Chapter 4

STRUCTURAL (ALGORITHMIC) ANALYSIS OF ALGEBRAIC PROOFS

Joseph M. Scandura and John H. Durnin

One of the most difficult problems in mathematics education is teaching students how to prove theorems. Traditionally, two basic methods have been used in the classroom. One method involves presenting the student with a wide variety of proofs and explaining or having the student explain why they work, in the hope that the learner will extract general methods that he can then apply in other situations. Another school of thought would have the student construct his own proofs that are then subject to criticism by the teacher and/or his peers. An extreme form of this approach is the well-known method used by R. L. Moore at Texas, in which the student not only proves his own theorems but also decides on which conjectures to try to prove (or disprove).

The first method of proof instruction depends heavily on induction from example. Success in using this method depends both on the wise selection of sample proofs and precautions to ensure that students do in fact attempt to infer general methods of proof rather than to simply memorize the samples. The second method of proof instruction most likely depends on the self generation of semantic models (and counter examples) of the theorems (conjectures) in question, as well as on the generation of proofs per se (such models correspond to structures in the sense of Chapter 2; cf. Scandura, 1971). This method more nearly parallels the types of capabilities needed in research mathematics. On the other hand, the method may not work well with relatively unmotivated students, and the instructor must be prepared to cover less material, at least initially.

Although both methods have been used successfully by many

---

1This research was supported by National Science Foundation Grant GW-7901 to Research for Better Schools.
teachers, there are perhaps more who have experienced something less than complete success—either because they do not know themselves what goes into proof making, and/or because they are unable to effectively impart to students the necessary know-how. In either case, we believe that qualitative improvements in proof-making instruction will require more precise specification of what it is that students must learn in order to prove theorems, irrespective of whether the primary goal is to teach the general nature of proof or to produce creative research mathematicians. Without such knowledge, the only reasonable alternatives seem to be informal, trial-and-error instruction by example and/or lots of experience and practice in theorem production and proof generation. (Even here, of course, more complete specification of what is to be learned would make possible wiser and more efficient selection of appropriate activities.)

We do not believe, however, that it will ever be possible to fully specify all that an ingenious theorem prover needs to know or might acquire through long years of study and practice. Nor, do we even think that this would be desirable, since new, creative insights undoubtedly depend to some extent on individual idiosyncrasies and capabilities. Nonetheless, we do believe that progress in this direction is possible and that the inability of many students to benefit from traditional instruction in proof making more than justifies attempts to develop alternative methods of instruction.

In recent years, a beginning has been made in this direction, primarily in computer-assisted instruction (e.g., Goldberg, 1971; Kane, 1972). For example, Suppes (1971) and his collaborators have provided students with an adaptive computer-controlled environment that checks each step in a proof as it is made to see if it is valid. To our knowledge, however, no one has attempted direct, systematic instruction in how to construct proofs. The primary reason is that no one has known exactly what it is that one should attempt to teach.

The purpose of the present study was to take a first step in this direction—to determine whether systematic, structural analyses of proof making are feasible. Specifically, our aim was to determine whether it is possible to construct algorithms that generate proofs of the theorems and exercises contained in Brumfiel, Eicholz, and Shanks' (1961) high school algebra text and that accomplish this in the same general way as might competent ninth-graders. The formulation of conjectures for possible proof or disproof was not considered.

We chose this text for analysis because it was one of the first to place heavy emphasis on proof in high school algebra. Also, because the text has become somewhat dated, our analysis is more likely to be viewed as a prototype (as we intend) rather than as a possible endorsement of one or another textbook.
1. **BACKGROUND**

In their analysis of geometry construction problems, Scandura, Durnin, and Wulfeck (Chapter 3) demonstrated the feasibility of systematic, rigorous structural (algorithmic) analyses of particular problem domains in mathematics. This research provided a general model for our analysis.

The geometry analysis of Chapter 3 was built on Polya's (1962) classification of straight-edge and compass construction problems in geometry and, in part, involved the precise formulation of three heuristics: The pattern of two-loci, the pattern of similar figures, and the pattern of auxiliary figures. After classifying the various construction problems (according to the heuristics involved in their solution), a sample of specific problems was identified within each class. Then, specific rules (procedures/algorithm) were devised whereby each of the sampled problems could be solved, one procedure for each type of problem. Next, parallels among these solution procedures were observed. Components of the solution rules and higher-order rules were identified that reflected these parallels (cf. Scandura, 1973), and from which the solution rules could be derived (by applying higher-order rules to component rules). The obtained set of component and higher-order rules was then checked with new problems to determine its adequacy (i.e., to see if the rules could be used to generate solution procedures for the new problems). The individual rules so identified were refined and modified as necessary. The resulting rules were sufficiently precise as to be programable on a computer.

In addition to the simplification afforded by Polya's (1962) preclassification of problems and identification of heuristics, the Scandura et al. (Ch. 3) analysis had several important limitations. First, the analysis was explicitly limited with regard to logical inference. At no point was an attempt made to show how various constructions might be logically justified. Second, some of the sampled tasks did not form "natural" units; no attempt was made to account for subproblems (cf. Chapter 2). This is an important limitation, since the identification of rules for constructing subproblem (goal) hierarchies (Newell & Simon, 1972) potentially could reduce the complexity of needed higher-order rules (cf. Chapter 3). Third, the only kinds of solution rules that could be generated using the identified higher-order rules were simple compositions (of component rules/inputs). In no case did a higher-order rule actually modify (e.g., generalize) an input rule. Clearly, other forms of derivation are also important in problem solving (e.g., solving problems by analogy).

In part, the present study was designed to overcome these limitations. Specifically: (1) logical inference formed the core of the analysis; (2) rules for breaking problems into more natural subproblems were included; (3) explicit attention was
given to higher-order relationships other than composition.

2. METHOD OF ANALYSIS

As was done in Chapter 3 the first step in our analysis was to become thoroughly familiar with the problem domain. In the present case, this meant first of all to properly interpret the tasks from a behavioral point of view. In proving theorems, one might think that the inputs are the premises (antecedents) and that the outputs are the final steps in the proofs (which are frequently the consequents). In reality, the inputs are the theorems and the (final) outputs are the proofs themselves. This point is implicitly recognized in Polya's (1962) distinction between "problems to solve" and "problems to prove".

After generally familiarizing ourselves with the contents of Brumfiel et al. (1961), we went through the proofs of all 67 theorems and 35 proof exercises in the text. The theorems were also proved independently, paying close attention to the processes we used (i.e., the processes that might be used by someone generally familiar with the content but not necessarily with all of the given proofs). In almost all cases, the proofs we derived were roughly the same as in the textbook.

Unlike the previous geometry analysis, we gave more explicit attention to the distinction between breaking a problem down into subproblems, and rule derivation. In particular, we were especially cognizant, in proving the theorems, as to when we generated portions of a proof, before even considering the rest, and when we knew exactly how to generate an entire proof, before actually attempting to generate it.

Although this distinction is not important computationally (insofar as deriving the proofs), it is important psychologically. As described in Chapter 2, the former corresponds to breaking problems down into subproblems and the latter to deriving new solution procedures (for given subproblems). Allowing for both possibilities brings together the subgoal hierarchy notion proposed by Newell and Simon (1972) and the rule derivation notion proposed by Scandura (1973). In the former case, problems are presumed to be solved by forming a hierarchy of subgoals and solving each of them in turn by directly applying available operators (component rules). A selected operator is assumed to be applied before the next one is determined (i.e., the various subgoals are considered one at a time). In simple rule derivation, on the other hand, new solution procedures are presumed to be derived in their entirety (by application of higher-order rules to other available rules) before any part of them is applied. This latter view corresponds essentially to "insight", in which a puzzled organism often suddenly sees how to proceed.

In truth, it would appear that both systematic step-by-step
planning and insight may occur during problem solving, and our 
aim was to come up with a characterization of underlying compe-
tence (rules) that reflects each of them. Thus, in our search 
for parallels among the proof-generating procedures identified, 
we were concerned with: (1) how the various types of theorems-
to-prove were broken down into subproblems, and (2) how solution 
procedures for solving various subproblems were derived. In the 
latter regard, we were particularly concerned with analogies 
among solution procedures in different algebraic systems (e.g., 
natural numbers and integers). 
A total of 24 basic problem definition, higher-order and 
lower-order rules are identified and described in the following 
text. The adequacy of the rules was checked, not only with re-
spect to the 67 theorems and 35 proof exercises in the book, but 
also with respect to additional sampled theorems concerning 
number systems that were not contained in the book. 
As in the geometry construction analysis, however, only 
critical portions of the basic rules were detailed. Although 
all of these rules are potentially programable on a computer, 
our aim was to identify rules similar to those that a competent, 
idealized ninth-grader might reasonably be expected to use in 
proving the given theorems, and to describe them in terms of 
"natural" cognitive units rather than in some specific program-
ing language. 
Furthermore, the analysis does not consider the growth of 
knowledge over time, as would be expected to take place, for 
example, as a result of proving a variety of similar theorems 
of the same type. One of the first author's dissertation 
students (Wulfeck) has extended the geometry analysis of Scandura 
et al. (Chapter 3) to deal with the latter problem. Specifical-
ly, the underlying higher- and lower-order rules identified in 
the original geometry analysis were subjected to further analysis 
as described in Chapter 2 to identify still more basic rule sets 
(called innate bases). Wulfeck repeated this process until a 
base level (rule set) was reached (see Chapter 14). 
In general, when such an analysis is performed, two impor-
tant things happen: (1) the individual rules become simpler, 
but (2) the resulting rule set as a whole becomes more powerful 
(than the set from which it was derived). The latter obtains 
in exactly the same sense that a set of higher- and lower-order 
rules can be used to solve more problems than the solution rules 
from which the set is derived. As this chapter goes to press, 
Wulfeck's analysis has been largely completed and is being used 
as a basis for exploring questions of optimization in adaptive 
instruction with respect to various types of problem-solving 
objectives. An overview of the theoretical approach is described 
in Scandura (1977; also Chapter 14).
3. INITIAL SUBDIVISION OF COMPOUND THEOREMS

In Brumfiel et al. (1961) there was no categorization of problems (theorems-to-prove) comparable to that given by Polya (1962), only the usual classification by chapter (whole numbers, integers, rationals, algebraic extension $\mathbb{Q} (\sqrt{2})$ over the rationals, and reals). Nonetheless, in our analysis it quickly became apparent that a number of the problems involved simpler subproblems; these compound problems did not form "natural units" in the sense described in Chapter 2.

Three general types of compound problems were identified. The corresponding (compound) theorems are listed in Table 1. (Note: "iff" means "if and only if.")

