A COMPUTER-ASSISTED INSTRUCTION PROGRAM FOR EXERCISES ON FINDING AXIOMS

by

Adele Goldberg and Patrick Suppes

TECHNICAL REPORT NO. 186

June 23, 1972

PSYCHOLOGY AND EDUCATION SERIES

INSTITUTE FOR MATHEMATICAL STUDIES IN THE SOCIAL SCIENCES

STANFORD UNIVERSITY

STANFORD, CALIFORNIA

A Computer-assisted Instruction Program

for Exercises on Finding Axioms*

by Adele Goldberg and Patrick Suppes

This paper describes an interactive computer-assisted system for teaching elementary mathematical logic, which was designed to handle formalizations of first-order theories suitable for presentation in a computer-assisted instruction (CAI) environment. The system provides tools with which the user can develop and then study a nonlogical axiomatic theory along whatever lines he specifies. These tools include a proof-checking program that allows the user to construct derivations by taking advantage of the theorem-proving capabilities of the computer.

Results of preliminary investigations using this computer-assisted teaching system in a manner designed to give the student greater control over the organization of his curriculum are summarized in Section 2, and Section 3 outlines initial studies on the uses of mechanical theorem provers in teaching about proof construction.

---

* This research has been supported by the National Science Foundation under Grant NSF GJ-443X2. We very much appreciate the assistance of Stephen A. Weyer in preparing this report for publication.

1. Basic Instructional Capabilities

The instructional system is written in the LISP programming language for a DEC PDP-10 computer at the CAI Laboratory of the Institute for Mathematical Studies in the Social Sciences (IMSSS), Stanford University. Figure 1, a block diagram of the basic system, views the program as an interpreter made up of two components. One, call it C, is a set of inference rules and proof procedures used in constructing derivations (DERIVE mode). It is defined in Language A (AXIOMATIZE mode). With A, the user is able to specify the vocabulary and axioms and to derive new rules of inference from axioms and established theorems. The names of the axioms, theorems and new rules, with instructions on how to use these rules, are learned by A and added to C, which is further augmented by the basic logical system. Specifically, this includes primitive rules to support a formalization of first-order predicate calculus with identity--quantifier rules, the rule of detachment (called "affirm the antecedent" and abbreviated AA), rules governing the identity relation, proper substitution (PS) and replacement, and procedures to construct conditional and indirect proofs (CP and IP).

Insert Figure 1 about here

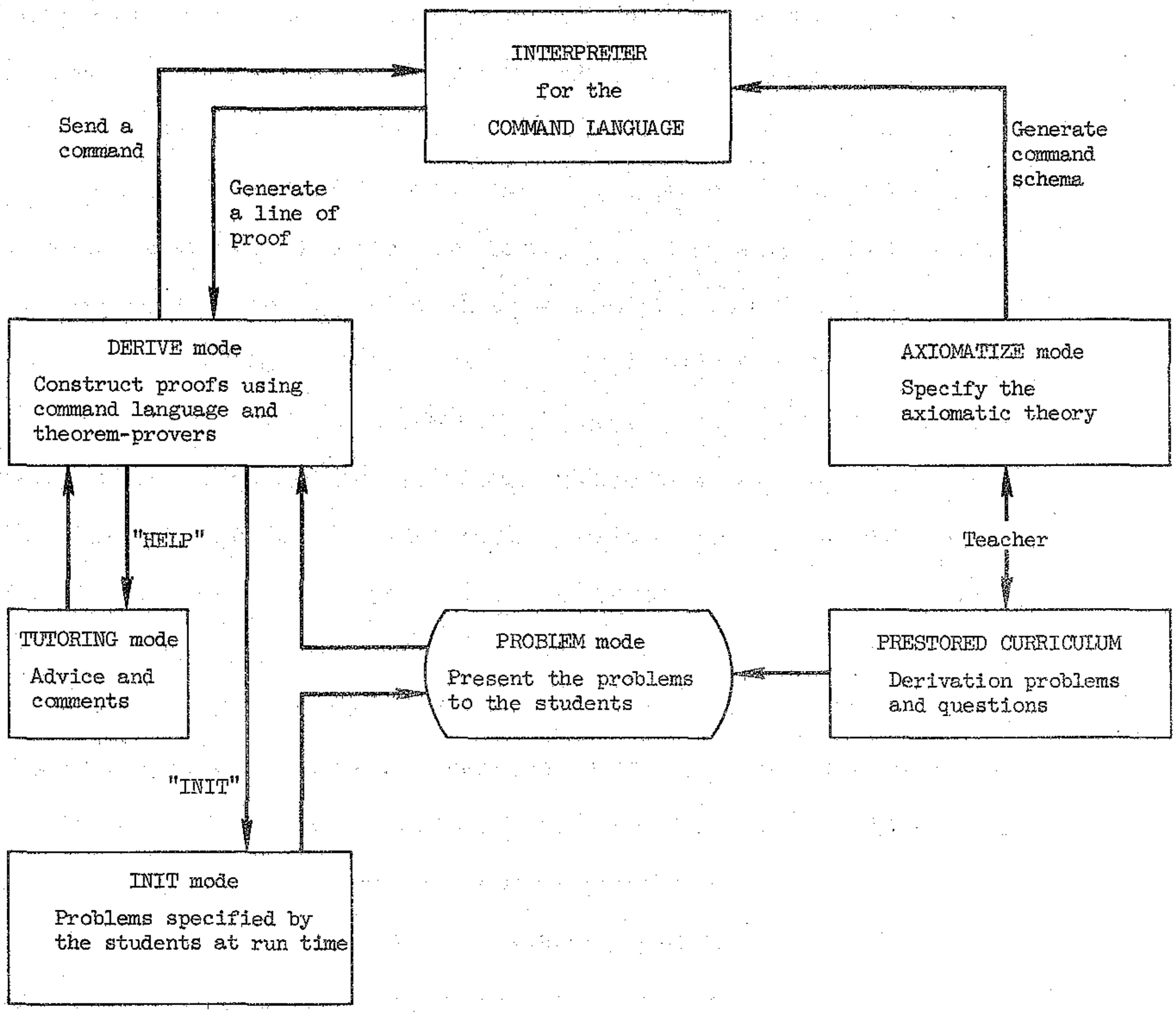

Fig. 1. Block diagram of the instructional system.

