-only-if, for all, there exists). However, theorems are often expressed less formally, that is, their wording departs significantly from this more formal style. For example, “$f + g$ is continuous at a point, provided f and g are” is informally stated. In contrast, “For all functions f, all functions g, and all real numbers a, if f is continuous at a and g is continuous at a, then $f + g$ is continuous at a” is its more formal counterpart.

Because theorems and definitions are so often expressed informally, there are probably some cognitive benefits to the practice. Indeed, we suspect one reason authors favor an informal style is to make theorems and definitions more “memorable,” more easily remembered and recalled. However, formally stated theorems can be relatively easily linked to what we call a proof framework, i.e., the portion of a proof that can be written without knowledge of the specific mathematical concepts involved and which determines whether a proof actually proves the theorem it claims to prove, rather than some other theorem. For example, the theorem mentioned above has the following simple proof framework: Proof: Let f and g be functions and let a be a number. Suppose f and g are continuous at a. . . . Thus, $f + g$ is continuous at a. QED.

Since a proof can contain subproofs, it may also contain nested proof framework structures. However, there is evidence that mid-level mathematics majors are unable to reliably unpack the logical structure of informally stated calculus theorems and definitions. Thus, it is unlikely that they can reliably recognize proof frameworks while validating proofs [21]. This suggests that explicit instruction, both in unpacking the logical structure of informally written statements and in the recognition and construction of proof frameworks would be very useful.

Proof frameworks, however, are not the only link between the (largely propositional calculus) language of proofs and the predicate calculus language of (outside) definitions and theorems. In validating a proof, one applies theorems and definitions to the mathematical situations arising within a proof. We found that many mid-level undergraduate mathematics majors, in a “bridge course” designed to introduce proofs and mathematical reasoning, were unable to reliably translate informally written mathematical statements into their predicate calculus equivalents – a process we call unpacking, i.e., associating with an informally worded mathematical statement a logically equivalent formal statement, including all those logical features that are often understood by convention, rather than explicitly expressed. Just 8.5% of the 94 unpacking attempts we collected from six small sections of a “bridge course” were successful. For example, given the following (incorrect, but meaningful) mathematical statement: For $a < b$, there is a c so that $f(c) = y$ whenever $f(a) < y$ and $y < f(b)$, students wrote such things as: For all $a < b$, there exists c such that for all f and for all y, $f(a) < y$ and $y < f(b)$ implies $f(c) = y$. Note that if f were a continuous function, this statement would be the Intermediate Value Theorem as stated in many calculus texts. However, the interchange of universal and existential quantifiers radically alters the meaning of the original statement. It now asserts

[7] Epp [4] has several sections on substitution into combinatorial formulas, but much less than we think students need.

the rather bizarre idea that in every interval (a, b) there is a peculiar number c, at which each nonconstant f takes on all values y between $f(a)$ and $f(b)$. [21, pp.136 - 139] This and other examples like it suggest that mid-level math majors are unable to understand and correctly apply quantified statements in context, and thus explicit instruction in their application would probably be useful. One might ask students of a particular theorem, “What does this theorem say here?”, about various mathematical situations.

Also, the language of proofs, and in particular, the meanings of a few words, might receive some explicit instruction, just as the meaning of “or” and “if and only if” are explicitly taught in logic. For example, “let” can be used in at least three distinct ways in proofs. In proving “For all numbers $\epsilon \ldots$”, a proof might include “Let ϵ be a number.” In that case one needs to be sure ϵ has not occurred earlier in the proof and that ϵ is not restricted in some way. In proving, “ $\ldots$ there is a $\delta \ldots$,” a proof might contain “Let $\delta = \frac{x}{2}$.” Here also δ should not have appeared earlier in the proof, but in contrast to the previous use of “let,” δ must arise from the previous portion of the proof and either be explicitly described or shown to exist. It would be clearer to say, “Set $\delta = \frac{x}{2}$,” but that is not common usage. The third use of “let” occurs in the proofs of theorems containing conditionals, e.g., “ $\ldots$ if f is continuous, then $\ldots$”. Here “Let f be continuous” means “Suppose f is continuous.” In this case, f can, indeed should have, occurred previously in the statement of the theorem or its proof. Furthermore, suppositions can be made about f more than once, as for example, when subsequently, an additional property about f is to be proved by contradiction. These three uses of “let,” for universal instantiation, assignment, and supposition, are not, to our knowledge, often explained to mathematics students, although we make a practice of doing so in “bridge courses[8] ”.