<table>
<thead>
<tr>
<th>TABLE 1 Types of Compound Theorems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Biconditionals</strong></td>
</tr>
<tr>
<td><strong>Rationals:</strong></td>
</tr>
<tr>
<td>$a/b &gt; c/b$ (b &gt; 0) iff $a &gt; c$</td>
</tr>
<tr>
<td>$a/b = c/d$ (b, d ≠ 0) iff ad = bc</td>
</tr>
<tr>
<td>$a/b = c/d$ (b, d ≠ 0) iff a/c = b/d</td>
</tr>
<tr>
<td>$a/b = c/d$ (b, d ≠ 0) iff (a+b)/b = (c+d)/d</td>
</tr>
<tr>
<td>$a/b = c/d$ (b, d ≠ 0) iff a/b = (a+c)/(b+d)</td>
</tr>
<tr>
<td><strong>$\mathbb{Q} (\sqrt{2})$:</strong></td>
</tr>
<tr>
<td>$a + b\sqrt{2} = 0$ iff $a = 0$ and $b = 0$</td>
</tr>
<tr>
<td>$a + b\sqrt{2} = c + d\sqrt{2}$ iff $a = c$ and $b = d$</td>
</tr>
<tr>
<td>$r + s = 0$ iff $r = 0$ or $s = 0$</td>
</tr>
<tr>
<td><strong>Reals:</strong></td>
</tr>
<tr>
<td>$a &gt; b$ iff $a + c &gt; b + c$</td>
</tr>
<tr>
<td><strong>Existence and Uniqueness</strong></td>
</tr>
<tr>
<td><strong>Rationals:</strong></td>
</tr>
<tr>
<td>For every $a/b$, $c/d ≠ 0$ there exists a unique $x$ such that $c/d \cdot x = a/b$</td>
</tr>
<tr>
<td>For every $a/b ≠ 0$ there exists a unique $x$ such that $a/b \cdot x = 1$</td>
</tr>
<tr>
<td>For every $a/b$, $c/d$ there exists a unique $x$ such that $a/b = c/d = x$</td>
</tr>
<tr>
<td><strong>$\mathbb{Q} (\sqrt{2})$:</strong></td>
</tr>
<tr>
<td>For every $a + b\sqrt{2}, c + d\sqrt{2}$ there exists a unique $x$ such that $x = (a+b\sqrt{2})/(c+d\sqrt{2})$ (c+d\sqrt{2} ≠ 0)</td>
</tr>
<tr>
<td><strong>Simple Antecedent, Compound Consequent</strong></td>
</tr>
<tr>
<td><strong>Wholes:</strong></td>
</tr>
<tr>
<td>If $ca &lt; cb$, then $a &lt; b$ and $c ≠ 0$</td>
</tr>
<tr>
<td><strong>Integers:</strong></td>
</tr>
<tr>
<td>If $x \cdot y = 0$, then $x = 0$ or $y = 0$</td>
</tr>
<tr>
<td><strong>Rationals:</strong></td>
</tr>
<tr>
<td>If $a/b ≠ 0$, then $a/b &gt; 0$ or $a/b &lt; 0$</td>
</tr>
</tbody>
</table>

Perhaps the easiest compound theorems to detect were the biconditionals (e.g., in the system of rationals $a/b > c/b$ [b > 0] if and only if $a > c$); next were theorems asserting both existence and uniqueness, for example, (rationals) for every $a/b$, $c/d ≠ 0$, there exists a unique $x$ such that $c/d \cdot x = a/b$; and third were theorems involving a compound consequent, where indirect proofs are required, for example, (whole numbers) if $ca < cb$, then $a < b$ and $c ≠ 0$.

In proving compound theorems, the first thing to do, invariably, was to break them down into a pair of subproblems. This
decomposition could be accomplished by applying the following problem definition rules:

**Biconditional Problem Definition Rule:**
If a theorem-to-prove is of the form "P if and only if Q"
break it down into the following theorems-to-prove
"If P, then Q" and "If Q, then P"

For example, application of this problem definition rule to
\[ a/b > c/b \ (b > 0) \text{ if and only if } a > c \ (\text{rationals}) \]
yields the pair of theorems:
If \( a/b > c/b \ (b > 0) \), then \( a > c \)
If \( a > c \), then \( a/b > c/b \ (b > 0) \)
(Note: In the following sections, subproblems are often denoted as ordered pairs of givens and goals. For example, the theorem-to-prove "If P, then Q" might be denoted, If P, then Q; If P, then Q + Proof.)

**Existence and Uniqueness Problem Definition Rule:**
If a theorem involves "there exists a unique", or some equivalent statement, break the theorem down into two subproblems, the first involving existence and the second involving uniqueness.

For example, application of the above rule, to
For every \( a/b, c/d \neq 0 \), there exists a unique \( x \) such that
\[ c/d \cdot x = a/b \ (\text{rationals}) \]
gives the following pair of theorems-to-prove:
For every \( a/b, c/d \neq 0 \), there exists an \( x \) such that
\[ c/d \cdot x = a/b \]
For every \( a/b, c/d \neq 0 \) such that \( c/d \cdot x = a/b, x \) is unique
(Note: Although it would not be difficult to do, it is beyond the scope of this analysis to include the grammatical detail that would be required in any complete statement of these problem definition rules.)
In the case of theorems involving a simple antecedent and a compound consequent, two related problem definition rules were required.

**Simple Antecedent - Compound Consequent Problem Definition Rule:**
(1) If a theorem is of the form
"If P, then Q and R",
break it down into the pair
"If not Q, then not P" and
"If not R, then not P"

(2) If a theorem is of the form
"If P, then Q or R",
break it down into the pair
"If P and not Q, then R"
"If P and not R, then Q"

For example, application of the former rule to
If \(ca < cb\), then \(a < b\) and \(c \neq 0\) (whole numbers)
gives the pair
If \(a \geq b\), then \(ca \geq cb\)
If \(c = 0\), then \(ca = cb\) (i.e., \(ca \geq cb\))

Similarly, application of the latter rule to
If \(x \cdot y = 0\), then \(x = 0\) or \(y = 0\)
gives
If \(x \cdot y = 0\) and \(x \neq 0\), then \(y = 0\)
If \(x \cdot y = 0\) and \(y \neq 0\), then \(x = 0\)

Although more "elegant" proofs are possible in some cases (e.g., with biconditionals, where proof steps are reversible), the applicability of the above problem definition rules seems universal. Breaking biconditionals into pairs of conditionals, breaking theorems of existence and uniqueness into existence theorems and uniqueness theorems, and breaking theorems involving compound consequents into pairs of theorems involving simple consequents, is equally appropriate whether the theorems are algebraic, geometric, or concerned with analysis. More important in the case of the theorems under consideration, this manner of formulating subproblems is consistent with the way a knowledgeable theorem prover might be expected to proceed.

The remainder of our analysis is simplified accordingly. We consider only those rules of competence necessary for proving the simple/basic theorems that remain after the above problem definition rules have been applied. (Insofar as teaching and learning is concerned, this amounts to assuming that students can successfully be taught how to break compound theorems into simpler theorems, and that once problems have been so defined, students will deal with each part separately [cf. Chapter 2].)

4. KEY FIRST STEP SUBPROBLEMS--DEFINITION RULES

In attempting to prove the basic theorems (implicit or explicit) in Brumfield et al. (1961) the importance of selecting an appropriate first step became almost immediately apparent. The first thing that had to be done in essentially every case, was to identify a key first step. In effect, each basic problem was broken down into a pair of subproblems, the first of which was to determine the key first step. (The second subproblem, of course, was to generate the remainder of the proof.) This fact led us to introduce the following first approximation to a new problem definition rule.
Key Step Problem Definition Rule (First Approximation):
Given a basic theorem-to-prove involving number systems
(the rule applies potentially to theorems other than just
those in Brumfiel et al., 1961), break the theorem into a
pair of subproblems {(theorem, theorem + appropriately re-
lated expression) and (theorem + appropriately related ex-
pression, theorem + proof)}. The first subproblem takes
the theorem as input and the first step, an expression
"appropriately related" to the theorem, as its goal.

As it stands, the statement of this problem definition rule con-
tains a "hooker". Just what is meant by "appropriately related"?

On further analysis, it turned out that the key first step
(i.e., the solution to the first subproblem) depends directly,
and often simply, on the form of the given theorem statement.
In this regard, there are six relevant types of basic theorems
in Brumfiel et al. (1961). Complete categorization of all
theorems by type is given in Table 2. For example, in proofs
of theorems involving Type 1, a simple relational antecedent and
a simple consequent, the key first step is simply the antecedent
—in every case. In the system of whole numbers, for instance,
the theorem "If x + a = a + b, then x = b", has "x + a = a + b"
as its first step. It is important to emphasize that this re-
relationship is obtained in all number systems, not only in the
system of natural numbers. The key first step in the proof of
the theorem "If x > 0, then −x < 0" (integers), for example,
is "x > 0".

The initial key step also follows directly from the theorem
statement in four other cases: Type 2, compound antecedent,
simple consequent; Type 3, existence of element denied; Type 4,
universally quantified statements of equality; and Type 5,
quickness of an element x satisfying a given equation. The
various types of key step subproblems and rules for generating
key first steps in these cases, along with examples, are given
in Table 3.

As can be seen from the table, the nature of the desired
output of the first subproblem (i.e., the subgoal/appropriately
related expression) can be specified in each of the above cases.
In Case 1, the appropriately related expression is the anteced-
ent; in Case 2, the compound antecedent; in Case 3, the assump-
tion of the existence of the denied element; in Case 4, the
assumption of the universally quantified equation, and in Case
5, the assertion of two distinguished elements satisfying the
given equation.³

In Case 6 (existence of elements), however, key first steps
could not be determined from theorem statements quite so easily.

³We were tempted in several cases to insert relation for equation but decided
to restrict ourselves to the latter in those Cases (4 and 5) where equations were
the only kinds of relations considered in Brumfiel et al. (1961).
### Table 2: Categorization of Theorems and Proof Exercises (E)

