DOCUMENT RESUME

ED 196 709 SE 033 690

| | |
|---|---|
| AUTHOR | Levine, Arvin; And Others |
| TITLE | Technical Problems in Implementing University-Level Computer-Assisted Instruction in Mathematics and Science: Second Annual Report. |
| INSTITUTION | Stanford Univ., Calif. Inst. for Mathematical Studies in Social Science. |
| SPONS AGENCY | National Science Foundation, Washington, D.C. |
| PUB DATE | 22 Mar 79 |
| GRANT | NSF-SED77-09698 |
| NOTE | 59p.; For related document, see ED 163 973. Contains marginal legibility in Appendix. |
| EDRS PRICE | MF01/PC03 Plus Postage. |
| DESCRIPTORS | *Computer Assisted Instruction; *Educational Research; Educational Technology; *Higher Education; Individualized Instruction; Logic; Mathematical Applications; *Mathematical Logic; Mathematics Education; *Mathematics Instruction; *Programing Languages; Science Instruction |
| IDENTIFIERS | *Mathematics Education Research; Proof (Mathematics); VOCAL Programing Language |

ABSTRACT

Difficulties in implementing the EXCHECK/Voice Oriented Curriculum Author Language (VOCAL) System, a general program designed for university-level computer-assisted instruction in mathematics and science written in the VOCAL language, are presented in terms of informal mathematical procedures, audio and prosodic features, and a schedule of proposed research. A full bibliography of presentations and publications supported by the National Science Foundation (NSF) during the grant, a sample proof, reduction proof procedures, and references are appended. (MP)

# TECHNICAL PROBLEMS IN IMPLEMENTING UNIVERSITY-LEVEL COMPUTER-ASSISTED INSTRUCTION IN MATHEMATICS AND SCIENCE: SECOND ANNUAL REPORT

by

Arvin Levine, Lee Blaine and Patrick Suppes

NSF Grant No. SED77 - 09698

March 22, 1979





INSTITUTE FOR MATHEMATICAL STUDIES IN THE SOCIAL SCIENCES

STANFORD UNIVERSITY

STANFORD, CALIFORNIA 94305

## Abstract

This report summarizes the research conducted at the Institute, under support of the National Science Foundation, concerning technical problems in implementing university-level computer-assisted instruction in mathematics and science. Two areas of research have been pursued:

1) Informal mathematical procedures. The development of sophisticated and efficient methods for students to use in interactive proof procedures. In particular, the research has concentrated on providing both low-level (e.g., 'intelligent' typing aids) and high-level (e.g., proof strategy mechanisms) procedures to enable students to follow standard mathematical practice as far as possible.

2) Audio and prosodic features. Audio research has concentrated on design considerations for the Mini-MISS machine, especially the problem of using interpolation of linear predictive coding of speech to achieve a low transmission rate without sacrificing speech quality. Also, an on-line prosodic generation procedure was implemented, and, improvements to the syntactic analysis and lexicon were made. Finally two studies of the MISS prosodic quality were carried out.

In addition to discussion of these topics, this report includes a full bibliography of presentations and publications supported by NSF during the first two years of this grant.

## Table of Contents

Appendix

## List of Figures

Figure | Page

## Acknowledgments

The authors wish to thank the following people for their contribution to this report: Mark Davis, Robert Laddaga, James McDonald, and William Sanders.

For their contributions to the research conducted at the Institute and described in this report we also thank: Barbara Anderson, George Black, Edward Bolton, Benjamin Cohen, Thomas Dienstbier, Carolyn Gramlich, Pentti Kanerva, Reed Letsinger, Ingrid Lindstrom, Sten Lindstrom, Lawrence Zaven Markosian, Jos Marlowe, Teri Pettit, Ronald Roberts and Marguerite Shaw.

The work described in this report was supported by the National Science Foundation under NSF Grant No. SED77-09698.

## 1 Introduction

This report summarizes the research conducted at the Institute for Mathematical Studies in the Social Sciences (IMSSS) at Stanford, under support of the National Science Foundation, concerning technical problems in implementing university-level computer-assisted instruction (CAI) in mathematics and science. We have pursued research in two areas relating to CAI: informal mathematical procedures and the use of audio messages with prosodic features. Our overall objective has been to research and develop the use of computers for instructional purposes. An overview of this project, together with a review of other CAI projects can be found in Suppes (1979).

In the area of informal mathematical procedures, we have developed sophisticated and efficient methods by which students can use interactive proof procedures in courses such as "Introduction to logic" and "Set theory". Part of the effort during this grant year has been devoted to modularizing and streamlining the programs that present and control the student courses and providing these courses with useful low-level aids such as command and formula recognition, on-line help and editing. A fuller discussion of the streamlining and student aids is given in Section 2.1.

Several different high-level procedures have been developed during this grant year. These procedures share a common goal of making it possible for the student to follow standard mathematical practice in doing proofs with the result that more of the routine work is shifted from the student to the computer. A procedure to execute proof strategies that reduce problems and goals to subproblems and subgoals is discussed below, in Section 2.2. Guidance to students attempting to solve problems is provided by utilizing the theorem prover as a knowledge base (see Section 2.3). A more sophisticated inference procedure was also added during this year (Section 2.4).

A new exercise type has been added to the curriculum during this grant year. This exercise type directs the student to provide finite objects satisfying given (logical) conditions. The structure of the exercise allows the computer to provide some automatic explanation of wrong answers. We discuss such exercises below, Section 2.5.

In the area of audio and prosodic features, a major avenue of research has been the design considerations for the Mini-MISS machine, especially the problem of using interpolation of linear predictive coding of speech to achieve a low transmission rate without sacrificing speech quality. Section 3.1 discusses these studies and an experiment we are conducting to determine the perceptual effect of a particular interpolation model.

Another focus of audio research has been on the improvement of the linguistic analysis and prosodic generation algorithms. A simple model of a "story-view" analysis of dialogue is being extended to generate paragraph prosodic contours (see Section 3.2.1). Improvements to the

syntactic analysis and prosodic generation routines to utilize semantic insights have been implemented (Section 3.2.2). An on-line prosodic generation procedure was implemented (see Section 3.2.3), providing the capability for production of audio messages generated during execution of the curriculum.

Two studies of the MISS prosodic quality were carried out. The first study examined the relative contributions to unnaturalness of different components of the method of word concatenation with prosodic manipulation as implemented in MISS (Section 3.3.1). The second evaluated prosodic quality by examining student use of audio in the course "Introduction to Logic" at Stanford University (Section 3.3.2). Further improvements to the audio generation are being planned in response to the results of these studies.

A substantial effort has been made to disseminate research results as described in this report, and engage in dialogue with other research groups. Project members have presented and published papers and lectures, and participated in panels dealing with both the theory and applications of research by us and others. This report includes a full bibliography (Appendix A) of these presentations and publications, supported by NSF, during the first two years of this grant.

## 2 Informal Mathematical Procedures

The work in informal mathematical procedures is being carried out as described in the section on proposed research in the original proposal (Suppes (1977), p. 82). Below, we review the work proposed, and describe the work accomplished during the second year.

### 2.1 General System work

We have implemented a complex system of instructional programs, EXCHECK, which presents the courses in introductory logic and set theory, and facilitates the development of curriculum for these courses. A primary goal of the design of EXCHECK is simplicity of use without sacrifice in the sophistication of the material to be taught or the modes used to present it. EXCHECK has been implemented so that lesson authors need not have any general knowledge of programming languages or techniques in order to write complex curriculum materials, and students are not presumed to have knowledge beyond that appropriate to the subject being taught. This implementation has been done in a modular way to facilitate program maintenance, modification and expansion (see below, Section 2.1.1). An important part of the modularization is the use of symbolic referencing of messages (see Section 2.1.2). We discuss some low-level student aids in Section 2.1.3, and the general input mechanism in Section 2.1.4.

#### 2.1.1 Description of the EXCHECK system

The EXCHECK system has a modular design in which various functions have been segregated into separate programs, as seen below, Figure 1. Using features inherent in the TENEX operating system at IMSSS, these programs run as a single large program with interacting components. All curriculum lessons are written in VOCAL (Voice Oriented Curriculum Author Language, Hinckley et al (1976)). The VOCAL compiler produces interpretable lesson code from VOCAL lessons. The EXCHECK driver interprets these lessons and governs each student's progress through the course. It may invoke a proof checker to present the derivation of proofs or the construction of explicit mathematical examples, described elsewhere in this report. The proof checker has access to a theorem prover and to REDUCE, a program for algebraic reductions Hearn (1973). The audio program is described in detail in Levine and Sanders (1978) and below, Section 3. Each course has its own parser for handling complex input specific to the course (e.g., formulas) which comes from the student.

The ability of EXCHECK to run as a dynamic configuration of programs depends upon the multiprocessing facility of TENEX to coordinate many core images ("forks") as one program, without the need to program explicit overlays. When a student logs in the EXCHECK course

```
              Vocal Compiler
             /     |      \ \
            /      |       \ Text-editor
           /       |        \
          / Excheck Driver   \
         / /       |      \   \
        / /        |       \   \
       / /  Proof-Checker   \   \
      / / /    |       |     \ \ \
     / / /     |       |      \ \ \
    Audio   Prover  Reduce  Parser
```

Figure 1. Diagram of the EXCHECK system.

driver is started and it decides which forks to assemble for that student. Each fork can be written in a different programming language, allowing programs written at other sites, such as REDUCE Marti et al (1977), to be readily incorporated for EXCHECK courses.

### 2.1.2 Symbolic referencing of messages and interfork communication

A wide variety of messages are transmitted to the student during a typical session. These include error messages, prompts for answers, information about possible answers and hints for doing proofs. These messages may be either displayed on the terminal or generated with audio through headphones. As these messages have begun to occupy a large amount of space in the programs, a facility was added to EXCHECK for referencing these messages symbolically from a file, as opposed to keeping the message itself in the core image. Beyond saving a significant amount of space, this feature simplifies the maintenance of messages and encourages the writing of more descriptive messages.

A great deal of information is passed between the various forks during an EXCHECK lesson. For example, the course driver and proof checker (which are separate forks) often need to access routines contained in the other fork. The simple fork structure makes it awkward for the proof checker to call routines in the driver fork, and also requires the driver to call the proof checking fork with the printing form of the LISP-style s-expressions which the proof checker must then read and interpret even though these s-expressions have almost always already parsed in the driver fork. We are altering the forks to share a large data area for lessons. This will allow the calling fork to pass only s-expression pointers through a fork call to provide the essential context for any routine in the other fork. By using this large data area we can also make the forks "co-routine" so that the proof checker can access driver routines. Using this data area, the VOCAL compiler can save pre-parsed s-expressions on disk files so that the driver need not parse them. The savings in execution time from not re-parsing s-expressions is substantial.

## 2.1.3 EXCHECK Course Driver

The EXCHECK course driver which interprets lessons prepared by the VOCAL compiler has three basic modes for curriculum presentation: LESSON, BROWSE and HELP. The LESSONs prepared for the main curriculum direct the general pattern of students' progress through courses. The BROWSE facility is intended to simulate a student's ability to skim forward or backward while reading a textbook or notes. The students can also review and manipulate material they have created, such as stored versions of their proofs. Special lessons are available to HELP students on points not covered by the general curriculum, such as use of the computer system, hours for teaching assistants, and descriptions of the inference rules available for doing a proof. A more complete description of the EXCHECK course driver and student aids is available in McDonald et al (1978).

## 2.1.4 VINPUT

All the input in EXCHECK is obtained using one general routine called VINPUT. The major advantage of using VINPUT is that it systematizes student interactions. Using a single routine simplifies the documentation of features, and minimizes the knowledge students must have of input routines. VINPUT also has features that reduce the amount of typing required and provide assistance in determining the options available. One such feature is input recognition. When a user terminates an input string, if that string is consistent with at least one legal command, then the program will automatically add characters until an ambiguity arises or some option is completed. The student can then continue typing to disambiguate the command, if necessary.