We turn now to context or “situatedness.” Beginning logic courses often seem to present logic very abstractly, in essence as a form of algebra, with examples becoming a kind of applied mathematics. Students are asked to convert natural language mathematical situations into symbolic form, analyze them, and then convert these back again to get the result. This kind of application does occur in validations, although often the links to symbolic logic are not so explicit. For example, a validator may be required to decide whether an assertion is warranted by recognizing it as the conclusion of a valid argument (written in natural language) with premises that are scattered through the previous portion of the proof. This is quite different from the abstract, symbolic setting of many exercises in beginning logic courses. Since there is evidence that the application of knowledge can be remarkably dependent on the situations in which it is learned[9], the reasoning involved in validation may depend as much on students' common sense background as on logic courses. Thus, it would probably be useful to teach logic in more realistic contexts, e.g., those provided by fragments of proofs. For example, students might be asked to select and justify one of several possible assertions that might follow from a proof fragment in which the premises of a valid argument are embedded.

Ways of Knowing Logic

Since the 1980s, research in mathematics education has concentrated on how students develop conceptual knowledge, rather than procedural knowledge – it has focused on understanding, rather than performance. However, in complex trains of thought such as validations, both conceptual and procedural knowledge appear to be inextricably integrated into cognition. In particular, in assenting to an assertion in a proof, a validator may simply recognize that it is the consequence of a familiar embedded logical argument, such as modus ponens. In that case, the validator's only conscious response may be to say “OK.” That is, whatever the validator does is below the conscious level of internalized speech or vision. One could think of the validator as activating a small schema corresponding to modus ponens, i.e., filling in the blank spaces between the connectives that form the pattern of modus ponens[10]. There may be other ways to view such automated

[8] Some “bridge course” texts do list some of the various ways the conditional (and biconditional) can be expressed in mathematics, e.g., if p, then q; p implies q; q, provided p; q, whenever p, p is sufficient for q; q is necessary for p [10].

[9] Within mathematics education, a growing number of researchers consider all learning as situated, i.e., concepts and procedures are inexorably linked to the context in which they are learned [2].

[10] Schemas have also been called scripts or frames and the blanks spaces referred to as variables or slots. They have been described as unconscious cognitive structures that underlie knowledge and skills. Schemas are active processors of knowledge

responses of the validator, but in any case, he or she is unaware of them in his or her conscious mind. Thus, the validator has no opportunity to check them, say, in the way a sequences of statements occurring in internalized speech might be broken into parts, rehearsed slowly and checked individually. However, the validator's conscious, automated response, "OK," may leave working memory largely free for other tasks.

This situation is somewhat analogous to that of an elementary student who simply "knows" that "8 times 7 is 56," rather than having to think "7 times 7 is 49 and 49 plus one more 7 is 56." This kind of procedural knowledge is very efficient, but if the student thought "8 times 7 is 57" and suspected an error, the only way to resolve this, would be to switch to a more conceptual approach.

Since, in validations, some logic appears to be applied in this procedural, automated way, it should be beneficial to teach a small part of beginning logic for procedural (as well as conceptual) knowledge, i.e., using drill and practice, especially in appropriate (mathematical) contexts.

A Sample Validation[11]

The following example illustrates some of the ideas discussed in this paper. For this purpose, we have selected a familiar calculus theorem, but are not suggesting what follows should be used in teaching calculus. Nor is this example meant to illustrate how a theorem or its proof might be invented since it does not consider such things as intuition, insight, visualization, false starts, or cognitive strategies.

Although the theorem we have selected is similar to Leron's Theorem 2.2 [12], our purpose is different. Leron presents the argument in a nontraditional, top-down, structured way for pedagogical purposes in order to facilitate students in grasping its basic structure, as well as the relationship between relevant mathematical concepts. We, on the other hand, give a traditional linear proof which facilitates the validation process.

