UC Office of the President
Stanford Technical Reports

Title

Computer-assisted Instruction at Stanford

Permalink

https://escholarship.org/uc/item/2t18k22q

Author

Suppes, Patrick

Publication Date

1971-05-19

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

COMPUTER-ASSISTED INSTRUCTION AT STANFORD

by

Patrick Suppes

TECHNICAL REPORT NO. 174

May 19, 1971

PSYCHOLOGY AND EDUCATION SERIES

Reproduction in Whole or in Part is Permitted for

any Purpose of the United States Government

INSTITUTE FOR MATHEMATICAL STUDIES IN THE SOCIAL SCIENCES

STANFORD UNIVERSITY

STANFORD, CALIFORNIA

Computer-assisted Instruction at Stanford*

Patrick Suppes

Stanford University
Stanford, California 94305

In most areas of technology perhaps the best way to forecast the operational applications in the coming decade is to look at the research and development efforts in the preceding decade. I am not certain how true this generalization is, but it is the approach I would like to use in considering the prospects for computer-assisted instruction during the seventies. I shall spend most of my time in discussing the history of our efforts at Stanford in this area, and only at the end of my lecture, will I attempt some specific forecasts of the future.

History of Computer-assisted Instruction at Stanford

In January 1963, the Institute for Mathematical Studies in the Social Sciences at Stanford University began a program of research and development in computer-assisted instruction. The Institute's program in computer-assisted instruction is under my direction and that of Richard C. Atkinson. We are both members of the Stanford faculty. In its initial inception, John McCarthy of the Department of Computer Science at Stanford played an important role in the design and operation of the Institute's computer facilities. The various research projects have been supported by the National Science Foundation, the United States Office of Education and the Carnegie Corporation of New York.

The initial instructional system in the Institute consisted of a medium-sized computer (a PDP-1) and six student stations placed within 100 feet of the computer. Each student booth contained two visual-display devices. The first was a random-access optical-display device developed for the laboratory by IBM Corporation that presented microfilmed source material on a 10-inch by 13-inch ground-glass screen. It was possible to encode the equivalent of a 512-page book (8-1/2 inch by 11-inch standard page) on microfilm and any page, or one-eighth of a page, could be displayed randomly within 1 second. The student responded to the display by using a light pen on the face of the screen itself. As the pen was touched to the screen, the coordinates of that position were sent to the computer for comparison with any predesignated areas of the screen. The accuracy of the light pen permitted identification of a 1/4-inch square on the screen. This device, which was the predecessor of the IBM-1500 system mentioned below, has been phased out and is no longer in the Institute.

The second display device, which is still in use, was developed for the Institute by the Philco-Ford Corporation. It is a cathode-ray tube, commonly called a "scope." It can display points of light in an area 10 inches high by 10 inches wide with 1,024 possible positions on both the horizontal and vertical

*Work on this paper has been supported by two NSF Grants to Stanford, G-18709 and GJ-443X. It was given orally at a conference on L'Homme et l'Informatique sponsored by the Institut de la Vie in Bordeaux, France in June 1970.

axes. In addition to individual points, 120 prearranged characters may be displayed in five different sizes. It is also possible to display vectors by simply identifying the end points. A typewriter keyboard is attached to the scope and may be used to send information from the student to the computer.

Until June 1968, the central computer was a PDP-1 designed by Digital Equipment Corporation. It has a 32,000 word core and a 4,000 word core, which can be interchanged within the 32 bands of a magnetic drum on files stored on two IBM-1301 disks. The two IBM-1301 disks were replaced by two IBM-2314 disks in the fall of 1968, and a PDP-10 became the main computer, although the PDP-1 is still in operation. The computer configuration as of June 1969 is shown in Figure 1.

1963-64

The first operational instruction program available in any form at all was a program in elementary mathematical logic. This program was first demonstrated on December 12, 1963, and two lessons consisting of 23 problems were run with four sixth-grade students on December 20, 1963. An additional two fifth graders were run for demonstration purposes on January 7, 1964. An occasional demonstration was given every month or so during the spring. More importantly, some 20 lessons giving a fairly detailed introduction to sentential logic were written and programmed during the spring. In the summer of 1964, these lessons, which were presented on the scopes, were run with two fifth-grade boys. One boy had 32 sessions for a total of more than 15 hours at the terminal, and the second boy had 38 sessions for a total of more than 36 hours at the terminal.

Because the logic program is the oldest and in many ways the most sophisticated of our CAI programs, a brief description of its curriculum content from year to year is included, beginning with 1964-65.

During the spring of 1964, preliminary experiments using first-grade mathematics material were also conducted in the Institute with 29 kindergarten children. Throughout 1964, staff members worked to write and code the computer CAI programs for first-grade and fourth-grade mathematics and for mathematical logic.

1964-65

During the 1964-65 school year, two groups of six first-grade children were given a preliminary version of the first-grade arithmetic program during the regular school year (September 14 to June 11, 1965). Two kindergarten children were given a revision of the first-grade program in the spring (March 15 to June 25, 1965).

By remote control, 41 fourth-grade children were given daily arithmetic drill-and-practice lessons on a teletype machine in their classroom at Grant School in the Cupertino Union School District (April 19 to June 4, 1965). This installation constituted an important first step in moving terminals from the Stanford campus to elementary schools, with direct connection from the computer to the terminals by telephone lines.

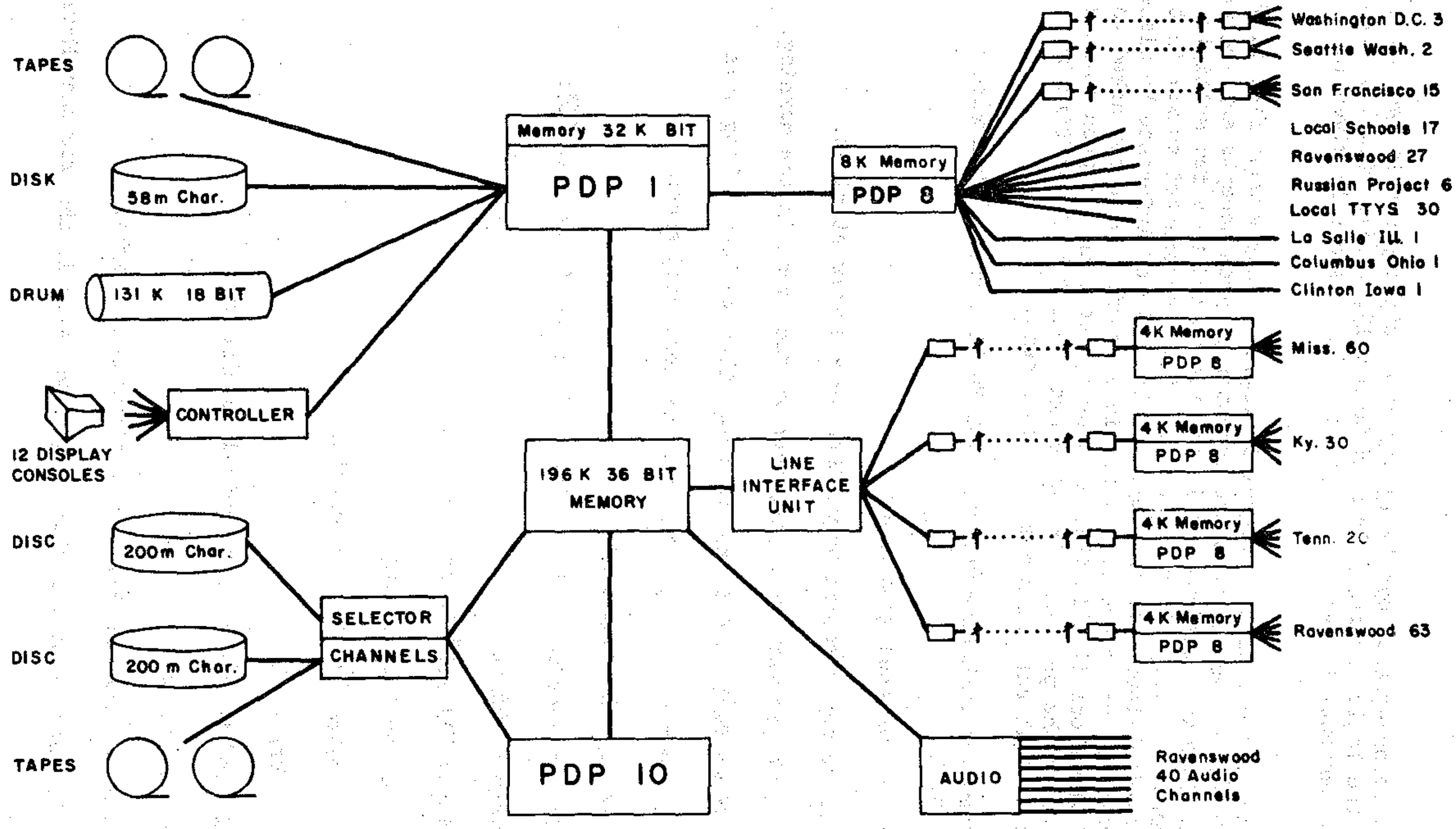

FIGURE 1. THE STANFORD CAI SYSTEM CONFIGURATION AS OF JUNE, 1969.

Two very capable second-grade boys (slightly under eight years in age) worked through a portion of the logic program (March 15 to May 19, 1965). The two students covered 29 lessons in sentential logic. A listing of the lessons is given in Table 1.

The most important feature of the program in logic already exemplified in the work beginning at the end of 1963 is that the computer program accepts any logically valid response of the student. The student is not restricted to a few multiple-choice answers, or more generally, there is not a unique constructed answer that must be given. The student inputs on the keyboard the rule of inference he wishes to apply to given premises, or to previous lines in a proof. He is not asked to type out the line of the proof itself; this is done by the computer upon command. Here are some examples of the program. In these examples, Rule AA--affirm the antecedent--is the classical rule of modus ponendo ponens.

The first two examples emphasize working with English rather than with mathematical sentences.

<u>Example</u> 1. Derive: We need good shoes.

Premise 1. If we buy sleeping bags, then we are warm at night.

Premise 2. If we are warm at night, then we feel good in the morning.

Premise 3. If we feel good in the morning, then we take a long walk.

Premise 4. If we take a long walk, then we need good shoes.

Premise 5. We buy sleeping bags.

In Example 1, the student would input "AA 1.5" to obtain as line (6):

6. We are warm at night.

He would next input "AA 2.6" to obtain:

7. We feel good in the morning.

After this would follow "AA 3.7" to obtain:

8. We take a long walk.

and finally "AA 4.8" to obtain the derived conclusion:

9. We need good shoes.

<u>Example</u> 2. Derive: Jack and Bill are not the same height.

Premise 1. If Jack is taller than Bob, then Sally is shorter than Mavis.