At any prompt, the student can type a "?" and then get a brief description of any command that was legal at that prompt. The mode entered by typing "?" is indicated to the student by prompting with "??". Typing a question mark ("?") after some input will show all the alternatives which are consistent with the input so far. The "?" feature interacts with recognition, so that if recognition halts, "?" can be used to examine the alternatives before continuing.

VINPUT also offers a means through which the student can recursively access HELP, BROWSE and other special features at any point in the course. In the example shown below, Figure 2, the student entered BROWSE while in HELP, which was entered while doing a PROOF, entered while the student was in (an earlier) HELP which was entered from the main lesson. The student obtained a list of the legal commands by typing "?", followed by more information about the default command, and, finally selected the default ("ok" took the student out of the mode entered with a "?").

In complicated situations, such as in Figure 2, it is crucial for the program to provide a student interface that is easy to understand and use.

*^L You are in BROWSE inside HELP inside a PROOF inside HELP
    inside your MAIN LESSON.
Where to? ( HELP1 ): ?

You are in BROWSE inside HELP inside a PROOF inside HELP
    inside your MAIN LESSON.

?? $ stay here, browse, help1, proof, help2, main lesson, logout
    [Student typed the ESC key to see a list of the options.]
?? HELP1$ will return you to the help lesson immediately above this
    browse level
    [Student asked for information about one of those options.]
?? ok$

Where to? ( HELP1 ): $
Returning to HELP above BROWSE...

Figure 2. A sample student dialog.

### 2.1.5 Data Collection

Procedures are being implemented and extended to collect and summarize considerable amounts of objective data on student proofs, including data on response latencies, and connect-times globally and locally for individual proofs. We will also collect and summarize data such as number and kind of student errors and inputs during proofs. These data will provide a basis for comparing versions of the theorem prover, and doing regression analyses on the relationship between errors and other objective variables. They will be of general use in the evaluation of the usefulness and naturalness of features of the EXCHECK system.

## 2.2 Proof Procedures and the Structuring of Proofs

Research in the area of proof procedures concentrates on developing the ability of the proof checker to interact with the student in the informal style common to standard mathematical practice. Fundamental progress in this area will also have direct applications to mathematically oriented courses in such disciplines as physics and chemistry.

We quote from the proposed research section (pages 41-42) of the original proposal:

> The first main component of the research in this area will be to redesign the EXCHECK proof checker to function not only as an interactive proof checker but also as an interactive theorem prover

capable of accepting and executing complex proof procedures or strategies. The use of such procedures will mean that substantially more of the routine work of presenting proofs will be shifted from the student to the program. A good example of this from standard mathematical practice is saying beforehand that a proof being presented will be by induction, by cases, by the Axiom of Extensionality, etc. Such proof procedures determine a global structure on the proof being presented. Attention is then selectively focused on the various subparts of the proof, and in this way the global structure of the proof is sketched out with the details added as needed.

During the second year the EXCHECK proof checker was expanded to function as an interactive theorem prover following the design worked out during the first year. This work is not yet complete but all the basic procedures needed for EXCHECK to function as an interactive theorem prover capable of accepting and executing proof strategies, particularly reductive proof procedures, have been implemented.

Reductive proof procedures reduce the current problem or goal to a set of subproblems or subgoals. For example, corresponding to the procedure for introducing a conditional into a proof, there is a reduction procedure for conditionals. When applied to a problem in which the goal is a conditional, the conditional reduction procedure will generate a new subproblem in which the antecedent of the conditional is a new assumption and the consequent of the conditional is the new subgoal. Schematically we write this as in Figure 3. In the reduction procedures shown here, the expression to the left of an '|=>' denotes the allowed assumptions and the expression to the right denotes the goal. In this figure, the upper line reads: prove $A \rightarrow C$ from $\Sigma$. The whole schema reads: reduce the problem 'prove $A \rightarrow C$ from $\Sigma$' to the subproblem 'prove C from $\Sigma$,A'.

$$\frac{\Sigma \quad |\!\Rightarrow \quad A \rightarrow C}{\Sigma, A \quad |\!\Rightarrow \quad C}$$

Figure 3. Sample reduction.

A major benefit of the use of interactive reductive proof procedures is the imposition of a global structure on students' proofs. The proof checking program uses the global structure to provide bookkeeping aid to the student, keeping track of the current context. The finer notion of context available to the program also greatly enhances the guidance capabilities of the program. Basic guidance is given by saying what the heuristic natural deduction theorem prover itself would do on the next step. This is usually presented to the student as a default option; i.e., the student can simply accept the

program's suggestion about what to do next by typing the ESC key. The example below illustrates some these benefits as well as highly interactive nature of the system.

### 2.2.1 Using the Heuristic Natural Deduction Theorem Prover

This section presents a proof in an elementary set theoretic system using the interactive heuristic natural deduction theorem prover. When a student is doing a proof, the initial goal is the theorem to be proven. As reductive proof procedures are applied, the proof is broken into subparts. The subparts constitute a hierarchy of contexts, in which previous results and assumptions from superior contexts are available in the current subpart. After each application of a reductive proof procedure the display is updated to show the new context.

Student input is underlined in <u>all</u> the examples in this report. Our comments (which the student does not see) are enclosed in square brackets and indented. Everything else in these examples is generated by the program while communicating with the student. We separate contexts into separate frames by horizontal lines. For this example we assume that the student has chosen a printing format in which the path of reductions to the current goal is always shown.

---

Derive: $\rho A \cup \rho B \subseteq \rho(A \cup B)$

*<u>red</u>uce
Which proof procedure? (definition) *<u>t</u>heorem
Number or Name (2.8.2) *<u>$</u>2.8.2

> [The student signals the program that he wishes to reduce the current goal. The program asks which proof procedure the student wishes to use, suggesting that he use a definition (here, the definition of subset) to eliminate the major defined symbol of the current goal. However, the student wishes to do the proof in a different manner, using a theorem instead of the default. The program looks through the available theorems to determine which might work. Based upon a pattern matching routine the program suggests theorem 2.8.2. (which states that if each of A,B is a subset of C then A B is a subset of C). The student chooses this default, and the program applies the reduction. The result is shown in the next display.]

15

---

Derive: $\rho A \cup \rho B \subseteq \rho(A \cup B)$

Show: (G2) $\rho A \subseteq \rho(A \cup B)$ and $\rho B \subseteq \rho(A \cup B)$

*reduce
Which proof procedure? (conjunction) *$conjunction

[The current goal is now a conjunction. The student chooses the default to break it into two subproblems and prove each conjunct separately.]

---

Derive: $\rho A \cup \rho B \subseteq \rho(A \cup B)$

Show: (G2) $\rho A \subseteq \rho(A \cup B)$ and $\rho B \subseteq \rho(A\ B)$

Show: (G3) $\rho A \subseteq \rho(A \cup B)$

*reduce
Which proof procedure? (definition) *$definition
Number or Name (subset) *$subset

[In the previous frame we saw that the original goal was reduced to a conjunction. The goal is now the first conjunct. The student chooses to reduce the current goal by applying the definition of subset.]

---

Derive: $\rho A \cup \rho B \subseteq \rho(A \cup B)$

Show: (G2) $\rho A \subseteq \rho(A \cup B)$ and $\rho B \subseteq \rho(A \cup B)$

Show: (G3) $\rho A \subseteq \rho(A \cup B)$

Show: (G5) $(\forall C)(C \in \rho A \rightarrow C \in \rho(A \cup B))$

*reduce
Which proof procedure? (universal) *eliminate
Which symbols? (∈) *$ ∈
What reducer? (theorem) *$theorem
Number or Name (powerset) *$powerset
Occurrences (all) *$all

[The student directs the program to eliminate the two occurrences of '∈' using the theorem of powerset.]

---

Derive: $\rho A \cup \rho B \subseteq \rho(A \cup B)$

Show: (G2) $\rho A \subseteq \rho(A \cup B)$ and $\rho B \subseteq \rho(A \cup B)$

Show: (G3) $\rho A \subseteq \rho(A \cup B)$

Show: (G5) $(\forall C)(C \in \rho A \rightarrow C \in \rho(A \cup B))$

Show: (G6) $(\forall C)(C \subseteq A \rightarrow C \subseteq A \cup B)$

*<u>boo</u>le (1) *<u>^G</u> $(\forall C)(C \subseteq A \rightarrow C \subseteq A \cup B)$

1 replace using th. powerset
(2) $(\forall C)(C \in \rho A \rightarrow C \in \rho(A \cup B))$
2 Df. subset
(3) $\rho A \subseteq \rho(A \cup B)$

[The current goal is an obvious truth of boolean algebra. With the notion of the current goal the student can get the formula corresponding to the current goal by typing '<u>^G</u>'. The '<u>^G</u>' feature greatly reduces the amount of typing and hence the number of typing mistakes. When the BOOLE procedure accepts the current goal, the procedures of the theorem prover work back through the previous goals. Additional lines are generated to satisfy previously unsatisfied goals as their subgoals become satisfied. Line (2) was automatically generated by the program because its only subgoal had just been satisfied by line (1). Similarly line (3) was automatically generated when (2) was used to satisfy its only subgoal.]

---

Derive: $\rho A \cup \rho B \subseteq \rho(A \cup B)$

Show: (G2) $\rho A \subseteq \rho(A \cup B)$ and $\rho B \subseteq \rho(A \cup B)$

Show: (G4) $\rho B \subseteq \rho(A \cup B)$

*<u>reduce</u>
Which proof procedure? (definition) *<u>$</u>definition
Number or Name (subset) *<u>$</u>subset

[The current goal is now the second conjunct from the original goal. The student is about to do this proof in the same manner as for the first conjunct. The next two frames show the similarity of the second sub-proof to the first.]

---

Derive: $\rho A \cup \rho B \subseteq \rho(A \cup B)$

Show: (G2) $\rho A \subseteq \rho(A \cup B)$ and $\rho B \subseteq \rho(A\ B)$

Show: (G4) $\rho B \subseteq \rho(A \cup B)$

Show: (G7) $(\forall C)(C \in \rho B \rightarrow C \in \rho(A \cup B))$

*reduce
Which proof procedure? (universal) *eliminate
Which symbols? ($\in$) *\$ $\in$
What reducer? (theorem) *\$theorem
Number or Name (powerset) *\$powerset
Occurrences (all) *\$all

---

Derive: $\rho A \cup \rho B \subseteq \rho(A \cup B)$

Show: (G2) $\rho A \subseteq \rho(A \cup B)$ and $\rho B \subseteq \rho(A\ B)$

Show: (G4) $\rho B \subseteq \rho(A \cup B)$

Show: (G7) $(\forall C)(C \in \rho B \rightarrow C \in \rho(A \cup B))$

Show: (G8) $(\forall C)(C \subseteq B \rightarrow C \subseteq A \cup B)$

*boole (4) *^G $(\forall C)(C \subseteq B \rightarrow C \subseteq A \cup B)$

4 replace using th. powerset
(5) $(\forall C)(C \in \rho B \rightarrow C \in \rho(A \cup B))$
5 Df. subset
(6) $\rho B \subseteq \rho(A \cup B)$
3,6fc (7) $\rho A \subseteq \rho(A \cup B)$ and $\rho B \subseteq \rho(A \cup B)$
7 Th. 2.8.2
(8) $\rho A \cup \rho B \subseteq \rho(A \cup B)$

Derive: $\rho A \cup \rho B \subseteq \rho(A \cup B)$

boole (1) $(\forall C)(C \subseteq A \rightarrow C \subseteq A \cup B)$
1 replace using th. powerset
(2) $(\forall C)(C \in \rho A \rightarrow C \in \rho(A \cup B))$
2 Df. subset
(3) $\rho A \subseteq \rho(A \cup B)$
boole (4) $(\forall C)(C \subseteq B \rightarrow C \subseteq A \cup B)$
4 replace using th. powerset
(5) $(\forall C)(C \in \rho B \rightarrow C \in \rho(A \cup B))$
5 Df. subset
(6) $\rho B \subseteq \rho(A \cup B)$
3,6fc (7) $\rho A \subseteq \rho(A \cup B)$ and $\rho B \subseteq \rho(A \cup B)$
7 Th. 2.8.2
(8) $\rho A \cup \rho B \subseteq \rho(A \cup B)$

Use QED to finish your proof.