<table>
<thead>
<tr>
<th>System</th>
<th>Type 1 Simple Antecedent, Simple Consequent</th>
<th>Type 2 Compound Antecedent, Simple Consequent</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Wholes</strong></td>
<td>( x + a = a + b \Rightarrow x = b )</td>
<td>( a &gt; b \land b &gt; c \Rightarrow a &gt; c )</td>
</tr>
<tr>
<td></td>
<td>( a + x = a + b \Rightarrow x = b )</td>
<td>( a &gt; b \land c \Rightarrow x &gt; b + d )</td>
</tr>
<tr>
<td></td>
<td>( a + x = b + a \Rightarrow x = b )</td>
<td>( a &gt; b \land b &gt; c \Rightarrow a &gt; b &gt; c )</td>
</tr>
<tr>
<td></td>
<td>( c = 0 \Rightarrow ca = cb )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( r - x = r - s \Rightarrow x = s )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x = a \Rightarrow a \Rightarrow x = a ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x = a \Rightarrow a \Rightarrow x = a ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a &gt; b \Rightarrow ca &gt; cb ) (c ( \neq ) 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a &gt; b \land a &gt; c \Rightarrow b + c )</td>
<td></td>
</tr>
<tr>
<td><strong>Integers</strong></td>
<td>( x + a = y + a \Rightarrow x = y )</td>
<td>( a &gt; 0 \land b &lt; 0 \Rightarrow ab &lt; 0 ) (E)</td>
</tr>
<tr>
<td></td>
<td>( 2</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>( 2x \leq 2x^2 ) (E)</td>
<td>( a &gt; b \land b &gt; c \Rightarrow a &gt; c ) (E)</td>
</tr>
<tr>
<td></td>
<td>( r &gt; s \Rightarrow r + t &gt; s + t )</td>
<td>( x \land y \lor z \lor x ) (E)</td>
</tr>
<tr>
<td></td>
<td>( r &gt; 0 \Rightarrow r^2 &gt; 0 ) (E)</td>
<td>( x \land y \lor z \lor x ) (E)</td>
</tr>
<tr>
<td></td>
<td>( x &gt; 0 \Rightarrow x^2 &gt; 0 ) (E)</td>
<td>( x &gt; y \Rightarrow 0 \land x \neq 0 \Rightarrow y = 0 ) (E)</td>
</tr>
<tr>
<td></td>
<td>( a \land b \Rightarrow a &gt; b ) &gt; 0 ( ) (E)</td>
<td>( x \land y \lor z \lor x ) (E)</td>
</tr>
<tr>
<td><strong>Rationals</strong></td>
<td>( a &gt; b \Rightarrow a &gt; b \Rightarrow b ) (b ( \neq ) 0)</td>
<td>( a &gt; b \land a &gt; b \Rightarrow a &gt; b ) (b ( \neq ) 0)</td>
</tr>
<tr>
<td></td>
<td>( a &gt; b \Rightarrow a &gt; b \Rightarrow b ) (b ( \neq ) 0)</td>
<td>( a &gt; b \land a &gt; b \Rightarrow a &gt; b ) (b ( \neq ) 0)</td>
</tr>
<tr>
<td></td>
<td>( a &gt; b \Rightarrow a &gt; b \Rightarrow b ) (b ( \neq ) 0)</td>
<td>( a &gt; b \land a &gt; b \Rightarrow a &gt; b ) (b ( \neq ) 0)</td>
</tr>
<tr>
<td></td>
<td>( a &gt; b \Rightarrow a &gt; b \Rightarrow b ) (b ( \neq ) 0)</td>
<td>( a &gt; b \land a &gt; b \Rightarrow a &gt; b ) (b ( \neq ) 0)</td>
</tr>
<tr>
<td>( \mathbb{Q} ) ( (\sqrt{2}) )</td>
<td>( a \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td>( a = c \land b = d \Rightarrow a + b \sqrt{2} = c + d \sqrt{2} )</td>
</tr>
<tr>
<td></td>
<td>( b \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( b \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( b \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x \neq 0 \Rightarrow y \neq 0 \Rightarrow xy \neq 0 ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x \neq 0 \Rightarrow y \neq 0 \Rightarrow xy \neq 0 ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x \neq 0 \Rightarrow y \neq 0 \Rightarrow xy \neq 0 ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x \neq 0 \Rightarrow y \neq 0 \Rightarrow xy \neq 0 ) (E)</td>
<td></td>
</tr>
<tr>
<td>( \mathbb{R} ) ( \mathbb{R} )</td>
<td>( a \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( b \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( b \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x \neq 0 \Rightarrow y \neq 0 \Rightarrow xy \neq 0 ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x \neq 0 \Rightarrow y \neq 0 \Rightarrow xy \neq 0 ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x \neq 0 \Rightarrow y \neq 0 \Rightarrow xy \neq 0 ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x \neq 0 \Rightarrow y \neq 0 \Rightarrow xy \neq 0 ) (E)</td>
<td></td>
</tr>
<tr>
<td>( \mathbb{Q} ) ( (\sqrt{2}) )</td>
<td>( a \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( b \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( b \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( b \neq 0 \Rightarrow a + b \sqrt{2} \neq 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x \neq 0 \Rightarrow y \neq 0 \Rightarrow xy \neq 0 ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x \neq 0 \Rightarrow y \neq 0 \Rightarrow xy \neq 0 ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x \neq 0 \Rightarrow y \neq 0 \Rightarrow xy \neq 0 ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x \neq 0 \Rightarrow y \neq 0 \Rightarrow xy \neq 0 ) (E)</td>
<td></td>
</tr>
<tr>
<td><strong>System</strong></td>
<td><strong>Type 3 Denial of Existence</strong></td>
<td><strong>Type 4 Universally Quantified Statements</strong></td>
</tr>
<tr>
<td><strong>Wholes</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \forall x, y, (0 + x) + y = y + x ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x, y, (1 \cdot x) \cdot y = y \cdot x ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a, b, c, a \cdot (b + c + d) = ab + ac + ad )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( r, s, r (a + l) = rs + r ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x, 2x = x + x )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x, x \cdot x = 0 )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a, a = a ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a, a = a ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( c, b &gt; a, c (b &gt; a) = cb &gt; ca )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a, b, c, a \cdot (b + c - e) = (a + b) - c )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a &gt; (b + c), a = (b + c) = (a + b) - c )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( a &gt; b &gt; c, (a - b) + c = a - (b - c) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x, 0 + x = x ) (E)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( x, \cdot 1 \times x = x ) (E)</td>
<td></td>
</tr>
</tbody>
</table>
Integers

\[ \forall x, (-1)x = -x \]
\[ \forall a, b, (-a) \cdot b = -(ab) \]
\[ \forall a, b, (-a) \cdot (-b) = ab \]
\[ \forall x, -(x) = x \]
\[ \forall a, b, -(a + b) = (-a) + (-b) \]
\[ \forall x, -x \cdot 0 = -x \]
\[ \forall x, y, x \cdot y = x + y \]
\[ \forall a, b, c, a \cdot (b - c) = (a + b) - c \text{ (E)} \]
\[ \forall a, b, c, a \cdot -(b + c) = (a - b) - c \text{ (E)} \]
\[ \forall a, b, c, a \cdot -(b - c) = (a - b) + c \text{ (E)} \]
\[ \forall a, b, c, a \cdot (b - c) = ab - ac \]

Rationals

\[ \neg \exists a/b, b \neq 0, a/b \cdot 0/1 = 1 \]
\[ \neg \exists r, r^2 = 2 \]
\[ \neg \exists r, r^2 = 3 \text{ (E)} \]
\[ \forall a/b, c/d \neq 0, a/b + c/d = a/b + d/c \text{ (b, c, d \neq 0)} \]
\[ \forall a/b, -(a/b) = -(a/b) \text{ (b \neq 0)} \]
\[ \forall a/b, -a/b = a/-b \text{ (b \neq 0)} \]
\[ \forall a/b, c/d, a/b \cdot c/d = ac/bd \text{ (bd \neq 0)} \]
\[ \forall a/b, c/d, a/b + c/d = (ad + bc)/bd \text{ (bd \neq 0)} \]
\[ \forall a/b, c/b, a/b + c/b = (a + c)/b \text{ (b \neq 0)} \]
\[ \forall a/1, a/1 = a \]
\[ \forall a \neq 0, a/a = 1 \]
\[ \forall a \neq 0, 0/a = 0 \]
\[ \forall a/b \neq 0, c/d \neq 0, \]
\[ (a/b \cdot c/d) (c/d \cdot a/b) = 1 \text{ (E)} \]
\[ \forall r, s, t, u, r/s - t/u = (ru - st)/su \text{ (E)} \]
\[ (su \neq 0) \]

\[ \mathbb{Q}(\sqrt{2}) \]

\[ \exists a + b\sqrt{2}, (a + b\sqrt{2})^2 = 3 \]

\[ \forall x, y, x \cdot y = x + y \text{ (E)} \]
\[ \forall x, y, x(-y) = -(xy) \text{ (E)} \]
\[ \forall x, y, (-x)(-y) = xy \text{ (E)} \]
\[ \forall a > 0, b > 0, \neg ab = fa + fb \]

<table>
<thead>
<tr>
<th>System</th>
<th>Type 5 Uniqueness</th>
<th>Type 6 Existence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wholes</td>
<td>[ \forall a, b, x = a + b, x \text{ is unique (b \neq 0)} ]</td>
<td>[ \forall a/b, c/d, c/d \cdot x = a/b, x \text{ is unique (b, d \neq 0)} ]</td>
</tr>
</tbody>
</table>

Integers

\[ \forall a/b, d/c, c/d \cdot x = a/b, x \text{ is unique (b, d \neq 0)} \]
\[ \forall a/b, c/d, a/b \cdot x = 1, x \text{ is unique (b, d \neq 0)} \]
\[ \forall a/b, c/d, a/b + c/d = x, x \text{ is unique (b, d \neq 0)} \]

Rationals

\[ \forall a + b\sqrt{2}, c + d\sqrt{2}, x + y\sqrt{2} = (a + b\sqrt{2}) + (c + d\sqrt{2}), x + y\sqrt{2} \text{ is unique (E)} \]
\[ \forall a + b\sqrt{2}, c + d\sqrt{2}, x + y\sqrt{2} = (a + b\sqrt{2})/(c + d\sqrt{2}), x + y\sqrt{2} \text{ is unique (c, d \neq 0)} \]

\[ \forall a + b\sqrt{2}, c + d\sqrt{2}, x + y\sqrt{2} = (a + b\sqrt{2})/(c + d\sqrt{2}), x + y\sqrt{2} \text{ is unique (c, d \neq 0)} \]
\[ \forall a + b\sqrt{2}, \exists x^2 + sx + t = 0, (a + b\sqrt{2})^2 + s(a + b\sqrt{2}) + t = 0 \]

Reals

Note: Key to logical notation

\[ \forall - \text{ For every} \]
\[ \exists - \text{ There exists} \]
\[ \neg - \text{ Not} \]
\[ \land - \text{ And} \]
\[ \Rightarrow - \text{ Implies} \]
| Case 1. **Simple antecedent, simple consequent** |
| Key Step Subproblem: (Simple antecedent, simple consequent (theorem); theorem + antecedent) |
| **Rule:** Write the antecedent. |
| **Example:** Subtheorem (wholes)--to-prove: (If \( x + a = a + b \) then \( x = b \); theorem + antecedent) |
| **Key Step (antecedent):** \( x + a = a + b \) |

| Case 2. **Compound antecedent, simple consequent** |
| Key Step Subproblem: (Compound antecedent, simple consequent (theorem); theorem + compound antecedent) |
| **Rule:** Write each relation in the antecedent on a separate line. |
| **Example:** Subtheorem (wholes)--to-prove: (If \( a > b \) and \( b > c \) then \( a > c \); theorem + compound antecedent) |
| **Key Steps (compound antecedent):** \( a > b \) \( b > c \) |

| Case 3. **Existence of element denied** |
| Key Step Subproblem: (Existence denied (theorem); theorem + existence) |
| **Rule:** Assume the existence of the element. |
| **Example:** Subtheorem (rationals)--to-prove: There does not exist a rational \( a/b \), \( b \neq 0 \) such that \( a/b \cdot 0/1 = 1 \) |
| **Key Step (existence):** Assume there is a rational \( a/b \), \( b \neq 0 \) such that \( a/b \cdot 0/1 = 1 \) |

| Case 4. **Universally quantified statements of equality** |
| Key Step Subproblem: (Universally quantified equation (theorem); theorem + equation) |
| **Rule:** Write the equation in the statement. |
| **Example:** Subtheorem (wholes)--to-prove: For all whole numbers \( a, b, c, d \), \( a(b + c + d) = ab + ac + ad \) |
| **Key Step (equation):** \( a(b + c + d) = ab + ac + ad \) |

