KNOWLEDGE-BASED ADAPTIVE CURRICULUM SEQUENCING FOR CAI: APPLICATION OF A NETWORK REPRESENTATION

by

Keith T. Wescourt, Marian Beard, and Laura Gould

TECHNICAL REPORT NO. 288
September 20, 1977

PSYCHOLOGY AND EDUCATION SERIES

To be published in the Proceedings of the National Association for Computing Machinery, 1977.

This research was supported by the Personnel and Training Research Programs, Psychological Sciences Division, Office of Naval Research under Contract No. N00014-76-C-0165. Contract Authority Identification Number, NR No. 154-381 and by the Navy Personnel Research & Development Center under Contract N00123-75-C-1528.

Approved for public release; distribution unlimited.

Reproduction in whole or in part is permitted for any purpose of the U. S. Government.

INSTITUTE FOR MATHEMATICAL STUDIES IN THE SOCIAL SCIENCES

STANFORD UNIVERSITY
STANFORD, CALIFORNIA
**Knowledge-based Adaptive Curriculum Sequencing for CAI: Application of a Network Representation**

Keith T. Wescourt, Marian Beard and Laura Gould

Institute for Mathematical Studies in the Social Sciences, Stanford University, Stanford, California 94305

Personnel & Training Research Programs
Office of Naval Research (Code 458)
Arlington, VA 22217

Approved for public release; distribution unlimited

**Abstract**

One aspect of tutoring skill for technical subjects is individualized, adaptive sequencing of the problems given to students as learning exercises. A Curriculum Information Network (CIN) describes the relationships between the problems in a CAI curriculum and the concepts and skills that they are intended to teach. It is a basis for selecting problems for each student with respect to his evolving knowledge of those concepts and skills. This paper describes the application of a semantic network to represent the complex relationships.
interrelationships among the skills in a CIN for the BASIC Instructional Program, a CAI problem-solving laboratory for introductory programming in the BASIC language. The semantic network is used in drawing complex inferences about the student's state of knowledge and the problems that are appropriate to present to him. Such inferences enable more skillful problem sequencing by the CAI system.
One aspect of tutoring skill for technical subjects is individualized, adaptive sequencing of the problems given to students as learning exercises. A Curriculum Information Network (CIN) describes the relationships between the problems in a CAI curriculum and the concepts and skills that they are intended to teach. It is a basis for selecting problems for each student with respect to his evolving knowledge of those concepts and skills. This paper describes the application of a semantic network to represent the complex interrelationships among the skills in a CAI for the BASIC Instructional Program, a CAI problem-solving laboratory for introductory programming in the BASIC language. The semantic network is used in drawing complex inferences about the student's state of knowledge and the problems that are appropriate to present to him. Such inferences enable more skillful problem sequencing by the CAI system.

1. Introduction

Since its inception, a major goal of computer-assisted instruction (CAI) has been to individualize instruction by making the teaching system's behavior contingent on a student's responses during prior interactions with the system. Some early adaptive CAI systems applied features of mathematical learning and decision theories to sequencing drill-and-practice curricula in elementary mathematics (Suppes & Morningstar, 1972) and in initial reading (Green & Atkinson, 1966). These systems generated problems of different types (e.g., addition problems with or without "carries") according to an estimate of the student's mastery of each type based on his prior history of correct and incorrect answers. As a result, more capable students could progress rapidly to harder problem types and remediation and review could be determined with respect to individual difficulties manifested with specific problem types.

More recently, features adapted from Artificial Intelligence (AI) systems have been applied to individualize CAI along other dimensions. Research on question answering and natural language has enabled systems such as SCHOLAR (Collins, Wartok, & Passafiume, 1975) to conduct spontaneous instructional dialogues, giving students great freedom of expression and providing exposure to a topic according to the student's demonstrated familiarity with it. Other systems, for example, SOPHIE (Brown & Burton, 1975), have been designed to monitor student problem-solving attempts and react to errors or requests for help with concepts that reflect the context in which the student has been working. Significant progress has been made in AI-based CAI toward understanding the student's behavior with respect to representations of the knowledge he acquires and uses, and not merely in terms of his performance on particular questions and problems.

Our research over the past few years has attempted to apply some AI techniques to individualized curriculum sequencing for complex scientific and technical subjects. Traditionally, instruction in these types of subjects has depended heavily on students solving large numbers of problems. This method has proven to be effective for forcing the integration of concepts and problem-solving skills described in lectures and readings. Our concern has been to individualize the sequencing of a curriculum of complex problems in order to improve the acquisition and integration of the underlying concepts and skills of the subject. The result of our initial efforts was the concept of the Curriculum Information Network (CIN), a means of describing the relationships among the problems in a curriculum in terms of procedural skills involved in solving them, and for modeling student learning with respect to those same skills (Barr, Beard, & Atkinson, 1976). A CIN and a problem-selection procedure were implemented in the BASIC Instructional Program (BIP), a CAI system that teaches the BASIC programming language at an introductory level. The present paper describes our recent research, in which we developed a semantic network representation for structural and pedagogical knowledge about BASIC and some inference techniques for using this network to increase the sophistication of CIN-based problem selection in the BIP system.