TABLE 1

List of 29 Logic Lessons, Spring of 1965

1. Rule AA (ponendo ponens). One-step proofs only.
2. Rule AA with two-step proofs.
3. Denials and Rule DC (tollendo tollens).
4. Rules AA and DC together in multi-step proofs.
5. Dominance and use of parentheses; law of double negation.
6. More on law of double negation.
7. Rule DD (tollendo ponens).
8. Truth and validity.
9. Truth and validity in relation to the law of double negation.
10. Analysis of inclusive "or" and validity of Rule DD.
11. Truth diagrams (analysis of the truth of compound sentences given the truth of the atomic sentences).
12. Truth-functional analysis of conjunction.
13. Rules of conjunction and simplification for inferences about conjunctions.
14. Truth-functional analysis of conditionals.
15. Truth tables.
16. Tautologies.
17. Relation between conditionals and logical arguments.
18. Valid arguments and tautologies.
19. P the denial of not P.
20. DeMorgan's laws.
21. Using DeMorgan's laws in derivations.
22. Hypothetical syllogisms.
23. More on hypothetical syllogisms.
24. Commutative laws for conjunctions and disjunctions.
25. Rule of addition (from P infer P or Q).
26. More on the rule of addition.
27. Disjunctive syllogisms.
28. More on disjunctive syllogisms.
29. Validity of disjunctive syllogisms.

Premise 2. Sally is not shorter than Mavis.

Premise 3. If Jack and Bill are the same height, then Jack is taller than Bob.

In this example, the student must use modus tollendo tollens, which we call Rule DC. 'DC' stands for the fact that we deny the consequent of the conditional premise. Thus in Example 2, the student who is responding correctly would input first "DC 1.2" to obtain:

4. Jack is not taller than Bob.

and then "DC 3.4" to obtain the derived conclusion:

5. Jack and Bill are not the same height.

Example 3. Derive: $y + 8 < 12$

Premise 1. $x + 8 = 12$ or $x \neq 4$

Premise 2. $x = 4$ and $y < x$

Premise 3. If $x + 8 = 12$ and $y < x$ then $y + 8 < 12$.

In this example, the student must use modus tollendo ponens, which we call Rule DD--deny a disjunct, as well as two rules dealing with conjunctions--the rule of conjunction (FC) for putting two sentences together to form a conjunction, and the rule of simplification for deriving one member of a conjunction, Rule LC to derive the left conjunct and Rule RC to derive the right conjunct. We show the steps of the derivation in one block, but it is to be emphasized that the student inputs only the rule abbreviations and the numbers at the left of each line.

LC 2	4. $x = 4$
DC 1.4	5. $x + 8 = 12$
RC 2	6. $y < x$
A 5.6	7. $x + 8 = 12$ and $y < x$
AA 3.7	8. $y + 8 < 12$

In these simple examples the possibilities for different proofs by different students are restricted, but already in this last example, the order of the lines can be changed, and the possibilities of variation increase rapidly as the complexity of the problems increases.

It should be mentioned that when the student makes an error, which means he attempts to take a logically invalid step, the computer program prints out the reason the step is in error and waits for him to make another move. For example, if the student attempts to apply Rule AA to a sentence in which the major connective is "and" rather than "if...then" the computer program simply prints out the message "line n is not a conditional sentence." The ability to analyze mistakes unerringly is an unusual feature of the logic program and rests upon the well-understood character of logical inference. In more diverse and open-ended subjects, the same unerring analysis of student errors is considerably more difficult.

In the spring of 1965, the second version of the logic program was prepared. This program was designed for an experimental class of 26 second graders run in the summer of 1965. Two variants of the program were written. One utilized English sentences throughout, while the other introduced logical symbolism. In both cases each new topic was introduced intuitively in English sentences. The purpose of writing two separate tracks was to determine whether the use of English sentences or abbreviated logical symbolism was easier for the students. In all, 20 lessons were prepared in both the English and symbolic tracks for the summer session. In addition, remedial lessons were prepared for each of the above lessons. The number of responses per lesson ranged from 10 for the introductory multiple-choice questions to 60 (including line numbers) for later lessons.

1965-66

During the 1965-66 school year, drill-and-practice teletype programs were conducted in three schools. In September, the arithmetic drill program at Grant School was expanded, with two teletypes for each of Grades 4, 5 and 6. On February 2, 1966, two more teletypes were added for third-grade classes. By the end of the year, of the 270 participating students, 62 were third graders, 76 were fourth graders, 70 were fifth graders and 62 were sixth graders. A detailed description of this first year of relatively large-scale operation of the drill-and-practice teletype program is to be found in Suppes, Jerman and Brian (1968).

On March 1, 1966, one teletype was installed at Ravenswood High School in the Sequoia Union High School District. The machine was used by seven arithmetic classes. About 60 students used the machine on alternate days.

During the spring of 1966, four teletypes were used for drill-and-practice work in spelling at Costano School in Ravenswood City School District. Audio was provided from the Institute's central computer facilities by a second telephone line and earphones. For an account of the research in spelling, see Knutson (1967) and Fishman, Keller and Atkinson (1968).

During 1965-66, work on tutorial programs was continued in both the mathematics and the mathematical logic programs. Two groups of four kindergarten children were given a revised version of the first-grade program during the regular school year (April 11, 1966 to June 10, 1966). Two groups of first-grade children were given a revised version of the second-grade program (April 11, 1966 to June 10, 1966). Drill-and-practice lessons in symbolic logic were given to 30 sixth-grade students one day each week from May 5, 1966 through June 9, 1966. A group of 7 fourth-grade students were given the same lessons in logic for a period of four weeks, one day each week (May 19, 1966 to June 9, 1966).

1966-67

In the summer of 1964, the Institute was granted a contract by the United States Office of Education to investigate the feasibility of teaching mathematics and reading as an integral part of an elementary-school program by using individualized, tutorial computer-assisted instruction over an extended period of time. The site chosen was the Brentwood Elementary School (Ravenswood City School District) in East Palo Alto, California. The Laboratory was housed in a specially built unit and was equipped with an IBM-1500 Instructional System operated by an IBM-1800 computer. (Use of this equipment was terminated on July 1, 1968). The first students were run on the system on October 27, 1966. For the 1966-67 school year, over 100 children, including all the first-grade students at Brentwood, participated in the project. Half of the students had daily computer-assisted instruction in mathematics, and the other half had daily sessions in reading. For a description of the reading program, see Atkinson (1967), Atkinson & Hansen (1966), and Wilson & Atkinson (1967).

In addition, the drill-and-practice program was expanded during 1966-67. Summary statistics are given in Table 2. In March 1967, two teletypes were put in operation at the Morehead State University Laboratory School in Morehead, Kentucky, more than 2,000 miles from Stanford. As in the case of other schools, the connection to the Institute's computer was by ordinary telephone line. Teletypes were added at other schools, mostly in California, so that the starting number of 877 students increased to slightly over 1,500 students at the end of the school year.

In addition, 31 students (average age about 10 years) at Walter Hays Elementary School in Palo Alto, California participated in a teletype program on symbolic logic and modern algebra. Lessons were prepared for two courses of study, sentential logic and elementary algebra. Both courses used the same logic program, but had separate introductory tracks for rule names and applications. For most of the year the sentential logic stressed derivations using symbols, and the algebra emphasized numerical derivations; however, rules from both were required for some proofs near the end of the year. Each child alternated his course of study from one day to the next; logic one day, algebra the next.

The logic program was intended to be self-contained as tutorial computer-assisted instruction at a teletype terminal, but students were able to question a staff member who was available in the teletype room when the logic program was running. Although a considerable amount of individual instruction was given to some students while they were working at the terminals, very little group instruction occurred.

TABLE 2

Stanford Programs in Computer-assisted Instruction

Program	66-67	67-68	68-69	69-70
Drill-and-practice mathematics				
Grades 1-8 (block structure)				
California	1,500	1,441	2,475	122
Iowa	-	640	-	-
Kentucky	-	1,632	1,060	-
Mississippi	-	640	2,113	-
Ohio	-	-	101	-
Washington	-	-	92	139
College level				
Tennessee (algebra)	-	-	206	183
Tutorial primary-grade mathematics	53	73	-	-
Tutorial reading, Grade 1	50	88	-	-
Drill-and-practice in initial reading				
Grades 1-3, Remedial 4-6				
California	-	-	442	642
Language Arts	-	-	-	30
Drill-and-practice mathematics				
Grades 1-6 (strands structure)				
California	-	-	-	1,713
Ohio	-	-	-	165
Washington, D. C.	-	-	-	39
Tutorial computer programming	-	-	115	177
Tutorial logic and algebra Grades 4-8	76	195	49	459
Tutorial problem-solving Grades 5,6	-	27	20	18
First- and second-year Russian	10	30	52	77

The format used for the logic problems was similar to that used in earlier years.

Lesson 1 of the sentential logic contained 19 problems that were written in symbolic format with two or three premises and that required one-step proofs applying modus ponendo ponens, a rule of inference familiar to all the students. As already indicated, the rule was abbreviated AA for <u>affirm</u> <u>the</u> <u>antecedent</u>. The students needed to know that 'R → S' meant 'if R then S', that 'R → S' was a conditional sentence whose antecedent was R and consequent was S, that 'P' was the abbreviation for 'premise', and that the use of AA required two line numbers with the line number of the conditional sentence followed by the line number of its antecedent. A period separated the two line numbers. After the teletype had printed what the student was to derive and the given premises, the typewheel positioned itself for the student's instructions. The student then typed the abbreviation for the rule and the line numbers required for its application. The next information printed by the teletype was either a valid step based on the student's input or an error message if the student had given instructions for an invalid step. The teletype proceeded to the next problem when the student had completed the desired derivation. An example from Lesson 1 is the following:

Derive:	L	
P	(1)	K → L
P	(2)	M
P	(3)	K
<u>AA1.3</u>	(4)	L.

The underlined phrase indicates what the student typed for this problem. The remainder of the typing was performed automatically under computer control.

Lesson 2 contained 8 more problems that had either two or three premises and that required only a one-step proof. Mathematical sentences were included, as well as the usual symbols of sentential logic. Each of the 7 problems in Lesson 3 had three premises and used Rule AA. Two-step problems were presented for the first time in this lesson.

The Rule of Conjunction was introduced in Lesson 4 as the rule that would Form a Conjunction (FC). The 17 problems in this lesson involved one-step, two-step, and three-step derivations using modus ponendo ponens and the Rule of Conjunction.

In Lesson 5, the Rule of Simplification was presented as two separate commands for the student to give the computer: to derive the Left Conjunct he typed LC, or to derive the Right Conjunct he typed RC with a designated line number to complete the instruction. For example:

Derive: R

P	(1)	$S \rightarrow R \& Q$
P	(2)	S
<u>AA1.2</u>	(3)	R & Q
<u>LC3</u>	(4)	R.

The underlined sections of the problem indicate the student's input for the derivation. There were 21 problems in Lesson 5 that involved one-step, two-step, and three-step derivations that used from one to five premises.