In the DERIVE mode, the interpreter takes on the role of a proof checker to verify that each step of a proof constructed by the user is a valid consequent of a set of statements of the theory. In such an interactive system, the user indicates a proof step by typing the name of a rule in C. The rule may refer to several previous lines of the proof and may include a reference to an occurrence of a term in a previous line. A properly formulated rule is a request to have the program generate a new line of the proof. The proof checker examines the format and intended application of each rule. If a rule is poorly formed, the user receives a message that explains what error was committed.

Figure 2 shows a simple example of a proof constructed within the system. Any information typed by the user is underlined; the rest is typed by the computer.

Insert Figure 2 about here

In the proof in Figure 2, the user typed WP to indicate that he wanted to type a working premise. Derived rule LC used in line (3) is a special case of the classical sentential rule of simplification; we use it to infer the left conjunct of a conjunction. The rule CP, for <u>conditional proof</u>, is a request to generate a conditional

```
PROVE      (P->Q)->((P & R)->Q)
:WP        (1)        P->Q
:WP        (2)            P & R
:2LC       (3)            P
:1.3AA     (4)            Q
:2.4CP     (5)        (P & R)->Q
:1.5CP     (6)   (P->Q)->((P & R)->Q)
CORRECT...
```

Fig. 2. Sample proof (underlining indicates student input).

statement from a working premise and some line already in the proof. The production of a line matching the problem expression is not sufficient to guarantee that the user has constructed a complete proof. Any working premises that have been introduced must be discharged in proper order by a conditional or indirect proof.

The proof checker can provide error messages for the rules it learns from the user. The same analysis routines that process rules and generate new lines of the proof are also used to compute error messages. These error-analysis procedures generate explanations of how each rule may be applied. The explanations include sample proof steps which, if possible, reference lines already in the user's proof. In this manner, the user can receive immediate feedback to help him reexamine the structure of the theory he himself has specified.

Sample recursive error messages for sentential logic are shown in Figure 3. The mnemonic DC refers to the principle of inference modus tollens. Hopefully no single student would ever make so many mistakes in a single exercise.

Insert Figure 3 about here

Labeling the user as either a teacher or student is

```
PROVE     (P->Q)->((NOT Q -> NOT P)->NOT P)
:WP          (1)          P->Q
:WP          (2)                NOT Q -> NOT P
:WP          (3)                          P
:3.1AA
          LINE 3 MUST BE A CONDITIONAL
:1.2AA
          LINE 2 MUST BE THE ANTECEDENT OF LINE 1
:1.3AA       (4)                          Q
:2.3DC       (5)                          NOT Q
:2.4.3IP
          LINE 2 WAS NOT THE LAST WORKING PREMISE INTRODUCED
:3.4.2IP
          LINE 4 IS NOT THE DENIAL OF LINE 2
:3.4.5IP     (6)                NOT P
:5.2CP
          LINE 5 IS NOT A WORKING PREMISE
:2.6CP       (7)          (NOT Q -> NOT P)-> NOT P
:1.7CP       (8)  (P->Q)->((NOT Q -> NOT P)->NOT P)
CORRECT...
```

Fig. 3. Sample error messages.

unnecessary. It is possible for the user to devise a curriculum for teaching others about the axiomatic system he has specified, so the user might be a teacher writing a computer-assisted course or a student experimenting with newly acquired skills in proof construction. Although the teacher may prescribe a curriculum, his student may interrupt the flow of exercises to make up his own exercises.

As shown in Figure 4, the student indicates his intention to alter the curriculum by typing the instruction INIT. He may avail himself of this option to test any ideas he has acquired about proof strategies or about the provability of some expression. He may repeat an exercise for practice, try more difficult derivations than those given him or prove lemmas that would help make the teacher-defined exercises easier to solve. He thus can alter the curriculum to suit his individual instructional needs. In Figure 4 we see how a student derived a new rule of inference from a lemma he has proved and then used the rule and two theorems to complete the exercise given to him. The two theorems are shown at the bottom of Figure 4. Error messages computed for the derived rule have also been included.

Insert Figure 4 about here

The student is not required to follow a particular solution path in order to construct a valid derivation. He is always free to explore within the bounds set by the inference rules, to ask questions about rules and to try any approaches he thinks are promising for finding a proof. However, the degree of student control over the curriculum can be changed at the teacher's discretion. By varying the degree of freedom given the student, we will in time be able systematically to test the comparative effectiveness of student-controlled versus fixed-branching CAI schemes.