2. Overview of the BIP system

2.1 Design

BIP is designed to teach elementary programming concepts and skills without lectures or a standard textbook. All of the resources available to students reside within the BIP system, illustrated in Figure 1, except for a manual written especially for the course. Furthermore, after a brief interactive introductory lesson, BIP presents no further text lessons on programming; instead, the student learns to program by solving programming problems using a BASIC interpreter built into the instructional system. For the purposes of our research, the value of BIP as a stand-alone CAI
representing the complete process by which a student understands a problem, determines solution algorithms, and implements it as acceptable BASIC code. The skills relate only to the coding process, many corresponding to a single line of code. For example, if a problem solution includes the lines

```
* ... TO INPUT T.
* 90 PRINT R * T
* ...
```

then its description in the CIN would include the skills, "assign numeric variable by -INPUT-" and "print numeric expression (operation on variables)".

Although the skills are, for the most part, defined in terms of syntactic constructs, HIP-I task selection does not reflect the student's knowledge of syntax, but instead depends on his knowledge of the semantics and pragmatics for using the skill appropriately. All purely syntactic errors are detected immediately by the HIP-I interpreter, which responds with explicit feedback describing the error and illustrating syntactically correct examples of the construct. These syntactic errors do not affect the model of the student maintained by HIP-I. This model is affected by logical errors which allow the student's program to run, but not to produce correct results. Many of these errors can be associated with semantically or pragmatically inappropriate use of the syntactic constructs described by HIP-I skills.

In HIP-I, the skills are grouped into about a dozen non-overlapping sets called techniques, such as simple printing, assignment, and conditional branching. The techniques themselves are linearly ordered according to judgments about the relative complexity of the skills they contain. The technique ordering is used in HIP-I as a constraint on the order in which major concepts are introduced and as a scale for determining whether problems available for remediating particular skills are appropriate to the student's overall progress.

2.4 Using a CIN for task selection

The general paradigm for applying a CIN to problem selection is:

1. From the model of the student's learning of all the skills, assemble sets of skills that correspond to relevant pedagogical considerations, such as "need further work," "ready to be learned," "already learned," etc.
2. Using these sets and other historical data (e.g., those tasks that this student has already completed), search the CIN for the task that uses a set of skills most congruent to a set that best satisfies overall pedagogical goals. For example, given the overall goal of reducing the total number of tasks students must complete to reach a level of competence, one might choose to search for a task with the most "need further work" and "ready to be learned" skills and the fewest "already learned" skills.

2.3 The CIN: Deleting tasks and skills

The primary knowledge base in HIP is a Curriculum Information Network. In the CIN of HIP-I, the version described by Barr, et al. (1976), each task in the curriculum is described in terms of a set of procedural skills necessary to complete it successfully (i.e., to write the BASIC program that is called for). About 50 skills are used to describe the entire HIP curriculum, with some tasks involving as few as one or two skills, and others more than ten. Each student's progress through the course is modeled with respect to his learning of the total set of skills. The skills defined in HIP-I in no way constitute a sufficient basis for representing the complete process by which a student understands a problem, determines solution algorithms, and implements it as acceptable BASIC code. The skills relate only to the coding process, many corresponding to a single line of code. For example, if a problem solution includes the lines

```
* ... TO INPUT T.
* 90 PRINT R * T
* ...
```

then its description in the CIN would include the skills, "assign numeric variable by -INPUT-" and "print numeric expression (operation on variables)".

Although the skills are, for the most part, defined in terms of syntactic constructs, HIP-I task selection does not reflect the student's knowledge of syntax, but instead depends on his knowledge of the semantics and pragmatics for using the skill appropriately. All purely syntactic errors are detected immediately by the HIP-I interpreter, which responds with explicit feedback describing the error and illustrating syntactically correct examples of the construct. These syntactic errors do not affect the model of the student maintained by HIP-I. This model is affected by logical errors which allow the student's program to run, but not to produce correct results. Many of these errors can be associated with semantically or pragmatically inappropriate use of the syntactic constructs described by HIP-I skills.

In HIP-I, the skills are grouped into about a dozen non-overlapping sets called techniques, such as simple printing, assignment, and conditional branching. The techniques themselves are linearly ordered according to judgments about the relative complexity of the skills they contain. The technique ordering is used in HIP-I as a constraint on the order in which major concepts are introduced and as a scale for determining whether problems available for remediating particular skills are appropriate to the student's overall progress.

