A General Learning Theory and its application to Schema Abstraction

John R. Anderson
Paul J. Kline
Carnegie-Mellon University

Charles M. Beasley, Jr.
Yale University

Send proofs to: John R. Anderson
Department of Psychology
Carnegie-Mellon University
Pittsburgh, Pa. 15213

This research was sponsored by the Personnel Training and Research Programs, Psychological Services Division, Office of Naval Research, under Contract No.: N00014-78-C-0725, Contract Authority Identification Number, NR No. 154-399 and grant NIE-G-77-0005 from the National Institute of Education. Some of the ideas presented developed out of discussion with David Nicolas and particularly Pat Langley.

A General Learning Theory and its application to Schema Abstraction

John R. Anderson
Paul J. Kline
Carnegie-Mellon University

Charles M. Beasley, Jr.
Yale University

Send proofs to: John R. Anderson
Department of Psychology
Carnegie-Mellon University
Pittsburgh, Pa. 15213

This research was sponsored by the Personnel Training and Research Programs, Psychological Services Division, Office of Naval Research, under Contract No.: N00014-78-C-0725, Contract Authority Identification Number, NR No. 154-399 and grant NIE-G-77-0005 from the National Institute of Education. Some of the ideas presented developed out of discussion with David Nicolas and particularly Pat Langley.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
Technical report 78-2

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A general learning theory and its application to schema abstraction.

5. TYPE OF REPORT & PERIOD COVERED
Final technical

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
John R. Anderson, Paul J. Kline, and
Charles M. Beasley, Jr.

8. CONTRACT OR GRANT NUMBER(s)
N00014-78-C-0725

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Department of Psychology
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
66153N
RR042-04-01 NR 154-399

11. CONTROLLING OFFICE NAME AND ADDRESS
Personnel and Training Research Programs
Office of Naval Research
Arlington, VA 22217

12. REPORT DATE
December 15, 1978

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office)

15. SECURITY CLASS. (of this report)
unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This research was also supported by NIE-G-77-0005 from the National Institute of Education

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Artificial Intelligence
Cognitive Psychology
Computer Simulation
Learning
Production Systems
Designation
Strengthening
Generalization
Discrimination
Schema Abstraction
Prototype Formation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ACT is a computer simulation program that uses a propositional network to represent knowledge of general facts and a set of productions (condition - action rules) to represent knowledge of procedures. There are currently four different mechamisms by which ACT can make additions and modifications to its set of productions as required for procedural learning: designation, strengthening, generalization, and discrimination. Designation refers to the ability of productions to call for the creation of new productions. Strengthening a production may have important consequences for performance, since a production's strength determines the amount of system resources that will be allocated to its processing. Finally, generalization and discrimination refer to complementary processes that produce better performance by either extending or restricting the range of situations in which a production will apply. These learning mechanisms are used to simulate experiments on schema abstraction by Franks and Bransford (1971), Hayes-Roth and Hayes-Roth (1977), and Medin and Schaffer (1978). The mechanisms are used to predict recognition trials to criterion, as well as final test recognition and classification. ACT successfully accounts for the effects of distance of instances from a central tendency, frequency of individual instances, and inter-item similarity.

Table of Contents

Abstract

ACT is a computer simulation program that uses a propositional network to represent knowledge of general facts and a set of productions (condition - action rules) to represent knowledge of procedures. There are currently four different mechanisms by which ACT can make additions and modifications to its set of productions as required for procedural learning: designation, strengthening, generalization, and discrimination. Designation refers to the ability of productions to call for the creation of new productions. Strengthening a production may have important consequences for performance, since a production's strength determines the amount of system resources that will be allocated to its processing. Finally, generalization and discrimination refer to complementary processes that produce better performance by either extending or restricting the range of situations in which a production will apply. These learning mechanisms are used to simulate experiments on schema abstraction by Franks and Bransford (1971), Hayes-Roth and Hayes-Roth (1977), and Medin and Schaffer (1978). The mechanisms are used to predict recognition trials to criterion, as well as final test recognition and classification. ACT successfully accounts for the effects of distance of instances from a central tendency, frequency of individual instances, and inter-item similarity.