In the present version of the instructional system, the teacher can block use of the INIT command, especially where consecutive experience with a set of exercises is desirable, or the teacher can indicate in the curriculum that a problem must be solved with or without the use of certain rules. Once the student has completed a valid solution, the proof checker chains through the sequence of rules that entered into the derivation to determine whether the student constructed the proof within restrictions specified by the teacher. If the student has not constructed a solution within the required restrictions, the proof checker will ask him to repeat the exercise.

```
DERIVE      (NOT P)->(P -> Q)
:INIT
          YOU CAN NOW REQUEST AN EXERCISE OR
          DERIVE A NEW RULE OF INFERENCE.
          TYPE  FIN  WHEN YOU ARE DONE.

:PROVE:  (P -> Q)->((Q -> R)->(P -> R))

:WP          (1)         P -> Q

:WP          (2)              Q -> R

:WP          (3)                   P

:1.3AA       (4)               Q

:2.4AA       (5)                   R

:3.5CP       (6)              P -> R

:2.6CP       (7)        (Q -> R)->(P -> R)

:1.7CP       (8)   (P -> Q)->((Q -> R)->(P -> R))

CORRECT...
NAME:LEM

:RULE
NAME:HS
FROM:LEM

:FIN
```

Fig. 4. Use of the INIT instruction.

```
NOW REDO THE EXERCISE YOU INTERRUPTED

DERIVE      (NOT P)->(P -> Q)

PS:TH2      ((NOT Q)->(NOT P))->(P -> Q)
::        (1)   ((NOT Q)->(NOT P))->(P -> Q)

PS:TH1      P->(Q -> P)
::P:NOT P
::Q:NOT Q
::        (2)   (NOT P)->((NOT Q)->(NOT P))

:1.2HS
        HS REQUIRES 3 LINE NUMBERS

:WP       (3)        NOT P

:1.2.3HS
        THE ANTECEDENT OF LINE 2 MUST BE
        THE CONSEQUENT OF LINE 1

:2.1.3HS  (4)        P -> Q

:3.4CP    (5)   (NOT P)->(P -> Q)

CORRECT...
```

Fig. 4, continued.

In summary, the instructional system was designed to increase the level of active participation by the student. A student can specify a first-order theory and build his own command language with which to construct proofs. He can then (a) make up his own exercises, (b) communicate with a proof checker to verify whether his solutions are correct, (c) work on an exercise without interference, i.e., he may try any solution path regardless of whether it approaches a successful proof, and (d) receive immediate feedback on errors. Furthermore, he can ask about previously learned material and receive advice and comments on his work which take into account only the material that he knows, i.e., material that he was taught or that he developed. This flexibility was made possible only by developing a system that was not restricted to a fixed curriculum or a fixed language for communicating with the students. Consequently, the system can be adapted to the instructional needs of a large class of users and a wide range of possible curriculums.

2. <u>The Finding-Axioms Exercises</u>

The possible uses for this system are varied. The resources of the program allow the user to axiomatize the theoretical structures of some elementary domain of science. For example, a graduate student in logic is using the program to study axiomatizations of elementary geometry; another is using it to develop a curriculum for the predicate calculus and elementary number theory.

A program like this is suitable for problem-solving tasks of varying levels of complexity, such as the Finding-Axioms Exercises, which are modeled after the famous R. L. Moore method of instruction. The students were either in an elementary course in logic at Stanford or 12-year-old seventh graders from a junior high school in Palo Alto, California. A description of this task and the results of actual classroom use of the instructional program follow.

A Finding-Axioms Exercise consists of a list of well-formed formulas of a formal theory. The student must select at most N formulas from which the rest can be derived and show that the selection is correct by carrying out the derivations. The student is encouraged to establish a definite order of the formulas to be proved so that one derived formula can be effectively used in deriving another.

For this exercise, the logical rules of inference already discussed were made available to the student. The only difference in format was the additional ability to reference the formulas in each exercise by their numbered position in the list. Each student was expected to infer the syntax for a well-formed expression from the formulas in the list.

At the time we introduced the Finding-Axioms Exercises, the students had learned a quantifier-free version of first-order logic, some elementary algebra and some Boolean algebra. Their teacher was an earlier version of the computer-assisted logic program in use at IMSSS since 1964 (Suppes and Binford, 1965; Suppes and Ihrke, 1970; Goldberg, 1971; Suppes, 1971). We could thus assume that the students were familiar with the operation of a teletypewriter (the student's input device) and with the manner in which proofs could be constructed on the computer. The students were adept at taking advantage of some of the more flexible features of the new instructional system, namely, with (a) naming and renaming formulas to find a suitable group to choose as the axioms, (b) making up lemmas to decrease the number of steps in some of the more complicated proofs, and (c) devising derivation problems with premises to test the effect of adding one or more axioms.

The four Finding-Axioms Exercises are shown in Figure 5. The students' previous experience with an axiomatization of the natural numbers made Exercise 1 conceptually easier and gave them an opportunity to adjust to the features of the new instructional system. Most of the college students completed all four exercises;

Exercise 4 was not given to the seventh graders because of the level of difficulty of the substitutions required.

Insert Figure 5 about here

In Tables 1-4 we show the axioms selected by each student for each exercise in the obvious matrix form. Variation in the selection of axioms occurred in all four exercises, probably least in the most trivial of the four, the first exercise. However, even here one, Student 11, managed to use only nine axioms. Tables 5-8 show the formulas used by each student to prove the theorems in each exercise.

Insert Tables 1-8 about here

In the case of Exercise 2, Students 12, 13 and 14, the seventh graders, did not finish, and the axioms they selected are not necessarily adequate--this is immediately obvious in the case of Student 13. These three young students did complete Exercise 3. The 11 college students, i.e., Students 1-11, all selected one of two sets of axioms; the choice was as evenly divided as possible with an odd number of students. Axioms 4 and 5 are definitional in character. By ordinary axiomatic standards the choice

Exercise 1. Natural Numbers

About 10 axioms seem to be needed. See what you can do.