In Lesson 6 there were 20 problems that contained two, three, or four premises using all the rules introduced up to that place in the curriculum. The problems required from one-step to four-step derivations. Another new rule, modus tollendo ponens, was introduced as the rule that would Deny a Disjunct (DD). For example:

Derive: D

P	(1)	A v (B & C)
P	(2)	D v -B
P	(3)	-A
<u>DD1.3</u>	(4)	B & C
<u>LC4</u>	(5)	B
<u>DD2.5</u>	(6)	D.

As before, the underlined sections indicate the student's typed work, and the teletype printed the remainder of the problem.

Lesson 7 contained 21 problems that required from one to four lines to solve problems based on two or three premises. Another new rule, modus tollendo tollens, was introduced as Deny the Consequent (DC). The underlined statements represent the student's work in the following problem:

Derive: R

P	(1)	N
P	(2)	$-R \rightarrow -S$
P	(3)	$N \rightarrow S$
<u>AA3.1</u>	(4)	S
<u>DC2.4</u>	(5)	R.

At approximately this stage in the curriculum (depending on each student's individual rate of progress), a multiple-choice mode was available for use at the teletype terminals. Two inserted lessons used this multiple-choice mode for review and practice on logical vocabulary. One new rule, Double Negation (DN), was introduced by using the multiple-choice mode for direct instruction. The first inserted lesson contained 20 problems and the second lesson 19 problems.

Lesson 8 contained 18 problems having from one to three premises and required one-step through four-step derivations to derive the conclusions. Practice in applying the Double Negation Rule was emphasized. For example:

Derive:	B	
P	(1)	$--(A \rightarrow B)$
P	(2)	A
<u>DN1</u>	(3)	$A \rightarrow B$
<u>AA3.2</u>	(4)	B.

The problems in Lesson 9 featured another new rule, Hypothetical Syllogism (HS). There were 21 problems in this lesson that required from one-step through five-step derivations. From one to three premises were provided for each problem. One problem required the use of an algebraic rule in its derivation. The rule of the Hypothetical Syllogism was applied in the following typical problem:

Derive:	$A \rightarrow D$	
P	(1)	$A \rightarrow B$
P	(2)	$B \rightarrow C$
P	(3)	$C \rightarrow D$
<u>HS1.2</u>	(4)	$A \rightarrow C$
<u>HS4.3</u>	(5)	$A \rightarrow D$.

Lesson 10 contained 27 problems with from one to five premises that required from one-step to twelve-step proofs for solution. Many applications of the algebra rules were necessary for the problems in this lesson. Also, the Law of Addition, Form a Disjunction (FD), was presented. This rule permitted the student to type the second part of a disjunction formula. The underlined sections indicate work typed by the student. For example:

Derive:	-S	
P	(1)	$S \rightarrow -(R \vee T)$
P	(2)	R
<u>FD2</u>	(3)	R v (<u>T</u>)
<u>DC1.3</u>	(4)	-S.

In Lesson 11 some of the 17 problems required derivations and some of the problems were presented in the multiple-choice mode. Those problems of the multiple-choice type reviewed the vocabulary and required the student to identify a certain type or part of a formula. The derivations contained from one to six premises with from two to twelve lines of rule applications for the solutions.

Lesson 12 combined both derivations and multiple-choice problems for the introduction of two new rules that applied the Commutative Laws. The first rule was called Commute Disjunction (CD), and the second rule was called Commute Conjunction (CC). There were 18 problems in this lesson; the nine derivations had either one or two premises and were one-step or two-steps in length. The Rule CD was applied as follows:

Derive: A v B

P (1) B

<u>FD1</u> (2) B v (<u>A</u>)

<u>CD2</u> (3) A v B.

Lesson 13 emphasized the combined use of algebra rules and logic rules. The 27 problems included both multiple-choice problems and derivations having one to three premises with as many as six lines of rule applications. The 16 problems in Lesson 14 followed the same format of combining multiple-choice problems with derivations that included the use of algebra rules.

The algebra curriculum was presented in much the same format to the students as the logic curriculum, with the exception that rules were introduced in a notebook written in a programmed format. This approach was initiated because there was no multiple-choice mode available when the algebra program started.

Directions written into the program instructed the student when to read the introduction and when to solve the problems for a new rule in his notebook. The student then used the answer section in his notebook to check his work. The first two pages of the notebook included the rule names for both the logic and algebra programs and examples of their application. Each student had his notebook at the teletype terminal available for reference each day.

Lesson 1 contained 10 problems in which the student practiced the Rule of Number Definition (ND). (Each positive integer greater than 1 is defined as its predecessor plus 1. Thus 2 = 1 + 1, 3 = 2 + 1, etc.) This rule was printed with a prefix that indicated which number the machine was to present and define. For example:

Derive: 6 = 5 + 1

<u>6ND</u> (1) 6 = 5 + 1.

The underlined section shows the student's command to the computer.

Lesson 2 presented 15 problems that required the student to apply the Rule of Number Definition and then the new rule, Definition (D), that allowed the definition of a particular number to be substituted for (the name of) the number in a given number sentence. A prefix number in front of the rule abbreviation indicated the number that was to be replaced by its definition, and a postfix number indicated which occurrence of the number in the given sentence was to be defined. For example:

Derive: 8 = ((5 + 1) + 1) +1

8ND (1) 8 = 7 + 1

7D1 (2) 8 = (6 + 1) + 1

6D1 (3) 8 = ((5 + 1) + 1) + 1.

Lesson 3 contained 20 problems using both the Rule of Number Definition and the Rule of Definition for two-step to four-step derivations. Lesson 4 provided further practice using the same rules for 15 problems that required three or four lines of proof.

In Lesson 5 a new rule, Commute Addition (CA), was introduced. To apply this rule to the previous line of the problem, a postfix number indicated which occurrence of the plus sign was used for the commutation. For example:

Derive: 7 = 1 + 6

7ND (1) 7 = 6 + 1

CA1 (2) 7 = 1 + 6.

For the 20 problems in this lesson, both the Rule of Number Definition and the Rule of Definition were used continuously.

Lesson 6 contained 23 problems that required as many as four steps of proof. The three rules available for algebra proofs were used. Lesson 7 provided further practice with the same rules. The 22 problems required as many as seven steps for a solution. Lesson 8 extended the use of the same three rules. The 13 problems needed as many as eight lines of proof for the derivation.

In Lesson 9 a new rule, Associate Addition to the Right (AR), was introduced. The student typed a postfix number to indicate which plus sign was to be dominant after applying Associate Addition to the Right. For example:

P (1) (4 + 3) + 1 = (4 + 3) + 1

AR2 (2) 4 + (3 + 1) = (4 + 3) + 1.

There were 20 problems in this lesson that needed as many as five steps of proof for solution.

Lesson 10 provided practice with all rules that had been presented. There were 21 problems that required as many as seven steps of proof. Lesson 11 contained 11 problems that provided further practice with the same rules.

Lesson 12 contained a new rule, Inverse Definition (ID). This rule put a number in place of its definition. A postfix number was required to indicate which occurrence of a number's definition was to be replaced by the number. For example:

P	(1)	$5 + 1 = 5 + 1$
<u>6ID2</u>	(2)	$5 + 1 = 6.$

The postfix 2 indicates that the second occurrence of the definition of 6 is to be replaced. There were 20 problems in this lesson and some required as many as seven steps of proof for a solution.

In Lesson 13 there were 17 problems that needed as many as six steps for a derivation. Lesson 14 introduced a new rule, Associate Addition to the Left (AL). This rule allowed the students to reassociate numbers to the left using the same format as Associate Addition to the Right. There were 17 problems in this lesson.

Thus, in these 17 algebra lessons a total of six algebraic rules of inference were introduced. The introduction of these rules gave the students experience with the sort of mathematical inferences that are widely used in elementary algebra and that are rather different from the rules of sentential inference. I emphasize again that the students were about 10 years of age.

1967-68

Seventy-three students continued in the 1967-68 tutorial mathematics program at the Brentwood Laboratory. A new mathematics curriculum was initiated for the second grade.

The drill-and-practice mathematics program expanded again during 1967-68. From the end of January, 1968, to the end of May, 1968, the enrollment jumped from 2,387 to 4,353 for 30 schools in California, Iowa, Kentucky, and Mississippi.

The 640 students enrolled in the drill-and-practice program in the Job Corps Center in Clinton, Iowa were high-school-age girls and older. These girls concurrently attempted to learn a trade and to earn a high school diploma. The majority of these students worked at the fourth-grade level.

In September 1967, 30 students at Stanford University enrolled in a course of computer-based elementary Russian for credit. Professor Joseph Van Campen of the Department of Slavic Languages at Stanford was responsible for the development of the computer-based Russian course. A control class received regular classroom instruction, attended a language laboratory, and submitted written homework assignments.

In a computer-based instruction class, regular classroom instruction was eliminated and work at Model-35 teletypes with Cyrillic keyboard and audiotapes with earphones was substituted. The students received instruction at the terminals for a period of 50 minutes per day, 5 days a week, throughout the entire academic year.

The statistical evaluation of the Russian course has been positive in two major aspects.

Of the 30 students who started the first-year computer-based course, 1 left during the first quarter, 3 left between the first and second quarters, 1 left during the second quarter, and 3 left between the second and third quarters. Two new students entered the computer-based section at the beginning of the second quarter. Of the 38 students enrolled for the autumn quarter in the regular Russian section, 10 left the course during the first quarter, 13 left between the first and second quarters, and 3 left between the second and third quarters. Four new students entered the regular section at the beginning of the third quarter, 1 of them having transferred from the computer-based class. Of the 30 students originally enrolled in the computer-based program, 22 (73 percent) finished all three quarters, whereas, of the 38 students in the regular class, only 12 (32 percent) finished the year's curriculum. This finding suggests that the computer-based course held the interest of the students much better than the regular course did. Probably because Russian is more difficult than French, Spanish, or German for American students, the dropout rate in Russian at Stanford and other universities is traditionally high.

Approximately 66 percent of the content of the final examinations for the autumn and winter quarters was identical for the computer-based and the regular Russian sections; the complete final examination for the spring quarter was identical for the two groups. The number of errors for each student, when the students are ranked according to their performance on the final examination, is shown in Figures 2, 3 and 4 for the fall, winter and spring quarters, respectively. Although the average number of errors was lower for the computer-based students in all three quarters--15.8 relative to 49.0 in the fall quarter, 21.8 relative to 25.8 in the winter quarter, and 53.0 relative to 71.1 in the spring quarter--the difference was statistically significant for the fall quarter (Mann-Whitney U test, $P < .001$) and the spring quarter ($P < .05$), but not for the winter quarter. Since the selection process resulting from the poorer students' leaving the regular course biases the results on the examination against the computer-based group, the superiority of the computer-based group on the spring examination is more impressive than the difference indicated by the average number of errors.

The logic and algebra tutorial program increased to 195 students in seven schools in California and Mississippi. This was the only program aimed mainly at very bright students and was offered as a supplement or enrichment to the regular mathematics program. An additional feature of the logic program was some work on problem solving aimed at obtaining a better understanding of the difficulties students encounter in solving word problems. This was done by giving the students a routine for computations to be performed by the computer program.