2.4 Using a CIN for task selection

The general paradigm for applying a CIN to problem selection is:

1. From the model of the student's learning of all the skills, assemble sets of skills that correspond to relevant pedagogical considerations, such as "need further work," "ready to be learned," "already learned," etc.
2. Using these sets and other historical data (e.g., those tasks that this student has already completed), search the CIN for the task that uses a set of skills most congruent to a set that best satisfies overall pedagogical goals. For example, given the overall goal of reducing the total number of tasks students must complete to reach a level of competence, one might choose to search for a task with the most "need further work" and "ready to be learned" skills and the fewest "already learned" skills.

2.3 The CIN: Deleting tasks and skills

The primary knowledge base in HIP is a Curriculum Information Network. In the CIN of HIP-I, the version described by Barr, et al. (1976), each task in the curriculum is described in terms of a set of procedural skills necessary to complete it successfully (i.e., to write the BASIC program that is called for). About 50 skills are used to describe the entire HIP curriculum, with some tasks involving as few as one or two skills, and others more than ten. Each student's progress through the course is modeled with respect to his learning of the total set of skills. The skills defined in HIP-I in no way constitute a sufficient basis for representing the complete process by which a student understands a problem, determines solution algorithms, and implements it as acceptable BASIC code. The skills relate only to the coding process, many corresponding to a single line of code. For example, if a problem solution includes the lines

```
* ... TO INPUT T.
* 90 PRINT R * T
* ...
```

then its description in the CIN would include the skills, "assign numeric variable by -INPUT-" and "print numeric expression (operation on variables)".

Although the skills are, for the most part, defined in terms of syntactic constructs, HIP-I task selection does not reflect the student's knowledge of syntax, but instead depends on his knowledge of the semantics and pragmatics for using the skill appropriately. All purely syntactic errors are detected immediately by the HIP-I interpreter, which responds with explicit feedback describing the error and illustrating syntactically correct examples of the construct. These syntactic errors do not affect the model of the student maintained by HIP-I. This model is affected by logical errors which allow the student's program to run, but not to produce correct results. Many of these errors can be associated with semantically or pragmatically inappropriate use of the syntactic constructs described by HIP-I skills.

In HIP-I, the skills are grouped into about a dozen non-overlapping sets called techniques, such as simple printing, assignment, and conditional branching. The techniques themselves are linearly ordered according to judgments about the relative complexity of the skills they contain. The technique ordering is used in HIP-I as a constraint on the order in which major concepts are introduced and as a scale for determining whether problems available for remediating particular skills are appropriate to the student's overall progress.

2.4 Using a CIN for task selection

The general paradigm for applying a CIN to problem selection is:

1. From the model of the student's learning of all the skills, assemble sets of skills that correspond to relevant pedagogical considerations, such as "need further work," "ready to be learned," "already learned," etc.
2. Using these sets and other historical data (e.g., those tasks that this student has already completed), search the CIN for the task that uses a set of skills most congruent to a set that best satisfies overall pedagogical goals. For example, given the overall goal of reducing the total number of tasks students must complete to reach a level of competence, one might choose to search for a task with the most "need further work" and "ready to be learned" skills and the fewest "already learned" skills.

2.3 The CIN: Deleting tasks and skills

The primary knowledge base in HIP is a Curriculum Information Network. In the CIN of HIP-I, the version described by Barr, et al. (1976), each task in the curriculum is described in terms of a set of procedural skills necessary to complete it successfully (i.e., to write the BASIC program that is called for). About 50 skills are used to describe the entire HIP curriculum, with some tasks involving as few as one or two skills, and others more than ten. Each student's progress through the course is modeled with respect to his learning of the total set of skills. The skills defined in HIP-I in no way constitute a sufficient basis for representing the complete process by which a student understands a problem, determines solution algorithms, and implements it as acceptable BASIC code. The skills relate only to the coding process, many corresponding to a single line of code. For example, if a problem solution includes the lines

```
* ... TO INPUT T.
* 90 PRINT R * T
* ...
```

then its description in the CIN would include the skills, "assign numeric variable by -INPUT-" and "print numeric expression (operation on variables)".

Although the skills are, for the most part, defined in terms of syntactic constructs, HIP-I task selection does not reflect the student's knowledge of syntax, but instead depends on his knowledge of the semantics and pragmatics for using the skill appropriately. All purely syntactic errors are detected immediately by the HIP-I interpreter, which responds with explicit feedback describing the error and illustrating syntactically correct examples of the construct. These syntactic errors do not affect the model of the student maintained by HIP-I. This model is affected by logical errors which allow the student's program to run, but not to produce correct results. Many of these errors can be associated with semantically or pragmatically inappropriate use of the syntactic constructs described by HIP-I skills.

In HIP-I, the skills are grouped into about a dozen non-overlapping sets called techniques, such as simple printing, assignment, and conditional branching. The techniques themselves are linearly ordered according to judgments about the relative complexity of the skills they contain. The technique ordering is used in HIP-I as a constraint on the order in which major concepts are introduced and as a scale for determining whether problems available for remediating particular skills are appropriate to the student's overall progress.
3. Present the task to the student and analyze his performance on it to update the model of his learning of the skills.