Validations can be done in various ways – they depend heavily upon the individual validator, his or her domain knowledge, and his or her knowledge of validation procedure.

We present a plausible, but imaginary, transcript detailing the main questions, answers, and comments of our (hypothetical) validator. This is followed by a (short) analysis of the logic needed for this validation. The transcript is written entirely as internalized speech, but an actual validator might also draw some pictures, think out loud, or visualize portions of the proof. For reference, we have numbered the sentences in the proof, using parentheses and brackets to enclose reference numbers and our comments.

Theorem. $f + g$ is continuous at a point, provided f and g are.

Proof: (1) Let a be a number and let f and g be functions continuous at a. (2) Let ϵ be a number greater than 0. (3) Note that $\frac{2}{2}$ is greater than 0. (4) Now because f is continuous at a , there is a δ_1 greater than 0, such that for any x_1 , if $|x_1 - a| < \delta_1$, then $|f(x_1) - f(a)| < \frac{2}{2}$. (5) Also there is a δ_2 greater than 0, such that for any x_2 , if $|x_2 - a| < \delta_2$, then $|g(x_2) - g(a)| < \frac{2}{2}$. (6) Let δ be the smaller of δ_1 and δ_2. (7) Note that δ is greater than 0. (8) Let x be a number. (9) Suppose $|x - a| < \delta$. (10) Then $|x - a| < \delta_1$, so $|f(x) - f(a)| < \frac{2}{2}$. (11) Also $|x - a| < \delta_2$, so $|g(x) - g(a)| < \frac{2}{2}$. (12) Now $|(f(x) + g(x)) - (f(a) + g(a))| = |(f(x) - f(a)) + (g(x) - g(a))| \leq |f(x) - f(a)| + |g(x) - g(a)| < \frac{2}{2} + \frac{2}{2} = \epsilon$. (13) Thus $|(f(x) + g(x)) - (f(a) + g(a))| < \epsilon$. (14) Therefore $f + g$ is continuous at a. QED

Our (hypothetical) transcript follows.

(1) "OK, a , f , and g can be introduced this way because they haven't been used before – this is the first line of the proof. Why is the argument starting with something about f and g when the theorem starts with $f + g$? Does the argument really prove the theorem? The theorem really means: For every number

that have been likened to small theories, procedures, and parsers. They are used in perception, text comprehension, memory, learning, and problem solving [17].

[11] This theorem and validation appear in [21, Appendix 1].

a and every function f and every function g , if f and g are continuous at a , then $f + g$ is continuous at a. [This is an instance of unpacking the original informal statement into a more formal one.] So the proof should start with an arbitrary a , f , and g and assume f and g are continuous at a. That amounts to (1). After some argument, the proof should end with $f + g$ is continuous at a, which turns out to be the last line, (14) . OK, if the argument checks out, the theorem is true." [That is, a proof framework obtained from the statement of the theorem agrees with the one embedded in this argument].

(2) "OK, ϵ hasn't come up before, so it's permissible to let it represent any number."

(3) [This comes from (2) and a bit of domain knowledge.] "Half a positive number is positive. So (3) is OK, but why say it?" [The answer to this would be helpful, but is not essential to the validation.]

(4) "This is supposed to come from 'f continuous at a' which means: For every δ greater than 0 there is a δ greater than 0 so that for all x , if $|x - a| < \delta$, then $|f(x) - f(a)| < \epsilon$. Here the symbols may mean something different from those in the proof. [This comes from domain knowledge, perhaps with some unpacking.] This says 'for every ϵ' so this part can be dropped with ϵ replaced by any positive number. So $\frac{2}{2}$ can replace ϵ . Also, the x can be renamed as x_1 and the δ as δ_1 throughout. That gives me (4)."

(5) "There is no 'because' here. But, it looks like (4) . So I can start with the definition of 'f is continuous at a' and change f to g. Just as before in 'ϵ greater than 0', I can replace ϵ by $\frac{2}{2}$. Also the x_2 can replace x and δ_2 can replace δ. So (5) is OK."

(6) "Now δ has not been used before in the proof, so it can represent any number made out of numbers already there." [It will turn out later that δ has to be treated this way, and cannot be arbitrary like ϵ.]