I Introduction

We are interested in understanding learning. For many years learning theory was practically synonymous with experimental psychology; however, its boundaries have shrunk to such an extent that they barely overlap at all with those of modern cognitive psychology. Cognitive psychologists, by and large, concern themselves with a detailed analysis of the mechanisms that underlie adult human intelligence. This analysis has gone on too long without adequate attention to the question of how these complex mechanisms could be acquired. In an attempt to answer this question, we have adopted one of the methodological approaches of modern cognitive psychology: Results of detailed experimental analyses of cognitive behaviors are elaborated into a computer simulation of those behaviors. The simulation program provides new predictions for a further experimental testing whose outcome is then used to modify the simulation and the whole process then repeats itself.

Our computer simulation is called ACT. The ACT system embodies the extremely powerful thesis that a single set of learning processes underlies the whole gamut of human learning--from children learning their first language by hearing examples of adult speech to adults learning to program a computer by reading textbook instructions.

In this paper we will give a general overview of the ACT learning theory and describe its application to research on abstraction of schemas. Elsewhere we have provided somewhat more technical discussions of the ACT system and described its application to other domains (Anderson, 1976; Anderson, Kline, and Lewis, 1977; Anderson, Kline and Beasley, 1977; Anderson, Kline and Beasley, in press).

A. The ACT System

In ACT knowledge is divided into two categories: declarative and procedural. The declarative knowledge is represented in a propositional network similar to semantic network representations proposed elsewhere (Quillian, 1969; Anderson and Bower, 1973; Norman and Rumelhart, 1975). While the network aspects of this representation are important for such

ACT processes as spreading activation, they are not important to the current learning discussion. For present purposes we will consider ACT's declarative knowledge as a set of assertions or propositions and ignore the technical aspects of its network representation.

ACT represents its procedural knowledge as a set of productions. The ACT production system can be seen as a considerable extension and modification of the production systems developed at Carnegie-Mellon (Newell, 1972, 1973; Rychener and Newell, 1977). A production is a condition - action rule. The condition is an abstract specification of a set of propositions. If a set of propositions can be found in the data base which meets this specification, the production will perform its action. Actions can both add to the contents of the data base and cause the system to emit observable responses.

ACT's productions can only have their conditions satisfied by *active* propositions. ACT's activation mechanism is designed such that the only propositions active are those that have recently been added to the data base or that are closely associated to propositions which have been added. Propositions are added to the data base either through input from the environment or through the execution of productions. Thus, this activation system gives ACT the property of being immediately responsive to changes in its environment or in its internal state.

ACT's basic control structure is an iteration through successive *cycles*, where each cycle consists of a production selection phase followed by an execution phase. On each cycle an APPLYLIST is computed which is a probabilistically defined subset of all of the productions whose conditions are satisfied by active propositions. The probability that a production will be placed on the APPLYLIST depends on the *strength* (s) of that production relative to the sum (S) of the strengths of all the productions whose conditions mention active elements; that is, this probability varies with s/S. Discussion of the process of assigning a strength to a production will be postponed until a later section; all that needs to be said here is that this strength reflects just how successful past applications of this production have been. Thus one component of the production-selection phase consists of choosing out of all the productions which could apply those which are the most likely to apply successfully. Further discussion of the details of production selection and execution is best conducted in the context of an example.

B. An Example Production System

Table 1 presents a set of productions for adding two numbers.[1] Let us consider how this production set would apply to the addition problem of 32 + 18. We assume this problem is encoded by a set of propositions which may approximately be rendered as:

```
The goal is to add 32 and 18
32 begins with a 2
The 2 is followed by a 3
32 ends with this 3
18 begins with a 8
The 8 is followed by a 1
18 ends with this 1
```

The above propositions encode the digits from right to left as is required by the standard addition algorithm.

The condition of P1 in Table 1 is satisfied by making the following correspondences between elements of the condition and propositions in the data base:

The goal is to add LVnumber1 and LVnumber2=The goal is to add 32 and 18
LVnumber1 begins with a LVdigit1 = 32 begins with a 2
LVnumber2 begins with a LVdigit2 = 18 begins with a 8

In making these correspondences, the variables LVnumber1, LVnumber2, LVdigit1, and LVdigit2 are bound to the values 32, 18, 2, and 8 respectively. The *LV* prefix indicates that these are local variables and can be bound to anything. Since they only maintain their binding within the production, other productions are not constrained to match these variables in the same way. The action of P1, *the subgoal is to add LVdigit1 and LVdigit2*, becomes,

[1] The productions presented in this paper are translations of the formal syntax of the implemented productons into (hopefully) more readable prose. The reader interested in the actual implementation details may request listings of the implemented versions and examples of their operation.

given the values of the variables, an instruction to place the proposition, *The subgoal is to add 2 and 8*, into the data base. This serves as a cue to productions that will actually add 2 and 8.