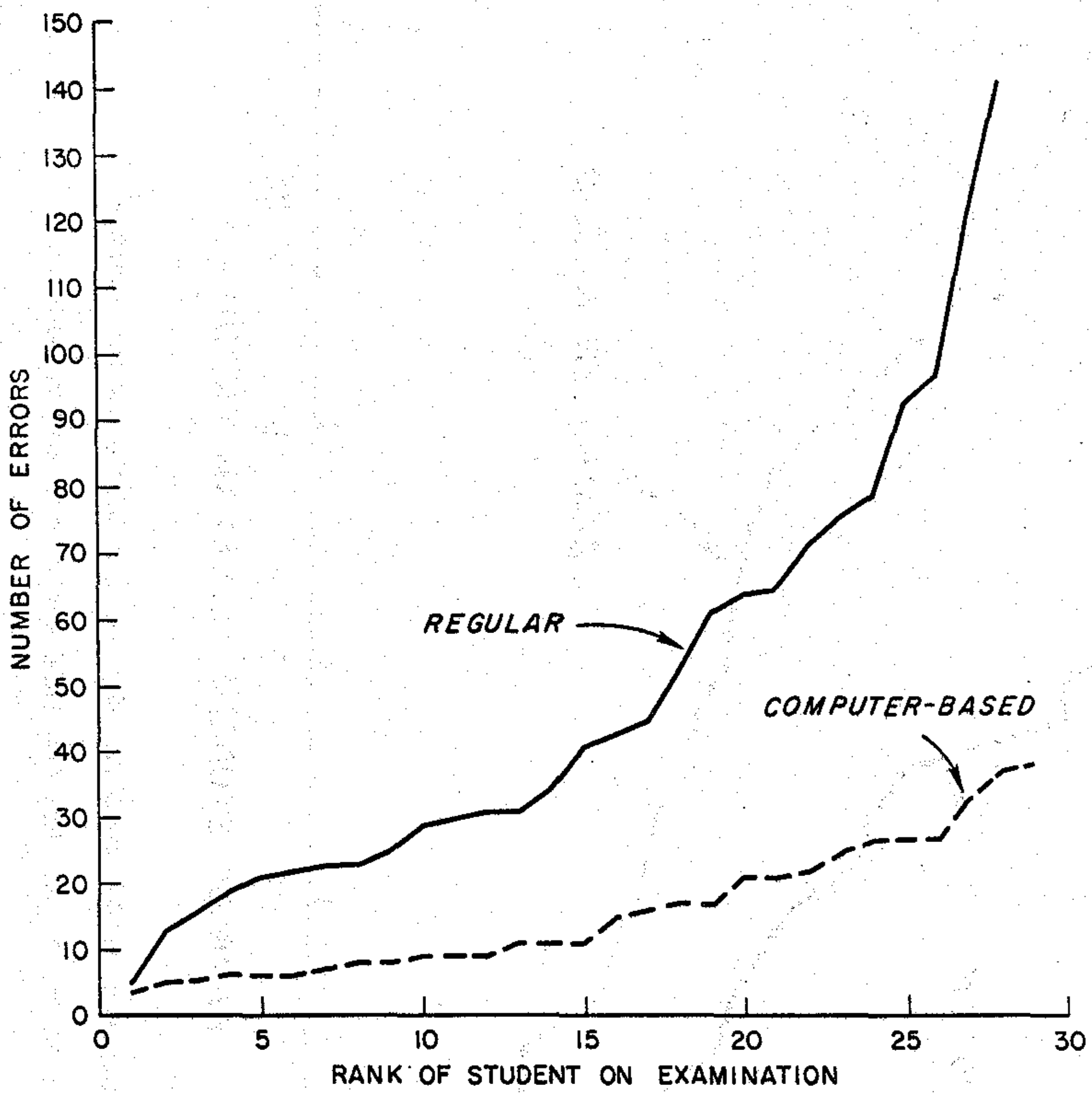

Figure 2. Student performance for the portion of the fall-quarter final examination in first-year Russian that was common to the computer-based and regular sections.

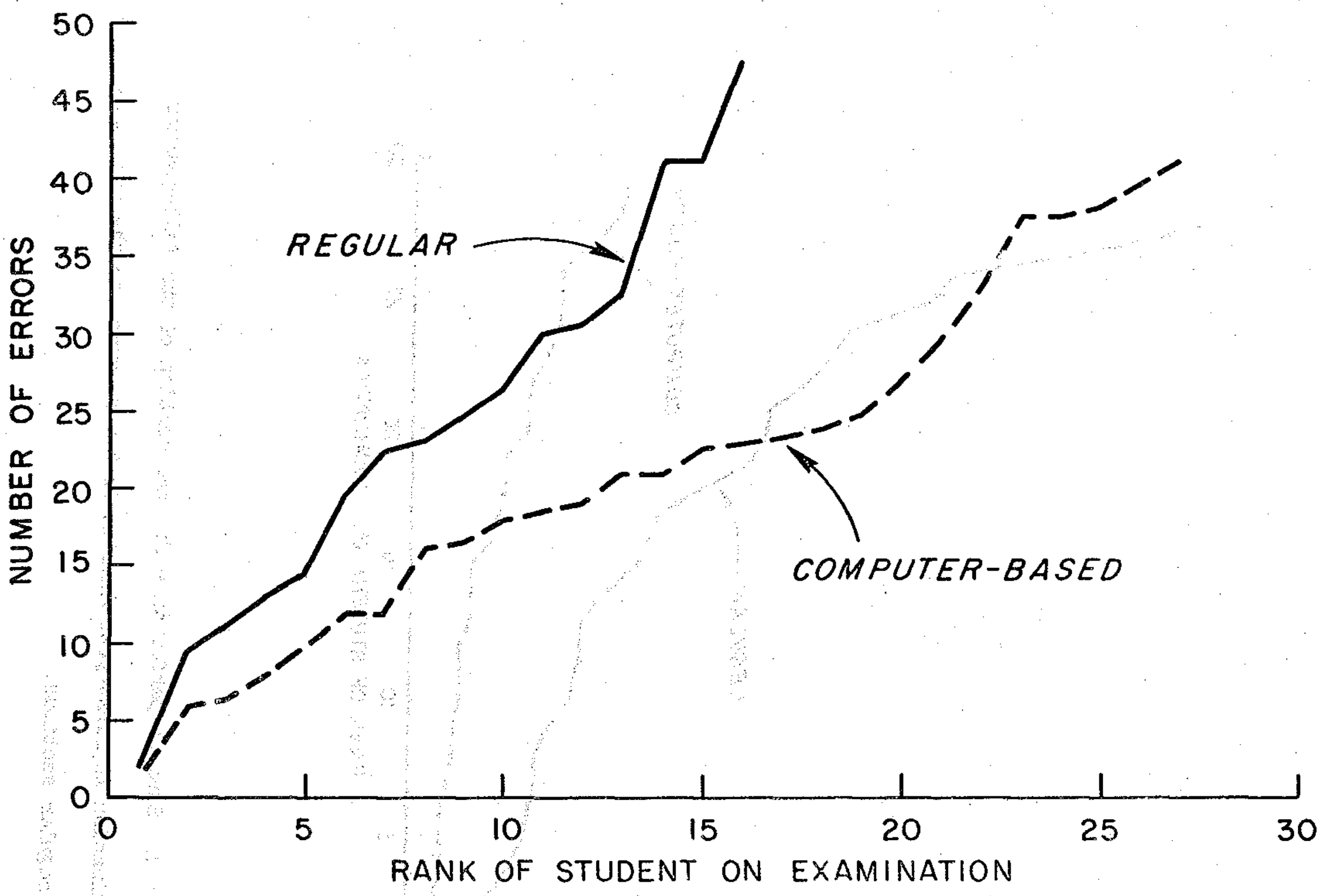

Figure 3. Student performance for the portion of the winter-quarter final examination in first-year Russian that was common to two sections.

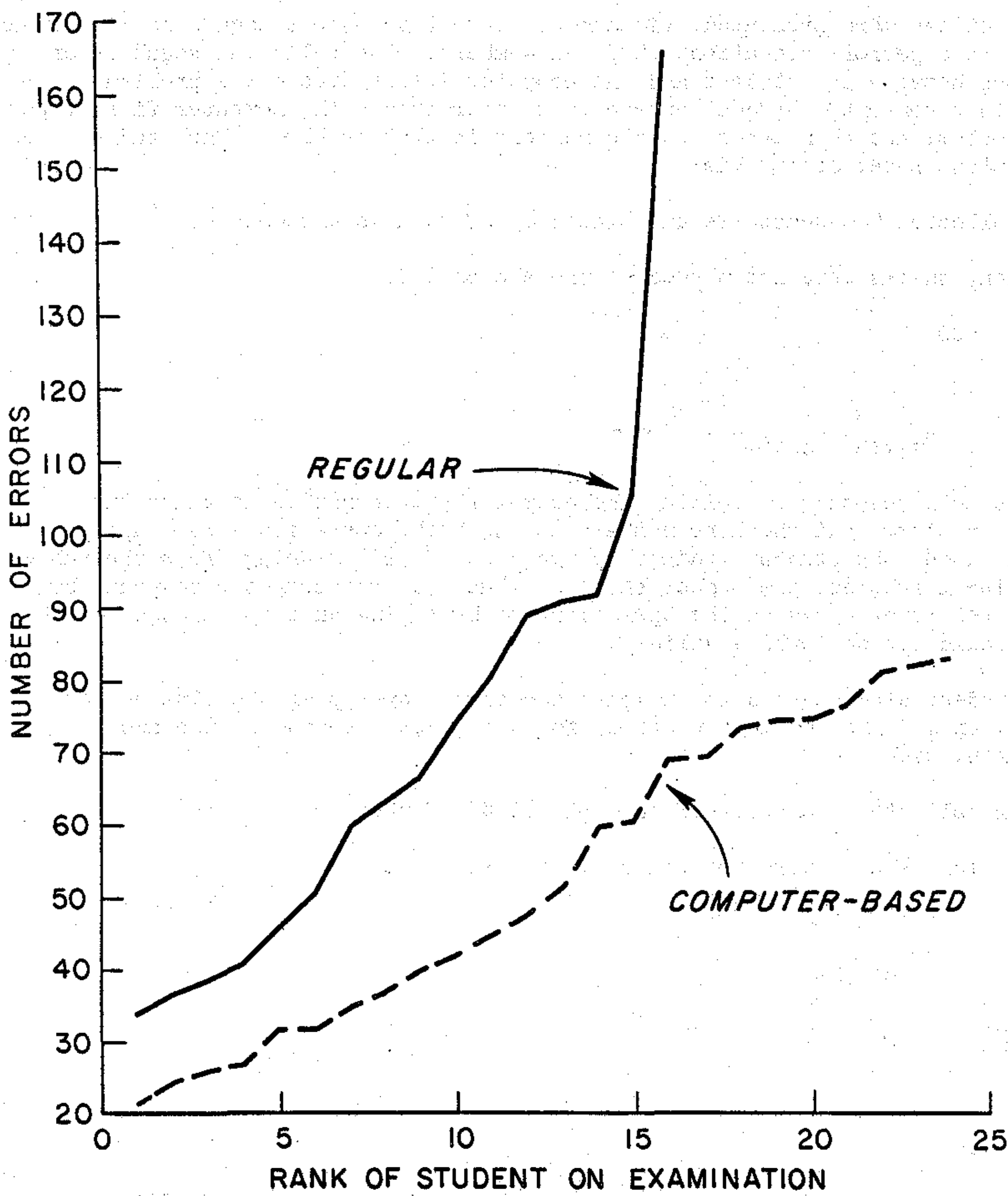

Figure 4. Student performance for the spring-quarter final examination in first-year Russian.