(7) [This comes from some domain knowledge.] "The smaller of two positive numbers is positive."

(8) "OK, x hasn't been used yet so it can represent any number."

(9) "This inequality is a statement because x , a , and δ already represent numbers and one can suppose any statement. But why?" [It turns out this supposition is part of proving $f + g$ is continuous at a . Knowing that would help the validator keep track of the argument, but it is not essential to the validation.]

(10) "The first part of this assertion follows from (9) and (6) [and simple domain knowledge – if a quantity is less than the smaller of two numbers, then it is less than either number separately.] The second part comes from the first part and the implication in (4) , replacing x_1 by x . This is permissible because (4) said 'for any x_1'. " [In addition, (10) is using modus ponens.]

(11) "The first part comes from (9) and (6) , just as in (10). The second part comes from the first part and (5) , just as in (10)."

(12) "The first equality comes from $(a + b) - (c + d) = (a - c) + (b - d)$. [Some domain knowledge.] The inequality comes from the triangle inequality, $|p + q| \leq |p| + |q|$, where $f(x) - f(a)$ replaces p and $g(x) - g(a)$ replaces q. Next the strict inequality comes from the last parts of (10) and (11) [and some domain knowledge, namely, $a + b < c + d$ can be inferred from $a < c$ and $b < d$]. The final equality is just arithmetic."

(13) "This follows from (12) [and several instances of the transitive property for real numbers, e.g.,

$a \leq b$ and $b < c$ implies $a < c$].”

(14) “This must depend on the definition of continuous at a, applied to $f+g$. That is, for every ϵ, there is a δ greater than 0, such that for every x , if $|x-a| < \delta$, then $|(f+g)(x) - (f+g)(a)| < \epsilon$.”

“Now I need to check that the definition of continuity is satisfied and that amounts to proving a theorem. [The validator, in effect, next constructs a proof framework.] Let ϵ be an arbitrary real number greater than 0. That’s (2) of the proof. Let δ be a real number greater than 0, which can depend on ϵ but not on x. That’s (6) and (7) of the proof. Let x be an arbitrary real number. That’s (8), which comes after (6), so δ doesn’t depend on x. Suppose $|x-a| < \delta$. That’s (9). Then, after some argument has been given, $|(f+g)(x) - (f+g)(a)| < \epsilon$. This doesn’t look like (13), but it means the same thing because $(f+g)(x)$ means $f(x)+g(x)$ and $(f+g)(a)$ means $f(a)+g(a)$.” [Here the validator does some notational unpacking.]

“So (14) is OK.”

“Now the entire argument is OK, and at the beginning, I checked that if the argument was OK, then it would prove the theorem.” [The validator decided, as part of checking (1) , that the proof framework of the argument was appropriate for the statement of the theorem.]

This completes the validation.

The proof in this example is fairly detailed – some might say overly so. Many proofs omit lines like (1) which serve to “introduce” a, f, and g. An expert validator would not need (1). The importance of (1) however is that it emphasizes the need to check that a, f, and g are arbitrary. If this were not the case – say f somehow depended on g or a – then an expert validator would notice the error even though (1) were missing. That is, he or she would act as if (1) were there should arbitrariness fail, but might not even notice its absence otherwise.

All or part of (12) might also be omitted from the proof. In that case, the validator might add the omitted portions to his or her own expanded version of the proof.

An actual validator might well carry out the validation process in this example much more quickly than the transcript can be read. On the other hand, if (12) were omitted from the proof, a relatively inexperienced student might take some time to realize that the key to proving (13) was the triangle inequality. Furthermore, if the need to insert an argument to justify (13) were not noticed until (14) was checked, another problem might arise. The checking of (14) is sufficiently complex that stopping for some time to justify (13) might overload working memory and cause part of the checking of (14) to be neglected. To guard against this sort of potential error, some validators make multiple passes through a proof.

The proof we have discussed here is short and “structurally simple,” consisting only of a brief proof within a proof. Many proofs are much longer and structurally more complex, and hence, require more complicated validations.