[All the subgoals have been satisfied and the proof is now conceptually complete. The program produces the completed proof and waits for the student to type QED. This allows the student to review the entire proof.]

### 2.2.2 Proposed research on the heuristic natural deduction theorem prover

In the coming year, we will refine and extend the heuristic natural deduction theorem prover. Obvious improvements can be made by extending the prover to accept complex proof procedures composed of simpler ones. One such procedure suggested by the example above is one that allows students to specify that the proof of a subgoal is `similar' (in the sense of using the same proof procedures) to the proof of a previous subgoal. Mechanisms will be added to facilitate proof procedures which operate by letting the prover run in an automatic mode, stopping only at preselected points -- for example, those in which no default is available, or those which involve a particular reduction.

We will also explore the possibility of substantially more sophisticated procedures. One example is a procedure which can choose the terms with which to prove existential formulas. Picking terms with desired characteristics is often the crux of a proof. Another example is a procedure that will make intelligent decisions about when to do proofs by cases, and what those cases should be. Another sophisticated procedure would allow one to do proofs in a manner similar to previously recorded proofs. This is a particularly important procedure, since there certainly will be proof procedures which can be demonstrated by clear examples, yet which lie beyond or outside the scope of our heuristic prover.

### 2.2.3 Evaluation of the heuristic natural deduction theorem prover

Evaluation of the interactive theorem prover in the third year of the grant will focus on student use. Where applicable, standard statistical tests (such as chi-square) will be used to analyze the data collected and evaluate hypotheses on student use of the theorem prover. If warranted by the data, more sophisticated (e.g., Markov) models of student behavior will be tested.

One important aspect of student use of the interactive prover is how often they use its more powerful features. For individual students, we want to know if use of powerful features increases, decreases or remains constant as they progress through the course. We will also examine the kind of use that is made of the prover: is it relatively uniform or dominated by individual differences with respect to amount of use or particular instances of use. Use that can be related to structural features of the theorems or proofs will be extremely valuable in evaluating the procedures used, and suggesting improvements.

Another broad area of concern is whether or not students are learning and using higher-level strategies and algorithms in completing their proofs. We will study this in part by including a facility for students to type sequences of commands as a single command. The use made of such a facility, and especially increases in such use, will be an indication of whether students are using and learning complex proof strategies.

## 2.3 Student aids

Research in this area is concentrated on developing procedures for providing guidance to students attempting to solve problems. Again we quote from the proposed research section of the original proposal (page 40):

> Essentially the same proof procedures and strategies used in the interactive theorem prover can be used by students to express plans for their proofs and also to experiment with finding proofs by successive reductions. Such thinking out of possible approaches to the proof is, of course, in keeping with standard practice and providing facilities to aid this process would considerably upgrade the instructional capability of the system. Guidance will be provided the student by examining the collection of proof procedures and strategies to find those applicable to his current partial proof.
>
> In the logic course guidance will be provided students by using a theorem prover to find continuations of their partially completed proofs.

In later stages of the research we will analyze methods for informally describing the continuations to determine those most effective pedagogically.

The interactive theorem prover as described in Section 2.2 makes it possible to provide several forms of guidance to the student. By comparing a student's current goals and assumptions with the consequents and antecedents of various theorems, reasonable alternatives for continuing the proof can be described in a fairly complete manner. The prover then provides guidance by suggesting different default reductions based upon these alternatives.

Other forms of guidance will be derived from the relationship between the general proof strategies of the theorem prover and heuristics which generate departures from those strategies. Of particular interest will be higher-level algorithms involving complex sequences of inference procedures (especially reductions). It is in such structural features that further guidance for the student can most profitably be rooted.

It should be noted that the interactive theorem prover is in itself a very effective (albeit incomplete) guidance mechanism. That is, the theorem prover with default reductions is generally effective in generating proofs, and presenting these defaults constitutes guidance to the student. Comparisons between proofs completed with and without additional guidance beyond the default reductions will therefore be of great interest. We will compare these student proofs on the basis of their similarity to traditional proofs, and on the basis of objective data on student proofs (see Section 2.1).

## 2.4 More Powerful Inference Procedures

Research in this area is concentrated on developing 'natural' inference procedures. We consider inference procedures to be natural when they justify the inferential steps actually made in standard mathematical practice, and, they are easy for students to understand and use. See pages 40 and 51-55 of the original proposal (Suppes (1977)) for fuller discussion of natural inference procedures. Some simple examples of natural inference procedures are the HYPOTHESIS and RAA (reductio ad absurdum) procedures familiar from standard mathematical practice and common logical systems. A more complex natural inference procedure is the IMPLIES procedure. It is used to derive results that follow by applying a previous result or definition. IMPLIES is described on pages 19-20 and 53-54 of the original proposal.

During the second year we added a new high-level natural inference procedure ESTABLISH to the EXCHECK system. It replaces the old VERIFY procedure which was described in the original proposal on pages 16-19 and 51-53. The fundamental difference between ESTABLISH and VERIFY is that ESTABLISH can be used to derive results that are consequences of prior results <u>in the theory under consideration</u> while VERIFY only can be

used to derive results that are logical consequences of prior results. In particular, in the set theory course, ESTABLISH can be used to decide simple set theoretic consequence while VERIFY is restricted to logical consequence. The examples in Figure 4 and Figure 5 illustrate the distinction between the two procedures.

```
        (i)  B ⊆ A
        (j)  A ⊆ B

*i,jverify *B = A
Using *axiom (Number or Name) *extentionality
Using *definition (Number or Name) *subset
Using *ok

i,j Verify  Using: Ax. Extensionality, Df. Subset
        (4)  B = A
```

Figure 4. A simple inference using VERIFY.

```
        (i)  B ⊆ A
        (j)  A ⊆ B

*i,jestablish *B = A
i,j Establish
        (4)  B = A
```

Figure 5. The same inference using ESTABLISH.

Because `A = B′ is a set-theoretic consequence but not a logical consequence of `A  B & B  A′ students using procedures that are restricted to logical consequence (such as VERIFY) must cite sufficient support to reduce the inference to a purely logical inference. In the example in Figure 4 the student must cite the axiom of extensionality and the definition of subset in order to use VERIFY. In Figure 5 the student need not cite any prior results or definitions when using ESTABLISH. Thus, ESTABLISH is considerably easier to use than VERIFY because students can directly express well understood set theoretic inferences without first analyzing them into purely logical inferences. Such analysis not only disrupts the student′s concentration, but also is difficult to do. Even the most experienced logicians and mathematicians (much less students) have difficulty ferreting out all the axioms, definitions, and theorems needed to reduce inferences within a theory to purely logical inferences.

The domain of ESTABLISH is theorems in elementary set theory but we do not expect it to prove all the theorems in the set theory course since it was designed to handle only the simple set theoretic inferences and theorems that occur in the course of larger proofs. It is difficult to give a good account of the scope of ESTABLISH. The most rigorous approach is to provide a general characterization of simple set

theoretic theorems and prove a completeness theorem stating that a formula is provable by ESTABLISH just in case it is a simple set theoretic theorem according to the characterization. There are two problems with this approach. The first is that there is no such characterization known to us (or, at the time of this writing, to anyone else), and, the second is that the procedures used by ESTABLISH are rather complex and any proof based upon induction over those procedures would also be rather complex.

We informally characterize the range of ESTABLISH as those theorems which are regarded as ‘simple’ by any standard text, e.g., Suppes (1960). More precisely, our set theory course (based upon Suppes’ book) contains approximately 600 theorems. If by ‘simple’ we mean the earlier theorems (the first third) then the result is that ESTABLISH will prove about 85% of those first 200 theorems. It should be noted that the equivalence between simple and early is rather rough. Some of the early theorems (e.g., the Schroeder-Bernstein theorem) are not simple. We give below (Figure 6) some examples of theorems that ESTABLISH can prove. ESTABLISH does more than simply prove theorems of boolean algebra. Note that none of the theorems below is a theorem of boolean algebra. In the following: ‘ ’ is used for power set, ‘ ’ is used for cartesian product, and ‘ ’ is used for both union and generalized union.

1) $\langle x,y\rangle = \langle u,v\rangle$ iff $x = u$ and $y = v$

2) If every element of B is a subset of C then $\bigcup B \subseteq C$

3) $A \subseteq B$ iff $\rho A \subseteq \rho B$

4) $A \times B = B \times A$ iff A is empty or B is empty or $A = B$

Figure 6. Some examples of theorems that ESTABLISH can prove.

The actual mechanisms used in ESTABLISH are too complex to fully describe here but the following explanation should give a good sense of the general process. ESTABLISH is based upon natural deduction heuristics and reduction procedures that simplify an inference and then use resolution or decision procedures as appropriate to attempt to decide the simplified formulas. During this reduction or simplification the needed axioms, definitions, or theorems are selected by the reduction procedures. ESTABLISH uses all the standard kinds of reductions, e.g., expanding defined notation.

The main step in ESTABLISH is the reduction or simplification process. Unless this is done with considerable care the result is likely to be an enormous formula in primitive notation. ESTABLISH controls the size of the resulting formulas by doing reductions in a particular order, by using proof strategies (complex sequences of reductions), by using special auxiliary algorithms to determine set theoretic truth or falsity where possible, by using special algorithms like TEQ and BOOLE where appropriate, and by using information from the

student about what methods should be used. The order of reductions was partially determined by experimentation on the set theory list of theorems.

During the third year of the grant, we will continue to study the relationship between traditional (textbook and classroom) proofs, and EXCHECK proofs. We expect to continue developing and improving the inference procedures of EXCHECK on the basis of those studies. We will be particularly concerned with identifying the structural similarities and differences between traditional proofs and those produced in EXCHECK, both by proctors and students. The structural information will be used to develop further modification and extension of powerful inference procedures.

## 2.5 Elementary exercises and dialogs

The work described is an extension of the work on ELFS described on pages 3-4 of the first annual report. It falls under both the Student Aids and More Powerful Inference Procedures sections of the original proposal.

A great many exercises in elementary mathematics take the form of the student having to specify finite objects satisfying given conditions. An important use of such exercises is in providing counterexamples to arguments (or statements in a theory).

For a large class of these exercises there are programs that will generate a tree of formulas from the original statement of the problem. We call such trees "verification trees". Essentially, the verification tree for a problem constitutes a reduction of the original (usually not directly verifiable) condition to a collection of directly verifiable conditions (the formulas at the leaves). These trees have the property that the failure of the formula at a node in the tree explains the failure of formulas at any of its ancestors.

For example, given the finite sets A and B the student might be asked to give an injection F from A into B, i.e., a function F that is one-to-one. If the student forgets to specify a value for an element of A, the first response to the student is that the domain of F isn't A. The student can then ask: " Why?". The program answers (going towards the leaves) that it is because A is not a subset of the domain of F. If the student again asks "Why" the program will give a specific element in A which is not in B. The program can give a specific counterexample since the routines that evaluate the formulas at the leaves provide counterexamples when those formulas fail.

The student can ask other questions besides "why" after an error. At the first response by the program ("The domain of F isn't A") the student can say: "So?" . The program then moves one step towards the root (the original statement of the conditions) and says: "Then F is not a map from A into B." The student can again say: "So?", to which the program responds: "F is not an injection from A to B."

Below we present the typescript of an actual interaction between a student and the exercise system. The problem is the one we have just discussed: to give an injection F from A into B.

A = {1, 2, 3}        B = {4, 5, 6, 7}

Give an injection F from A into B.

Specify: the function F

Let: F = {<1,4>, <2,5>}

F = {<1,4>,<2,5>}

Let: ok

[Here the program is called to evaluate the student's response. Its initial response is:]

Wrong, since it is not true that Dom(F) = A

[The student has forgotten to include 3--an element of the domain A--in specifying the function F. ]

[The student now asks:]

: why

[The program responds:]

Because it is not true that $A \subset \mathrm{dom}(F)$

[The dialogue continues until the student understands what his error was and why it led to an incorrect solution to the original problem.]