1. X+1=Y+1 -> X=Y
2. NOT X+1=0
3. (NOT X=Y) -> (NOT X+1=Y+1)
4. X+0=X
5. NOT X+1=X
6. X+(Y+1)=(X+Y)+1
7. X+Y=Y+X
8. 0+X=X
9. Zx0 = 0
10. Z x(Y+1)=(Z x Y)+Z
11. Z x Y=Y x Z
12. 0 x Z=0
13. Z x 1=Z
14. NOT 0=1

Fig. 5. Finding-Axioms Exercises.

Exercise 2. Order Properties

Find 4 of the 8 statements that will serve as axioms. For intuitive purposes, think of P as greater than and Q as equal to or greater than.

1. X P Y & Y P Z -> X P Z

2. X P Y -> NOT Y P X

3. NOT X P X

4. X Q Y -> NOT Y P X

5. NOT X P Y -> Y Q X

6. X Q Y OR Y Q X

7. X Q Y & Y Q Z -> X Q Z

8. X Q X

Fig. 5, continued.

Exercise 3. Lattices

This exercise is close to the earlier work on Boolean algebra. The operations are the Boolean operations and the relation Q is like the relation of subset. But there is no complementation. We let ^ stand for intersection, and V for union.

1. X Q Y & Y Q Z -> X Q Z [Transitivity]

2. X Q Y & Y Q X -> X = Y [Antisymmetry]

3. X Q X [Reflexive]

4. X Q Y -> X ^ Y = X

5. X ^ Y = X -> X Q Y

6. X ^ X = X [Idempotent]

7. X V X = X

8. X ^ Y = Y ^ X [Commutative]

9. X V Y = Y V X

10. X ^ (Y ^ Z)=(X ^ Y)^ Z [Associative]

11. X V (Y V Z)=(X V Y)V Z

12. X ^ (X V Y)= X [Absorption]

13. X V (X ^ Y) = X

14. X Q Y & X Q Z -> X^Y = X^Z

15. X Q Y -> X V Y=Y

16. X V Y = Y -> X Q Y

17. X ^ Y Q X

18. X Q X V Y

19. X Q Z & Y Q Z -> X V Y Q Z

Fig. 5, continued.

20. X Q Y & X Q Z -> X Q Y ^ Z

21. X ^ Y Q X V Y

22. X ^ (X ^ Y) = X ^ Y

23. X ^ (Y ^ Z) Q Y

24. X ^ Y Q X V Z

25. X Q (Y ^ Z) V X

Exercise 4. Betweenness

B(X,Y,Z) means that Y is between X and Z on a line segment. We still call it betweenness when X=Y or Y=Z. Find 5 of the 11 statements as axioms.

1. B(X,X,X)

2. B(X,Y,X) -> X=Y

3. B(X,Y,Z) -> B(Z,Y,X)

4. X=Y -> B(X,Y,Z)

5. B(X,Y,W) & B(Y,Z,W) -> B(X,Y,Z)

6. (NOT Y=Z & B(X,Y,Z) & B(Y,Z,W)) -> B(X,Y,W)

7. B(X,Y,Z) & B(X,W,Z) -> B(Y,W,Z) OR B(W,Y,Z)

8. B(X,Y,Z) & B(Y,X,Z) -> X=Y

9. (B(X,Y,Z) OR B(Y,Z,X)) OR B(Z,X,Y)

10. ((NOT Y=Z) & B(X,Y,Z) & B(Y,Z,W)) -> B(X,Z,W)

11. B(X,Y,X) -> B(Z,Y,X)

Fig. 5, continued.

TABLE 1. EXERCISE 1

AXIOMS

Formula no. \ Student no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	X		X	X	X	X	X	X	X		X	X	X	X
2	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3		X							X	X				
4	X	X	X	X		X	X	X	X	X	X	X	X	
5	X	X	X	X	X	X	X	X	X	X	X	X	X	X
6	X	X	X	X	X	X	X	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X	X	X	X	X	X	X	X
8					X									X
9	X	X		X	X	X	X	X	X	X	X		X	X
10	X	X	X	X	X	X	X	X	X	X	X	X	X	X
11	X	X	X	X	X	X	X	X	X	X	X	X	X	X
12			X									X		
13	X	X		X	X	X	X	X		X		X	X	X
14			X											

TABLE 2. EXERCISE 2

AXIOMS

Formula no. \ Student no.	1	2	3	4	5	6	7	8	9	10	11	12*	13*	14*
1												X	X	
2	X		X	X			X					X		
3														X
4	X	X	X	X	X	X	X	X	X	X	X	X	X	
5	X	X	X	X	X	X	X	X	X	X	X	X		X
6		X			X	X		X	X	X	X		X	X
7	X	X	X	X	X	X	X	X	X	X	X			X
8													X	

TABLE 3. EXERCISE 3

AXIOMS

Formula no. \ Student no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1									X					
2														
3														
4	X	X	X	X	X		X	X	X	X	X		X	X
5		X	X	X		X	X	X		X	X	X	X	X
6	X					X						X		
7														
8	X	X	X	X	X	X	X	X	X	X	X	X	X	X
9	X	X	X	X	X	X	X	X	X	X	X	X	X	X
10	X	X	X	X	X	X	X	X	X	X	X	X	X	X
11	X	X	X	X	X	X	X	X	X	X	X	X	X	X
12		X	X	X		X	X			X	X	X	X	X
13	X	X	X	X			X		X	X	X		X	X
14														
15					X	X		X				X		
16	X							X						
17					X									
18					X									
.														
.														
.														
25									X					