After the execution of P1 the first element of the condition of production P2 is satisfied:

The subgoal is to add LVdigit1 and LVdigit2=The subgoal is to add 2 and 8

The remaining condition of P2 matches a proposition in the data base about integer addition:

LVsum is the sum of LVdigit1 and LVdigit2 = 10 is the sum of 2 and 8

The action of P2 adds to the data base *The subgoal is to put out 10.*

The next production to apply is P5 which is matched as follows:

The subgoal is to put out LVsum = The subgoal is to put out 10
The subgoal is to add LVdigit1 and LVdigit2=The subgoal is to add 2 and 8
LVsum is greater than 9 = 10 is greater than 9
LVsum is the sum of LVdigit3 and 10 = 10 is the sum of 0 and 10

The action of P5 writes out *0* as the first digit in the answer, places a proposition in the data base, *The subgoal is to do the next digits after 2 and 8*, to the effect that this column is finished, and sets a,carry flag.

Insert Table 1 about here

It is worth considering why no other production besides P5 can apply. All the conditions of production P3 match, but P5 contains all the conditions of P3 plus two additional propositions. Because its condition contains more elements, P5 is applied rather than P3. This illustrates the principle of specificity - if two productions match but the condition of one of them is a subset of the condition of the other, then the production with the larger number of conditions (more specific) will apply instead of the production with fewer conditions (more

Table 1

A Set of Productions for Adding Two Numbers

```
P1:      IF  the goal is to add LVnumber1 and LVnumber2
             and LVnumber1 begins with a LVdigit1
             and LVnumber2 begins with a LVdigit2
       THEN  the subgoal is to then add LVdigit1 and LVdigit2

P2:      IF  the subgoal is to add LVdigit1 and LVdigit2
             and LVsum is the sum of LVdigit1 and LVdigit2
       THEN  the subgoal is to put out LVsum

P3:      IF  the subgoal is to put out LVsum
             and the subgoal is to add LVdigit1 and LVdigit2
       THEN  write LVsum
             and the subgoal is to add the digits after LVdigit1 and LVdigit2

P4:      IF  the subgoal is to put out LVsum
             and the subgoal is to add LVdigit1 and LVdigit2
             and there is a carry
             and LVsum1 is the sum of LVsum plus 1
       THEN  write LVsum1
             and the subgoal is to do the digits after LVdigit1 and LVdigit2
             and remove the carry flag

P5:      IF  the subgoal is to put out LVsum
             and the subgoal is to add LVdigit1 and LVdigit2
             and LVsum is greater than 9
             and LVsum is the sum of LVdigit3 and 10
       THEN  write LVdigit3
             and the subgoal is to do the next digits after LVdigit1 and LVdigit2
             and set the carry flag

P6:      IF  the subgoal is to put out LVsum
             and the subgoal is to add LVdigit1 and LVdigit2
             and there is a carry
             and LVsum is greater than 9
             and LVsum is the sum of LVdigit3 and 9
       THEN  write LVdigit3
             and the subgoal is to do the digits after LVdigit1 and LVdigit2

P7:      IF  the subgoal is to put out the digits after LVdigit1 and LVdigit2
             and the LVdigit1 is followed by a LVdigit3
             and the LVdigit2 is followed by a LVdigit4
       THEN  the subgoal is to add LVdigit3 and LVdigit4

P8:      IF  the subgoal is to add the digits after LVdigit1 and LVdigit2
             and the goal is to add LVnumber1 and LVnumber2
             and LVnumber1 ends with the LVdigit1
             and LVnumber2 ends with the LVdigit2
       THEN  the goal is satisfied
```

general). Productions P4 and P6 do not apply because there is no carry into the first column. One might wonder why P1 or P2 do not apply again since their conditions were satisfied once by data base elements that have not been changed. The current version of the ACT production system does not allow production conditions to match twice to exactly the same data-base propositions. This constraint serves to avoid unwanted repetitions of the same productions and thus some of the danger of infinite loops.

Production P7 applies next, adding *The subgoal is to add 3 and 1 to* the data base so that the next column can be added. Production P2 next applies, finds the sum, and adds *The subgoal is to put out 4* to the data base. Production P4 adds the carry to LVsum and writes out the second digit of the answer, 5. P8 then applies, noting that the problem is finished.