Three main theoretical entities bear on this task-selection paradigm. First, there is the representation of knowledge to be learned, which is the skills in the CIN. Second, there is a theory of learning, which maps performance on tasks onto changes in the student's knowledge of the skills. Finally, there is a theory of instruction, which for any state of knowledge determines the next task that is "best" for the student to work on.

3. Applying a network representation of knowledge to task selection

Our initial use of a CIN in BIP-I for selecting tasks has demonstrated the successful application of the CIN paradigm (Harr, et al., 1976). However, the representation of programming knowledge and the model of student learning used in BIP-I are very rudimentary. Most obviously, BIP-I's grouping of skills into techniques is an oversimplification of the actual interrelations between skills. The technique groups do not provide a sufficient basis for anticipating a student's performance in new contexts based on his performance in related contexts--an important aspect of a human tutor's skill in selecting tasks for his student. Likewise, the student model, consisting of counters for each skill, does not differentiate various levels of skill mastery indicated by the amount of difficulty a student encounters in completing tasks. We therefore undertook to design a new CIN-based task-selection procedure for a BIP-II system, incorporating both a more detailed representation for the knowledge underlying the curriculum and more complex assumptions for modeling student learning. The remainder of this paper will focus on the new representation we have developed and how it is used in the BIP-II system.

In considering alternative representations for the knowledge underlying a task, we recognized that the most powerful approach would be a procedural representation sufficient to synthesize task solutions (see, for example, Brown, Barton, Miller, Dekker, Purcell, Hausmann, & Bobrow, 1975, and Carr & Goldstein, 1977). However, the state-of-the-art in program synthesis and analysis techniques has not yet advanced to a point where a manageable system could be implemented for automatically solving programming problems like those in the BIP curriculum. Thus, we decided to extend the original concept of a set of skills by embedding the skills in a network representation describing the structural and pedagogically significant relations between them. The network relationships allow inferences that potentially add sophistication to both the process of task selection and of interpreting student performance to update the student model. For example, unlearned skills that are deemed to be analogous to other skills that are already learned can be given lower priority for inclusion in the next task to be presented. Or, if such skills occur in a task that a student quits, then they can be taken as less likely sources of his difficulty than unlearned skills that are analogous to other skills that are already known to be troublesome for that student.

3.1 The BASICNET

Rather than basing the network of knowledge to be learned directly on the BIP curriculum, we built the skill relationships on a general representation for BASIC programs. From a general analysis of the BASIC language, guides to BASIC programming, and the skills and techniques of BIP-I, we developed a network representation for BASIC programming constructs (the BASICNET), a simplified portion of which is shown in Figure 2.

The node names are self-explanatory; the links (relationships) are Kind, Component, Hardness, and (mutual functional) Dependency. The section of the BASICNET shown specifies that there are two kinds of control structures, and expresses a judgment that the conditional kind is harder to learn than the unconditional. There are two kinds of conditional structures, and FORNEXT is harder than IFTHEN. The components of an IFTHEN statement are the words "IF" and "THEN" with the Boolean condition and the line number in the appropriate places. For the purposes of this illustration, the BOOLEAN consists of a numeric expression (NEXPR), a relational operator (REL), and another NEXPR; among the three kinds of NEXPS, numeric literals (NILIT) are easiest, and numeric variables (NVAR) and simple arithmetic expressions (SIMARITH) are increasingly hard.

Note that the downward links in the figure provide information like that found in a BPS notation for BASIC, while the horizontal links provide pedagogical information specifying relative difficulty, analogy, and dependency. The opinions expressed by the horizontal links are necessarily general and do not always hold for all students in all stages of learning. For example, an arithmetic expression is generally a harder construct than a numeric variable because it often includes a variable itself, but observation indicates that
using a statement such as PRINT 64 tends to be an easier task for a beginner than using PRINT N.
This implies that, ultimately, the pedagogical relationships between concepts must sometimes be a function of the student's state of learning at the time the relationships are to be used. We have chosen not to tackle this refinement in the BASICET underlying the BP-II system.

3.2 List notation for the BASICET

A simplified notation of the list notation we use to represent the portion of the BASICET in Figure 2 is:

\[
\text{(CONTROL/CONTROL STRUCTURE} K \quad \text{(UNCONDITIONAL CONDITIONAL))} \\
\text{(CONDITIONAL} K \quad \text{(IF THEN FOR NEXT))} \\
\text{(IF THEN} C \quad \text{("IF" DOolean "THEN" LIBRARY) H \quad \text{(FOR NEXT))} \\
\text{(FOR NEXT} C \quad \text{(FOR NEXT))} \\
\text{(FOR} D \quad \text{(NEXT))} \\
\text{(IF X X EXPRESSION REL. EXPRESSION))} \\
\text{(NEXT X \quad \text{MUTI SVAR SIMARITH))} \\
\text{(HVAR H \quad \text{(VAR) H (SLIT))} \\
\text{(NVAR H \quad \text{(SIMARITH) A \quad \text{(SVAR) S \quad \text{(SVAR))}}}
\]

The A links specify that numeric literals are analogous to string literals (SLIT), and that numeric variables are analogous to string variables (SVAR). The REL link says that NVAR and SVAR are similarly difficult. (The information about SLIT and SVAR is found in another part of the BASICET.) The notation here is simply that used to express property lists in LISP.