: why

Because it is not true that $(\forall x)[x \in A \rightarrow x \in \mathrm{dom}(F)]$

: why

Because $3 \in A$ and not $(\exists y)$ $\langle 3,y\rangle \in F$

[Having seen the reasons why his response was wrong the student now investigates the set-theoretic consequences of his error.]

: so

it is not true that $(\forall x)[x \in A \rightarrow x \in \mathrm{dom}(F)]$

: so

it is not true that $A \subset \mathrm{dom}(F)$

: so

it is not true that Dom(F) = A

: so

it is not true that F: A → B

: so

it is not true that F: A inj B

: so

The original problem was to show that F: A inj B

As the example shows, the deductive chain of reasons and consequences generated by the "verification tree" gives the exercise system the ability to state reasons for its assertions when it is asked "why", and, to state the set-theoretic consequences of its assertions when "so" is demanded. The highly structured information in the verification tree prov the semantic base for a dialog with the student in which the pr can explain to the student what is wrong with the answer. It s ou be noted that more complex forms of explanation are available. In particular the program could have said at the beginning that: "Because 3 is not given a value by F, the domain of F is not A and hence F is not a function from A to B."

In the coming year, we will implement a number of exercises similar to the example shown here, and will explore ways to refine and extend their use. The exercises implemented will range from such simple problems as specifying different types of functions and relations to complex problems such as those dealing with boolean algebras. Many of these exercises will be incorporated as help lessons, because they provide for a highly focused analysis of individual concepts. These exercises will also be refined to provide an optional analysis of correct answers, and to generate correct examples as hints. We will also experiment with algorithms to find the best explanation for a student's error by choosing the appropriate node in the verification tree from which to begin the dialog.

## 3 Audio and Prosodic Features

The work on Audio and Prosodic Features is proceeding largely as outlined on page 83 of the original project proposal (Suppes (1977)). We quote from that page:

> ... the syntactic parser and contour-generation producers would be reimplemented for efficient on-line generation of synthesized speech.
>
> In the second year, we would have one operational remote MINI-MISS machine ..., continue to implement new LPC techniques, and simulate the concatenation of affixes to root words. We would continue to study the relation of the quality of the synthesized speech and the strain on the listener in understanding the speech. We would add the semantic analysis and story-view analysis to the total prosodic analysis and continue improvements to the lexicon.

The implementation and testing of the MINI-MISS machine has been postponed in favor of additional testing of data compression techniques in order to insure that the machine produces the best quality speech, utilizing the most sophisticated and up-to-date techniques and components possible. A MINI-MISS machine could be built now, but it could not be built to utilize the expected data reduction improvements without resorting to an overly complex design configuration to accommodate the different possible reduction methods. We discuss this issue in more depth below, Section 3.1.

In the first year of the grant we initiated work on the concatenation of affixes to root words, which had immediate application to the maintenance of curriculum. This necessitated delaying work on the on-line generation of prosodic features until the current year. A fuller discussion of the schedule change is presented in the first annual report, Blaine et al (1978). A procedure providing on-line generation capability has been implemented. We describe this procedure below in Section 3.2.3. Evaluation of this procedure will proceed in the next year of the grant. We implemented a simple model of the prosodics of a "story-view" of the structure of text and will continue to develop this model for implementation into the prosodic generation for the curriculums. We incorporated the analysis of some semantic features into the syntactic analysis and into the parameter generation routines instead of constructing a separate semantic component. These topics are discussed below in Section 3.2.1 and Section 3.2.2.

We evaluated different aspects of the word concatenation algorithm in a rating experiment which is described below, Section 3.3. Evaluation of the effectiveness of audio messages in the course in Elementary Logic at Stanford is also proceeding.

### 3.1 Mini-Miss and data rate studies

We proposed to build and test a Mini-MISS machine during this year of the grant but we have decided to defer the actual construction of the machine pending further research on data compression and interpolation of sound data. Construction of the Mini-MISS machine is within our capability at any time we chose to implement the current best techniques in compression and interpolation. However, we foresee substantial improvements to these techniques in the near future whose incorporation will make Mini-MISS a considerably more useful machine. We have chosen instead to concentrate during this grant year on further research and experimentation to increase the degree of compression and the usefulness of interpolation for such a machine. We are attempting, during this grant, to reduce the transmission and storage bit rate from our current 9,000 bits per second to a rate of 3,000 to 4,000 bits per second. This attempt will be brought to a close in the third grant year with the best compression we can achieve. At that time, we will proceed to construction and testing of a Mini-MISS machine employing this reduction. While other LP systems (e.g., Texas Instruments "Speak and Spell", see below Section 3.1.4) have lower transmission rates than we are envisioning, the usefulness of Mini-MISS is predicated on maintaining higher quality speech (necessitating higher data rates) than these other systems are concerned with achieving.

In addition to the primary advantage of rate reduction for the Mini-MISS there are also two secondary considerations which impel us to further research on compression and interpolation, rather than proceeding to construction of the machine. A successful interpolation algorithm will also enable us to perform better word formation through affix concatenation and also to smooth inter-word junctures to sound more continuous. The second consideration in deferring the building of the Mini-MISS is that micro-chips for parts of the speech synthesis algorithm may become commercially available in the near future and these chips will simplify the design of the Mini-MISS machine.

#### 3.1.1 Design considerations for the Mini-MISS machine

We list here our basic design considerations for the Mini-MISS machine. Following the list is a discussion of these criteria in more depth. Criteria 1, 2 and 5 were integral to the design of the original MISS machine, the third (data rate) was deliberately not in the original design and the fourth (interpolation) will be implemented in the MISS machine as a way of perfecting the design for the Mini-MISS.

1) It must be capable of synthesizing a Linear Predictive (LP) encoding of speech in real time.

2) It must be capable of performing prosodic manipulations for sentence formation by word concatenation.

3) It must have a sufficiently low data

transmission rate, so that the speech data together with curriculum display commands can be sent to it over a (leased) phone line, i.e., at a combined data rate of approximately 4800 bits per second.

4) It must be able to perform interpolation of some format of compressed LP coefficients: reflection, area, log-area ratio or formant.

5) It should be micro-coded but should not contain more elaborate hardware facilities than are needed for satisfying the other design criteria.

Linear Prediction has proven itself (see Sanders and Laddaga (1976) and Laddaga and Sanders (1977)) as a high quality, efficient technique for speech synthesis. The manipulation of fundamental frequency, duration and loudness when concatenating words to form phrases and sentences imparts needed naturalness to the process of generating audio messages for a curriculum. While recording entire phrases provides more natural sounding speech, the requirements for storage and recording of new phrases make this approach too costly and time consuming. Also there is less flexibility in the design, production, maintenance and revision of curriculum materials using recorded phrases, due to the high turn-around time required for recording the phrases. These two criteria, the use of LP and the prosodic capability, are critical to the usefulness of the Mini-MISS machine.

The importance of a low transmission rate is that it allows a remote student terminal to access an audio lesson without requiring a second phone line since both audio and display commands can be sent over the same phone line. If a higher bit rate for the audio messages was used, a second phone line would be needed for remote access to transmit the audio messages. In addition to the transmission rate, a low bit rate permits less expensive storage of speech data at remote sites, thus providing the potential for more extensive use of the audio capability. In order to provide the lower transmission rate, the Mini-MISS must be capable of interpolating LP parameters with little degradation of quality. Interpolation is necessary because only a small decrease in the current bit rate could be accomplished by using a more elaborate coding of the parameters to optimize the information content of the coding on a frame-by-frame basis. We discuss interpolation below, Section 3.1.2.

Our final criterion is that the Mini-MISS be micro-coded but not "over-designed." We could utilize expensive and sophisticated micro-processors to build a machine that would have more capability than any single design for interpolation of LP coefficients with prosodic capability could require. This would be a misplaced effort, since such a machine would be largely unused and excessively expensive. We intend to design appropriate components for a particular synthesis (de-compression/interpolation) algorithm. In particular, we are conducting an experiment, described below, Section 3.1.5, to determine the parameters and utility of a specific interpolation method. The use of micro-coding is still important since this decreases development time

and cost, and, provides flexibility in adding improvements within the chosen technique.

### 3.1.2 Data compression and interpolation

We are investigating the use of interpolation for data compression of LP coded speech sounds. Interpolation is a method to reduce the redundancy of information contained in LP analyzed speech across long sequences of data points. It is thus an approximation to the original LP representation, but one that can preserve the perceptually important features of that representation. Section 3.1.5 describes our experiments investigating the relationship between the parameters of interpolation and perception of speech quality. The importance of determining the form of interpolation to the design of the Mini-MISS machine is discussed below, Section 3.1.3.

There are several representations for LP coding of speech which are mathematically equivalent (See Makhoul (1975) for a general introduction to LP, and Levine and Sanders (1978) for a discussion of the different representations.). Interpolation of LP coded speech is dependent on the representation of the LP "transfer function" (Makhoul (1975)) which is shown in equation 1.

$$A(z) = \frac{G}{1 - \sum_{i=1}^{p} \left[ a_i * z^{-i} \right]} \qquad (1)$$

In equation 1, A(z) is the transfer function, G is the amplitude of the original signal, p is order of the polynomial (in our case usually 12), the a's are the constant coefficients of the function, and z is a formal parameter. Equation 2 gives the form of the transfer function in terms

$$A(z) = \frac{G}{\mathrm{prod}_{i=1}^{p} \left[ 1 - R_i * z^{-1} \right]} \qquad (2)$$

of it complex roots, $R_i$, which occur in complex conjugate pairs. Each of the (complex) $R_i$, can be represented, as in equation 3, in polar coordinate form as a modulus, r, and an angle, $\sigma$.

$$R_i = r * e^{j*\sigma} \qquad [j=\mathrm{sqrt}(-1)] \qquad (3)$$

$$F_i = \frac{\sigma}{2\pi} \qquad (4)$$

$$B_i = \frac{-1}{2\pi} * \log_e(r) \qquad (5)$$

The "pseudo-formant" frequencies, $F_i$, and bandwidths, $B_i$, for each complex conjugate pair is computed as in equations 4 and 5 where the frequency and bandwidths are numerically expressed as a fraction of the sampling frequency. Since the LP coding is itself an approximation to the original speech signal, the $F_i$ and $B_i$ are not necessarily the actual formants of the signal, hence the term "pseudo-formants". We will return to our method of assigning the pseudo-formants obtained from the LP coding to particular formant slots. Since the order of the LP model is even, real roots also occur in pairs and provide gross spectral shaping. They will also be considered pseudo-formants here.

The transformation to pseudo-formants from other LP representations (such as reflection coefficients) is exact and invertable. The purpose of doing the conversion is that the pseudo-formant frequencies are smooth functions of time and thus likely candidates for interpolation techniques. The bandwidths due to the vocal tract are also quite smooth functions of time but LP analysis is sensitive to small positional variations. However, errors made in the estimate of the bandwidths are less important perceptually than errors in estimating the formant frequencies. This relative unimportance to perception makes it feasible to interpolate bandwidths even when they have large variances.

There are several considerations which are consequences of this approach to LP interpolation. It is critical to have an accurate and robust polynomial root finder in order to deal with the particular polynomials that result from the LP model. Our original root finder was modified to yield reliable results by saturating arithmetic overflow and underflow. It is also critical to shrink the unit circle in the domain of the polynomial to provide a convergent solution for the initial affricate /dj/ as in `jack' (and similar sounds). A convergent solution of the polynomial would not otherwise be available.

A further consideration for interpolation is numerical representation of real roots of the LP polynomial. While a pair of broad band complex poles can meet on the real axis with little effect on the frequency response, our present representation of the real poles is discontinuous at the point of approach to the real axis. We are proceeding to work on representing these poles in a way that is suitable for interpolation.