| Case 5. **Uniqueness of element satisfying given equation** |
| Key Step Subproblem: (Uniqueness theorem; theorem + 2 equations each with different distinguished element) |
| **Rule:** Assume two "distinguished" elements \( x_1, x_2 \) satisfying equation. |
| **Example:** Subtheorem (integers)--to-prove: Given \( x \) such that \( x = a + r \), \( x \) is unique |
| **Key Steps (2 equations):** \( x_1 = a + r \) \( x_2 = a + r \) |

| Case 6. **Existence of element(s) (equation) satisfying given equation (equation form)** |
| Key Step Subproblem: (Existence theorem; theorem + solution/postulated element) |
| **Rule:** Solve the given equation using the real-number solution module and substitute. (Solve the given equation form to determine unknown coefficients using the real-number module and substitute.) |
| **Examples:** Subtheorem (rationals)--to-prove: For all rationals \( a/b, c/d \), there exists an \( x \) such that \( a/b - c/d = x \) |
| **Key Step (solution):** Reason from \( a/b - c/d = x \) via real-number module to get \( a/b + -c/d = x \); substituting gives \( a/b - c/d = a/b + c/d \) |
| **Subtheorem (extension of rationals)--to-prove:** For every \( a + b\sqrt{2} \) in \( \mathbb{Q}(\sqrt{2}) \) where \( a \) and \( b \) are rational, there exists a quadratic equation \( x^2 + sx + t = 0 \) with \( s \) and \( t \) unknown rationals, such that \( (a + b\sqrt{2})^2 + s(a + b\sqrt{2}) + t = 0 \) |
| **Key Steps (solution):** Reason from \( (a + b\sqrt{2})^2 + s(a + b\sqrt{2}) + t = 0 \) \( a^2 + 2b^2 + sa + t + (2ab + sb)\sqrt{2} = 0 \) to get \( 2ab + sb = 0 \) \( s = -2a \) \( a^2 + 2b^2 + sa + t = 0 \) \( t = a^2 - 2b^2 \) Substituting gives \( (a + b\sqrt{2})^2 + (-2a)(a + b\sqrt{2}) + (a^2 - 2b^2) = 0 \) |

*The real-number solution rule plays a special role in generating key steps in existence proofs. The rationale for its inclusion is described in the text.*
In proofs of the Type 6 theorems in Brumfiel et al. (1961), as often occurs in mathematics texts emphasizing proofs, the key first steps typically appeared mysteriously, without apparent reason—the result presumably of immediate "insight" (or, prior and unspecified trial and error).

At least in the case of the Brumfiel et al. (1961) theorems, however, the nature of these key steps and their derivations can be specified. Although they often appear to come out of nowhere, the key first steps satisfy general goal requirements and can be generated algorithmically (i.e., by rule). In particular, the key first steps (postulated elements) involve real number solutions of given equations (or equation forms) involving one or more unknowns. These key first steps are not derived logically, but rather are generated via informal algebraic manipulation in which individual steps are not logically justified.

**TABLE 4**

**Real-Number Solution Rule**

1. Write the equation.
2. If the unknown is squared, transform the equation so that one side equals 0 and go to 19.
3. If the equation involves division, transform the equation, so that the divisors become the multipliers on the opposite side.
4. If subtraction is indicated change - to + and write the additive inverse in place of the expression behind the - sign.
5. If $\sqrt{2}$ is in the equation, go to 14.
6. If the designated $x$ is by itself on one side of equation, go to 11.
7. Multiply both sides of the equation by the denominators, if any, distribute terms and simplify by adding like terms with numerical coefficients.
8. Transform the equation so that all terms involving $x$ are added on one side of the equation and those not involving $x$ are added on the other side; change signs of those terms moved from one side to the other.
9. Factor out $x$.
10. Multiply both sides of the equation by the multiplicative inverse of the coefficient of $x$, simplify both sides of the equation and go to 6.
11. If the expression opposite the designated $x$ contains another unknown, consider the equation still unsolved; select the other equation obtained from 17, substitute the obtained value for $x$ into this equation, consider the other unknown as the designated $x$ and go to 7.
12. If the expression opposite the designated $x$ contains no other unknown term, then consider it a real-number solution.
13. If all real-number solutions for the unknowns have been obtained, go to 21; otherwise substitute the real-number solution obtained into the other unsolved equation, designate the remaining unknown $x$ and go to 7.
14. Multiply and square expressions so indicated.
15. Commute and associate terms so that all terms involving $\sqrt{2}$ are added together on both sides of the equation.
16. Factor out $\sqrt{2}$.
17. Set those expressions that are the coefficients of $\sqrt{2}$ on either side of the equation equal to each other, and set the expressions that are not the coefficients of $\sqrt{2}$ on either side of the equation equal to each other.
18. If one of the two equations obtained from 17 contains only one unknown, select it, and go to 6; otherwise select any one of the two equations, designate one of unknowns $x$ and go to 6.
19. Apply the quadratic formula $ax^2 + bx + c = 0 \quad x = (-b \pm \sqrt{b^2 - 4ac})/2a$ and $x = (-b - \sqrt{b^2 - 4ac})/2a$ for real solutions only.
20. Simplify by factoring, clearing squared values from the radical and dividing.
21. Substitute the simplified real-number solutions into the original equation and stop.
Although one would not want to automatically assume a priori mastery of proofs in the real number system (i.e., the ability to generate proofs of theorems in this system), it is quite reasonable to assume that our idealized subject (in view of eight or more years exposure to ordinary arithmetic) has intuitive mastery of manipulations that are legal in the system of real numbers. More specifically, we assume that the subject knows how to solve linear, quadratic, and simple rational equations. The real number solution rule, which represents this capability, is given in Table 4.

We illustrate below how this rule operates with respect to the theorem "For all $a + b \sqrt{2}, c + d \sqrt{2}$ (c, d $\neq 0$), there exists an $x + y \sqrt{2} = (a + b \sqrt{2}) \div (c + d \sqrt{2}) [\mathbb{Q}(\sqrt{2})]$". Step by step derivation of the key first step,

$$x_0 + y_0 \sqrt{2} = (a + b \sqrt{2}) \div (c + d \sqrt{2}),$$

where $x_0, y_0$ are known, goes as follows:

Step 1.  $x + y \sqrt{2} = (a + b \sqrt{2}) \div (c + d \sqrt{2})$
Step 3.  $(x + y \sqrt{2})(c + d \sqrt{2}) = (a + b \sqrt{2})$
Step 14.  $xc + yc \sqrt{2} + xd \sqrt{2} + 2dy = a + b \sqrt{2}$
Step 15.  $xc + 2dy + (yc \sqrt{2} + xd \sqrt{2}) = a + b \sqrt{2}$
Step 16.  $xc + 2dy + (yc + xd) \sqrt{2} = a + b \sqrt{2}$
Step 17.  $yc + xd = b$
Step 18.  $yc + xd = b$
Step 8.  $xd = b - yc$
Step 10.  $x = (b + yc)/d$
Step 11.  $([b - yc]/d)c + 2dy = a$
Step 7.  $bc + yc^2 + 2d^2y = ad$
Step 8.  $-yc^2 + 2d^2y = ad + -bc$
Step 9.  $y (-c^2 + 2d^2) = ad + -bc$
Steps 10 and 12.  $y_0 = (ad + -bc)/(2d^2 + -c^2)$

Real Number Solution:

Step 13.  $([ad + -bc]/[2d^2 + -c^2])c + xd = b$
Step 7.  $adc + -bc^2 + 2xd^3 + -xc^2d = 2bd^2 + -bc^2$
Step 8.  $2xd^3 + -xc^2d = 2bd^2 + -ad^2$
Step 9.  $(2d^3 + -c^2d)x = 2bd^2 + -ad^2$
Steps 10, 12, 13.  $x_0 = (2bd + -ac)/(2d^2 + -c^2)$

Real Number Solution:

Step 21.  $([2bd + -ac]/[2d^2 + -c^2]) +$
$([ad + -bc]/[2d^2 + -c^2]) \sqrt{2} =$
$(a + b \sqrt{2}) \div (c + d \sqrt{2})$

To summarize, the phrase "appropriately related" in the above problem definition rule can be given a precise meaning for all six types of theorems. In the first five cases, the key first steps are identical or closely related to expressions in given theorem statements (e.g., antecedents). In Case 6, they are more indirectly related and must be derived.

In effect, the first of the subproblems obtained on appli-
cation of the above problem definition rule can be specified precisely in each case. The given is the theorem statement, and the goal (desired output) can be characterized in terms of the specified "appropriate" relationship (i.e., key first steps). Put somewhat differently, given a theorem-to-prove, the first subproblem can be fully and precisely determined (via the problem definition rules) from the form of the theorem.

Because there are only six possibilities, it is easy to see how the above "approximation" to the key step problem definition rule might be rigorized: Either represent the problem definition rule in terms of six specialized definition rules (with suitably restricted domains of applicability), one for each case. Or, introduce internal decisions (e.g., Is the theorem of the form simple antecedent, simple consequent?) into a single, more general definition rule that has the same effect. One such key step problem definition rule is shown in Table 5.

<table>
<thead>
<tr>
<th>TABLE 5</th>
<th>Key Step Problem Definition Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2. If the theorem involves an antecedent and a simple consequent, go to (1-2).</td>
<td></td>
</tr>
<tr>
<td>3-6. If the theorem involves existence, go to (3-6).</td>
<td></td>
</tr>
<tr>
<td>4. If the theorem involves universally quantified statements of equality, go to (4).</td>
<td></td>
</tr>
<tr>
<td>5. If the theorem involves uniqueness of an element, go to (5).</td>
<td></td>
</tr>
<tr>
<td>(1-2). Define the subproblems: (theorem, theorem + antecedent) and (theorem + antecedent, theorem + proof) and STOP.</td>
<td></td>
</tr>
<tr>
<td>(4). Define: (theorem, theorem + equation) and (theorem + equation, theorem + proof) and STOP.</td>
<td></td>
</tr>
<tr>
<td>(5). Define: (theorem, theorem + 2 equations) and (theorem + 2 equations, theorem + proof) and STOP.</td>
<td></td>
</tr>
<tr>
<td>(3-6). If the theorem involves denial, go to (3).</td>
<td></td>
</tr>
<tr>
<td>(6). Define: (theorem, theorem + real-number solution) and (theorem + real-number solution, theorem + proof) and STOP.</td>
<td></td>
</tr>
<tr>
<td>(3). Define: (theorem, theorem + existence) and (theorem + existence, theorem + proof) and STOP.</td>
<td></td>
</tr>
</tbody>
</table>

(For clarification of that part of Table 5 that deals with proof completion subproblems (theorem + key step, theorem + proof) see the discussion in the next section.) Anyone moderately skilled at constructing programs can devise others.

Although higher-order rules are important in many structural analyses, they do not seem to play a central role in the above. Simple lower-order rules are sufficient for identifying key first steps in 5 cases; in the sixth case the real-number solution rule applies directly. Higher-order rules enter more naturally in the next section.