Instructions were presented, via computer, to teach the students how to command the computer to perform operations on given numbers. The following sequence of interactions between the student and the computer illustrates how a problem is solved in this context. Student entries are underlined. The computer first types out the problem, and then types out the numbers in that problem. The student sees on the printout sheet before him:

Tom collected 500 seashells and placed 43 of them in a showcase.

How many shells were not placed in the showcase...

G (1) 500

G (2) 43

"G" stands for "given number."

The student then responds by telling the computer the operation he wants the computer to perform, and the line numbers to which the operation should apply. In the present case, the student ordinarily types out "1.2S" meaning "from the number shown on line 1 subtract the number shown on line 2". The computer responds by typing the result of applying the operation, or by typing an error message if the operation could not be applied validly.

The student also learned to indicate the answer by typing the line number followed by an X. The complete protocol for a correct response in the above example, then, might be:

Tom collected 500 seashells and placed 43 of them in a showcase.

How many shells were not placed in the showcase...

G (1) 500

G (2) 43

<u>1.2S</u> (3) 457

<u>3X</u>

Correct

Similar notation was used for the other three rational operations of addition, multiplication and division. A detailed report of this study is to be found in Suppes, Loftus and Jerman (1969). A subsequent study is reported in Loftus (1970).

1968-69

The project in East Palo Alto schools shifted major emphasis from tutorial programs to drill-and-practice programs in elementary mathematics and reading. All eight elementary schools in the Ravenswood district were involved. Forty teletype terminals modified for audio were used in the reading program for Grades 1, 2, and 3, and 50 teletype terminals were used for arithmetic in Grades 1 to 6.

The total number of students enrolled in the drill-and-practice program in elementary arithmetic grew to over 6,000 during the year. Again, summary data are shown in Table 2.

Under the direction of Professor Atkinson, an initial reading program was designed and prepared that would complement any classroom reading series by providing drill and practice in the basic subskills for the complex task of learning to read. This program, pioneering the use of digitized audio, was made available to 442 first-, second- and third-grade students.

A remedial mathematics course for college students was prepared and run with students at Tennessee State University. The drill-and-practice program emphasized computational skills in arithmetic and elementary algebra. The program included sections on concept development as well. Students spent 20 minutes a day on terminals and the remainder in regular class sessions. The terminal installation consisted of 20 teletypes and a PDP-8 computer serving as a multiplexing device, connected by high-speed phone line to the Institute's central computer at Stanford.

The students who began work in logic and algebra on terminals in their school in 1966-67 continued during 1967-68 and 1968-69. By June of 1969 they were about twelve years in age and had proved most of the standard elementary theorems that hold for ordered fields. A list of the theorems used in the curriculum is given in Table 3. At this point their intensive work terminated because of their graduation from elementary school.

The second-year computer-based Russian course consisted of 113 lessons and was offered for credit at Stanford University through the Department of Slavic Languages and Literature. Disk-generated, computer-generated individualized review sessions and analyses of student performance were initiated and preliminary tests were made in the computer-based generation of sentences from individual vocabulary items.

For the autumn quarter, 1968, 19 students enrolled in the computer-based class and 11 enrolled in the conventional class; for the winter quarter, 1969, 18 students enrolled in the computer-based class and 8 enrolled in the conventional class; for the spring quarter, 1969, 15 enrolled in the computer-based class and 6 enrolled in the conventional class. The results paralleled those for the previous year.

The system was expanded during the year, as shown in Figure 1, into a network that included students in Kendall School for the Deaf in Washington, D. C. on the east coast and students in the Special Education Unit of the University of Washington in Seattle on the west coast.

TABLE 3

Logic and Algebra Course Outline

1968-1969

Second Year*

Recognition of true and false sentences; recognition of types of sentences; equations; inequalities.

Review of atomic and molecular sentences; conditionals; when a conditional is true.

AA: affirm the antecedent;
Truth value of conditionals as related to truth value of antecedent and consequent.

ND: number definition.

WP-CP: working premise and conditional proof.

Valid rule of inference.

CE: commute equals.

AE: add equals.

SE: subtract equals.

LT: rule of logical truth.

RE: replace equals (long form).

RE: replace equals (short form).

CA: commute addition axiom
$A + B = B + A$
short form of CA.

AS: associate-addition axiom
$(A + B) + C = A + (B + C)$;
AR: associate addition right;
AL: associate addition left.

Z: zero axiom
$A + 0 = A$.

N: negative number axiom
$A + -B = A - B$.

AI: additive inverse axiom
$A + -A = 0.$

Theorem 1: $0 + A = A$
Theorem 2: $(-A) + A = 0$
Theorem 3: $A - A = 0$
Theorem 4: $0 - A = -A$
Theorem 5: $0 = -0$

Theorem 6: $A - 0 = A.$
Theorem 7: $A + B = A + C \rightarrow B = C$
Theorem 8: $A + B = C \rightarrow A = C - B$
Theorem 9: $A = C - B \rightarrow A + B = C.$
Theorem 10: $A + B = 0 \rightarrow A = -B$

Theorem 11: $A = -B \rightarrow A + B = 0$
Theorem 12: $A + B = A \rightarrow B = 0.$
Theorem 13: $-(-A) = A.$
Theorem 14: $(-(A + B)) + B = -A$
Theorem 15: $-(A + B) = (-A) - B$

Theorem 16: $(-A) - B = (-B) - A$
Theorem 17: $-(A - B) = B - A$
Theorem 18: $(A - B) - C = A + ((-B) - C)$
Theorem 19: $(A - B) - C = A - (B + C)$
Theorem 20: $A + (B - A) = B$

Theorem 21: $A - (A + B) = -B$
Theorem 22: $(A - B) + (B - C) = A - C.$

CM: commute multiplication axiom
$A \times B = B \times A.$

MU: multiplication-by-unity axiom
$A \times 1 = A.$
Theorem 30: $1 \times A = A.$

ME: multiply equals.
DE: divide equals.

MI: multiplicative inverse axiom
$\neg A = 0 \rightarrow A \times (1/A) = 1;$
Theorem 31: $\neg A = 0 \rightarrow (1/A) \times A = 1.$

U: unity axiom
$\neg 1 = 0.$
IP: indirect proof.

FR: axiom for fractions
$\neg B = 0 \to A/B = A \times (1/B)$

Theorem 32: $1/1 = 1$
Theorem 33: $A/1 = A$

Theorem 34: $\neg B = 0 \ \& \ A \times (1/B) = C \to A = B \times C$
Theorem 35: $\neg A = 0 \ \& \ B = 1/A \to A \times B = 1$

Theorem 36: $B = 1 \ \& \ \neg A = 0 \to A \times B = A$

DL: distributive axiom;
Theorem 37: $A \times 0 = 0$.

MS: associate-multiplication axiom
$(A \times B) \times C = A \times (B \times C)$;
MR: associate multiplication right;
ML: associate multiplication left.

Theorem 38: $A \times B = 0 \ \& \ \neg A = 0 \to B = 0$
Theorem 39: $\neg A = 0 \to 0/A = 0$
Theorem 40: $\neg A = 0 \ \& \ A \times B = A \times C \to B = C$

Theorem 41: $\neg B = 0 \ \& \ A = B \times C \to A \times (1/B) = C$
Theorem 42: $\neg A = 0 \ \& \ A \times B = 1 \to B = 1/A$
Theorem 43: $\neg A = 0 \ \& \ A \times B = A \to B = 1$
Theorem 44: $(A + B) \times (C + D) = (A \times C + A \times D) + (B \times C + B \times D)$
Theorem 45: $A \times (-B) = -(A \times B)$

Theorem 46: $(-A) \times B = -(A \times B)$
Theorem 47: $(-A) \times B = A \times (-B)$
Theorem 48: $(-A) \times (-B) = A \times B$
Theorem 49: $A \times (B - C) = A \times B - A \times C$
Theorem 50: $-A = (-1) \times A$

Truth assignment mode

Counterexample mode

Axiom 13: $A < B \to \neg B < A$
Theorem 60: $\neg A < A$
Problems using counterexample mode

Theorem 61: $A = B \to \neg A < B \ \& \ \neg B < A$
Theorem 62: $A < B \to \neg A = B \ \& \ \neg B < A$

Axiom 14: $A < B \to A + C < B + C$
Theorem 63: $A < 0 \to 0 < -A$
Theorem 64: $0 < -A \to A < 0$
Theorem 65: $A + B < B + C \to B < C$

Theorem 66: $A < B \rightarrow -B < -A$
Theorem 67: $-B < -A \rightarrow A < B$
Theorem 68: $A + (-B) < A + (-C) \rightarrow C < B$
Theorem 69: $C < B \rightarrow A + (-B) < A + (-C)$
Theorem 70: $(A < 0 \,\&\, B < C) \rightarrow A \times C < A \times B$

Theorem 71: $(A < 0 \,\&\, A \times B < A \times C) \rightarrow C < B$
Theorem 72: $(0 < A \,\&\, A \times B < A \times C) \rightarrow B < C$
Theorem 73: $0 < 1$
Theorem 74: $A < 0 \rightarrow (1/A) < 0$

Axiom 15: $(A < B \,\&\, B < C) \rightarrow A < C$

Theorem 75: $(0 < A \,\&\, B < 0 \,\&\, C < 0) \rightarrow A \times B < B \times C$
Theorem 76: $(A < 0 \,\&\, 0 < B \,\&\, 0 < C) \rightarrow A \times B < B \times C$
Theorem 77: $0 < (A/B) \rightarrow 0 < A \times B$
Theorem 78: $0 < A \times B \rightarrow 0 < (A / B)$

Axiom 16: $\neg A = B \rightarrow A < B \vee B < A$
Theorem 79: $(\neg A = B \,\&\, \neg A < B) \rightarrow B < A$

Third Year*

Theorem 80: $Z + B = 0 \rightarrow Z = -B$
Theorem 81: $Z + B < 0 \rightarrow Z < -B$

Definition: $A > B \leftrightarrow B < A$
Definition: $A \geq B \leftrightarrow A > B \vee A = B$