The other major consideration for using the formant and bandwidth representation of LP is that a polynomial root solver does not order the roots in any natural way. This fact discouraged previous researchers from considering formant frequencies and bandwidths as a practical method of encoding the LP model. We have developed a number of (automatic) heuristic constraints which order pole pairs so that formants are numbered in an appropriate manner. The heuristic can overrule the simple frequency ordering to preserve the smoothness of the first 4 (and sometimes 5) pseudo-formants. Any resulting broad band pole pairs are assigned to the highest available pseudo-formants after the real pole pair (if it exists) is assigned to the 6th pseudo-formant. Unvoiced speech regions are analyzed with fewer poles (3 or 4) than voiced regions, so the extra (zero) poles are assigned in a heuristic manner.

This approach to ordering the formants yields a great deal of smoothness in formants 1 through 5. Thus, they are well suited to linear interpolation. Formant 6 concentrates much of the frame-to-frame variability in the LP coefficients, and so is poorly represented by linear interpolation. This situation is somewhat paradoxical because the broad band pole pair is the least perceptually significant and a simple representation should be sufficient. Efforts are underway to simplify the representation of pseudo-formant 6.

### 3.1.3 Interpolation and the design of Mini-MISS

The design of the Mini-MISS machine will critically depend on the arithmetic requirements of the interpolation and data compression methods we adopt. The current MISS machine has a humming machine (Levine and Sanders (1978)), which is essentially a general purpose microcomputer whose function is to manage the speech data being transmitted from the PDP-10 and to perform the prosodic calculations. In the Mini-MISS the humming machine would also have to do the interpolation calculations and whatever coefficient transformations are necessary for those calculations, followed by a final conversion to LP reflection coefficients for synthesis by the digital filter part of the machine.

One example of how the design of the humming machine depends on the compression methods used is the question of how much special purpose processor power is required for interpolation. Different special purpose sub-components may be needed depending on which representation of the LP coefficients (e.g., formant and bandwidth) is used for interpolation since that choice will determine the arithmetic complexity of converting from this representation to reflection coefficients for digital filtering. For example, some conversions would require a fast divider while others would not.

In addition to the important data rate reduction which a good interpolation method will provide, it will also be useful for affix concatenation and word junction smoothing. We are investigating word formation by concatenating affixes to word stems as part of this grant. Important work on affixing was done during the first year of the grant (see Blaine et al (1978)). A major improvement will result from the availability of interpolation to smooth the boundary of the stem and affix when they are concatenated. One of the conclusions of the experiment rating relative unnaturalness of word concatenated utterances (reported below, Section 3.3.1) was that the use of individual words contributed the most to the overall unnaturalness of concatenated speech. Here again, interpolation can help to overcome the unnaturalness of word junctures and provide smoother sounding utterances.

### 3.1.4 Commercial availability of speech synthesis components

There are currently two other speech synthesis units which may become commercially available in the near future. Telesensory Systems Inc. (TSI) is developing a pair of chips which will perform the digital filter portion of the Mini-MISS. As mentioned above, the Mini-MISS will contain a filter section which is fed hy the humming machine. If the filter chips are available from TSI in time to be included in our final plan for the Mini-MISS, including it will simplify the design. Since the current MISS microcoded filter is reasonably adequate for Mini-MISS as well the availability of the TSI chips is not critical to the successful completion of the grant.

Texas Instruments has recently begun marketing "Speak and Spell", which is a device using LP coded speech to speak words from a limited vocabulary and spell them out loud. This device is oriented towards drilling spelling at low cost. While the technological sophistication of the synthesis chip in "Speak and Spell" is considerable, it has internal hardware constraints which make it unsuitable for the quality of audio synthesis which is necessary for our CAI applications. Furthermore, the chip is designed in such a way that these constraints can not be overcome by incorporating the chip into a larger synthesis system.

### 3.1.5 Experimental design for compression and interpolation experiment

We stated above (Section 3.1.1) that one design criterion for Mini-MISS was the ability to keep speech quality high at a data rate of 4,000 bits per second (or lower). We are conducting an experiment to test the usefulness and parameters of linear interpolation of a 'formant and bandwidth' representation of LP coefficients as a method for reducing the data transmission rate. An experiment using the judgements of human subjects is necessary to determine the parameters of interpolation and thereby to establish the usefulness of the method. Once the experiment is completed the design for the micro-coding of Mini-MISS can be completed and the hardware construction will quickly follow.

An approximation of real data by a simple smooth contour, such as linear interpolation of LP coded speech, is crucially dependent on how the error between the smooth contour and the real data is measured. Current practice in LP interpolation is to define the error as a function of the spectral difference between the model and the real data. A problem for this measure is that it can easily lead to perceptually inconsistent results. There is much disagreement between spectral differences and perception since the the speech perception space is not necessarily convex nor is the difference measure necessarily Euclidean. Perceptual distance limens for formant frequency and formant bandwidth correlate poorly with their corresponding spectral differences. While spectral difference has long been used by the engineering community in the design of audio amplifiers, the spectral differences in that domain are on the order of 1% or less. Much larger differences are necessary to achieve a low frame rate interpolation technique, and at the larger magnitudes the spectral and perceptual differences diverge. Without

using a perceptual model, the spectral difference method can not take advantage of possibly greater interpolations in particular circumstances without risking an unacceptable number of bad predictions.

The experiment we are performing involves a simple perceptual model of speech representation which employs pseudo-formants and bandwidths (see above Section 3.1.2). Present perceptual models are based on very simple phenomena, such as the perception of steady state vowels. The prediction of complex speech phenomena from such models is not at all straight-forward. Our studies during this grant have indicated that development of a good general perceptual model is still quite far off, given the current state of research. As an alternative to a complex model, we are trying to develop partial decision criteria for constraining the utilization and interaction of the simple perceptual models and the general techniques such as spectral distance measures. In the course of our experiments, we will specify some of these decision criteria and examine the quality of resultant interpolated speech.

We have developed a successful pseudo-formant acoustic model that separates smoothly varying LP parameters from the irregular parameters of standard LP analysis. This success allows us to define the error between the smooth contour and the real data in terms of this model. In equation 6,

$$\alpha = (1/n) * \sum_{i=0}^{n-1} \left[ \frac{S_{ti} - a*(ti-t0) - b}{S_{ti}} \right]^2 \qquad (6)$$

is defined as the mean squared fractional difference between the real data and the line which starts at time `t0', at position `b' with slope `a'. $S_{ti}$ is the coefficient (in this particular LP representation) at time ti. In this equation, `a' is the only parameter available for minimizing the perceptual distance between the smooth line and the real data.

$$a = \frac{\displaystyle\sum_{i=1}^{n-1} \left[ \frac{(S_{ti} - b)*(ti-t0)}{S_{ti}} \right]}{\displaystyle\sum_{i=1}^{n-1} \left[ \frac{(ti-t0)^2}{S_{ti}} \right]} \qquad (7)$$

Equation 7 gives `a' as the optimal slope found by setting the derivative of equation 1 (with respect to a) equal to zero. Thus, `a' minimizes $\alpha$.

Although the mean squared fractional error, as given in equation 6, is a tractable specification of an interpolation criterion, simple averaging can lead to unacceptable situations. If the real data is well approximated by a straight line for a long period but then diverges sharply from its previous path, averaging will cause the straight line to continue well past the point of divergence. Equation 8 provides an additional error definition for the interpolation.

$$\beta = \max_{i\_0}^{n-1} \left[ \frac{S_{ti} - a*(ti - t0) - b}{S_{ti}} \right]^2 \qquad (8)$$

We define $\beta$ in equation 8 to be the maximum component of $\alpha$. By using to compute `a′ and using both $\alpha$ and $\beta$ in the decision criteria shown in equation 9,

Decision criteria (reject if):

$$\alpha > \text{threshold}_{\alpha} \text{ or } \beta > \text{threshold}_{\beta} \qquad (9)$$

we can keep both the average and the maximum error within acceptable bounds.

The formant/bandwidth representation of the $S_{ti}$ has four disjoint numerical regions: .5 > S > 0, S = 0, 0 > S > -.25 and -.25 > S > -.5. This provides the interpolation with an additional decision criterion: interpolated pseudo-formants and bandwidths never cross region boundaries.

Another issue that will be examined as part of this experiment is the allocation of the total acceptable approximation errors among the various formants and bandwidths. Not all formants (or bandwidths) will be equally sensitive to errors of interpolation, but without a general perceptual model it is difficult to decide what the best division is. A series of experiments conducted by Flanagan (summarized in Flanagan (1972)) provides a basis for initial allocations to the first and second formant frequencies and bandwidths. Initial allocations to the other parameters have been based on informal listening tests. The experiment we are currently conducting will improve these allocation assignments, which are crucial to a successful interpolation.

## 3.2 Linguistic analysis

We have implemented one simple model of a "story-view" analysis of dialogue text, and we are beginning work on the generation of paragraph prosodic contours using simple models of the "story-view" of the texts

in our curriculum. We have not directly pursued semantic analysis of sentences, but we have increased the sophistication of our syntactic analysis to accomplish many of the goals that would be derived from semantic studies. We have implemented a procedure for adding on-line generation of audio messages in the curriculum.

### 3.2.1 Story-View and paragraph prosodic contours

The DIALOG program generates audio messages from a prepared text file using multiple voices derived from a single recorded vocabulary. It serves as a tool to determine the parameters of a simple model of "story-view" prosodics. While there are many differences between speakers, or even between different utterances by a single speaker (for example, in the phonetic content of utterances), the model we are investigating specifies some constraints which can be implemented in a prosodic generation system such as ours. The constraints which are manipulated in this program isolate the pitch, speed and loudness of each sentence as being capable of defining separate voices. The pitch of a sentence is defined as the maximum fundamental frequency of any word in the sentence; speed is the factor by which the entire sentence is speeded or slowed from the neutral sentence assignment; loudness is the factor by which the loudness contour for the entire sentence is modified. Sentential prosodic generation procedures are left unchanged by manipulation of these constraints in DIALOG, the results of the sentential procedures are simply relativized to the above constrains. The model implemented in DIALOG indicates some perceptually important parameters for determining continuity and discontinuity of discourse and is therefore useful in developing a "story-view" model.

We are beginning to implement and test a simple model of paragraph scope prosodic contours in the curriculum. This model assigns constraints on the sentential prosodic assignment algorithms from limited information derived from the (abstract) structure of paragraphs, and, syntactic and semantic heuristics based on sentence length and explicit semantic markings in the text. The model manipulates the same constraints, pitch, speed and loudness, as the DIALOG program.

### 3.2.2 Semantic and Syntactic Analysis

We proposed to add a semantic analysis to the linguistic analysis routines which provide the prosodic parameters for intonation synthesis. That proposal implied a separation of the semantic and syntactic components of language analysis. We have decided instead to pursue the improvement and expansion of our syntactic and assignment components to include semantic considerations rather than explicitly add semantics to our system. This decision is based on results of our early work to include semantic components.

In the process of adding an explicit semantic component to our analysis we decided to incorporate local, somewhat ad-hoc uses of semantic information into the syntactic analysis to make it more sensitive to semantic and lexical facts. Also the assignment of

prosodic parameters from the syntactic structure of a sentence has been adjusted to make it more sensitive to certain semantic features. The syntactic component handles pronouns in a more complex way, deciding when to consider them as empty anaphors and when as content words. The assignment is now sensitive to these anaphors and also to comma intonation. These adjustments of the syntactic and assignment components have been fairly effective in providing the naturalness we expected to obtain from adding explicit semantics.

### 3.2.3 On-line generation

We have implemented a procedure this year, called BSPEAK, which enables us to generate prosodic contours for sentences which are at least partially syntactically analyzed. This procedure largely answers the need for on-line generation of audio with reasonable intonation. For generating audio messages for proof summarization, a parameterized message, such as "$\alpha$ proves that $\beta$ holds when $\gamma$ is also true" is syntactically analyzed and stored (where $\alpha$, $\beta$ and $\gamma$ are string variable place holders for potentially complex phrases). Substitutions for these variables are also stored in a partially syntactically analyzed form. In the sample parameterized message given here, we could, for example, substitute "theorem 31" for $\alpha$, "the conditional P implies Q" for $\beta$, and, "for all x, x implies $\delta$" for $\gamma$. In this example, $\delta$ is itself a parameterized message; $\delta$ is a string variable, and might be substituted by "S of x".