5. PROOF COMPLETION SUBPROBLEMS

So far, nothing has been said about the proof completion subproblems generated by the above problem definition rule. These subproblems take theorems, together with key first steps, as inputs; and, roughly speaking, they take completed proofs as outputs (goals). The exact form of the subgoals of the proof completion subproblems, however, depends on the type of theorem.
involved. In Cases 1 and 2, for example, the subgoals are the
consequents of given theorems and the steps leading up to them.
That is, to complete the proof of a given theorem, it is suffi-
cient to generate the consequent from the given theorem/first step (antecedent) via logical operations.

In each Case 3 proof completion subproblem, the subgoal is
to find a contradiction (and the steps leading to it). Although
there is no subgoal criterion that is both simple and specific,
the general idea is to generate a proof statement that contra-
dicts basic assumptions (e.g., $0 = 1$), or an earlier assumed
statement or part thereof (e.g., assume $\sqrt{2} = p/q$ where $p$ and $q$
are relatively prime and show that 2 divides both $p$ and $q$). In
all, there were only two different types of contradiction in

Brumfiel et al. (1961).

The Case 4 subgoals involve finding equations with identical
expressions on both sides (e.g., $a = a$, $0 = 0$, $ab + ac + ad$
$= ab + ac + ad$). In the example of Table 3, this involves manipu-
lating the left side to make it identical with the right. In
other examples both sides are modified. In all cases, however,
each operation acts to reduce the difference between the two
sides in a manner not unlike means-ends analysis (Ernst & Newell,
1969). Furthermore, only reversible modifications are allowed.
The psychological significance of the latter restriction is that
of knowing, whenever reversible steps are used, that it is pos-
sible to start from equations having identical expressions on
both sides, and to generate the initial (assumed) equation by
applying the same steps in reverse order. A less sophisticated
way to characterize competence (sufficient to the task) would
be to introduce a problem definition rule that operates on Case
4 proof completion subproblems and generates a pair of subpro-
blems—one to generate equations with identical expressions on
both sides and the second to go in the reverse direction.

In Case 5, the subgoal of each proof completion subproblem
is to derive an equation showing explicitly that the two hypo-
thesized "unique" elements are identical (e.g., $x_1 = x_2$). Each
subgoal in Case 6 also involves deriving an equation with iden-
tical expressions on both sides.

Although the above level of subproblem definition seemed
appropriate for the initial decomposition, further analysis was
often necessary in order to solve the proof completion subpro-
blems. In particular, some of the relations and operations of
the key steps and goals were defined in terms of primitives in

Brumfiel et al. (1961).

Thus, the relation $a > b$ was defined as $a = b + k$ (with
special cases $a > 0 = a = 0 + a$; $a < 0 = a + -a = 0$) and $a | b$
(a divides b) was defined as $b = ak$ ($2 \nmid b \rightarrow b = 2k + 1$). The
operations $- \text{ and } \div (\slash )$ were defined in terms of addition and
multiplication, respectively (i.e., $k + (a - b) = c \rightarrow k + a =
c + b$ and $k (a \div b) = c \rightarrow ka = cb$).
In attacking the proof completion subproblems, then, we checked first to see if the relations and operations were defined. If not, we transformed the first steps and goals (of proof completion subproblems) into corresponding first steps and goals in which the relations (> , | ) and operations (− , ÷) were defined in terms of more primitive relations (=) and operations (+ , ×).

Redefinition of the proof completion subproblems was accomplished by the following:

Redefinition Rule:

Inputs: Proof completion subproblems (Theorem + first step, goal) that involve undefined relations and/or operations,

Outputs: Proof completion subproblems that involve corresponding defined relations and/or operations,

Rule:

1. If > is involved, apply a > b → a = b + k (k > 0);
   a > 0 → a = 0 + a; 0 > a + 0 = a + −a (−a > 0) and justify corresponding steps in proof "by definition".
2. If a | b (2 ∤ x) is involved, apply a | b → b = ak
   (2 ∤ x → x = 2k + 1) and justify corresponding steps in proof "by definition".
3. If − or ÷ (/) is involved, apply k + a − b = c + k + a =
   c + b; k · (a ÷ b) = c → k a = c · b for b ≠ 0
   (k · a/b = c/d → k · a = b · c/d for b, d ≠ 0) and justify corresponding steps in proof "by definition".
4. Reapply the above steps until all possible relations and/or operations are defined.

For instance, consider the proof completion subproblem associated with the theorem (rationals) "If (a + b) /b = (c + d) /d, then a/b = c/d" (i.e., consider the first step (a + b) /b = (c + d) /d and the consequent a/b = c/d). Application of the redefinition rule to this subproblem yields the defined first step:

(a + b) d = b (c + d) by definition of/
and the defined consequent:

ad = bc by definition of/

Similarly, application of the redefinition rule with respect to the Type 4 theorem (whole numbers) "For all a, b > c,

a + (b − c) = (a + b) − c" yields the defined first step:

a + (b − c) + c = a + b by definition of −

The goal steps of Type 4 theorems are necessarily defined.

The redefinition rule, of course, only applies to certain proof completion subproblems, in particular to those involving >, | (divides) and/or − or ÷. For example, the proof completion subproblem corresponding to the theorem "For every x, x · 0 = 0" is already defined in the above sense.

Since applying the redefinition rule to proof completion
subproblems always yields equivalent subproblems that are defined, the remainder of our analysis concerns the competence necessary for solving defined proof completion subproblems. Unlike key first step rules, proof completion procedures depend on the number system involved, as well as on the type of theorem. Some operations that would be allowed in proving theorems about rationals, for example, could not be used in proving theorems about whole numbers. In general, the whole number system is more restrictive in this sense than any of the other number systems considered. Because of this type of restriction, the theorem "For all a, b, c, a - (b + c) = (a - b) - c", for instance, is a satisfactory statement about rationals but not about whole numbers. For whole numbers, the stipulation must be added that a > (b + c); otherwise subtraction is undefined. Similarly, some of the operations associated with the reals (and algebraic extensions of the rationals) cannot be used in proving theorems about the rationals.

In view of present emphases in lower school mathematics curricula, we assumed in our analysis that idealized ninth-graders who are capable of proving the Brumfiel et al. (1961) theorems have a greater intuitive feel for both the arithmetic operations and the corresponding logical justifications associated with the rationals than for either the more restricted operations on integers and whole numbers or the special operations associated with the reals and the algebraic extension \( \mathbb{Q}(\sqrt{2}) \). For this reason, we selected the system of rational numbers as the main basis for our analysis. (We were tempted, initially, to build only on the positive rationals but decided that most contemporary mathematics curricula give sufficient attention, by eighth-grade, to the basic operations with signed numbers, to justify consideration of the rationals as a whole.)

### Table 6

**Proof Completion Rules for Rationals**

<table>
<thead>
<tr>
<th>1. Simple antecedent, simple consequent</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Theorem, defined antecedent (key step) and defined consequent</td>
</tr>
<tr>
<td><strong>Output:</strong> Completed proof</td>
</tr>
<tr>
<td><strong>Rule:</strong> 1. If the consequent contains an undistributed term, distribute and give reason (i.e., distributivity) in proof; if it contains an additive term or additive inverse (identity) term(s) not in the first step, add the term(s) to both sides of the equation and indicate addition of equals in the proof.</td>
</tr>
<tr>
<td>2. If the consequent contains a multiplicative term or multiplicative inverse (identity) term (or squared expressions) not in the first step equation, multiply the term to (square) both sides of the equation and indicate multiplication of equals in the proof. (If two different products = 0 ( \neq 0 a ), set the products equal to each other.)</td>
</tr>
<tr>
<td>3. If the equation is the same as the consequent, stop; otherwise associate, combine, and distribute terms where necessary and simplify the equation by applying one of the following as needed: ( a + b = c + b = a + c ; a \cdot b = c\cdot b = a \cdot c ) (for ( b \neq 0 )); ( (a - b) + b = a ); ( a + b + a ) (for ( b \neq 0 )); ( b \cdot a/b = a ) (for ( b \neq 0 )); ( a + -a = 0 ), ( a + 0 = a \cdot 1 + a ) and indicate in the proof which is used.</td>
</tr>
<tr>
<td>4. Reapply Step 3.</td>
</tr>
</tbody>
</table>
2. Compound antecedent, simple consequent

Input: Theorem, defined antecedent and defined consequent
Output: Completed proof

Rule: 1. If one term ≠ 0 in the antecedent is not in the consequent and multiplication is not indicated in either the antecedent or consequent (or subtraction is indicated on both sides of relation in the consequent), apply substitution \(a = b + k_1, b = c + k_2 \rightarrow a = c + (k_1 + k_2)\) or \(a = bk_1, b = ck_2 \rightarrow a = c k_1 k_2\), and distribute, if necessary (or apply \(b = c + k + b - c = k\) for subtraction in the consequent, substitute again and reapply Step 1). Indicate substitution in the proof.

2. If the consequent is a sum of terms in the defined key step equations add the equations, distribute terms as needed and indicate addition of equals (and distributivity) in the proof.

3. If the consequent is a product (squared or a reciprocal of an element in the antecedent) elect from the key step the term > 0 or additive inverse of the term < 0, multiply it to (square) both sides of the defined key step obtained from the other key step and indicate multiplication of equals in the proof. Distribute terms, apply \(a \cdot b/a = b\), for \(a \neq 0\), and indicate these in the proof, as needed.

4. If the product was obtained by multiplying by the additive inverse, add the additive inverse(s) of the (nonconstant, k) product(s) to both sides of the equation, apply \(a + "a = 0\) and \(0 + a = a\) and indicate these the in the proof, until the equation is the same as the consequent.

5. If the key step equation is \(x \cdot y = 0\) and \(x \neq 0\) set \(x \cdot y = x \cdot 0\) apply cancellation and indicate this in the proof.

6. If one relation in the antecedent is ≠ and the other is ≠ \(\bigwedge\) apply disjunction to \(a = b\) or \(a > b\) or \(a < b\) and indicate this in the proof.

Denial of Existence

Input: Theorem, defined key step
Output: Completed proof

Rule: 1. If the key step contains a multiplicative expression with 0 in it, take the product and simplify using \(a \cdot 0 = 0\) and indicate the rule in the proof.

2. If the key step assumes there exists a rational \(r\) such that \(r^2 = 2(3)\), redefine \(r\) as \(a/b\) where \(a, b\) are relatively prime.

3. Through substitution of \(a/b\) for \(r\) and multiplication by \(b^2\), show \(2|a^2 - 3b^2\). Apply \(2|x^2 - 2|x\). \(3|\) interchange \(x\) and \(3\). Indicate at each step of the proof the rule used.

4. If a contradiction is obtained, stop; otherwise substitute \(2k (3k)\) for \(a\) in \(a^2/b^2 = 2 (3)\). Indicate this in the proof and apply Step 3 to show \(2b^2\) \((3b^2)\).

4. Universally quantified equations

Input: Theorem, defined key step
Output: Completed proof

Rule: 1. If the defined key step equation involves additive (multiplicative) inverses, add (multiply) the immediate inverse of the expression having the least notation to both sides of the equation. If \(x \cdot 0 = 0\) is the key step add \(x \cdot 0\) to both sides of the equation; if \(2x\) is the first step apply \(2 + 1 + 1\). Indicate addition (multiplication) of equals in the proof.