TABLE 4. EXERCISE 4

AXIOMS

Formula no. \ Student no.	1	2	3	4	5	6	7	8	9	10	11
1											
2			X	X		X			X	X	X
3		X	X	X		X		X	X	X	X
4								X			
5		X	X	X		X		X	X	X	X
6		X								X	
7											
8		X						X			
9		X	X	X		X		X	X	X	X
10			X	X		X			X		X
11											

TABLE 5. EXERCISE 1

THEOREMS

Student no.	Formula no.	Proved using formulas no.
1	3	1
	8	4,7
	12	9,11
	14	5,12
2	1	3
	8	4,7
	12	9,11
	14	4,5
3	13	4,7,10,11,12
	8	4,7
	9	11,12
	3	1
4	8	4,7
	12	9,11
	14	4,5,7
	3	1
5	4	7,8
	12	9,11
	3	1
	14	4,5
6	3	1
	8	4,7
	14	5,8
	12	9,11
7	8	4,7
	3	1
	12	9,11
	14	5,7,8
8	14	4,5
	8	4,7
	12	9,11
	3	1
9	8	4,7
	14	5,8
	13	8,9,10
	12	9,11
10	1	3
	8	4,7
	12	9,11
	14	4,5
11**	8	4,7
	12	9,11
	13	8,9,10
	3	10
	14	2,8
12	9	11,12
	8	4,7
	3	1
	14	2,4,5
13	8	4,7
	12	9,11
	14	4,5
	3	1
14	4	7,8
	12	9,11
	3	4,7,8,9,10
	14	4,5

TABLE 6. EXERCISE 2

THEOREMS

Student no.	Formula no.	Proved using formulas no.	Student no.	Formula no.	Proved using formulas no.
1	1	2,4,5,7	8	8	6
	3	2		3	4,8
	6	2,5		2	L1,6
	8	4,5		1	2,4,5,6,7
2	8	6	9	2	L1,L4,4
	3	4,6		3	2
	2	4,6		8	3,5
	1	2,4,5,7		1	L3,4,5,7
3	6	2,5	10	1	2,4,5,7
	3	2		3	2
	8	6		6	2,5
	1	2,4,5,7		8	3,5
4	3	2	11	8	6
	8	3,5		3	4,8
	6	2,5		2	L1,4,6
	1	2,4,5,7		1	L3,4,5,7
5	8	6	12	3	2
	3	4		8	5,3
	1	4,5,6,7		6*	
	2	1,3		7*	
6	2	4,6	13	3	4,8
	1	2,4,5		2	1,3
	3	2		5*	
	8	3,5		7*	
7	3	2	14	8	3,5
	8	3,5		1*	
	6	3,5		2	1,3
	1	2,4,5,7		4*	

TABLE 7. EXERCISE 3

THEOREMS

Student no.	Formula no.	Proved using formula no.	Student no.	Formula no.	Proved using formula no.
1	15	4,8,9,13	8	2	9,15
	5	8,9,13,16		19	L1,11,15
	1	4,5,10		20	L2,4,10
	2	9,15		3	L3
	3	5,6		7	3,15
	7	3,15		6	3,4
	18	7,11,16		22	6,10
	12	4,18		17	5,8,22
	14	4		18	7,11,16
	22	6,10		12	4,18
	17	5,8,22		13	9,15,17
	19	11,15,16		21	9,11,13,16
	20	4,5,8,10		14	4
	21	9,11,13,16		23	5,6,8,10
	23	5,6,8,10		24	5,8,10,12
	24	9,11,13,16		25	5,9,12
	25	5,9,12		1	4,5,10
2	1	4,5,10	9	2	4,8
	2	4,8		3	9,13,25
	6	12,13		6	3,4
	3	5,6		7	6,13
	16	5,12		5	6,8,13,25
	15	4,9,8,13		14	4
	7	3,15		15	4,8,9,13
	14	4		16	6,9,25
	17	13,16		17	6,13,25
	18	5,12		18	6,9,25
	19	11,15,16		19	11,15,18
	20	4,5,8,10		20	4,5,8,10
	21	1,17,18		12	4,6,9,25
	22	6,10		21	5,9,10,12
	23	8,10,17		22	4,8,17
	24	1,17,18		23	8,10,17
	25	9,18		24	10,12,23

TABLE 7, continued.

3	15	4,8,9,13	10	1	4,5,10
	16	5,12		2	4,8
	6	12,13		6	8,12,13
	7	12,13		7	3,15
	2	4,8		14	4
	14	4		17	5,6,8,10
	24	9,11,13,16		18	7,11,16
	17	7,24		19	11,15,16
	18	6,24		20	4,5,8,10
	25	6,9,24		21	1,17,18
	23	24,10,8,7		22	10,6
	22	6,10		23	8,10,17
	21	24		24	1,17,18
	1	5,11,12,15		25	9,18
	3	7,16		15	4,8,9,13
	19	11,15,16		16	5,12
	20	4,5,10		3	5,6
4	1	4,5,10	11	6	12,13
	14	4		7	12,13
	2	4,8		3	5,6
	20	4,5,8,10		2	4,8
	19	11,15,16		14	4
	15	4,8,9,13		22	6,10
	6	12,13		23	5,6,8,10
	3	5,6		17	5,6,8,10
	7	3,15		18	5,12
	18	7,11,16		15	4,9,13
	17	5,6,8,10		16	5,12
	21	5,9,10,12		25	7,9,11,16
	22	6,10		24	5,8,10,12
	23	5,6,8,10		21	5,9,10,12
	24	11,13,16		19	11,15,16
	25	7,9,11,16		20	4,5,10
	16	5,12		1	4,5,10

TABLE 7, continued.