When the message is generated, substitutions for variable strings are made and the entire partially analyzed sentence is sent to the BSPEAK procedure which efficiently generates prosodic parameters and directs the MISS machine to begin playing the sounds. Using the substitutions suggested above, the sample message would read

"theorem 31 proves that the conditional P implies Q holds when
$\alpha$.........                $\beta$.........................

for all x, x implies S of x is also true."
$\gamma$.........................
$\delta$.....

This sentence would require a relatively large amount of time (on the order of several minutes with a heavily loaded system) for syntactic analysis if it were to be parsed as such, but since each smaller string was analyzed at the time of storage (some with variables) no additional syntactic analysis is necessary. Since BSPEAK uses a context free grammar to generate prosodic parameters, it can infer reasonable parameters for the un-analyzed portions of the message in light of the structure of the message. Therefore, even if one or more of the variables in the example above had not been pre-stored with a syntactic analysis, BSPEAK would still have been able to generate a reasonable set of prosodic parameters for the sentence.

### 3.3 Study of prosodic quality

We have utilized two means of evaluating the quality of prosodically modified word-concatenated utterances. We conducted an experiment, Sanders et al (1978), in which twenty naive subjects rated eight variations of twenty sentences. The variations of the sentences isolated the pitch, duration and loudness components of our prosodic model. The other means of evaluating prosodic quality we used is the modeling of student use of audio in the course, "Introduction to Logic."

#### 3.3.1 Rating experiment

LP encoding separates speech into four component contours: pitch, duration (rate), loudness, and spectrum. There is a generally recognized hierarchy of intonational effects, in which pitch contour has first place, duration next, loudness third. Segmental information is outside the hierarchy. The purpose of this study was to discover, for the current MISS system, the relative contributions to unnaturalness made by these factors, in order to appropriately direct efforts to improve the quality of word concatenated speech.

Twenty sentences were chosen from a collection of sentences used in a course taught at the Institute. They had all been recorded by the speaker who also recorded the lexical words. The sentences selected contained a full range of English phonemes, and represented a variety of syntactic types: declaratives, questions, and imperative sentences. Word boundaries in each recorded sentence were marked as accurately as could be determined by graphic and perceptual means. The sentence texts were parsed by MISS's parser, and the word durations, amplitudes, and pitch contours were calculated. The recorded sentence was then modified to have some artificial (MISS generated) characteristics while retaining some of its original character.

Eight different treatments of each sentence were prepared.

1. The recorded LPC processed sentence (recorded).
2. The recorded sentence with MISS's amplitude modification replacing the original (artificial amplitude).
3. The recorded sentence with MISS's duration assignments replacing the original word durations (artificial duration).
4. The recorded sentence with MISS's pitch contour replacing the original pitch contour (artificial pitch).
5. The recorded sentence with spectral coefficients from individual words as stored in the English dictionary replacing the original coefficients (artificial coefficients).
6. The sentence as produced by lexical word concatenation with prosodic modification of pitch, duration and amplitude (artificial prosody).
7. The recorded sentence with artificial pitch, artificial duration, artificial amplitude, and artificial coefficients replacing the originals (artificial composite).
8. The recorded sentence with a previous version of MISS's

pitch contour replacing the original pitch contour (old pitch).

The subjects were first given a definition of naturalness and were told to rate the sentences they were about to hear from 1 to 9, with 0 and 10 as overflow categories. We chose to use naive subjects to avoid the possibility that previous exposure to such speech might have influenced the experienced listener's notion of naturalness.

The means of each treatment (averaged over sentences and subjects) showed significant and interesting differences. The mean for each treatment was tested against the other means using a t-test. The t-test showed that while the recorded version did not differ significantly from the artificial amplitude version and that the artificial prosody sentences was neither better nor worse than the artificial composite sentences, all the other differences in the means were statistically significant below the .001 level, so that we can order the mean scores in six groups as shown in Figure 7.

| Group | Treatment | Mean | S.D. |
|---|---|---|---|
| 1. | recorded | 6.705 | 1.889 |
| | artificial amplitude | 6.565 | 1.812 |
| 2. | artificial duration | 5.443 | 2.102 |
| 3. | artificial pitch | 4.540 | 2.098 |
| 4. | old pitch | 4.113 | 2.093 |
| 5. | artificial coefficients | 2.320 | 1.742 |
| 6. | artificial composite | 1.840 | 1.372 |
| | artificial prosody | 1.835 | 1.420 |

Figure 7. Means and Standard Deviation of the Naturalness Scores.

These preferences indicate that, as expected, the recorded sentences are more natural than the artificial prosody sentences. More importantly, the significance of the preference ordering shows which artificial elements contribute the most to the perceived unnaturalness of artificial prosody sentences. Substituting artificial spectral coefficients for the original sentence coefficients makes the greatest difference in naturalness of any of the single changes.

### 3.3.2 Analysis of student preferences

The course, "Introduction to Logic", is given three times per year for five units of university credit and taken by approximately 250 Stanford students each year. To test the effectiveness of audio in our curriculum, we offered a choice of display-only and display-with-audio

courseware. We observed the relative naturalness of the prosodic generation by giving students in the logic course either the recorded phrases (LS-mode) or prosodically generated sentences (P-mode) as the audio version of the course and allowing them to use the display-only version if they wished. Students were encouraged to try both versions and to switch between versions whenever they found it helpful. They were informed that their grades in no way depended on which version of the course they used or how often they switched. By statistically measuring and modeling the relative preferences of LS-mode versus display, and P-mode versus display-only (DPY-mode) we estimate the relative naturalness of the two modes of MISS synthesis. Three experiments were conducted to collect data on student preferences. In the first experiment we compared only LS and DPY modes; in the second, we compared LS and P modes via the relative preference for DPY mode; in the third, we allowed a free choice of the three modes.

In the first experiment, students were split into two groups, one beginning in LS-mode, the other in DPY-mode for an initial segment of the course. Both groups were then switched to the alternate mode for a second segment of the course. For these initial segments, the student had no choice over which mode (DPY or LS) was presented in which lesson. Both groups were free to choose either mode at each "log-in" from then on.

The results of the first experiment, shown in Figure 8, indicated that audio was not an overwhelming favorite; only about half the log-ins for the portion of the course in which students had a choice were audio log-ins. The percentage of audio log-ins actually begins higher but decreases as students progress through the course. The percentage of audio chosen was not particularly affected by the amount of forced exposure to audio (3 lessons in Winter '77, 1 lesson in Spring '77) they had.

| Term | Lessons with no choice | With choice | Log-ins (choice) | Percent in audio |
|---|---|---|---|---|
| Winter '77 | 1-6 | 7-18 | 1278 | 49 |
| Spring '77 | 1-2 | 3-20 | 2742 | 48 |

Figure 8. Results of the first audio/display experiment.

The second experiment was very similar to the first, except that half of each of the two groups was using LS-mode when they received audio, while the other half was using P-mode. The results for this experiment, shown in Figure 9, are somewhat clouded by the fact that in the autumn quarter there were errors in the course software and curriculum which affected only the audio versions. By the winter quarter these problems had been mostly corrected for P-mode, but not for LS-mode (hence the poor showing of LS-mode versus P-mode for that quarter). The results of this experiment indicate that P-mode is as acceptable to students as LS-mode, despite the difference in quality. A

| School Term | Lessons Forced | Lessons Choice | Logins w/choice | Percent of Logins | Comment |
|---|---|---|---|---|---|
| Spring '77 | (none) | 21-29 | 889<br>763 | 30 LS-mode<br>26 P-mode | See Figure 8 |
| Autumn '77 | 1-2 | 3-29 | 1759<br>1941 | 20 LS-mode<br>22 P-mode | Errors in audio curric. |
| Winter '78 | 1-2 | 3-29 | 1601<br>1524 | 17 LS-mode<br>34 P-mode | Errors fixed in P-mode only |

Figure 9. Results from LS-mode and P-mode vs. DPY-mode experiment.

significant reason why students favored DPY-mode can be attributed to the errors which existed in the audio version of the curriculum. Results for Spring '78 and Autumn '78 are currently being analyzed.

A third experiment was performed with the first four lessons (approximately four hours) of the logic course over a two week period in autumn 1977, and repeated in 1978. For the four-lesson course, 64 Stanford freshmen were given a three-way choice: DPY-mode, LS-mode and P-mode, with a brief explanation of what the modes involved. These students constitute the only group so far to have a choice between LS-mode and P-modes in audio, in addition to the option of choosing DPY-mode. The results from these two years are summarized in Figure 10. The percentage of audio log-ins remained fairly constant in both years, but the preference for audio, the number of students who logged-in more times in one of the audio modes than in display, was greater in 1978. Similarly, the preference for LS-mode over P-mode, among those whose preference was for audio, was also considerably higher in 1978. P-mode compared respectably in a head-to-head confrontation with LS-mode in this experiment.

| | | | Number of Students | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Preferred | | | Preferred (Audio) | | |
| School year | total log-ins | Percent in Audio | Audio | Dpy | Mixed | LS-Mode | P-Mode | Mixed |
| '77 | 402 | 62 | 25 | 33 | 6 | 18 | 7 | - |
| '78 | 357 | 69 | 43 | 16 | 5 | 36 | 5 | 2 |

Figure 10. Results from three-way free choice experiment.

Another experiment is presently under way (Winter '79) in which the students' tendency to use audio is compared to the extent of previous, forced exposure to audio. For this experiment students have a choice only between P-mode and DPY-mode. The students are divided into four groups:

1) short exposure to P-mode then long exposure to DPY-mode;

2) short exposure to P-mode then medium exposure to DPY-mode;

3) short exposure to DPY-mode then long exposure to P-mode;

4) short exposure to DPY-mode then medium exposure to P-mode.

By "short exposure" we mean approximately a half hour; "medium exposure" is about three hours, and, long exposure is nine hours. These times are mean student times from previous quarters of the course for the lessons where the exposure to P-mode or DPY-mode was forced. This experiment will be repeated for Spring '79 using better prosodic techniques that have been developed during this grant.

# 4 Schedule of Proposed Research

## 4.1 Informal Mathematical Procedures

The schedule given here for completing the research proposed under the informal mathematical procedures portion of the original proposal is a more explicit version of the schedule on page 82 of the original proposal.

We will continue the development of natural inference procedures and will extend the implementation of the interactive theorem prover to handle complex proof procedures. In particular, the basic procedures used in the set theoretic version of ESTABLISH will be refined and extended to enhance its performance on more difficult inferences in set theory. Complex reduction sequences (proof strategies) will be implemented. The finite structure mechanisms including the dialog procedures will be extended and implemented in the logic course.

Sophisticated mechanisms for providing guidance in the set theory course based upon suggesting applicable reductions or strategies will be extended and refined. These mechanisms will also be implemented for the logic course. Guidance based upon re-execution of stored proofs will be implemented in the set theory course to provide guidance for those cases beyond the capability of the other guidance procedures.

A major focus of our work in the third year will be the evaluation of the interactive theorem prover, powerful inference procedures and guidance mechanisms. For more detailed descriptions of the evaluation procedures, see the appropriate sub-sections of Section 2. In addition, work will continue on streamlining and improving the course driver, EXCHECK, to facilitate the student's interaction with the computer. Dissemination of research results will be carried on through continued publications, presentations and conferences.

## 4.2 Audio and Prosodic Features

The schedule for completing research originally proposed for the third year of the grant is substantially the same as that given on page 83 of the original technical proposal. We will implement concatenation algorithms and additional prosodic analysis and generation algorithms. We will continue to study the quality of the synthesized speech as described on pages 65-66 of the original proposal.

In the next year of this grant we will complete the specification of design criteria for the MINI-MISS, including data compression and interpolation algorithms. Important specification information will be

derived from the results of an experiment on interpolation (see Section 3.1.5 above) to be completed in this third year. We will construct the MINI-MISS machine satisfying the design criteria and test it in the field. We will also continue to implement new LPC techniques (see pages 58-59 of the original proposal) and work will continue on improving the algorithm for digitizing individual words (see page 63 of the original proposal).