2. If the expressions on both sides of the equation are the same, stop; otherwise commute, associate and distribute terms (apply \(x = 1 \cdot x\), \(-x = (-1) \cdot x\), \((ab) = (b)\) as needed before distributing) and simplify the equation by applying one of the following as needed: \(a + 0 = a\), \(a + "a = 0\) or \(0 + a = 0\)\), \(a - 1 = a\), \(a/a = 1\) (for \(a \neq 0\)), \((a - b) + b = a\), \((a + b) \cdot b = a\) (for \(b \neq 0\)), \(a/b + c/b = (a + c)/b\) (for \(b \neq 0\))

3. Reapply Step 2.

5. Uniqueness

Input: Theorem, defined key step
Output: Completed proof

Rule: 1. Set expressions equal to the same element equal to each other and indicate substitution in the proof.

2. Apply \(a + b = c + b = a = c\) or \(a \cdot b = c \cdot b = a = c\) for \(b \neq 0\) and indicate cancellation in the proof.

3. If unique elements \(x_1\) and \(x_2\) are equal to each other, stop; otherwise reapply Step 2.
6. **Existence**

**Input:** Theorem, defined key step  
**Output:** Completed proof  
**Rule:**  
1. Combine terms through multiplication as needed.  
2. If the expressions on both sides of the equation are the same, stop; otherwise commute, associate and distribute terms in order to simplify the equation by applying one of the following as needed:  
   \[ b \cdot \frac{a}{b} = a \text{ (for } b \neq 0), \quad a \cdot \frac{a}{a} = a, \quad a \cdot a = a, \quad a \cdot 0 = 0 \text{.} \]  
3. Reapply Step 2. 

*The italicized parts of the rules are applicable in most cases; whereas the parts that are not italicized are applicable only in special cases. In the statements, expressions refer to one or the other side of an equation and terms to additive and multiplicative elements of an expression.*

**These rules are applicable only after they have been proven.**

The six proof completion rules for the rationals in Table 6 correspond to defined versions of the six types of proof completion subproblems given above. Each proof completion rule identifies the competence necessary for one to get from a defined first step to a defined goal (e.g., consequent). 

1. For example, application of the simple antecedent, simple consequent rule of Table 6 to the defined first step is sufficient to complete the proof of the above theorem (rationals) "If \((a + b)/b = (c + d)/d\), then \(a/b = c/d\)." Thus, starting with \((a + b)d = b(c + d)\), distributivity of Step 3 of the rule gives \(ad + bd = bc + bd\).  
Step 3 also gives \(ad = bc\) by cancellation.  

Since \(ad = bc\) is the same as the defined consequent, the proof is complete.  

Application of the proof completion rules in Table 6 for Cases 2 through 6 are illustrated below.  

2. **Compound antecedent, simple consequent**  
   **Theorem:** If \(a/b \neq 0\) and \(a/b \neq 0\) then \(a/b > 0\).  
   **Defined key steps:** \(a/b \neq 0\), \(a/b \neq 0\)  
Applying Step 5 of the Case 2 proof completion rule of Table 6, the trichotomy law for rationals  
\(a/b = 0\) or \(a/b > 0\) or \(a/b < 0\)  
is stated and we obtain \(a/b > 0\) by disjunction  
Since this result is the same as the consequent, the proof is finished.  

3. **Denial of existence**  
   **Theorem:** There does not exist a rational \(r\) such that \(r^2 = 2\).  
   **Defined key step:** Assume there exists \(r\) such that \(r^2 = 2\).  
Step 2 of the Case 3 proof completion rule of Table 6 gives
\[ r = a/b \text{ where } a, b \text{ are relatively prime} \]
\[ a^2 / b^2 = 2 \text{ by substitution and squaring} \]
\[ a^2 = 2b^2 \text{ by multiplication of equals} \]
\[ 2a^2 = b \text{ by definition of } | \]
\[ 2a = b \text{ by } 2|x^2 | \rightarrow 2|x \text{ (cf. exercise, Integers, Table 2)} \]
\[ 2k = a \text{ by definition of } | \]
\[ (2k)^2 = 2b^2 \text{ by substitution} \]
\[ 4k^2 = 2b^2 \text{ by squaring and multiplication of equals} \]
\[ 2k^2 = b^2 \text{ by multiplication of equals} \]
\[ 2b^2 \text{ by definition of } | \]
\[ 2b \text{ by } 2|x^2 | \rightarrow 2|x \]

2|a and 2|b contradicts the assumption that a and b are relatively prime. This completes the proof.

4. Universally quantified equations

Theorem: For all r/s, t/u, r/s - t/u = (ru - st)/su
 Defined key step: \[ r = s \left( (ru - st)/su + t/u \right) \]

By distributivity of Step 2 of the Case 4 proof completion rule of Table 6 we get \[ r = s (ru - st)/su + st/u \]. Application of Step 2 of the rule gives

\[ r = (ru - st)/u + st/u \text{ by b \cdot a/b + a} \]
\[ r = [(ru - st) + st]/u \text{ by a/b + c/b + (a + c)/b} \]
\[ r = ru/u \text{ by (a - b) + b + a} \]
\[ r = r \cdot 1 \text{ by a/a + a} \]
\[ r = r \text{ by a \cdot 1 + a} \]

Since the expressions on both sides of the equation are the same the proof is finished.

5. Uniqueness

Theorem: For all a/b, c/d such that a/b - c/d = x, x is unique

Defined key steps: \[ a/b = (x_1 + c/d) \]
\[ a/b = (x_2 + c/d) \]

Applying Step 1 of the uniqueness proof completion rule of Table 6 we obtain:

\[ x_1 + c/d = x_2 + c/d \text{ by substitution} \]
\[ x_1 = x_2 \text{ by cancellation} \]

Since \[ x_1 = x_2 \], the proof is finished.

6. Existence

Theorem: For all a/b, c/d (b, d \neq 0), there exists an x such that a/b - c/d = x

Defined key step: \[ a/b = [a/b + c/d + c/d] \]

Application of Step 2 of the existence proof completion rule of Table 6 gives

\[ a/b = [a/b + 0] \text{ by } -a + a + 0 \]
\[ a/b = a/b \text{ by } a + 0 + a \]

Since the expressions on both sides of the equation are the same, the proof is finished.

To summarize so far, the above analysis is based on the
assumption that proof completion subproblems are first redefined (so that initially given relations and operations are defined). Subsequent analysis was restricted to the rationals, and rules were identified by which the remaining steps in each type of theorem could be generated. (Note: An alternative way of completing the proofs would be to introduce a simple higher-order composition rule similar to those in Chapters 2 and 3. In those cases where relations and/or operations are undefined, the composition rule would operate on a simpler version of the redefinition rule and a proof completion rule, and generate composites of these two rules. In turn, the composite rule would first redefine relations and operations in the key first steps, and then generate the remaining steps in the proof. The termination of such proofs can be made explicit by applying the inverse of the redefinition rule to the defined output generated by the above composite rule.)

Six proof completion rules, parallel to those given in Table 6, can be devised for each of the other number systems. In the case of the whole numbers and the integers, the corresponding operations and decisions (relations) are relatively restricted. For example, the operations of subtraction \((a - b)\) and division \((a \div b)\) in the whole number system may be applied, respectively, only where \(a > b\) and \(b\mid a\) (\(b\) is a factor of \(a\)). Furthermore, there are no additive or multiplicative inverses in the system (except for 0 and 1). With the integers, the only restrictions that are relevant involve division.

With the algebraic extension \(\sqrt{2}\) over the rationals and the reals, on the other hand, not only are all of the rational operations and relations applicable but, in addition, there are other allowable operations/relations as well. For example, addition and multiplication with \(\sqrt{2}\) and corresponding relations such as \(\sqrt{2} > 0\) and \(\sqrt{2} \neq \text{rational}\) are allowable in the algebraic extension. In addition, operations such as \(\frac{n}{\sqrt{x}}\) (nth root), \(\left(\frac{n}{\sqrt{a}}\right)^n = a\) (exponential rules), and completion of the square are allowable in the reals.

Knowledge of the former type (e.g., restricted operations) may be represented in terms of the corresponding rational number operators with suitably restricted domains. (In the case of the inverse operators, the domains would be limited to one of the single elements 0 or 1.) Knowledge of the latter type would be represented simply in terms of new rules (operations) and relations.

More important for present purposes, it is easy to envisage higher-order substitution (restriction) and generalization rules by which the sets of proof completion rules, associated with the various number systems, can be derived from the corresponding proof completion rules associated with the rationals. Thus given: (1) a theorem involving the whole number system/Integers, including the corresponding defined proof completion sub-
problem, (2) the corresponding restricted operations and decisions, and (3) that proof completion rule (rationals) associated with the given type of theorem; the needed proof completion rule (whole numbers or integers) can be derived by simply substituting the restricted operations and decisions for the corresponding ones in the proof completion rule for the rationals. For example, if applied to a Case 4 proof completion subproblem over the whole numbers, and the fourth proof completion rule of Table 6, the higher-order substitution rule would generate the following proof completion rule for whole numbers.

1. If \( x \cdot 0 = 0 \) is the key step, add \( x \cdot 0 \) to both sides of the equation and indicate addition of equals in the proof; if \( 2x \) is the first step, apply \( 2 \cdot 1 + 1 \).

2. If the expressions on both sides of the equation are the same, stop; otherwise commute, associate and distribute terms (apply \( x + 1 \cdot x \) where required before distributing) and simplify the equation by applying one of the following as needed: \( a + 0 = a \); \( a \cdot 1 = a \); \( a \cdot 0 = 0 \) (after proven); \( (a - b) + b = a \) (for \( a > b \)); \( (a \div b) \cdot b = a \) (for \( b \mid a \)) and indicate in the proof which is used.

3. Reapply Step 2.

The higher-order generalization rule would operate on similar inputs, but rather than substituting new operations and decisions for old ones it would simply add the new operations/decisions allowable in the new system (i.e., the algebraic extension or the reals). For instance, application of the higher-order generalization rule to a proof completion subproblem over the algebraic extension \( \mathbb{Q}(\sqrt{-2}) \) involving existence, and the sixth proof completion rule of Table 6, would yield the following proof completion rule for the extension \( \mathbb{Q}(\sqrt{-2}) \).

1. Combine terms through multiplication and squaring \( \sqrt{-2} (-\sqrt{-2}) \) as needed.

2. If the expressions on both sides of the equation are the same, stop; otherwise commute, associate and distribute rational terms and \( \sqrt{-2} \) terms in order to simplify the equation by applying one of the following, as needed:
   \[ b \cdot a/b = a \] (for \( b \neq 0 \)), \( a + -a = 0 \), \( a + 0 + a \),
   \( a \cdot 0 = 0 \) (after proven), \( a \cdot 1 = a \), \( (a - b) + b = a \)
   and indicate in the proof which justification is used.