3.3 The BASICET and INF skills

After the BASICET was defined, each skill in INF's CIN was represented in terms of a subnet. First, the structure of each skill was described, in list notation like the following (where Skill 42 is "conditional branch, comparing a numeric literal with a numeric variable"):

\[
\text{(SKO42 (IF THEN) \quad \text{(BOOLEAN - NEXT EXP NVAR))}}
\]

Skill 42 is represented as an instance of IF THEN (see Figure 2), in which the BOOLEAN component is further specified as consisting of the relation between a numeric literal (the first NEXT EXP component of the BOOLEX) and a numeric variable (the second NEXT EXP). The REL is left uninstantiated, since Skill 42 does not specify the kind of comparison to be made between the two. Thus, any REL is appropriate.

Skill 43 is "conditional branch, comparing a simple numeric expression with a numeric variable." Its structure is:

\[
\text{(SKO43 (SKO42 \quad \text{(NEXT EXP SIMARITH) }}
\]

The notation is read "Skill 43 is identical to Skill 42 except that the first instance of NEXT EXP should be SIMARITH," which is exactly what the English description of the skill says. Skill 46 ("conditional branch, comparing two numeric variables") is represented as:

\[
\text{(SKO46 (SKO42 \quad \text{(NEXT EXP \quad \text{NVAR)})}}
\]

again reflecting the minimal difference between the related skills.

3.4 Skill sets

Based on the notation for skill structures, skills were grouped together into ten major skill sets representing printing, numeric assignment, string assignment, IF-THEN, FOR-NEXT, etc. Each skill set was formed by starting with a skill, not described in terms of any other skill—like Skill 42 above—and all other skills (43, 46, etc.) described in terms of it, or described in terms of other members of the set. As might be expected, there was a degree of congruence between the ten skill sets and the technique groupings of BP-I.

3.5 The SKILLSSET

Within each skill set, pairs of skills were examined to find their minimal difference. If the nodes by which they differ are linked in the BASICET, then links were used to define a relation between the skills. If the nodes by which two skills differ do not have a direct link, relations were sought at increasingly higher levels of the BASICET.

For instance, since the BASICET shows NVAR to be harder than SLIT, and SIMARITH harder than NVAR, it follows that Skill 46 is harder than 42, and 43 is harder than 46. The relationships determined in this manner between all pairs of skills comprise the SKILLSSET, a knowledge representation that can be directly expressed in a CIN and used for task selection. The SKILLSSET, like the BASICET, can be expressed in LISP property list notation. (The underlining in the following example emphasizes the relationships being discussed here.)

\[
\text{(SKO42 H \quad \text{(SKO44 \quad \text{SKO46 \quad \text{SKO47) A \quad \text{(SKO47)}})} \quad \text{P \quad \text{(SKO43 \quad \text{SKO36 \quad \text{SKO39)}})} \quad \text{(SKO43 \quad \text{SKO47 \quad \text{SKO48)}} \quad \text{P \quad \text{(SKO46 \quad \text{SKO36 \quad \text{SKO39)}})} \quad \text{(SKO46 \quad \text{SKO47 \quad \text{SKO48)}} \quad \text{P \quad \text{(SKO43 \quad \text{SKO36 \quad \text{SKO39)}})}
\]

The P links shown here are Prerequisite links; like the Hardness links, these are a matter of pedagogical opinion. The P links, however, appear only in the SKILLSSET, not in the BASICET, and express judgments that are more specific to the INF course than those expressed in the BASICET. A few of the skills (e.g., those involving the use of built-in BASIC functions such as INT and 

\[\text{D} \]

did not fall into skill sets since they seemed not to be describable in terms of any other skill. These are related within the SKILLSSET only by means of P links.

3.6 INF-II task-selection procedure

The new task-selection procedure for the INF-II system, designed to use the relationships between skills expressed in the SKILLSSET, is identical to the technique-based method in its overall design: A set of skills appropriate to the student's current level of understanding is generated, a set of tasks using some of these skills is
identified, the rest of those tasks (by some criteria) is presented, and the student model is updated based on the student's performance and self-evaluation on the task. The major difference between the two methods is that by using an expanded set of relations between skills in the SKILLSET, the new procedure can make more intelligent inferences both in updating the student model and in generating the set of skills to be involved in the student's next task.

The following is a simplified description of the process by which a task is selected at any point during instruction, given the student's state of knowledge of all the skills. The procedure integrates a number of a priori reasonable pedagogical heuristics about how to vary the relative difficulty of tasks to optimize learning as performance varies and about how to teach a network of knowledge (e.g., breadth-first vs. depth-first exposure).