Let us now examine the above validation for uses of logic or logic- related topics. For example, in (1), a validator must know which of the three uses of “let” is intended, unpack the informally written statement of the theorem to sort out the quantifiers, the premises, and the conclusion, construct a proof framework, and check that the first and last lines of the proof agree with this framework. These are all things not taught in traditional logic courses. In addition, as the validation proceeds, there are a number of instances of substitution combined with application of the definition of continuous, e.g., in (4), (5), (10), (12), and (14). This is also not taught in traditional logic courses. The argument is largely concrete – it deals with arbitrary, but fixed instances of a, f, g, x, and ϵ. The logic used seems confined to universal instantiation, e.g. in (1), (2), and (8), and modus ponens, e.g., in (10), (11), and (13).

References

[1] Baddeley, A. (1995). Working memory. In M. S. Gazzaniga (Ed.), *The Cognitive Neurosciences* (pp. 755-764), Cambridge, MA: MIT Press.

[2] Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning, *Educational Researcher*, 18(1), 32-42.

[3] diSessa, A. (1993). Toward an epistemology of physics. *Cognition and Instruction*, 101 (2 & 3), 105-225.

[4] Epp, S. (1990) *Discrete Mathematics with Applications*, Belmont, CA: Wadsworth.

[5] Garnham, A. & Oakhill, J. (1994). *Thinking and Reasoning*, Oxford, UK: Blackwell.

[6] Gries, D. & Schneider, F. G. (1995). A new approach to teaching discrete mathematics. *Primus*, 5(2), 113-138.

[7] Gries, D. & Schneider, F. G. (1995). Teaching math more effectively, through calculational proofs. *American Mathematical Monthly*, 102(8), 691-697.

[8] Hadamard, J. (1945). *An Essay on the Psychology of Invention in the Mathematical Field*, Princeton, NJ: Princeton University Press.

[9] Hanna, G. (1989). Proofs that prove and proofs that explain. *Proceedings of the Thirteenth International Conference for the Psychology of Mathematics Education*, Vol. 2, Paris, France, 45-51.

[10] Kurtz, D. C. (1992). *Foundations of Abstract Mathematics*, New York, NY: McGraw-Hill.

[11] Lamport, L. (1995). How to write a proof. *American Mathematical Monthly*, 102(7), 600-608.

[12] Leron, U. (1983). Structuring mathematical proofs. *American Mathematical Monthly*, 90(3), 174-184.

[13] Martin, W. G. and Harel, G. (1989) Proof frames of preservice elementary teachers, *Journal for Research in Mathematics Education*, 20(1), 41-51.

[14] Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information, *Psychological Review*, 63, 81-97.

[15] Nisbett, R. E., Gong, G. T., Lehman, D. R., & Cheng, P. W. (1987). Teaching reasoning. *Science*, 238, 625-631.

[16] Rips, L. J., (1994). *The Psychology of Proof: Deductive Reasoning in Human Thinking*, Cambridge, MA: MIT Press.

[17] Rumelhart, D. E. (1980). Schemata: The building blocks of cognition. In R. Spiro, B. Bruce, & W. Brewer (Eds.), *Theoretical issues in reading comprehension.* Hillsdale, NJ: Erlbaum.

[18] Schoenfeld, A. H. (1985). *Mathematical problem solving.* New York, NY: Academic Press.

[19] Schoenfeld, A. H. (1988). When good teaching leads to bad results: The disasters of "well-taught" mathematics courses. *Educational Psychologist*, 23(2), 145-166.

[20] Selden, A. and Selden, J. (1987). Errors and misconceptions in college level theorem proving. *Proceedings of the Second International Seminar on Misconceptions and Educational Strategies in Science and Mathematics*, Vol. III, Cornell University, Ithaca, NY, 456-471.

[21] Selden, J. and Selden, A. (1995). Unpacking the logic of mathematical statements. *Educational Studies in Mathematics*, 29(2), 123-151.

[22] Stenning, K. and Oaksford M. (1993). Rational reasoning and human implementations of logic. In K. I. Manktelow and D. E. Over (Eds.), Rationality: Psychological and Philosophical Perspectives, London: Routledge.

[23] Stewart, J. (1991). Calculus (2nd edition), Pacific Grove, CA: Brooks/Cole.

Tennessee Technological University
Mathematics Education Resources Co.