3. Reapply Step 2.

With regard to the reals, notice that there are two different real-number system rules, the real-numbers solution rule, and the proof completion rule over the reals. Each rule plays a distinct role in generating proofs. The reals solution rule is used to generate "candidate" elements (equations) in existence proofs of all types. In turn, these elements are systematically checked via appropriate proof completion rules to see if they satisfy the postulated existence conditions. The solution rule for the reals is relatively more complex than the reals proof.
completion rule because the former applies in a wider variety of computational situations. On the other hand, although it applies in a more limited set of situations, the proof completion rule generates reasons (justification) for steps in addition to the steps themselves.

The fact that we have assumed the availability of the real-number solution rule may raise some question in the reader's mind as to why we based our analysis of the proof completion subproblems on the rationals rather than on the reals. Our main reasons were two in number:

1. As suggested in the preceding paragraph, there is a difference between simple intuitive computational ability without explicit understanding of why each operation/decision can be applied, and the ability not only to perform, but also to justify the use of such operations/relations. While others may disagree, we felt that it was reasonable to assume that capable ninth-grade students have the former capability but not the latter.

2. By basing our analysis on the rationals it was possible to illustrate two kinds of higher-order rules, one involving substitution (restriction) and the other generalization. If the reals had been used, only restriction would have been required.

While on the subject of higher-order rules, we note parenthetically that the above higher-order restriction and generalization rules can be extended in a natural way to apply in a broader range of situations than indicated. Perhaps the most trivial generalizations, for example, would involve new number systems (e.g., different algebraic extensions) in which the higher-order rules might be used to generate proof completion rules for the new systems. Further generalizations might involve quite different areas of application that deal only indirectly with number systems. For example, the above substitution rule is directly analogous to, and could be generalized to include, the substitution rules identified in Scandura et al. (1971; also see Chapter 11). The latter were concerned with such things as changing bases in numeration systems. Even in everyday life we solve new problems by analogy to (substitution in rules for dealing with) familiar situations (e.g., by substituting a stone for a hammer where the latter is not available), or by generalizing learned rules (e.g., generalizing from one or two instances of dishonest behavior to general distrust).

4This is over and above the complexity introduced simply because we spelled out the individual steps in greater detail in Table 5 than in describing the proof completion rules.
6. SUMMARY AND COMPUTATIONAL ADEQUACY OF THE RULE SET

By way of summary, we have identified a total of 24 rules in our analysis: 10 problem definition rules, 12 lower-order (proof-generating) rules, and 2 higher-order derivation rules. The 24 basic rules and their specific functions are summarized in Table 7.

<table>
<thead>
<tr>
<th>Type/Rule</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem Definition Rules</td>
<td>Each Reformulates or Breaks Down Given Theorems-to-prove (TTP) into Sub TTP</td>
</tr>
<tr>
<td>D1 Compound Problem Definition Rules</td>
<td>Breaks Compound TTP into Simple TTP</td>
</tr>
<tr>
<td>Biconditional (P if and only if Q)</td>
<td>Breaks compound biconditional TTP into pairs of conditional TTP (If P, then Q; If Q, then P)</td>
</tr>
<tr>
<td>D2 Existence and Uniqueness</td>
<td>Breaks compound existence and uniqueness TTP into an existence sub TTP and a uniqueness sub TTP (There exists an X; X is unique)</td>
</tr>
<tr>
<td>(There exists a unique....)</td>
<td></td>
</tr>
<tr>
<td>D3 Simple Antecedent-Compound Consequent (If P, then Q or R)</td>
<td>Breaks compound simple antecedent-compound consequent TTP into pairs of TTP (If P and not Q, then R; if P and not R, then Q)</td>
</tr>
<tr>
<td>Six Key Step Definition Rules</td>
<td>Each breaks simple TTP into pairs of sub TTP (key step subproblem and proof completion subproblem)</td>
</tr>
<tr>
<td>KD1 Simple antecedent-simple consequent</td>
<td>1. (theorem, theorem + antecedent ) (theorem + antecedent, completed proof (through consequent) )</td>
</tr>
<tr>
<td>KD2 Compound antecedent-simple consequent</td>
<td>2. (theorem, theorem + compound antecedent ) (theorem + compound antecedent, completed proof (through consequent) )</td>
</tr>
<tr>
<td>KD3 Existence denied</td>
<td>3. (theorem, theorem + assumed existence ) (theorem + assumed existence, completed proof (through contradiction) )</td>
</tr>
<tr>
<td>KD4 Universally quantified equations</td>
<td>4. (theorem, theorem + equation ) (theorem + equation, completed proof (through equation with identical expressions on both sides) )</td>
</tr>
<tr>
<td>KD5 Uniqueness of element</td>
<td>5. (theorem, theorem + assumed two elements x₁ and x₂ ) (theorem + assumed two elements, completed proof (through x₁ = x₂) )</td>
</tr>
<tr>
<td>KD6 Existence of element</td>
<td>6. (theorem, theorem + postulated element ) (theorem + postulated element, completed proof (through equation with identical expressions on both sides) )</td>
</tr>
<tr>
<td>RD Redefinition Rule</td>
<td>Redefines proof completion subproblems involving undefined relations/operations in terms of corresponding proof completion subproblems involving defined relations/operations</td>
</tr>
</tbody>
</table>
Proof-Generating Rules

<table>
<thead>
<tr>
<th>Proof-Generating Rules</th>
<th>Generate Key Steps or Completed Proofs</th>
</tr>
</thead>
<tbody>
<tr>
<td>KG1 Simple antecedent-simple consequent</td>
<td>Generate Key Step for the Six Types of Theorems</td>
</tr>
<tr>
<td>KG2 Compound antecedent-simple consequent</td>
<td>1. generates antecedent</td>
</tr>
<tr>
<td>KG3 Existence denied</td>
<td>2. generates compound antecedent</td>
</tr>
<tr>
<td>KG4 Universally quantified equations</td>
<td>3. generates assumption of existence</td>
</tr>
<tr>
<td>KG5 Uniqueness of element</td>
<td>4. generates given equation</td>
</tr>
<tr>
<td>KG6 Existence of element (real-number solution rule)</td>
<td>5. generates assumption of two elements</td>
</tr>
<tr>
<td></td>
<td>6. generates postulated element</td>
</tr>
</tbody>
</table>

Six Proof Completion Rules for Rationals

<table>
<thead>
<tr>
<th>Proof Completion Rules</th>
<th>Generate completed proofs for the six types of theorems over the rationals</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG1 Simple antecedent-simple consequent (rationals)</td>
<td>1. generates completed proof thru consequent (rationals)</td>
</tr>
<tr>
<td>CG2 Compound antecedent-simple consequent (rationals)</td>
<td>2. generates completed proof thru consequent (rationals)</td>
</tr>
<tr>
<td>CG3 Existence denied (rationals)</td>
<td>3. generates completed proof thru contradiction (rationals)</td>
</tr>
<tr>
<td>CG4 Universally quantified equations (rationals)</td>
<td>4. generates completed proof thru equation with identical expressions on both sides (rationals)</td>
</tr>
<tr>
<td>CG5 Uniqueness of element (rationals)</td>
<td>5. generates completed proof thru ( x_1 = x_2 ) (rationals)</td>
</tr>
<tr>
<td>CG6 Existence of element (rationals)</td>
<td>6. generates completed proof thru equation with identical expressions on both sides (rationals)</td>
</tr>
</tbody>
</table>

Higher-Order Rules

Given proof completion rule (rationals) and restrictions or operations/relations in new system, generates corresponding proof completion for new system.

H1 Substitution (Restriction) Rule

Generates proof completion rules for wholes/ integers (given restrictions in new system)

H2 Generalization Rule

Generates proof completion rules for algebraic extension/reals (given additional operations/relations in new system)

These rules, together with the special restrictions, operations, and relations associated with the various number systems, are computationally sufficient for proving all of the theorems and exercises in Table 2. In this regard, we implicitly assume an adequate control mechanism by which the rules may be appropriately combined (see Chapter 2). Furthermore, to the extent that these rules accurately reflect idealized competence with respect to some specifiable population, they collectively provide a potential basis for simulating individual human behavior. (The population compatibility requirement is necessary to insure that the rule set can be used effectively to assess the specific
knowledge available to individuals in the target population, see Chapters 5, 8, and 9.)

In this section, we shall limit ourselves to illustrating the computational sufficiency of the rule set. The reader may want to check the rule set with respect to other theorems/exercises in Table 2.

Example 1: Consider the theorem (rationals) "For every \( a/b, \ c/d \) in the rationals, there exists a unique \( x \) such that \( c/d \cdot x = a/b. \)" When presented with this theorem-to-prove, the first rule that is applied is the Compound Existence and Uniqueness Problem Definition Rule (D2). This gives a pair of simple theorems-to-prove. The first simple theorem involves existence (i.e., For every \( a/b, \ c/d \neq 0 \) there exists an \( x \) such that \( c/d \cdot x = a/b \)) and the second, uniqueness (i.e., Given \( x \) such that \( c/d \cdot x = a/b, \ x \) is unique).

Then the existence theorem-to-prove is broken down by the Existence Key Step Problem Definition Rule (KD6) into a key step subproblem and a proof completion subproblem. Next, the key step subproblem is solved by applying the real-number solution rule to obtain \( c/d \cdot ad/bc = a/b \) (i.e., \( x = ad/bc \)). Since the proof completion subproblem involves an undefined operation (i.e., division, \( / \)), Redefinition Rule (RD) is applied, giving a corresponding proof completion subproblem in which the operations/relations are defined. Specifically, the defined key step is \( c/d \cdot ad = a/b \cdot bc. \) Finally, the Proof-Generating Rule (CG6) is applied to the defined proof completion subproblem (key step) and the existence theorem is proved.

The simple uniqueness theorem is proven in similar fashion, only this time the Type 5 rules play the main role. The main difference between the two proofs is that the uniqueness proof involves a pair of elements \( x_1 \) and \( x_2 \) that are assumed in the first step to satisfy the equation \( c/d \cdot x = a/b. \) In this case, the first step rule (KG5) generates the equations \( c/d \cdot x_1 = a/b \) and \( c/d \cdot x_2 = a/b. \) Then, the Proof-Generating Rule (CG5) is applied to the proof completion subproblem completing the proof; it begins by generating the equation \( c/d \cdot x_1 = c/d \cdot x_2 \) and ends with the final step \( x_1 = x_2. \)

Example 2: In proving the theorem (whole numbers) "If \( x + a = a + b \) then \( x = b, \)" the Case 1 Key Step Problem Definition Rule (KD1) is applied first. This rule breaks the problem into a key step subproblem and a proof completion subproblem. The key step subproblem is solved by applying the Case 1 Key Step Rule (KG1), giving \( x + a = a + b. \) The operations/relations of the proof completion subproblem are already defined, so no redefinition is needed. However, since this subproblem involves the whole numbers rather than the rationals, a new proof comple-

5The redefinition rule applies only where / refers to the operation of division.
An appropriate proof completion rule for the whole numbers can be derived by applying the Higher-Order Substitution Rule (H1) to the restrictions associated with the whole number system and the Case 1 Proof Completion Rule (CG1) for rationals. The result is a restricted simple antecedent—Simple Consequent (Case 1) Proof Completion Rule for the whole numbers. Finally, the newly derived rule is applied to the key step \( x + a = a + b \), generating the remaining steps (and reasons) in the proof ending with \( x = b \).