Theorem 82: $Z + B > 0 \rightarrow Z > -B$
Theorem 83: $AX = 0 \,\&\, \neg A = 0 \rightarrow X = 0$
Theorem 84: $AX > 0 \,\&\, A > 0 \rightarrow X > 0$
Theorem 85: $AX > 0 \,\&\, A < 0 \rightarrow X < 0$

Theorem 86: $AX < 0 \,\&\, A > 0 \rightarrow X < 0$
Theorem 87: $AX < 0 \,\&\, A < 0 \rightarrow X > 0$
Theorem 88: $AX + B = 0 \,\&\, \neg A = 0 \rightarrow X = -(B/A)$
Theorem 89: $AX + B > 0 \,\&\, A > 0 \rightarrow X > -(B/A)$
Theorem 90: $AX + B > 0 \,\&\, A < 0 \rightarrow X < -(B/A)$

Theorem 91: $AX + B < 0 \,\&\, A > 0 \rightarrow X < -(B/A)$
Theorem 92: $AX + B < 0 \,\&\, A < 0 \rightarrow X > -(B/A)$
Theorem 93: $X + Y = A \rightarrow X = A - Y \,\&\, Y = A - X$
Theorem 94: $\neg A = 0 \,\&\, AX + BY = 0 \rightarrow X = -(B/A)Y$
Theorem 95: $\neg A = B \,\&\, X + Y = 0 \,\&\, AX + BY = C \rightarrow Y = C/(B-A)$
Theorem 96: $\neg A = B \,\&\, X + Y = 0 \,\&\, AX + BY = C \rightarrow X = C/(A-B)$

---

*Second Year and Third Year refer to the curriculum as laid out for a year's tutorial work and assume that the first year of curriculum has already been completed. The Third Year does not represent a full year, but rather only the first part.

A tutorial program in computer programming was initiated with high school students in Woodrow Wilson High School in San Francisco. Two languages, SIMPER and LOGO, were taught. The nonstandard languages were developed especially to provide an introduction to some of the basic ideas underlying computer programming. The terminal installation consisted of 15 teletypes.

1969-70

Fewer terminals were in the system than in the previous year, because two of the major centers (Mississippi and Kentucky) continued with systems of their own, and because there was a decrease in the level of federal support to other schools.

The block program for Tennessee State University was continued throughout the year. New course material was prepared. The content of the program included a review of arithmetic and intermediate algebra in a drill-and-practice mode.

The number of instructional programs increased as shown in Table 2. In the spring, the block version of the drill-and-practice mathematics program (for a detailed description of the block version, see Suppes, Jerman and Brian (1968) or Suppes and Morningstar (1969)) was replaced by a new program, which I now describe.

During the summer of 1968, development began on a major revision of the drill-and-practice program in arithmetic. The revised program evolved when attention was diverted from a program that could duplicate and expedite classroom procedures for a given grade to a program that could provide the most efficient drill for a given individual from the start of Grade 1 through the end of Grade 6. The question used to determine what types of problems a child should receive on a drill changed from "What grade is the child in?" and "What is usually taught at that grade level?" to "What concepts has this child mastered?" and "What should this child learn next?"

Attention to the child rather than to the classroom resulted in a reorganization of the drill-and-practice material in elementary-school mathematics into ungraded strands. The student, working on several strands simultaneously, begins at the bottom of a strand and moves upward on each strand as a function of his ability to perform correctly on that strand. Since movement along a strand depends on the student, the level of performance on one strand relative to the level of performance on other strands creates a problem set for one student different from the problem set for another student. Thus, unlike in the traditional classroom, each student is solving a different set of problems, and each set of problems contains problem types from each strand appropriate to the ability level of the student involved.

The strand system consists of three major elements:

1. A curriculum structure that classifies the problems appropriate for an elementary-school mathematics program;

2. A set of rules for determining the problems to be presented to each student;

3. A set of rules to define the progress of a student through the structure.

The present curriculum structure contains 15 strands. Each strand includes all problem types of a given concept, e.g., fractions, equations, or of a major subtype of a concept, e.g., horizontal addition, vertical multiplication, presented in Grades 1-6. Table 4 shows the 15 strands and the portion of the six-year curriculum for which they are appropriate.

Within each strand, problems of a homogeneous type, e.g., all horizontal addition problems with a sum between zero and five, are grouped into equivalence classes. Each strand contains either 5 or 10 classes per half year with each class labeled in terms of a grade-placement equivalent. A problem count of problem types occurring in three major elementary-school mathematics texts and data collected during the past three years of the drill-and-practice program at Stanford were used to arrange the equivalence classes in an increasing order of difficulty and to insure that new skills, e.g., regrouping in addition, were introduced at the appropriate point in the curriculum.

In addition to the ordering of the problems within a strand, we must know how much emphasis is needed on each strand at a given point in the year. To determine this, we divided the curriculum into 12 parts, each corresponding to half a year. A probability distribution was defined for the proportion of problems on each strand for each half year. Both the problem count from the three textbooks and the average latency for problem types based on past data were used to characterize the curriculum distribution. The final proportions in terms of time and problems for each half year for each strand are shown in Table 5, with the exception of Strand 15 (problem solving) which is handled separately.

A student's progress through the strand structure is a function of his performance on each strand. As certain criteria of performance are satisfied for a given strand, the equivalence class from which the student is receiving problems changes, with a corresponding change in the student's grade placement on the strand. The criterion for a given equivalence class is a function of the strand and half year of which that class is a member.

For each equivalence class the criterion is stated in terms of three integers, W, Y and Z. After every Y problems on a strand the student's performance is examined; if he did W or fewer problems correctly, he moves down one equivalence class; if he did more than W and fewer than Z problems correctly, he stays at the same equivalence class; if he did at least Z problems correctly, he moves up one equivalence class. An exception to the criterion for movement is made when a student is presented problems from a given equivalence class for the first time. In such a case, a check is made after the first three problems; if the student did all three incorrectly, he moves down one equivalence class.

The calculation of the values of W, Y and Z for each equivalence class involved the combination of known facts, estimated facts and several assumptions. Knowing the amount of time a student would spend doing problems during a half year,

TABLE 4

Content and Duration of Each Strand

Strand	Content	Grade range
1	Counting and place value	1.0-7.0
2	Vertical addition	1.0-6.0
3	Horizontal addition	1.0-3.5
4	Vertical subtraction	1.5-6.0
5	Horizontal subtraction	1.0-3.5
6	Equations	1.5-7.0
7	Horizontal multiplication	2.5-5.5
8	Vertical multiplication	3.5-7.0
9	Fractions	3.5-7.0
10	Division	3.5-7.0
11	Large numbers and units of measure: time, money, linear measure, dozen, liquid measure, weight, Roman numerals, metric measure	1.5-7.0
12	Decimals	3.0-7.0
13	Commutative, associative, and distributive laws	3.0-7.0
14	Negative numbers	6.0-7.0
15	Problem solving	3.0-7.0

and estimating the average latency from presentation of a problem to a response from the student for each problem type (equivalence class), we estimated the number of problems a student would receive from each strand during a half year. Then, assuming that a student has an average probability correct of .70, the values of W, Y, and Z were computed so that a student would be expected to increase his grade placement by .5 on all strands during a half year of time at the computer terminal.

A programmed tutorial course in BASIC was added as another computer programming course during the year. Students in both Wilson High School and the Stanford Medical Center were enrolled in the program. In addition, a small number of students at Wilson High School took a more advanced programming course in the computer language AID.

Due to federal cut-backs in the 1969-70 school year, the initial reading program was operative in only two Ravenswood elementary schools. Selected kindergarten students, all first and second graders, and remedial third-through-sixth-grade students took part in the program on a daily basis.

The mathematics and logic programs were continued for the students at Kendall School for the Deaf in Washington, D. C. Thirty special lessons were written and used by deaf students at Kendall School during April, May and June. The lessons included the grammar of single noun phrases and an introduction to interrogative transformations. One teletype terminal was located in a school in Cupertino School District in California for handicapped children who were trainable.

The first- and second-year computer-based Russian language courses were offered for credit by Stanford University. In the autumn quarter, 49 students registered for the first-year course and 31 registered for the second-year course. The spring quarter ended with 39 first-year students and 22 second-year students. Approximately 90 students were turned away from the first-year course in September because of lack of facilities to accommodate more than 80 students. While the second-year course continued to be revised, the first-year course ran without further changes.

Using the experience gained from the logic and algebra course for bright elementary-school students, we prepared and tested an introductory college course in elementary mathematical theories with Stanford students. As in the case of the Stanford Russian courses, the bulk of the instruction took place at teletype terminals. A revised version of the material on sentential logic and the algebra of ordered fields was used. In addition, axioms for Boolean algebra in quantifier-free form were presented and the students were asked to prove a number of elementary theorems. The emphasis throughout the course is on getting practice in proving theorems. To this end the students were encouraged to experiment with various lines of attack on a theorem and to view the teletype output as creating a "work space" or scratch pad for thinking through a proof. Because the program formally checks the correctness of a proof, false starts and blind alleys remain part of the output and provide us with an unparalleled opportunity to study the methods of attack tried by the students.

On Monday, May 18 and Friday, May 22, 1970 the Institute made what is perhaps the first use of a communication satellite to distribute CAI. The demonstration was important because it proved that satellite distribution of CAI through low-cost satellite ground stations had the potential for making CAI as accessible to isolated rural areas as to large cities.

Prediction for the Seventies

I turn now to some predictions about the development of computer-assisted instruction for the coming decade. Rather than engage in general long-range speculations that will not be realized for another fifty years, I shall attempt to make my predictions relatively concrete and definite and pertinent to the coming decade. As a reference, I shall use the history of our efforts at Stanford. In a general way I have divided my projections for the future into four parts. The first part deals with research on dialogue and the interaction between student and computer program. The second deals with the theoretical problem of building an adequate psychological model of the student. The third deals with some operational predictions concerning the simplest applications and their spread during the next ten years. Finally, the last part deals with the social and cultural impact of the continued spread in use of computers for instruction.

1. Research on dialogue

Without giving the subject much reflection, one might think the appropriate model for a dialogue should be Socrates at work in the Platonic dialogues, but it does not take much perusal of Plato's writings to recognize that this is not a serious pedagogical or psychological model of how an instructive or tutorial conversation should take place. The real problem is that we do not have a good intellectual model that is well enough developed for the interaction between a tutor and his pupil. We therefore do not have a sharply defined analytical model that we can plan to simulate in formulating powerful computer programs. The central difficulty in the area of interaction between student and program is not the clumsiness or limitations of the computer, but our ignorance in understanding in any explicit way the character of a successful dialogue. A large number of topics being studied either as a part of computer-assisted instruction or as part of artificial intelligence should contribute to a deeper understanding of the nature of dialogue. I shall mention only a few special topics, since I see no point in trying to deal with this difficult problem in a general way.