**STEP 1:** Create a set called NEED, consisting of skills that will be sought in the next task. Look for "trouble" skills first (those in tasks that the student quit), then for analogies to learned skills, then for inverse-prerequisites of learned skills. As soon as a group of such skills is found, stop looking.

**STEP 2:** Remove from the NEED set those skills that have unlearned prerequisites. Add those skills to the NOTREADY set (which may be used later).

**STEP 3:** Given a NEED set, find the most appropriate task that involves some of the skills.

(a) Assemble GOODLIST, those tasks that have the desired number of NEEDed skills. (This number increases if the student is consistently successful, decreases if he has trouble.)

(b) If no GOODLIST can be created, make a new NEED set consisting of the prerequisites of the skills in NOTREADY. If no new NEED set can be created, then the curriculum has been exhausted; otherwise, GOTO 3a.

(c) Find the "best" task: if the student is doing well, find the task in GOODLIST that has the fewest learned skills; if he is progressing more slowly, find the task with the fewest unseen skills. Remove the selected task from GOODLIST.

**STEP 4:** See if the selected task is otherwise appropriate.

(a) If none of the skills in the selected task have unlearned prerequisites, stop looking and present the task. (END)

(b) If any skills have unsatisfied prerequisites, reject the task and add those skills to the NOTREADY set.

(c) If GOODLIST is exhausted, change (usually reduce) the critical number of NEEDed skills, and GOTO 3a. Otherwise, using the rest of GOODLIST GOTO 3c.

As an example of the inferences made in the generation of the NEED set, let us assume that skills 42 and 61 ("FOR, NEXT loops with literal as final value of index") are under consideration. Skill 61 is represented as

$$(\text{SK061} \land \text{SK062}) \land (\text{SK042})$$

The Prerequisite relationship specified that 42 must be learned before a task involving 61 can be presented. The structure enforced by the P links relating pairs of skills gives the presentation of tasks some degree of order, and is designed to prevent too-rapid progress or drastic jumps in difficulty. The hardness links, in contrast, are used to facilitate progress for a student doing well, by allowing some skills to be considered "too easy" for inclusion in the NEED set. Such skills are not inferred to be learned; they are simply not sought actively by the selection algorithm.

As an example using the skills described here, suppose that a student has successfully completed a task using Skill 43, although he has not yet seen Skill 42. (The fact that 43 is harder than 42 does not force 42 to be presented first; only P links force such order.) When the task-selection procedure is next run, the next set of NEED skills, it will "infer" that 42 is now too easy to become part of that set, since something harder than 42 has already been learned.

Furthermore, since 42 is now considered too easy to look for, Skill 61 can now be sought. If the student successfully completes a task involving 61, the student model will be updated to show that 61 has been learned, and by inference, that its unseen prerequisite 42 has also been learned. These kinds of inferences (by which skills can reach too-easy or learned states without the student actually having seen them in a task) can of course be contradicted by direct observation or by other inferences if the student has difficulty. For example, the unseen prerequisite of a given skill may change its state from "too easy" to "in trouble" if the student quits (gives up on) a task involving the given skill. The next task selection would attempt to find a task using that prerequisite skill in such a case.

3.7 BIP-II performance

The BIP-II task-selection procedure has been implemented with parameters (e.g., numbers of skills sought and thresholds determining when an unsought skill is "too easy" to be included in the NEED set) that can be changed readily. The system can therefore be used to explore the effectiveness of somewhat different pedagogical heuristics for task sequencing. We used this capability in conjunction with a simulation system we developed to create a version of BIP-II that we expected would be effective for a range of student abilities. Recently, we collected data from a group of about 27 students who used this BIP-II system. At this time, the data have not been extensively analyzed, but we can provide a general summary and report our subjective observations.
1. Principal Civilian Advisor for Education and Training
   Attn: Dr. William L. Malay
   Naval Training Command, Code OOA
   Pensacola, FL 32508

2. Dr. Alfred F. Smode, Director
   Training Analysis & Evaluation Group
   Department of the Navy
   Orlando, FL 32813

3. Chief of Naval Education and Training Support (OIA)
   Pensacola, FL 32509

4. Capt. H. J. Connery, USN
   Navy Medical R&D Command
   NNMC, Bethesda, MD 20014

5. Navy Personnel R&D Center
   Code 01
   San Diego, CA 92152

6. Navy Personnel R&D Center
   Attn: Dr. James McGrath (Code 306)
   San Diego, CA 92152

7. A. A. Sjoholm, Head, Technical Support
   Navy Personnel R&D Center (Code 201)
   San Diego, CA 92152

8. Navy Personnel R&D Center
   Attn: Library
   San Diego, CA 92152

   Head, Section on Medical Education
   Uniformed Services, Univ. of the Health Sciences
   6917 Arlington Road
   Bethesda, MD 20014