5	2	4
	14	4
	25	9,18
	5	8,17
	16	18
	12	4,18
	13	9,15,17
	23	8,10,17
	20	4,5,10
	19	11,12,16
	1	4,8,23
	3	13,25
	4	12,15
	3	5,6
	6	3,4
	22	6,10
	7	3,12
6	16	5,12
	4	12,15
	3	5,6
	7	10,15
	1	11,15,16
	2	9,15
	17	5,6,8,10
	13	9,15,17
	14	4
	18	5,12
	19	11,15,16
	20	4,5,10
	24	9,11,13,16
	21	24
	22	4,8,17
	23	8,10,17
	25	9,18
12	22	6,10
	14	4
	16	5,12
	3	5,6
	2	3,4,6,8,14
	1	4,5,10
	18	7,11,16
	19	10,11,15,16
	7	3,15
	20	4,5,10
	17	5,6,8,10
	21	1,17,18
	23	8,10,17
	25	9,18
	4	5,12
	13	9,15,17
	24	1,17,18
13	6	3,4
	2	4,6,8,10
	7	3,15
	1	4,5,10
	14	4,8,10
	16	5,12
	20	4,5,10
	17	9,12,13,16
	18	5,12
	19	4,11,15,16
	22	4,8,17
	3	5,12,13
	15	4,8,9,13
	21	1,16,17,18
	23	8,10,17
	24*	
	25*	

TABLE 7, continued.

7	3	5,12,13	14	7	3,15
	6	3,4		1	4,5,10
	25	5,9,12		14	4
	1	4,5,10		2	4,8,10
	24	8,10,12		16	5,12
	14	4		17	3,9,13,16
	2	4,8		18	5,12
	18	5,12		15	4,8,9,13,17
	15	4,8,9,13		6	7,12
	16	5,12		3	5,12,13
	7	15		19	4,9,10,11, 12,15,16
	17	9,11,16		20	4,5,10
	19	11,15,16		22	6,10
	20	4,5,10		21	1,17,18
	21	1,17,18		23	8,10,17
	22	6,10		24*	
	23	8,10,17		25*	

TABLE 8. EXERCISE 4

THEOREMS

Student no.	Formula no.	Proved using formulas no.	Student no.	Formula no.	Proved using formulas no.
2	1	9	8	1	4
	2	1,3,5,8		2	3,4,8
	4	2,3,9		11	2,3,4
	10	3,6		6	L5,L4,3,8,9
	11	2,3,4		10	L1,L2,3,8,9
	7	2,3,4,6,9,10		7*	
3	1	9	9	1	9
	8	2,5		8	2,5
	4	1,2,3,9		11	2,3,9
	11	2,3,4		4	1,3,11
	6	3,10		7	L1,3,8,9,10
	7	L2,2,3,9,10		6	L1,3,5,8,9,10
4	4	2,3,9	10	1	9
	1	4		4	L1
	11	2,3,4		8	2,5
	8	2,5		11	2,3,4
	6	3,10		7	L3,L4,L5,3,5,9
	7	2,3,4,5,9,10		10	L1,2,3,5,6,9
6	1	9	11	4	2,3,9
	11	L1,2,3		1	4
	4	1,3,11		11	2,3,4
	6	3,10		8	2,5
	8	2,5		6	3,10
	7	L1,3,8,9,10		7*	

of Formulas 2 and 7 as the remaining axioms is somewhat surprising; certainly they are not a choice that would be found in any text dealing with these elementary order properties.

Exercise 3 on lattices produced the greatest variety of axioms. Seven students of the 14 used the same set, but the remaining seven students produced five additional sets of axioms. Student 8 introduced the following three lemmas to shorten his proofs.

Lemma 1: (NOT X Q Y)-> (NOT X V Y = Y)

Lemma 2: (NOT X Q Y)-> (NOT X ^ Y = X)

Lemma 3: (X=Y)-> (X Q Y) & (Y Q X)

Because of its greater difficulty, Exercise 4 was completed by only six of the 11 college students. (The college course required completion of this exercise only by those students who wanted the highest grade.) Students 8 and 11 selected adequate axioms, but did not complete the proofs. Various students introduced and used the following five lemmas.

Lemma 1: B(X,X,Y)

Lemma 2: B(X,Y,Y)

Lemma 3: NOT (B(Y,W,Z) OR B(W,Y,Z))->
NOT B(Y,W,Z)

Lemma 4: NOT B(Z,Y,X)-> NOT B(X,Y,Z)

Lemma 5: NOT B(Z,Y,W)-> NOT Z=Y

We have looked at the data on the order in which formulas that were not selected as axioms were proved by each student. We also have looked at the sequence of rejection of formulas as possible axioms, but the data are too elaborate to reproduce here, and thus we have restricted ourselves to points that seem of particular interest.

In conversations with the students we found that they experienced some difficulty in crossing over from one exercise to another when identical symbols were used (namely, Q). There was also an initial desire on the part of the seventh graders to use theorems from one exercise for proving theorems in another. This confusion in distinguishing between the distinct theories might stem from the CAI environment in which they had been studying earlier, one in which the curriculum was designed so that each axiom introduced and each theorem proved were always available for use in subsequent proofs.