**Example 3:** In the case of the theorem (reals) "For every \( a > 0, b > 0 \), \( \sqrt{ab} = \sqrt{a} \sqrt{b} \)," the first rule applied is the Case 4 Problem Definition Rule (KD4). The obtained key step subproblem is solved by applying the Case 4 Key Step Rule (KG4), which gives \( \sqrt{ab} = \sqrt{a} \sqrt{b} \) as the key first step.

Since the proof completion subproblem involves real numbers, the Higher-Order Generalization Rule (H2) is applied to the Case 4 Proof Completion Rule (CG4) for the rationals and the special operations/relations associated with the real-number system. Application of H2 has the effect of adding the operation of raising to the nth power to Step 1 of rule CG4 and the operations \( \sqrt[n]{a} + a \) and completion of the square to Step 2. As before, the final step involves applying the generalized proof completion rule (reals) to the key step \( \sqrt{ab} = \sqrt{a} \sqrt{b} \). The final step in the proof is \( ab = ab \).

Although we have not seriously attempted to define its outer limits, the rule set summarized in Table 7 is sufficient for proving many theorems and exercises not given in Brumfiel et al. (1961). Clearly, for example, any statement that is a theorem in one system (e.g., wholes) is necessarily also a theorem in any more constrained system (e.g., reals). The identified rule set can obviously be used to generate proofs of such theorems irrespective of whether or not the are made explicit in Brumfiel et al. (1961).

The following are just a few additional theorems not explicitly stated in Brumfiel et al. (1961) that can be proven via the rule set:

1. If \( a \cdot x = a \cdot b \) then \( x = b \) (whole numbers)
2. If \( a \cdot b = b \), then \( a = 1 \) (whole numbers)
3. If \( a \div x = b \div x \) (\( x \neq 0 \)), then \( a = b \) (integers)
4. There does not exist a rational \( r \) such that \( r^2 = 5 \) (rationals)
5. For every \( a/b, c/d \), \( a/b + (-c)/d = a/b - c/d \) (rationals)
6. For every pair of real numbers \( r_1, r_2 \), there exists a unique \( r \) such that \( r_1 - r_2 = r \) (reals)

There are also some less trivial statements that can be proven using the identified rules. Consider the following theorem over the whole numbers "For all \( a > b, c > d, (a - b) +"
(c - d) = (a + c) - (b + d)." In this case, application of the Universally Quantified Equations Problem Definition Rule (KD4), the Key Step Rule (KG4), the Redefinition Rule (RD), the Universally Quantified Equations Proof Completion Rule (CG4) and the Substitution Rule (HL) yields the following proof:

1. (a - b) + (c - d) = (a + c) - (b + d)
   key step
2. (a - b) + (c - d) + b + d = a + c
   by definition of -
3. ((a - b) + b) + ((c - d) + d) = a + c
   by commutativity and associativity
4. a + ((c - d) + d) = a + c
   by (a - b) + b = a (Step 2 of CG4 whole numbers)
5. a + c = a + c
   by (a - b) + b = a (i.e., (c - d) + d = c)

Another theorem that can be proved in a similar manner is, "For all a, b, -(a - b) = b - a" over the integers. In this case, rules KD4, RD, CG4, and HL are required.

Furthermore, the addition of special restrictions and/or operations/relations associated with new systems (e.g., Q (√3)) (which may be represented as rules) would make it possible to prove theorems over such systems. In this case, the same higher-order rules could be used. For example, the following are some statements about Q(√3) that are provable in the rule set:

1. There does not exist a + b √3 such that (a + b√3)² = 2
2. a + b√3 = 0 if and only if a = 0 and b = 0
3. For all a + b √3 and c + d√3 ≠ 0, there exists a unique x + y √3 such that (a + b√3)/(c + d√3) = x + y√3

The rule set does, of course, have its limitations, particularly as it stands. It cannot be used to prove such (relatively) deep theorems as the Fundamental Theorem of Arithmetic. This theorem was stated but left unproven in Brumfield et al. (1961). In fact, the rule set fails even on some apparently easy theorems (e.g., If 2|ab, then 2|a or 2|b [integers]).

In some cases, however, the rule set can easily be extended so that it does work. For example, although the rule set fails with the compound theorem "If 2|ab, then 2|a or 2|b (integers)," it does not fail by much. Without citing all of the rules involved, the identified rules are sufficient for generating the following proof steps for the associated simple theorem, "If 2|ab and 2|a, then 2|b"

1. 2|ab
2|a

2. ab = 2k₁
   a = 2k₂ + 1

3. (2k₂ + 1) b = 2k₁
   2bk₂ + b = 2k₁

Type 2 proof completion rule

For the integers

Only two steps remain to complete the proof, namely:

4. b = 2k₁ - 2bk₂
5. b = 2(k₁ - bk₂) (i.e., 2|b)

In particular, the existing Compound Antecedent, Simple Conse-
quent Proof Completion Rule lacks two simple, but in this case, important operations. To overcome this limitation it would be sufficient to incorporate the inverse of the subtraction rule \(a + b = c + b = c - a\) and distributivity over subtraction.

Finally, we note that although all of the theorems in Table 2 can be proven via the rule set, some of them required the introduction of special-purpose operations. For example, the real number theorem, "If \(ax^2 + bx + c = 0\), then \(x = (-b \pm \sqrt{b^2 - 4ac})/2a\)" is the only one that required the completion of the square procedure. In particular, this bit of knowledge (including both the domain and the operational aspect of the rule) would have to be available, as special knowledge about the reals, in order for the higher-order generalization rule to generate a proof completion rule sufficient for completing the proof (starting with \(ax^2 + bx + c = 0\)).

7. CONCLUSION AND IMPLICATIONS

As a result of our analysis, it would appear that the specific competencies necessary for proving relatively substantial classes of theorems can be identified. Moreover, given information concerning the procedures used to prove a representative sample of theorems, such competencies can be identified in a reasonably systematic manner.

A total of 24 rules (procedures) are sufficient for generating proofs of all 67 theorems and 35 exercises of Brumfiel et al. (1961), plus an undetermined number of other theorems about number systems. Among the theorems that cannot be so proved is the Fundamental Theorem of Arithmetic, which intuitively seems to require a different method of proof.

Of the basic 24 rules, 10 are relatively simple problem definition rules, 12 are proof-generating rules, and 2 are higher-order rules. The problem definition rules construct subproblems and are directly analogous to the subgoal hierarchies of Newell and Simon (1972).

There are two types of proof-generating rules (i.e., rules for solving problems): key step-generating rules and proof completion rules. The six key step generating rules are independent of particular number systems and correspond respectively to six types of theorems. All but one are extremely simple; this exception generates postulated elements (trick steps) to be tested in existence proofs.

The six corresponding proof completion rules, on the other hand, apply only with the rationals. Corresponding proof completion rules for each of the other number systems can be generated from the rationals' rules and specific information about the other number systems, by applying either the restriction or the generalization higher-order rule. Although the methods of representa-
tion differ, the proof completion rules operate as in means-ends 
/analysis (Ernst & Newell, 1969) to reduce the differences be-
tween given and goal states.

In spite of the noted similarities, the overall analysis 
differs from those proposed by Newell and Simon (1972) and/or 
Ernst and Newell (1969) in at least two important ways:

1. In means-ends analysis, production systems, and state 
space methods generally (Nilsson, 1971), solution procedures 
generated essentially by composing individual operations 
rules, productions). In the structural learning theory (cf. 
Scandura, 1973; Chapter 2, this book), new rules also may be 
generated in other ways. The present research demonstrates this 
fact in the case of restriction and generalization of proof 
generating procedures.

2. The methods by which individual rules are assumed to 
interact in generating behavior (and learning) also differ. In 
production systems, for example, it is assumed that rules (pro-
ductions) are tested one by one. If a state satisfies the domain 
condition of a rule, then it is applied. If not, the next rule 
is tested. This particular mechanism makes no specific provision 
for learning (other than to add new productions). In the 
structural learning theory, rules interact in a more general 
way. For example, allowance is made both for solving subpro-
blems in turn, which corresponds to sequential application of 
rules, and for deriving/generating (i.e., learning) new solution 
rules, in a manner that corresponds to "insight" (see Chapter 2).

The major practical implications of the research fall into 
two major categories: (1) those involving the method of analysis 
generally and, (2) those involving the specific rule set identi-
fied. The general method of analysis is suggestive, for ex-
ample, both for research in artificial intelligence and for 
curriculum development. In each case the major goal is to iden-
tify the competencies needed to perform satisfactorily on a 
given task domain; the proposed method of analysis provides a 
quasystematic means of accomplishing this.

The specific results obtained also have obvious implica-
tions for artificial intelligence and education. Most good 
programmers, for example, would have relatively little diffi-
culty in implementing the identified rule set and basic control 
structure. (Wulfeck has already done this with the geometry 
analysis of Chapter 3.) Moreover, in addition to their demon-
strated value in proving theorems about number systems, the 
problem definition and higher-order rules identified might 
reasonably be expected to play a useful role in other proof-
making domains.

A skilled mathematics educator could also use the present 
analysis to advantage, for example, in preparing instructional 
materials for teaching proof making. Just as the programmer 
would have to translate the present analysis into a form a
computer can compile, the text writer (or classroom teacher) would have to translate the analysis into a form suitable for the learner. In the latter case, for example, it is not necessary that any particular rule be learned in a particular way, say either by exposition or by discovery. The important thing is that the rules are learned. In this regard, it is our contention that if text writers and/or teachers knew exactly what it is that must be learned in order to prove theorems, then he or she would be able to do a more effective job of promoting learning than otherwise.

It also is worth noting that a text based on the obtained rule set would be quite different than the text by Brunfiel et al. (1961), even though the present rule set was derived from the latter. At a gross level, for example, proofs involving the rationals would enter much earlier than in the Brunfiel text and would provide a basis for comparison throughout. More generally, the various types of proof and relationships among theorems and systems would be made explicit for the learner rather than left to chance discovery.

One final comment by way of conclusion: Some readers might object to our analysis on the grounds that no apparent attempt has been made to relate the identified rules to what is sometimes referred to as general cognitive structure. This is true, however, only in a limited and unavoidable sense. There is no one cognitive structure to which the present rule set might be related, in the sense that individual differences in knowledge that do not directly involve the problem domain (say, the text) are irrelevant insofar as explaining and predicting basic capabilities with respect to this domain. Relationships to broader cognitive structure make a difference only with respect to problems beyond the given domain. In this case, however, structural analysis of a broader domain that includes such problems would deal specifically with such relationships.

This is not to say, however, that structures (on which rules operate) of the sort introduced in Chapter 2 may be ignored. In fact, they are implicit in the domains of the rules identified. Several of the rules identified, for example the higher-order rules, may be thought of as operating on entire numbers systems (i.e., structures) and aspects thereof (cf. Scandura, 1971, 1973, Chapter 5; also see Chapter 15, Section 3.5, this book). Although the central role such systems play was not emphasized, a precise specification of these rules would necessarily involve the properties which collectively define such systems.
REFERENCES