10. LCDR J. W. Snyder, Jr.
    F-14 Training Model Manager
    VF-124
    San Diego, CA 92025

11. Dr. John Ford
    Navy Personnel R&D Center
    San Diego, CA 92152

12. Dr. Worth Scanland
    Chief of Naval Education & Training
    NAS, Pensacola, FL 32508

13. Dr. Richard A. Pollak
    Academic Computing Center
    U.S. Naval Academy
    Annapolis, MD 21402

Army

1. Technical Director
   U.S. Army Research Institute for the Behavioral & Social Sciences
   5001 Eisenhower Avenue
   Alexandria, VA 22333

2. Armed Forces Staff College
   Attn: Library
   Norfolk VA 23511

3. Commandant
   U.S. Army Infantry School
   Attn: ATSH-I-V-IT
   Fort Benning, GA 31905

4. Commandant
   Attn: EA
   U.S. Army Institute of Administration
   Fort Benjamin Harrison, IN 46216

5. Dr. Beatrice Farr
   U.S. Army Research Institute
   5001 Eisenhower Avenue
   Alexandria, VA 22333

6. Dr. Frank J. Harris
   U.S. Army Research Institute
   5001 Eisenhower Avenue
   Alexandria, VA 22333

7. Dr. Ralph Dusek
   U.S. Army Research Institute
   5001 Eisenhower Avenue
   Alexandria, VA 22333

8. Dr. Leon Nawrocki
   U.S. Army Research Institute
   5001 Eisenhower Avenue
   Alexandria, VA 22333
1 Dr. Joseph Ward
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

1 Dr. Milton S. Katz, Chief
Individual Training & Performance
Evaluation Technical Area
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

1 Col. G. B. Howard
U.S. Army
Training Support Activity
Fort Eustis, VA 23604

1 Col. Frank Hart, Director
Training Management Institute
U.S. Army, Bldg. 1725
Fort Eustis, VA 23604

1 HQ USAREUR & 7th Army
ODCSOPS
USAREUR Director of GED
APO New York 09403

1 Dr. James Baker
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Air Force
1 Research Branch
AFMPC/DFMYP
Randolph AFB, TX 78148

1 AFHRL/AS (Dr. G. A. Eckstrand)
Wright-Patterson AFB
Ohio 45433

1 Dr. Ross L. Morgan (AFHRL/ASR)
Wright-Patterson AFB
Ohio 45433

1 Dr. Marty Rockway (AFHRL/TT)
Lowry AFB
Colorado 80230

1 Instructional Technology Branch
AFHRL
Lowry AFB, CO 80230

1 Dr. Alfred R. Fregly
AFOSR/NL, Bldg. 110
Bolling AFB, DC 20332

1 Dr. Sylvia R. Mayer (MCIT)
HQ Electronic Systems Division
LG Hanscom Field
Bedford, MA 01730

1 Air University Library
AUL/ISE 76-443
Maxwell AFB, AL 36112

1 Director, Office of Manpower
Utilization
HQ, Marine Corps (Code MPU)
BFB, Bldg. 2009
Quantico, VA 22134

1 Dr. A. L. Slafkosky
Scientific Advisor (Code RD-1)
HQ, U.S. Marine Corps
Washington, DC 20380

1 AC/S, Education Programs
Education Center, MCDEC
Quantico, VA 22134

Coast Guard
1 Mr. Joseph J. Cowan, Chief
Psychological Research Branch (G-P-1/62)
U.S. Coast Guard Headquarters
Washington, DC 20590

Other DoD
1 Dr. Harold F. O'Neill, Jr.
Advanced Research Projects Agency
Cybernetics Technology, Room 623
1400 Wilson Blvd.
Arlington, VA 22209
1 Dr. Robert Young
Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

12 Defense Documentation Center
Attn: TC
Cameron Station, Bldg. 5
Alexandria, VA 22314

1 Military Assistant for Human Resources
Office of the Director of Defense Research & Engineering
Room 3D129, The Pentagon
Washington, DC 20301

1 Director, Management Information Systems Office
CSD, M&RA
Room 3B917, The Pentagon
Washington, DC 20301

Other Government

1 Dr. Vern Urry
Personnel R&D Center
U.S. Civil Service Commission
1900 E Street, NW
Washington, DC 20415

1 Dr. Andrew R. Molnar
Science Education Dev. & Res.
National Science Foundation
Washington, DC 20550

1 Dr. Marshall S. Smith
Associate Director
NIE/OPEPA
National Institute of Education
Washington, DC 20208

1 Dr. Joseph L. Young, Director
Memory & Cognitive Processes
National Science Foundation
Washington, DC 20550

Miscellaneous

1 Dr. John R. Anderson
Department of Psychology
Yale University
New Haven, CT 06520

1 Dr. Scarvia B. Anderson
Educational Testing Service
3445 Peachtree Road, NE - Suite 1040
Atlanta, GA 30326

1 Prof. Earl A. Alluisi
Department of Psychology (Code 287)
Old Dominion University
Norfolk, VA 23508

1 Dr. Gerald V. Barrett
Department of Psychology
University of Akron
Akron, OH 44325

1 Dr. John Seeley Brown
Bolt Beranek and Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

1 Jacklyn Caselli
ERIC Clearinghouse on Information Resources
School of Education - SCRTD
Stanford University
Stanford, CA 94305