To find a suitable set of axioms for a given exercise, the students had to experiment with different combinations of the formulas, perhaps trying to construct proofs for formulas already designated as axioms. In doing this, they sometimes made an error that we term "cycling"--proving Formula A from B designated as an axiom, and then proving B from the already established Theorem A. A trial-and-error heuristic for choosing axioms was developed by the students from this cycling problem: if the proofs for two formulas depend on one another, assume that one should be an axiom. We offered another heuristic: to be able to prove things about the relationships between the variables, pick as axioms those formulas that define the relations in terms of one another, not just singularly.

The longest proofs were of formulas (7) and (10) of Exercise 4. Each of the 11 college students attempted to find a proof of Formula (7). The mean length of these proofs was 45 lines, with the shortest, which used three lemmas, being 39 lines, and the longest being 77 lines without using any lemmas. Only three students proved Formula (10), with proofs of 17, 32 and 43 lines. The rather elegant shortest proof of 17 lines is shown as Figure 6.

Insert Figure 6 about here

The general response of the college students to these exercises was surprisingly positive. Even though the Finding-Axioms Exercises were considerably harder than the earlier exercises in the course, all of the students expressed a clear preference for the Finding-Axioms Exercises, because of the apparent greater interest in the exploratory search for axioms as opposed to straight

```
:PROVE: 10
PROVE      (((NOT Y=Z)& B(X,Y,Z))& B(Y,Z,W))-> B(X,Z,W)

:WP         (1)         ((NOT Y=Z)& B(X,Y,Z))& B(Y,Z,W)

:1RC        (2)         B(Y,Z,W)

:AXB      B(X,Y,Z)-> B(Z,Y,X)
X::Y
Y::Z
Z::W        (3)         B(Y,Z,W)-> B(W,Z,Y)

:3,2AA      (4)         B(W,Z,Y)

:1LC        (5)         (NOT Y=Z)& B(X,Y,Z)

:5RC        (6)         B(X,Y,Z)

:AXB      B(X,Y,Z)-> B(Z,Y,X)
X::X
Y::Y
Z::Z        (7)         B(X,Y,Z)-> B(Z,Y,X)

:7,6AA      (8)         B(Z,Y,X)

:5LC        (9)         NOT Y=Z

:9CE1       (10)        NOT Z=Y

:AXB      (((NOT Y=Z)& B(X,Y,Z))& B(Y,Z,W))-> B(X,Y,W)
W::X
X::W
Y::Z
Z::Y        (11)        (((NOT Z=Y)& B(W,Z,Y))& B(Z,Y,X))-> B(W,Z,X)

:10,4FC     (12)        (NOT Z=Y)& B(W,Z,Y)

:12,8FC     (13)        ((NOT Z=Y)& B(W,Z,Y))& B(Z,Y,X)
```

Fig. 6. Shortest student proof for formula (10), Exercise 4.
AXA: (((NOT X=Y)& B(Z,X,Y))& B(X,Y,W))->B(Z,X,W)
AXB: B(X,Y,Z)-> B(Z,Y,X).

```
:11.13AA   (14)       B(W,Z,X)

:AXB      B(X,Y,Z)-> B(Z,Y,X)
X::W
Y::Z
Z::X       (15)       B(W,Z,X)-> B(X,Z,W)

:15.14AA   (16)       B(X,Z,W)

:1.16CP    (17)   (((NOT Y=Z)& B(X,Y,Z))& B(Y,Z,W))-> B(X,Z,W)

CORRECT...
```

Fig. 6, continued.

derivations from given axioms or premises.

The reaction of the 12-year-old students was rather different. They had had extensive training in proving theorems from given axioms, and they felt uneasy at not knowing immediately where to begin, or if once they started, whether their initial subset of axioms was actually adequate for what they wanted to prove. The currect revision of the course consequently includes counterexamples and very elementary Finding-Axioms Exercises almost from the beginning.

3. Theorem Provers for Instructional Use

The reader might question the instructional worth of having a student prove complex theorems within the framework of a proof checker which, as so far illustrated, expects the user to construct rigorous proofs. As discussed in Section 2, the limitation of the program for teaching mathematics is just this requirement that the student construct an explicit formal proof for every theorem. The routine steps of more advanced mathematical work must be compressed and eliminated from the student's explicit focus of concern in order to provide adequate time to concentrate on the crucial conceptual steps in a given proof.

A significant contribution of the system to the development of more advanced mathematics courses in a CAI environment is the use of theorem provers for instructional purposes. With theorem provers, the student can instruct the program to move from one point to another in the proof without explicitly carrying out the mediating steps. The intention is that the skipped steps be modest and of the right level of difficulty for mechanical theorem provers of a noninteractive nature. An example is repeated use of the commutative and associative laws in a fashion that is common in elementary algebraic arguments.

Using the instructional system as a research tool, we have been able to study possible roles mechanical theorem provers can play in the construction of mathematically valid proofs by the student. As modes of operation for the use of theorem provers, we introduced generalized interchange laws and instantiation rules for each axiom and proved theorem, as well as an instruction called SHOW. For example, whenever a new expression in the form of an identity or biconditional is established as a true statement of a given theory, and a name is assigned to it, the program is capable of performing both the substitution and replacement rules in one step.