In addition to the experiment on interpolation, evaluations of the quality of MISS speech production and prosodic assignments will continue. Dissemination of the results of our research will be carried on through continued publications, presentations and conferences. Especially important will be the presentation of our findings on data rate reduction.

# Appendix A

## Dissemination of Research

An important part of the work carried out under this grant has been the dissemination of the results and conclusions of the research done under support of the grant. Papers and presentations have been given at national and regional conferences concerning computers (ACM), education (CECC and ADCIS), and, speech research (ASA). Below we list the publications and presentations of the principal investigator, Patrick Suppes, and the publications and presentations of the other staff members. Some of these publications and presentations are also referenced in the technical sections of this report and appear in the report bibliography. They are included here for completeness.

### Published works of Patrick Suppes

1977, Learning theory for probabilistic automata and register machines, with applications to educational research. H. Spada & W. F. Kempf (Eds.), Structural models of thinking and learning (Proceedings of the 7th IPN-Symposium on Formalized Theories of Thinking and Learning and their Implications for Science Instruction). Bern: Hans Huber Publishers, Pp. 57-79.

1977, A survey of contemporary learning theories. In R. E. Butts & J. Hintikka (Eds.), Foundational problems in the special sciences (Part two of the Proceedings of the Fifth International Congress of Logic, Methodology and Philosophy of Science, London, Ontario, Canada, 1975). Dordrecht: Reidel, Pp. 217-239.

1977, The future of computers in education. In, Computers and the learning society. (Hearings before the Subcommittee on Domestic and International Scientific Planning, Analysis and Cooperation, of the Committee on Science and Technology, U.S. House of Representatives, Ninety-fifth Congress, First Session, October 4, 6, 12, 13, 18, and 27, [No. 47].) Washington: U.S. Government Printing Office, Pp. 548-569.

1978, Impact of research on education: Some case studies. Washington, D.C.: National Academy of Education, 672 pp.

1978, Performance models of undergraduate students on computer-assisted instruction in elementary logic. Instructional Science, 7, 15-35 (with I. Larsen and L. Z. Markosian).

1978, Steps toward a variable-free semantics of attributive adjectives, possessives, and intensifying adverbs. In K. E. Nelson (Ed.), Children's language (Vol. 1). New York: Gardner Press, Pp. 81-115 (with E. Macken).

1978, The role of global psychological models in instructional technology.

In R. Glaser (Ed.), Advances in instructional psychology (Vol. 1). Hillsdale, N. J.: Erlbaum, Pp.229-259 (with E. Macken & M. Zanotti).

1978, The historical path from research and development to operational use of CAI. Educational Technology, (April) 9-12 (with E. Macken).

1978, La informatica en la educacion. Nonotza, Revista de Difusion Cientifica Tecnologica y Cultural 13, IBM de Mexico.

1979, Current trends in computer-assisted instruction. In M. C. Yovits (Ed.), Advances in Computers (Vol. 18), Academic Press, New York.

1979, Variable-free semantics for negations with prosodic variation. In E. Saarinen, R. Hilpinen, I. Niiniluoto, & M. P. Hintikka (Eds.), Essays in honour of Jaakko Hintikka. Dordrecht: Reidel, Pp. 49-59.

## Unpublished works of Patrick Suppes

### 1977

| | |
|---|---|
| August 27 | Some global models of learning and performance, meeting of American Psychological Association, San Francisco |
| November 9 | Computer-assisted Instruction in University-level Mathematics, Department of Mathematics Fall Colloquium Series, San Francisco State University |
| November 19 | Computers in Education, First Western Educational Computing Conference, San Francisco |
| December 12 | Computer-assisted Instruction, Tata Institute for Fundamental Research, Bombay, India |

### 1978

| | |
|---|---|
| January 19 | Research on Computerized Instruction, School of Education, Stanford University |
| January 26 | The Future of Computer-assisted Instruction, International Symposium of Informatics, Mexico City |
| February 22 | Variable-free Semantics for Natural Language, School of Social Sciences of California, Irvine |
| March 6 | Past, Present and Future Educational Technologies, Third World Mathematics Conference, Khartoum, Sudan |
| April 7 | Some Remarks on the Semantics of Natural Language, Language, Mind, and Brain National Interdisciplinary Symposium, Gainesville, Florida |
| April 13 | Variable-free Semantics for Natural Language, Department of Linguistics, University of Texas, Austin, Texas |

| | |
|---|---|
| April 26 | Future Trends in Computer Assisted Instruction, Computer Science Colloquium, University of California, Berkeley |
| May 6 | Computer-assisted Instruction in Community Colleges, Northern California Community College Computer Consortium, Sierra College, Rocklin, California |
| May 20 | Computers and Productivity in Education, Campus Conference, Stanford Alummi Association |
| May 25 | Computer-assisted Instruction at the University Level, California Polytechnic State University, San Luis Obispo |
| July 13 | Logic and Set Theory in Schools, Australian Logic Teachers Association National Conference and Australasian Association for Logic Annual Conference, The University of Queensland |
| July 13 | Analysis of Student Trajectories in CAI Courses, Kelvin Grove College of Advanced Education, Brisbane, Australia |
| July 14 | Computer-based Logic Instruction, A.A.L./A.L.T.A., University of Queensland, St. Lucia, Brisbane, Australia |
| July 14 | The Semantics of Natural Language in Logic Courses, A.A.L./A.L.T.A., University of Queensland |
| July 14 | Computer-assisted Instruction in the Schools and Colleges in the United States, Kelvin Grove College of Advanced Education, Brisbane, Australia |
| July 21 | Variable-free Semantics for Natural Language, Department of Philosophy, The Australian National University, Canberra |
| August 31 | Inductive Logic and Its Applications. International Institute of Philosophy, Dusseldorf, Germany |
| September 23 | Educational Technology and the Future of Education, Teachers College, Columbia University, New York |
| November 28 | Transportability of Curriculums, Digital Users Group, DECUS, U.S. Fall Symposium, San Francisco |
| December 1 | Panel Discussion: Reactions to "Values and Evaluation," presented by Michael Scriven. Far Western Philosophy of Education Society, 27th Annual Meeting, Palo Alto, California |
| December 1 | Teaching Logic and Set Theory by Computer, Logic Colloquium, Group in Logic and the Methodology of Science, University of California, Berkeley |
| December 6 | Panel Discussion: Computer-based Courses. ACM meeting, Washington, D.C. |

1979

February 27 The future of Computers in Education: The Dean's Lecture, Computer-Based Education: Mission of the Future. Meeting of the Association for the Development of Computer-Based Instructional Systems, San Diego, Ca.

## Published works of other project members

1977, Blaine, L. and R. L. Smith, "Intelligent CAI: The role of the curriculum in suggesting computational models of reasoning", Proceedings of the 1977 Annual Conference, Association for Computing Machinery, Seattle.

1978, Blaine, L., A. Levine, R. Laddaga and P. Suppes, Technical problems in implementing university-level computer-assisted instruction in mathematics and science: First annual report, Technical Report No. 293, Institute for Mathematical Studies in the Social Sciences, Stanford University.

1978, Davis, M. and T. Pettit, Using VOCAL: A guide for authors. Technical Report No. 296, Institute for Mathematical Studies in the Social Sciences, Stanford University.

1978, Levine, A. and W. R. Sanders, The MISS speech synthesis system. Technical Report No. 299, Institute for Mathematical Studies in the Social Sciences, Stanford University.

1978, Levine, A. and W. R. Sanders, "The MISS speech synthesis system", Proceedings of the IEEE-ICASSP conference, Washington, D.C.

## Unpublished works of other project members

### 1977

First Western Educational Computing Conference, San Francisco.

Blaine, L. "The Excheck System".

Acoustic Society of America 94th Meeting, Miami, Florida.

Laddaga, R. and W. R. Sanders, "Testing recognition of computer-generated speech with elementary school children".

Levine, A., "Variation in sentence duration due to lexical content, semantic emphasis, and text structure".

### 1978

Western Educational Computing Conference, Anaheim.

Blaine, L. and J. McDonald, "Interactive processing in the EXCHECK system of natural mathematical reasoning."

Blaine, L., M. Davis, S. Lindstrom and R. Laddaga, "Courses taught using the EXCHECK system."

Markosian, L. Z., W. R. Sanders, H. Seropian and J. van Campen, "A language instruction program for Armenian using computer-synthesized speech."

McDonald, J., L. Blaine, J. Marlowe and R. Roberts, "The EXCHECK system."

Laddaga R., J. Marlowe, T. Pettit, and L. Markosian, "The VOCAL system."

Sanders, W. R., A. Levine and R. Laddaga, "The MISS audio system."

Acoustic Society of America, 96th Meeting, Honolulu, Hawaii.

Levine, A. "Musical intervals in the elaboration of underlying tonal melodies".

W. R. Sanders, C. M. Gramlich and A. Levine, "The sensitivity of LPC synthesized speech to the imposition of artificial pitch, duration, loudness and spectral contours".

1979

Computer-Based Education: Mission of the Future
February, 1979 meeting at San Diego of the
Association for the Development of Computer-Based Instructional Systems

W. R. Sanders, A. Levine and R. Laddaga, "The Use of MISS in Computer-Based Instruction".

Other presentations

Levine, A. Six lectures on "Computational linguistics, speech synthesis and CAI," presented at Bar Ilan University, Ben Gurion University, Hebrew University, University of Tel Aviv, the Weitzman Institute; Israel.

49

## Appendix B

### Sample Proof

The following is a sample proof in an elementary set theoretic system which demonstrates features of the interactive heuristic natural deduction theorem prover in addition to those illustrated in Section 2.2. A slightly different method of display has been selected by the student, so that only the immediate goal is printed in any frame. All student input is underlined (the continuation of the command being supplied by the program). The horizontal lines separate different frames (previous material disappears from the screen).

---

Derive: $\neg(A \in B \text{ and } B \in A)$

*reduce
Which proof procedure? (negation)*reductio ad absurdum
Formula to be contradicted (formula)*theorem *regularity

If A is nonempty then $(\exists x)[x \in A \ \& \ (\forall y)(y \in x \rightarrow y \notin A)]$

Specify for A? *{A,B}

[The student decides to use a proof by reductio ad absurdum for this derivation. (This is not the default procedure.) The program asks for a formula to be used as one half of the contradiction; the negation of the given formula will be the other half. An option exists which allows the student to defer proving the contradiction. Since the student has chosen to contradict a theorem, only the negation of the theorem needs to be proved.]

---

Show: (G2) $\neg(\{A,B\} \neq 0 \rightarrow (\exists x)[x \in \{A,B\} \ \& \ (\forall y)(y \in x \ \& \ y \notin \{A,B\})])$

Assume (1) $A \in B$ and $B \in A$

*reduce
Which proof procedure? (negation) *full negation

[The option "negation" takes a negated (complex) goal and produces a non-negated formula, using elementary logic. For example, from $\neg(P \rightarrow Q)$ we get $(P \ \& \ \neg Q)$. The option "full negation" carries out this procedure recursively throughout the formula.]

---

Show: (G3) $\{A,B\} \neq 0 \;\&\; (\forall x)[x \in \{A,B\} \rightarrow (\exists y)(y \in x \;\&\; y \in \{A,B\})]$

Assume (1) $A \in B$ and $B \in A$

*reduce
Which proof procedure? (conjunction) *$conjunction

[The typing of the ESC key alone (represented as $) at any point selects the default (listed in parentheses).]

---

Show: (G4) $\{A,B\} \neq 0$

Assume (1) $A \in B$ and $B \in A$

*establish
(2) *^G $\{A,B\} \neq 0$

[With Establish, the student derives the first conjunct. Typing a control-G (^G) selects the current goal for the formula. When the inference to the current goal is accepted, the program goes on to the next goal which is, in this case, the second of the two conjuncts from the goal (G3).]

---

Show: (G5) $(\forall x)[x \in \{A,B\} \rightarrow (\exists y)(y \in x \;\&\; y \in \{A,B\})]$

Assume (1) $A \in B$ and $B \in A$

*reduce
Which proof procedure? (universal) *$universal
Which variable do you want for x? (x) *$x

[The student has the choice of variable to use in the goal.]