This example illustrates a number of important features of the ACT production system.

(1) Individual productions act on the information in long-term memory. They communicate with one another by entering information into memory.

(2) Productions tend to apply in sequences where one production applies after another has entered some element into the data base. Thus the action of one production can help evoke other productions.

(3) The condition of a production specifies an abstract pattern of propositions in the data base. The more propositions that a condition requires in its pattern, the more difficult it is to satisfy that condition. Similarly, the more a condition relies on constants instead of variables to describe its pattern, the more difficult it is to satisfy that condition.

II Learning in ACT

ACT can learn both by adding propositions to its data base and by adding productions. It can also learn by modifying strengths of propositions and productions. We will concentrate here on the learning that involves productions. Production learning tends to involve the more significant events of cognitive restructuring. It is also through production learning that ACT accounts for schema abstractions.

Productions can be added to the data base in one of two ways. They can be added by deliberate designation as in the encoding of instructions or they can be encoded by spontaneous restructuring of productions in response to experience. We will talk about two varieties of spontaneous restructuring, generalization and discrimination. There is another spontaneous process, strengthening, which adjusts strengths of productions in response to their record of success. Our discussion of learning will be divided to three subsections - one to describe the deliberate designation, another to describe generalization and discrimination, and a third to describe the mechanisms of strength adjustment.

A. Designation

Productions can designate the creation of other productions in their action just as they can designate the creation of propositional structure. We will illustrate the basic idea with an example. Consider how ACT might assimilate the following rules defining various types of LISP expressions (adapted from the second chapter of Weissman, 1967):

1. If an expression is a number it is an atom.
2. If an expression is a literal (a string of characters) it is an atom.
3. If an expression is an atom it is an S-expression.
4. If an expression is a dotted pair, it is an S-expression.
5. If an expression begins with a left parenthesis, followed by an S-expression, followed by a dot, followed by an S-expression, followed by a right parenthesis, it is a dotted pair.

After receiving this instruction ACT will have the sentences expressing these rules represented in its data base. However this representation, by itself, does not allow it to perform any of the cognitive operations that would normally be thought of as demonstrating an "understanding" of these rules. In order to obtain such an understanding, a means of integrating these rules into ACT's procedural knowledge is required. Since these rules have the form of conditionals (antecedent implies consequent), they can be translated in a fairly straightforward manner into the condition-action format of productions. Table 2 illustrates four ACT productions for performing such a translation.[2] Production P9 handles the antecedents of the first four conditionals. For example, P9 matches the segment *If an expression is a number*...of rule (1) by binding LVword to the word *number* and LVconcept1 to the concept @NUMBER that ACT considers underlies that word. Its action is to save the proposition *An object is a @NUMBER* for the condition of a new production.

Insert Table 2 about here

Production P10 is responsible for actually building the productions encoding these rules. It obtains the actions of these new productions from its own processing of the consequent parts of the rules, while the conditions of these new productions have already been identified, so P10 only needs to retrieve them. For example, in the case of rule (1), P10 applies after P9, matching the remainder of the sentence... *it is an atom*. The local variables LVword and LVconcept receive values of *atom* and @ATOM, respectively, in the process of matching. The action of P10 builds the production:

```
P13:      IF  an object is a @NUMBER
        THEN  it is an @ATOM
```

Production P13 is the mechanism by which ACT can actually make the inferences authorized by rule (1).

Productions P11 and P12 are responsible for processing complex conditionals like (5). P11

[2] These productions and some others in this paper embody some clearly over-simplified notions about language comprehension; a more adequate treatment would only distract attention from the learning processes which are the matters of present interest, however. For a discussion of language processing within the ACT framework see Anderson, Kline, and Lewis (1977). (One complication necessary to any complete analysis of language comprehension is, nevertheless, being observed in some of the examples in this paper -- the distinction between words and the concepts underlying them.)