Let me mention some of the things we are planning under this general heading for the logic and mathematics programs I have described as part of our activity at Stanford during the past decade. Perhaps the central limitation of these programs at the present time is their requirement that the student construct an explicit formal proof for every theorem. Somehow 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. Published mathematical proofs, even in relatively elementary textbooks, are far from formally complete. We must close the gap between this formal incompleteness and the theoretical conception of a proof in the formal sense. The most promising approach to this central problem in the development of more advanced mathematics courses in CAI 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. The steps in these moves are modest and of the right level of difficulty for theorem provers; they cover the many routine steps that are tedious and far too boring for the student to make explicitly if he is called upon to prove any genuinely interesting theorems. For example, repeated use of the commutative and associative laws in a fashion that is common in elementary algebraic arguments would be turned over to the theorem prover to execute. The same remark applies to all standard arguments using sentential or predicate logic. Once the student has learned the elements of sentential and predicate logic, the routine applications may properly be assigned to the theorem prover by the student.

As one mode of operation for the use of theorem provers, we introduce an additional instruction into the proof procedures, an instruction called show. In this case the student inputs what he wants the theorem prover to show; he also indicates the preceding theorems and axioms from which the intermediate result should be derived. Our theorem prover is of sufficient power to take these intermediate steps, but not adequate to take the larger steps required for an entire proof. There is good reason to believe that this will probably be the situation for several years. My own feeling is that the instructional use of theorem provers is perhaps one of the best operational arenas in which to develop and improve on the results accomplished thus far. Without a facility such as a theorem prover I see little hope of being able to give self-contained courses that catch the spirit of more advanced parts of mathematics in the sense of requiring the student to give proofs of the main theorems.

A second and closely related activity for which theorem provers are a necessary ingredient is that of monitoring a 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. Again, at least in elementary and semi-elementary domains of mathematics, there is hope of concretely realizing programs of this sort. The data base is simple, namely, the elementary mathematical theory, together with the data on the student's current attempt at a proof. Investigations of ways in which to complete the proof begun by the student are in such contexts not overly difficult. The theorem prover searches for a way to complete the proof and then gives the student a hint of the next step to take when he has run out of conjectures himself. Preliminary work that we have begun on this line of attack seems promising. I do not for a moment underestimate the problems of extending our work to more complex bodies of mathematics. I do think it is an important direction for developing richer mathematical courses in a computer-based environment.

In many respects we can expect to make the most rapid progress in the domain of mathematics, because of its limited data base, the formality and explicitness of its language, and our own very explicit understanding of the structure. The development of tools to provide aids and hints in other domains will not be a simple matter. There is currently a variety of attacks on the development of good question-answering systems. Although adequate systems are still far from available, it seems likely that the development of question-answering systems for use in instructional settings will be an important part of research in CAI during the seventies.

I would like briefly to mention some of my own work in this area, especially work conducted in conjunction with Dr. Hélène Bestougeff of the University of Paris. Dr. Bestougeff and I are attempting to write a question-answering system with certain features that have previously been missing in the literature and that we think are probably highly desirable for future progress in this domain. The central objectives of our study can be described very simply. We are attempting to define for the question inputs and answers a machine-independent grammar and semantics such that when the program is constructed we can prove a theorem asserting that every question is answered correctly. Of course, by saying that every question is answered correctly we mean that every question is answered correctly relative to the data base. Without an explicit grammar for the fragment of a natural language used for the input questions and without an explicit semantics for this fragment, it is impossible to prove a formal theorem about the nature of the question-answering system. As in other domains of science, there is also a hope that by introducing a deeper structure into the question-answering system--such as the kind introduced by an explicit grammar and semantics--we shall be able to handle more efficiently and develop more easily the actual system itself. Whether or not my conjectures about this direction of development are correct, there seems to be little doubt that progress in this area will be a significant part of CAI work in the coming decade.

Closely related to what I have said about question-answering systems is the whole domain of developing genuine voice-to-voice interaction between student and program. We are beginning to have an understanding of natural language grammar and semantics for fragments of natural languages, adequate to produce a reasonable line of talk on the part of the computer. By making the grammar and semantics probabilistic, we can also avoid the stereotyping that otherwise would be a disturbing character of the computer talk.

Important work on speech recognition by Dr. D. Rajagopal Reddy and others has taken place during the past decade. Reddy and his group are able to recognize in reasonable time a vocabulary of up to about 500 words. While it is true that the machine power required for this recognition is awe-inspiring and far too great for operational applications in CAI, there is reason to hope that fairly soon at least a small vocabulary may be recognized easily. Then we shall be in a position to have a genuine voice-to-voice interaction between computer and program with the beginnings of a genuine verbal dialogue.

2. <u>Research on model of the student</u>

The bulk of our research at Stanford and research conducted elsewhere on student performance in computer-based courses is at a very empirical level. Roughly speaking, the behavioral research falls into two classes. One class of studies is concerned with external evaluation by achievement test data of the comparative performance of students in CAI. In these studies, control groups of a comparable nature either receive ordinary instruction or ordinary instruction without benefit of supplementary CAI work. The second class of studies is concerned to analyze the detailed performance of student's responses in a CAI course. Regression studies of item difficulty, reported extensively in Suppes, Hyman and Jerman (1967) and in Suppes, Jerman and Brian (1968), are typical. The structural variables that are defined as independent variables in these linear regression studies involve almost without exception complexities of the subject matter and the curriculum, not postulated complexity of the student. Structural assumptions about the student only enter through consideration of the dependent variable, which in these studies has been either probability of a correct response or latency of response.

It might be thought that the developments in cognitive psychology, especially the structuralism of Piaget and others, would provide a basis for going beyond sheer empiricism in considering student responses. Unfortunately, however, it does not take an extended perusal of the literature in cognitive psychology to determine that the models are not sufficiently developed in a mathematical fashion to provide a genuine tool for the analysis of data. Perhaps the best way to put the matter is that the current cognitive theories are simply not specific and definite enough in their formulation of basic assumptions to lead to specific predictions. There simply are not the tools in the writings of Piaget, nor in those of Bruner and others, adequate to provide predictions of differential difficulty over a selection of items drawn from some complex domain like that of elementary arithmetic or elementary foreign-language learning. I do not claim that Piaget or Bruner, for example, have stated that they offer such tools. I merely make explicit the fact that such tools are not available in the theoretical work they have as yet offered us.

I do think, however, that within the general tradition of stimulus-response psychology, tools of an adequate precision and complexity are now available for at least the elementary parts of skill subjects, such as mathematics and foreign-language learning. Concerning models for foreign-language learning, some preliminary, but at least specific, models are offered by Crothers and me (1967), along with extensive tests of these models. In a more recent and promising vein, we have begun to use probabilistic automaton models to study the performance of students in elementary arithmetic. Theoretical formulation of these models is begun in Suppes (1968) and is currently under active development in our work at the Institute. Our objective is to build a large probabilistic automaton model for the individual student, with individual parameters reflecting his state of learning and performance, and then to present instruction differentially and contingently so as to change the values of critical parameters in the model of the individual student in a manner that can be characterized in one specific sense as optimal. I shall not enter into details here, but I do emphasize that I think the task of building an automaton to model the individual student, even in a subject as well defined as elementary arithmetic, is far from trivial. We are currently successfully testing models for individual parts of arithmetic of this sort, and these models have a full information-processing capacity. But, putting together a common model for a given student across, say, the skills of addition, subtraction, multiplication and division is already proving to be a more complex and difficult problem than we had originally anticipated. While I would not want to be overly sanguine about the depth of what we shall be able to accomplish in the next few years, I do think this is the direction in which we must move. The computer program must have a more sophisticated and complex representation of the student in order to provide instruction that is properly tailored to the individual student.

3. Some operational predictions

It is not possible to construct an adequate data base from which to make a serious prediction about the operational use of CAI in the United States or in any other country during the course of the next decade. Certainly in the past, sanguine predictions have been made that have turned out not to be true, and I do not want to engage in any overly optimistic forecasts in the present discussion. There are, however, already signs that the effort, at least in the United States, will be substantial during the seventies, and by this I mean that a substantial effort will be made on the part of school systems and not simply on the part of research centers like ours at Stanford.

I do think that the following prediction is a reasonable one. By 1980, 15 percent of the students in the United States at all grade levels will be in daily contact with a computer for some aspect of their instruction. At the elementary-school level this will probably be especially in the areas of reading and mathematics. One large-scale operational system is already installed in New York City, and the city of Chicago is in the process of making a similar installation that will become operational in the fall of 1971. A number of smaller school systems have already purchased systems. The list is too long to enumerate here. It is on the basis of the above information that I predict that over the course of the next ten years at least 15 percent of the students will have such involvement. My forecast is the same for secondary schools and for colleges, but let me pursue the analysis a bit for the case of the elementary school. There are approximately a million elementary-school classrooms in the United States. Fifteen percent of these is approximately 150,000. The ordinary classroom has between 25 and 30 students. For drill and practice in mathematics or in reading, one terminal per classroom would be an appropriate allocation. During the seventies, the cost per terminal will probably be about $3,000. This means that by the end of the decade we will incur a cost of approximately half a billion dollars to service 15 percent of these students. In terms of current school costs in the United States, this is not an unrealistic allocation. Certainly the expenditure of fifty million dollars a year for ten years is a relatively modest expenditure, considering the enormous concentration on basic reading and mathematics skills in the elementary school and the fact that about a billion-and-a-half dollars is being allocated each year primarily for such concentration as part of Title I of the Elementary-Secondary Act of 1965.

I believe that similar forecasts can be made for the secondary schools and colleges. However, in the case of the colleges, the use may be somewhat different. For example, the student may operate in more of a tutorial mode as in the case of the Russian and logic courses at Stanford described earlier. I have recently been involved in the implementation of a tutorial course in basic English at the college level. This course is designed for students who are not able to pass a standard placement examination upon entrance into college and who need remedial work in grammar and composition. At Stanford we have also been involved in similar work in remedial mathematics. The work described at Tennessee State University is an example.

4. Social and cultural impact of CAI

In the Hellenistic world of, say, 100 B.C., a scholar who wanted to read and study literature or science in a domain of his interest was able to do so only in a small number of places. He could go, for example, to Alexandria and work in the great library and museum. He could also find papyri in other great cities such as Athens and Syracuse. Unless he were a man of great wealth, he would have few of these papyrus manuscripts in his own house. With the development of printing 1500 years later, it became possible (starting in the sixteenth century) for a man of ordinary affluence to acquire a substantial library for his personal use. In the twentieth century, even a person of modest means has access to large libraries with extensive holdings in most domains of science and literature. The bulk of the population in Europe and the United States is within relatively east traveling distance to a library of some serious proportions. The cultural impact of this slow, but increasing accessibility of learning has without question been enormous and one of the most important features of modern culture.