---

Show: (G6) $x \in \{A,B\} \rightarrow (\exists y)(y \in x \;\&\; y \in \{A,B\})$

Assume (1) $A \in B$ and $B \in A$

*reduce
Which proof procedure? (conditional) *$conditional

---

Show: (G7) (∃ y)(y ∈ x & y ∈ {A,B})

Assume (1) A ∈ B and B ∈ A
Assume (3) x ∈ {A,B}

*reduce
Which proof procedure? (existential) *cases

Case 1 Assumption *x = A
Conclusion *^G(∃ y)(y ∈ x & y ∈ {A,B})

Case 2 Assumption *x = B
Conclusion *^G(∃ y)(y ∈ x & y ∈ {A,B})

[The instance of y that would satisfy the goal depends upon what x is. From (3), x may be either A or B. Given this, the student decides to do a proof by cases: showing that (G7) follows no matter whether x is A or x is B.]

---

Show: (G8) x = A   x = B

Assume (1) A ∈ B and B ∈ A
Assume (3) x ∈ {A,B}

*3 establish
(4)^Gx = A ∨ x = B

[The student must first prove the disjunction that justifies breaking the proof into cases.]

---

Case 1:
Show: (G9) (∃ y)(y ∈ x & y ∈ {A,B})

Assume (1) A ∈ B and B ∈ A
Assume (3) x ∈ {A,B}
Assume (5) x = A

*reduce
Which proof procedure? (existential) *$existential
Which variable do you use for y? (y) *B

[In the first case, the student wants to show that if x equals A, then the conclusion follows. The student selects the proper instantiation of the goal to work for, using B as the required term.]

---

Case 1:
Show: (G10) $B \in x$ & $B \in \{A,B\}$

Assume (1) $A \in B$ and $B \in A$
Assume (3) $x \in \{A,B\}$
Assume (5) $x = A$

*<u>1,5 est</u>ablish
(6) *<u>^G</u>B $\in x$ & $B \in \{A,B\}$
6 EG (7) $(\exists y)(y \in x$ & $y \in \{A,B\})$

[The goal is now an obvious truth of set theory, which is proven by ESTABLISH. Once the student proves (6), the goaling machinery can use the inverse of the reduction procedure for existentials, already specified by the student, to satisfy the previous goal. This finishes the first case, where x is A.]

---

Case 2:
Show: (G11) $(\exists y)(y \in x$ & $y \in \{A,B\})$

Assume (1) $A \in B$ and $B \in A$
Assume (3) $x \in \{A,B\}$
Assume (8) $x = B$

*<u>red</u>uce
Which proof procedure? (existential) *<u>$</u>existential Term *<u>A</u>

[Once the first case is finished, the student goes on to the second, where x = B. That case is dealt with in a similar fashion in succeeding frames.]

---

Case 2:
Show: (G12) $A \in x$ & $A \in \{A,B\}$

Assume (1) $A \in B$ and $B \in A$
Assume (3) $x \in \{A,B\}$
Assume (8) $x = B$

*<u>1,7 est</u>ablish
(9) *<u>^G</u>A $\in x$ & $A \in \{A,B\}$
8 EG (10) $(\exists y)(y \in x$ & $y \in \{A,B\})$

---

Derive: $\neg(A \in B \text{ and } B \in A)$

Assume (1) $A \in B$ and $B \in A$
Establish
(2) $\{A,B\} \neq 0$
Assume (3) $x \in \{A,B\}$
3 Establish
(4) $x = A \vee x = B$
6 EG (7) $(\exists y)(y \in x \ \& \ y \in \{A,B\})$
9 EG (10) $(\exists y)(y \in x \ \& \ y \in \{A,B\})$
4,7,10 Cases
(11) $(\exists y)(y \in x \ \& \ y \in \{A,B\})$
3,11 CP (12) $x \in \{A,B\} \rightarrow (\exists y)(y \in x \ \& \ y \in \{A,B\})$
12 UG (13) $(\forall x)[x \in \{A,B\} \rightarrow (\exists y)(y \in x \ \& \ y \in \{A,B\})]$
2,13 FC (14) $\{A,B\} \neq 0 \ \& \ (\forall x)[x \in \{A,B\} \rightarrow (\exists y)(y \in x \ \& \ y \in \{A,B\})]$
14 Full Negation
(15) $(\{A,B\} \neq 0 \rightarrow (\exists x)[x \in \{A,B\} \ \& \ (\forall y)(y \in x \ \& \ y \notin \{A,B\})])$
1,15 Contradiction using Th. Regularity
(16) $\neg(A \in B \text{ and } B \in A)$

Use QED to finish your proof

[Since (G12) is the last unsatisfied goal, the goal machinery can use the inverses of the reduction procedures, already specified by the student, to finish the proof. This is done by generating formulas to satisfy those goals all of whose sub-goals have already been satisfied. Thus, once (9) is inferred using existential generalization, (10) is inferred by the proof by cases. Lines (11) through (14) are inferred using the proof procedures: Conditional, Universal, Conjunction and Full Negation. Finally, the overall goal is derived in a reductio ad absurdum from the contradiction between (14) and the theorem of regularity.]

---

## Appendix C

### Reduction Proof Procedures

The following is a partial list of the proof procedure reductions available in the interactive heuristic natural deduction theorem prover. Section 2.2, above, contains a discussion of proof reductions. The expression to the left of an `|=>' denotes the allowed assumptions and the expression to the right denotes the goal. Hence, in the first example, the upper line reads: prove A  C from +. The whole schema reads: reduce the problem `prove A  C from +' to the subproblem `prove C from +,A'.

```
conditional
          Σ     |=>   A → C
         -----------------
          Σ,A   |=>   C

biconditional
          Σ     |=>   A ↔ B
         -----------------
          Σ     |=>   A → B   ∧   B → A

conjunction
          Σ     |=>   A ∧ B
         -----------------
          Σ     |=>   A
          Σ     |=>   B

disjunction
          Σ     |=>   A1 v  . .  v An
         -----------------
          Σ     |=>   Ai

cases
          Σ         |=>   C
         -----------------
          Σ         |=>   A1 v A2 v . . . v An
          Σ,A1      |=>   C
                 . . .
          Σ,An      |=>   C

universal
          Σ     |=>   (∀ x) P(x)
         ----------------------
          Σ     |=>   P(x)  where x is neither free nor subscripted in Σ

existential
          Σ     |=>   (∃ x) P(x)
         ----------------------
          Σ     |=>   P(t)  where t is any term
```

55

uniqueness

$$\frac{\Sigma \quad \mid\!\Rightarrow \quad (\exists!\, x)\, P(x)}{\Sigma \quad \mid\!\Rightarrow \quad (\exists\, x)\, P(x) \quad \& \quad (\forall\, x, y)(\, P(x) \wedge P(y) \rightarrow x = y\,)}$$

negation

$$\frac{\Sigma \quad \mid\!\Rightarrow \quad \neg\neg A}{\Sigma \quad \mid\!\Rightarrow \quad A}$$

$$\frac{\Sigma \quad \mid\!\Rightarrow \quad \neg(\, A \wedge B\,)}{\Sigma \quad \mid\!\Rightarrow \quad \neg A \vee \neg B}$$

$$\frac{\Sigma \quad \mid\!\Rightarrow \quad \neg(\, A \vee B\,)}{\Sigma \quad \mid\!\Rightarrow \quad \neg A \wedge \neg B}$$

$$\frac{\Sigma \quad \mid\!\Rightarrow \quad \neg(\, A \rightarrow C\,)}{\Sigma \quad \mid\!\Rightarrow \quad A \wedge \neg C}$$

$$\frac{\Sigma \quad \mid\!\Rightarrow \quad \neg(\, A \leftrightarrow B\,)}{\Sigma \quad \mid\!\Rightarrow \quad \neg(\, A \rightarrow B\,) \quad \vee \quad \neg(\, B \rightarrow A\,)}$$

$$\frac{\Sigma \quad \mid\!\Rightarrow \quad \neg(\forall\, x)\, P(x)}{\Sigma \quad \mid\!\Rightarrow \quad (\exists\, x)\, \neg P(x)}$$

$$\frac{\Sigma \quad \mid\!\Rightarrow \quad \neg(\exists\, x)\, P(x)}{\Sigma \quad \mid\!\Rightarrow \quad (\forall\, x)\, \neg P(x)}$$

$$\frac{\Sigma \quad \mid\!\Rightarrow \quad \neg(\exists!\, x)\, P(x)}{\Sigma \quad \mid\!\Rightarrow \quad \neg(\exists\, x)\, P(x) \quad \vee \quad (\exists\, x, y)(\, P(x) \wedge P(y) \wedge x \neq y\,)}$$

reductio ad absurdum

$$\frac{\Sigma \quad \mid\!\Rightarrow \quad C}{\Sigma, \neg C \quad \mid\!\Rightarrow \quad \text{contradiction}}$$

implies

$$
\begin{array}{l}
\Sigma \ \mid\Rightarrow \ C[t] \\
\hline
\Sigma \ \mid\Rightarrow \ (\forall x)(\ A[x] \rightarrow C[x]\ ) \qquad \text{usually definition or theorem} \\
\Sigma \ \mid\Rightarrow \ A[t]
\end{array}
$$

In particular, identities and memberships can be reduced by applying the appropriate theorem or definition, e.g.:

$$
\begin{array}{l}
\Sigma \ \mid\Rightarrow \ A \in \rho B \\
\hline
\Sigma \ \mid\Rightarrow \ (\forall x)(\ x \in \rho B \rightarrow x \subseteq B\ ) \qquad \text{theorem of powerset} \\
\Sigma \ \mid\Rightarrow \ A \subseteq B
\end{array}
$$

57

## References


1. Blaine, L., A. Levine, R. Laddaga and P. Suppes, Technical problems in implementing university-level computer-assisted instruction in mathematics and science: First annual report, Technical Report No. 293, Institute for Mathematical Studies in the Social Sciences, Stanford University, 1978.

2. Flanagan, J. L. Speech Analysis, Synthesis and Perception 2nd edition, Springer-Verlag, New York, 1972.

3. Hearn, A. C. REDUCE User's Manual. Salt Lake City, Utah: University of Utah, Department of Computer Science, 1973.

4. Hinckley, M., R. Laddaga, J. Prebus and R. L. Smith, VOCAL: Voice Oriented Curriculum Author Language. Stanford, Calif.: Stanford University, Institute for Mathematical Studies in the Social Sciences, 1976.

5. Laddaga, R. and W. R. Sanders, 1977, Testing recognition of computer generated speech with elementary school children, JASA 62.S1 (abstract).

6. Levine, A. and W. R. Sanders, The MISS speech synthesis System, Technical Report 299, IMSSS, Stanford CA 94305, 1978.

7. Makhoul, J. 1975, Linear prediction: A tutorial review, Proc. of the IEEE, 63.4 pp.561-80.

8. Marti, J., A. Hearn, M. Griss, and C. Griss, 1977, Standard Lisp Report, University of Utah.

9. McDonald, J., L. Blaine, J. Marlowe and R. Roberts, The EXCHECK system. Western Educational Computing Conference, Anaheim, 1978.

10. Sanders, W. R. and R. Laddaga, 1976, Use of LPC synthesized speech in an instructional application, JASA 60.S1 (abstract).

11. Sanders, W. R., A. Levine, and C. Gramlich, 1978, The sensitivity

of LPC synthesized speech quality to the imposition of artificial pitch, duration, loudness and spectral contours, JASA, 64.S1 (abstract).

12. Suppes, P. Technical Problems in Implementing University-level Computer-assisted Instruction in Mathematics and Science. A research proposal submitted to the National Science Foundation, 1977.

13. Suppes, P. 1979, Current trends in Computer-assisted instruction, appearing in M. C. Yovits (Ed.), Advances in computers (Vol. 18), Academic Press, New York.

14. Suppes, P. Axiomatic set theory. New York: Van Nostrand, 1960.

59