Table 2

A Set of Productions for Encoding Rules about LISP expressions

```
P9:         IF  there is a sentence beginning:  "IF an expression is a LVword..."
                and LVconcept is the concept for LVword
          THEN  save an object is a LVconcept for a new condition

P10         IF  the sentence ends:  "...it is a LVword"
                and LVconcept is the concept for LVword
                and LVcondition is the saved condition
         THEN BUILD IF    LVcondition
                  THEN  it is a LVconcept

P11:        IF  there is a sentence beginning:  "IF an expression begins with a LVword..."
                and LVconcept is the concept for LVword
          THEN  save IF an object begins with an LVconcept for a new condition
                and LVconcept is the last concept

P12:        IF  the sentence continues:  "...followed by a LVword"
                and LVconcept is the last concept
                and LVconcept1 is the concept for LVword
          THEN  add the LVconcept1 is before a LVconcept to the new condition
                and LVconcept1 is the last concept
```

processosos thc first *begins* phrase and P12 each subsequent *followed by* phrase. After the antecedent of the conditional has been entirely processed, production P10 will apply to process the consequent and the designate a production. In the case of rule (5) this production would be:

```
P14:        IF  an object begins with a @LEFT-PARENTHESIS
                and the @LEFT-PARENTHESIS is before a @S-EXPRESSION
                and the @S-EXPRESSION is before a @DOT
                and the @DOT is before a @S-EXPRESSION
                and the @S-EXPRESSION is before a @RIGHT-PARENTHESIS
          THEN  it is a @DOTTED-PAIR
```

This designation process serves in any learning situation as the initial means of introducing productions into the system. Once productions are introduced, the generalization and discrimination processes can operate to create new productions. The designating productions in Table 2 are quite sophisticated. However, one can also propose much more primitive designating productions. For instance, it would not be unreasonable to propose that a child has the following production which encodes a simple principle of reinforcement:

```
P15:        IF  LVevent occurs just before ACT performs LVaction
                and LVaction is followed by reinforcement
  THEN BUILD        IF  LVevent
                  THEN  LVaction
```

B. Generalization and Discrimination

It is the ability to perform successfully in novel situations that is the hallmark of human cognition. For example, *productivity* has often been identified as the most important feature of natural languages, where this refers to the speaker's ability to generate and comprehend utterances never before encountered. Traditional learning theories are generally considered inadequate to account for this productivity and ACT's generalization abilities must eventually be evaluated against this same standard.

While it is possible for ACT to designate new productions to apply in situations where existing ones do not, this kind of generalization requires having designating productions that correctly anticipate future needs. It is plausible that ACT could have such designating productions to guide its generalizations in areas in which it possesses some expertise. However, there are many situations where it would be unreasonable to assume such expertise. For this reason, ACT has the ability to create new productions automatically that are generalizations of its existing productions. This ability, while less powerful than the ability to designate generalizations, is applicable even in cases where ACT has no reliable expectations about the characteristics of the material it must learn.

We will use an example from the schema abstraction literature to illustrate ACT's automatic generalization mechanism. Figure 1 illustrates the stimuli from Experiments 3 and 4 of Franks and Bransford (1971). The 12 figures on the left hand side of the figure were presented to subjects for study. We will assume that *S*s designate productions to recognize each stimulus. So for the first stimulus item subjects would designate the following production:

```
P16:      IF   a triangle is to the right of a circle
               and a square is to the right of a heart
               and the first pair is above the second pair
        THEN   this is an instance of the study material
```

For the third stimulus the following production would be designated:

```
P17:      IF   a circle is to the right of a triangle
               and a square is to the right of a heart
               and the first pair is above the second pair
        THEN   this is an instance of the study material
```

From these two productions a generalization can be formed that captures what these two productions have in common. This involves deleting terms on which the two productions differ and replacing these terms by local variables. Thus, we have the following