The automatic generalization rule for formulas in the form of a conditional represents a definite savings in the number of steps required to complete a proof, as illustrated in Figure 7 by the derivation of line (7) from

(4) and Theorem A, line (8) from (5) and Theorem A, and line (9) from (6), (7), (8), and Axiom TRA. The proof depends directly on one previous theorem and one axiom, which are shown at the bottom of the figure and which are part of a system of constructive plane geometry. The user simply references the name of the formula and an ordered list of proof lines whose conjunction is a substitution instance of the antecedent of the formula. The program can then generate the corresponding instance of the consequent. If substitutable variables occurring in the consequent do not occur in the antecedent, the program will ask the student to complete the desired substitution. The ability to derive new inference rules extends this instantiation rule in enabling the student to detach from a theorem a formula that would otherwise only be obtainable by iterative application of <u>modus ponens</u> on an instance of that theorem.

Insert Figure 7 about here

Our efforts to interface the instructional system with the theorem-proving program of Allen and Luckham (1970) have been moderately successful. Basically, our idea is to let the student type a line into the proof, thereby claiming that it is a valid inference from the work

```
PROVE       (((NOT Y=Z)& B(X,Y,Z))& B(Y,Z,W))-> B(X,Z,W)
:WP         (1)       ((NOT Y=Z)& B(X,Y,Z))& B(Y,Z,W)
:1LC        (2)       (NOT Y=Z)& B(X,Y,Z)
:2LC        (3)       NOT Y=Z
:2RC        (4)       B(X,Y,Z)
:1RC        (5)       B(Y,Z,W)
:3CE1       (6)       NOT Z=Y
:4THA       (7)       B(Z,Y,X)
:5THA       (8)       B(W,Z,Y)
:6.8.7TRA
            (9)       B(W,Z,X)
:9THA       (10)      B(X,Z,W)
:1.10CP     (11)  (((NOT Y=Z)& B(X,Y,Z))& B(Y,Z,W))-> B(X,Z,W)
CORRECT...
```

Fig. 7. A proof using general substitution rules.
Theorem A: B(X,Y,Z)->(B(Z,Y,X)
Axiom TRA: (((NOT X=Y)& B(W,X,Y))& (B(X,Y,Z))->B(W,X,Z)

he has already done. He then calls on the theorem prover to verify his claim. The student also must indicate which lines already in the proof and what instances of axioms and theorems he thinks should enter into the theorem prover's computations. Formula (10) from the fourth Finding-Axioms Exercise is presented as an example of using the SHOW rule (Figure 8). The SHOW rule, together with those discussed above, were used to eliminate ten lines from the minimum student proof shown in Figure 6.

Insert Figure 8 about here

A second and closely related activity in which theorem provers are useful is that of monitoring the student's activity while he is in the process of searching for a proof and then giving him hints of how he may complete the proof he has begun. At least in elementary domains of mathematics this role of a theorem prover has already been implemented as an instruction called HELP. A heuristically based theorem prover was designed to perform the work the student is expected to do, i.e., it constructs proofs in the elementary theory of Abelian groups. By taking the steps of the student's partial or erroneous work into account when searching for a solution, the theorem prover can compute various ways to complete the student's

```
PROVE      (((NOT Y=Z)& B(X,Y,Z))& B(Y,Z,W))-> B(X,Z,W)

:WP        (1)        ((NOT Y=Z)& B(X,Y,Z))& B(Y,Z,W)

:1RC       (2)        B(Y,Z,W)

:2AXB      (3)        B(W,Z,Y)

:SHOW      (4)        ((NOT Z=Y)& B(W,Z,Y))& B(Z,Y,X)
FROM LINES OF THE DERIVATION?
::1,3
FROM AXIOMS OR THEOREMS?
::AXB    B(X,Y,Z)-> B(Z,Y,X)
X::X
Y::Y
Z::Z
::
OK?  Y
LINE 4 IS OK

:4AXA      (5)        B(W,Z,X)

:5AXB      (6)        B(X,Z,W)

:1,6CP     (7)   (((NOT Y=Z)& B(X,Y,Z))& B(Y,Z,W))-> B(X,Z,W)

CORRECT...
```

Fig. 8. Proof of Formula (10), Exercise 4, using the SHOW instruction.
AXA: (((NOT X=Y)& B(Z,X,Y))& B(X,Y,W))-> B(Z,X,W)
AXB: B(X,Y,Z)-> B(Z,Y,X)

task. From this information, the instructional system can generate a tutorial dialogue aimed at helping the student construct a successful proof. The details of using a theorem prover as a proof analyzer to help a student continue his work is dealt with elsewhere (Goldberg, forthcoming).

List of References

Allen, J., & Luckham, D. An interactive theorem-proving program. In B. Meltzer & D. Michie (Eds.), Machine Intelligence 5. New York: American Elsevier, 1970.

Goldberg, A. A generalized instructional system for teaching elementary mathematical logic. Technical Report No. 179, October 11, 1971, Stanford University, Institute for Mathematical Studies in the Social Sciences.

Goldberg, A. Computer-assisted instruction: The application of theorem-proving to adaptive response analysis. Doctoral dissertation, University of Chicago, Chicago, Illinois, forthcoming.

Suppes, P. Computer-assisted instruction at Stanford. Technical Report No. 174, May 19, 1971, Stanford University, Institute for Mathematical Studies in the Social Sciences.

Suppes, P., & Binford, F. Experimental teaching of mathematical logic in the elementary school. The Arithmetic Teacher, 1965, 12, 187-195.

Suppes, P., & Ihrke, C. Accelerated program in elementary-school mathematics--the fourth year. Psychology in the Schools, 1970, 7, 111-126.