It is reasonable to ask ourselves if the same will be true of the slow, but inevitable spread of computer facilities. What can we anticipate? I do not want to attempt to forecast all the dimensions of development, but just to concentrate on that concerned with instruction. I believe the most important social change that will begin in the seventies, but not have a major impact until after that decade, will be the placement of computer terminals in homes and the availability of a wide range of courses for the continuing education of adults. We are already formulating plans at the Institute for a large-scale experiment on the use of computers installed in homes for instructional purposes during the seventies. Initially, we have been thinking of two sorts of students. One sort is citizens who need additional education in basic skills and vocational training in order to complete their education. At the present time only about 70 percent of the population completes their secondary school education. In hospitals, in factories, in businesses and in government there is substantial employment of individuals who are blocked from further advancement because of their lack of education and who now perceive the advantages of completing secondary school and possibly taking additional work. We would like to have an organized set of courses that would allow students who are now in their mature years and who are fully employed to complete secondary school requirements.

The second sort of students include professional people who already have a relatively high degree of formal education, but who wish to develop additional skills, such as mastery of an additional language or acquisition of technical skills like those of computer programming or statistics.

It is not possible to predict how successful such courses would be or how well they would be received by the individuals initially in the experiment, but they do represent a development that seems almost inevitable. It will be an important aspect of CAI in the seventies to identify those skills and subject matters that adults will want to learn or acquire in the privacy of their own homes.

Bringing computer terminals into the home is in one sense the ultimate act of decentralization in education. It can apply not only to adults, but also to children. A social problem of the future is the extent and nature of such decentralization. The answers will depend on social and cultural, rather than technical considerations.

A bibliography of articles and technical reports on research in the Institute relevant to CAI is included.

Bibliography

Atkinson, R. C. Reading instruction under computer control. American School Board Journal, 1967, 155, 16-17.

Atkinson, R. C. Learning aspects of computer-assisted instruction. In R. W. Gerard (Ed.), Computers and education. New York: McGraw-Hill, 1967. Pp. 11-63.

Atkinson, R. C. Instruction in initial reading under computer control: The Stanford project. Technical Report No. 113, July 14, 1967, Stanford University, Institute for Mathematical Studies in the Social Sciences.

Atkinson, R. C. Computerized instruction and the learning process. American Psychologist, 1968, 23, 225-239.

Atkinson, R. C. The role of the computer in teaching initial reading. Childhood Education, 1968, 44, 464-470.

Atkinson, R. C. Computer-based instruction in initial reading. Proceedings of the 1967 invitational conference on testing problems. Princeton, N. J.: Ed. Testing Svc., 1968. Pp. 55-66.

Atkinson, R. C. Learning to read under computer control. Programmed Learning and Educational Technology: British Journal of the Association for Programmed Learning, 1968, 5, 25-37.

Atkinson, R. C. The computer as a tutor. Psychology Today, January, 1968.

Atkinson, R. C. A reply to Professor Spache's article "A reaction to computer-assisted instruction in initial reading." Reading Research Quarterly, 1968, 3, 418-420.

Atkinson, R. C. Computer-assisted learning in action. Proceedings of the National Academy of Sciences, 1969, 63, 588-594.

Atkinson, R. C., Fletcher, J. D., Chetin, H. C., and Stauffer, C. M. Instruction in initial reading under computer control: The Stanford project. Educational Technology, 1971, in press.

Atkinson, R. C., and Hansen, D. Computer-assisted instruction in initial reading: The Stanford project. Reading Research Quarterly, 1966, 2, 5-25.

Atkinson, R. C., and Paulson, J. A. An approach to the psychology of instruction. Psychological Bulletin, 1971, in press.

Atkinson, R. C., and Wilson, H. A. Computer-assisted instruction. Science, 1968, 162, 73-77.

Crothers, E., and Suppes, P. Experiments in second-language learning. New York: Academic Press, 1967.

Fishman, E. J., Keller, L., and Atkinson, R. C. Massed vs. distributed practice in computerized spelling drills. Journal of Educational Psychology, 1968, 59, 290-296.

Fletcher, J. D., and Atkinson, R. C. An evaluation of the Stanford CAI program in initial reading (grades K through 3). Technical Report No. 168, March 12, 1971, Stanford University, Institute for Mathematical Studies in the Social Sciences.

Jamison, D., Suppes, P., and Butler, C. Estimated costs of computer assisted instruction for compensatory education in urban areas. Educational Technology, 1970, 10, 49-57.

Jerman, M., and Suppes, P. A workshop on computer-assisted instruction in elementary mathematics. The Arithmetic Teacher, 1969, 16, 193-197.

Knutson, J. M. Spelling drills using a computer-assisted instructional system. Technical Report No. 112, June 30, 1967, Stanford University, Institute for Mathematical Studies in the Social Sciences.

Loftus, E. J. F. An analysis of the structural variables that determine problem-solving difficulty on a computer-based teletype. Technical Report No. 162, December 18, 1970, Stanford University, Institute for Mathematical Studies in the Social Sciences.

Rodgers, T. S. Linguistic considerations in the design of the Stanford computer-based curriculum in initial reading. Technical Report No. 111, June 1, 1967, Stanford University, Institute for Mathematical Studies in the Social Sciences.

Suppes, P. Modern learning theory and the elementary-school curriculum. American Educational Research Journal, 1964, 1, 79-94.

Suppes, P. Problems of optimization in learning a list of simple items. In M. W. Shelly, II, and G. L. Bryan (Eds.), Human judgments and optimality. New York: Wiley, 1964. Pp. 116-126.

Suppes, P. The formation of mathematical concepts in primary-grade children. In A. H. Passow and R. R. Leeper (Eds.), Papers from the ASCD eighth Curriculum Research Institute, 1964. Pp. 99-119.

Suppes, P. Towards a behavioral foundation of mathematical proofs. In K. Ajdukiewicz (Ed.), The foundations of statements and decisions: Proceedings of the international colloquium on methodology of science, September 18-23, 1961. Warszawa: PWN - Polish Scientific Publ., 1965. Pp. 327-341.

Suppes, P. On the behavioral foundations of mathematical concepts. Monographs of the Society for Research in Child Development, 1965, 30, 60-96.

Suppes, P. Mathematical concept formation in children. American Psychologist, 1966, 21, 139-150.

Suppes, P. Towards a behavioral psychology of mathematical thinking. In J. Bruner (Ed.), Learning about learning (a Conference Report). Washington, D. C.: U. S. Govt. Printing Off., 1966. Pp. 226-234.

Suppes, P. The psychology of arithmetic. In J. Bruner (Ed.), Learning about learning (a Conference Report). Washington, D. C.: U. S. Govt. Printing Off., 1966. Pp. 235-242.

Suppes, P. Applications of mathematical models of learning in education. In H. O. A. Wold (Scientific Organizer), Model building in the human sciences. Entretiens de Monaco en Sciences Humaines, Session 1964. Monaco: Union Europienne D'Editions, 1966. Pp. 39-49.

Suppes, P. Accelerated program in elementary-school mathematics--the second year. Psychology in the Schools, 1966, 3, 294-307.

Suppes, P. The axiomatic method in high-school mathematics. The role of axiomatics and problem solving in mathematics. The Conference Board of the Mathematical Sciences. Washington, D. C.: Ginn, 1966. Pp. 69-76.

Suppes, P. The uses of computers in education. Scientific American, 1966, 215, 206-221.

Suppes, P. On using computers to individualize instruction. In D. D. Bushnell and D. W. Allen (Eds.), The computer in American education. New York: Wiley, 1967. Pp. 11-24.

Suppes, P. Some theoretical models for mathematics learning. Journal of Research and Development in Education, 1967, 1, 5-22.

Suppes, P. The psychological foundations of mathematics. Les modeles et la formalisation du comportement. Colloques Internationaux du Centre National de la Recherche Scientifique. Paris: 1967. Pp. 213-242.

Suppes, P. The teacher and computer-assisted instruction. National Education Association Journal, 1967, 56, 15-32.

Suppes, P. The case for information-oriented (basic) research in mathematics education. In J. M. Scandura (Ed.), Research in mathematics education. Washington, D. C.: NCTM, 1967. Pp. 1-5.

Suppes, P. Computer technology and the future of education. Phi Delta Kappan, 1968, 44, 420-423.

Suppes, P. Systems analysis of computer-assisted instruction. In G. J. Kelleher (Ed.), The challenge to systems analysis. New York: Wiley, 1970. Pp. 98-110.

Suppes, P. Computer-assisted instruction: An overview of operations and problems. In A. J. H. Morrell (Ed.), Information processing 68, proceedings of IFIP congress 1968, Edinburgh. Vol. 2. Amsterdam: North-Holland, 1969. Pp. 1103-1113.

Suppes, P. Stimulus-response theory of finite automata. Journal of Mathematical Psychology, 1969, 6, 327-355.

Suppes, P. Stimulus-response theory of automata and tote hierarchies: A reply to Arbib. Psychological Review, 1969, 76, 511-514.

Suppes, P. Probabilistic grammars for natural languages. Synthese, 1970, 22, 95-116.

Suppes, P. Semantics of context-free fragments of natural languages. Technical Report No. 171, March 30, 1971, Stanford University, Institute for Mathematical Studies in the Social Sciences.

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

Suppes, P., and Groen, G. Some counting models for first-grade performance data on simple addition facts. In J. M. Scandura (Ed.), Research in mathematics education. Washington, D. C.: NCTM, 1967. Pp. 35-43.

Suppes, P., and Hansen, D. Accelerated program in elementary-school mathematics--the first year. Psychology in the Schools, 1965, 2, 195-203.

Suppes, P., and Hill, S. Mathematical logic for the schools. The Arithmetic Teacher, 1962, 9, 396-399.

Suppes, P., Hyman, L., and Jerman, M. Linear structural models for response and latency performance in arithmetic on computer-controlled terminals. In J. P. Hill (Ed.), Minnesota symposia on child psychology. Minneapolis: Univ. Minn. Press, 1967. Pp. 160-200.

Suppes, P., and Ihrke, C. Accelerated program in elementary-school mathematics--the third year. Psychology in the Schools, 1967, 4, 293-309.

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

Suppes, P., and Jerman, M. Computer-assisted instruction. The Bulletin of the National Association of Secondary School Principals, 1970, 54, 27-40.

Suppes, P., Jerman, M., and Brian, D. *Computer-assisted instruction: Stanford's 1965-66 arithmetic program.* New York: Academic Press, 1968.

Suppes, P., Loftus, E., and Jerman, M. Problem-solving on a computer-based teletype. *Educational Studies in Mathematics*, 1969, *2*, 1-15.

Suppes, P., and Morningstar, M. Computer-assisted instruction. *Science*, 1969, *166*, 343-350.

Suppes, P., and Morningstar, M. Technological innovations: Computer-assisted instruction and compensatory education. In F. Korten, S. Cook, and J. Lacey (Eds.), *Psychology and the problems of society.* Washington, D. C.: Amer. Psychol. Assoc., 1970. Pp. 221-236.

Wilson, H. A., and Atkinson, R. C. Computer-based instruction in initial reading: A progress report on the Stanford project. Technical Report No. 119, August 25, 1967, Stanford University, Institute for Mathematical Studies in the Social Sciences.