1 Century Research Corporation
4113 Lee Highway
Arlington, VA 22207

1 Dr. Kenneth E. Clark
College of Arts & Sciences
University of Rochester
River Campus Station
Rochester, NY 14627

1 Dr. Allan M. Collins
Bolt Beranek and Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

1 Dr. John J. Collins
Essex Corporation
201 N. Fairfax Street
Alexandria, VA 22314

1 Dr. Ruth Day
Center for Advanced Studies in the Behavioral Sciences
202 Junipero Serra Blvd.
Stanford, CA 94305
1 Dr. John D. Carroll  
Psychometric Laboratory  
Davie Hall, 013A  
University of North Carolina  
Chapel Hill, NC 27514

1 ERIC Facility-Acquisitions  
4833 Rugby Avenue  
Bethesda, MD 20014

1 Major I. N. Evonic  
Canadian Forces Personnel Applied Research Unit  
1107 Avenue Road  
Toronto, Ontario, CANADA

1 Dr. Victor Fields  
Department of Psychology  
Montgomery College  
Rockville, MD 20850

1 Dr. Edwin A. Fleishman  
Advanced Research Resources Organization  
8555 Sixteenth Street  
Silver Spring, MD 20910

1 Dr. John R. Frederiksen  
Bolt Beranek and Newman, Inc.  
50 Moulton Street  
Cambridge, MA 02138

1 Dr. Vernon S. Gerlach  
College of Education  
146 Payne Bldg. B  
Arizona State University  
Tempe, AZ 85281

1 Dr. Robert Glaser, Co-Director  
University of Pittsburgh  
3939 O'Hara Street  
Pittsburgh, PA 15213

1 Dr. M. D. Havron  
Human Sciences Research, Inc.  
7710 Old Spring House Road  
West Gate Industrial Park  
McLean, VA 22101

1 CDR Mercer  
CNET Liaison Officer  
AFHRL/Flying Training Division  
Williams AFB, AZ 85224

1 HumRRO/Western Division  
Attn: Library  
27857 Berwick Drive  
Carmel, CA 93921

1 HumRRO/Columbus Office  
Suite 23, 2601 Cross Country Drive  
Columbus, GA 31906

1 Dr. Lawrence B. Johnson  
Lawrence Johnson & Associates, Inc.  
2001 S Street, NW - Suite 502  
Washington, DC 20009

1 Dr. Arnold F. Kanarick  
Honeywell, Inc.  
2600 Ridgeway Pkwy.  
Minneapolis, MN 55413

1 Dr. Roger A. Kaufman  
203 Dodd Hall  
Florida State University  
Tallahassee, FL 32306

1 Dr. Steven W. Keele  
Department of Psychology  
University of Oregon  
Eugene, OR 97403

1 Dr. David Klahr  
Department of Psychology  
Carnegie-Mellon University  
Pittsburgh, PA 15213

1 Dr. Robert R. Mackie  
Human Factors Research, Inc.  
6730 Corton Drive  
Santa Barbara Research Park  
Goleta, CA 93017

1 Dr. William C. Mann  
Information Sciences Institute  
University of Southern California  
4676 Admiralty Way  
Marina Del Rey, CA 90291

1 Dr. Leo Munday  
Houghton Mifflin Co.  
P.O. Box 1970  
Iowa City, IA 52240
Dr. Thomas G. Sticht  
Associate Director, Basic Skills  
National Institute of Education  
1200 19th Street, NW  
Washington, DC 20208

Prof. Fumiko Samejima  
Department of Psychology  
Austin Peay Hall 304C  
University of Tennessee  
Knoxville, TN 37916

Dr. Meredith Crawford  
5605 Montgomery Street  
Chevy Chase, MD 20015

Dr. Nicholas A. Bond  
Department of Psychology  
Sacramento State College  
6000 Jay Street  
Sacramento, CA 95819

Dr. James Greeno  
Learning R&D Center  
University of Pittsburgh  
3939 O'Hara Street  
Pittsburgh, PA 15213

Dr. Frederick Hayes-Roth  
The Rand Corporation  
1700 Main Street  
Santa Monica, CA 90406

Dr. Robert Sternberg  
Department of Psychology  
Yale University  
Box 11A, Yale Station  
New Haven, CT 06520

Dr. Walter Schneider  
Department of Psychology  
University of Illinois  
Champaign, IL 61820

Dr. Richard B. Millward  
Department of Psychology  
Hunter Laboratory  
Brown University  
Providence, RI 02912