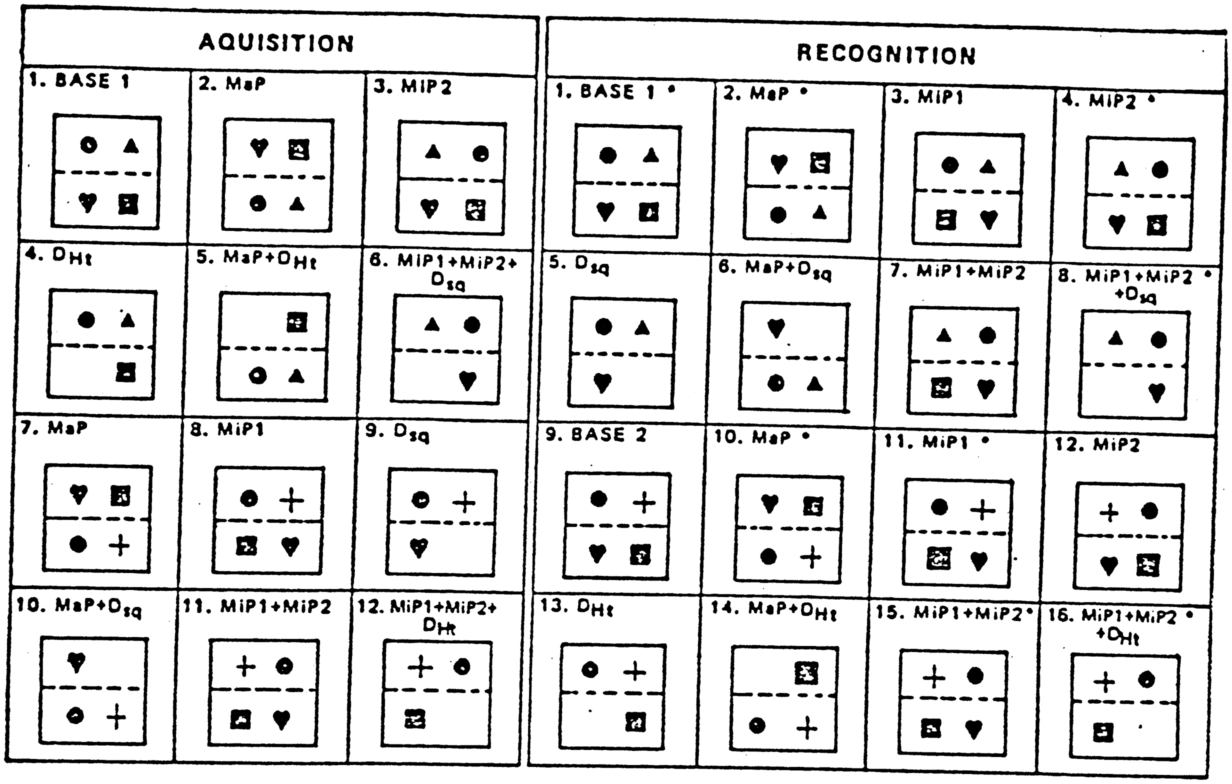

Figure 1: The material used in Franks and Bransford (1971).

generalization:[3]

```
P18:      IF   a LVshape1 is to the right of a LVshape2
               and a square is to the right of a heart
               and the first pair is above the second pair
        THEN   this is an instance of the study material
```

This generalization can be thought of as an attempt on ACT's part to arrive at a more

[3] As discussed in detail elsewhere (Anderson, Kline, & Beasley, in press) there can be many different maximal common generalizations. In this case there is another maximal common generalization besides P18. This generalization preserves the information that there is a *triangle* and a *heart* in both stimuli but consequently looses information about the position of the shapes. This generalization could be rendered in our approximate syntax as:

```
 IF there is a triangle
 and there is a heart
 and a square is to the right of a heart
 and the second pair is below another pair
 THEN this is an instance of the study material
```

In our simulations we will be working with the first generalization.

general characterization of the study material. Note that ACT's generalization mechanism needs only two examples to propose a generalization.[4] This generalization does not replace the original two but rather co-exists with them as an alternate means of characterizing the stimulus set. Which production will actually produce the response depends on the strength mechanism that we will describe shortly.

Restrictions are needed on how many elements can be deleted in making a generalization. Consider, ACT's representation for the sixth stimulus from the Franks and Bransford set:

```
P19:      IF  a circle is to the right of a triangle
              and a heart is to the right of a blank
              and the first pair is above the second pair
        THEN  this is an instance of the stimulus material
```

If we allowed this stimulus to be generalized with stimulus 1 (P16) we would get the following generalization:

```
P20:      IF  a LVshape1 is to the right of a LVshape2
              and a LVshape3 is to the right of a LVshape4
              and the first pair is above the second pair
        THEN  this is an instance of the stimulus material
```

This production will accept any array of geometric objects as an instance of the study material. While it is conceivable that any possible array may be an experimental stimulus, this seems like too strong a generalization to make just on the basis of these two examples. Therefore, a limit is placed on the proportion of constants that can be replaced by variables. In the current system no more than half of the constants in the production with least constants can be replaced by variables in a generalization. The terms that ACT considers

[4] This feature of generalization (two instances to make a generalization) fits well with the following observation about inductions which has been attributed to George Miller (by E. Smith, personal communication): "Suppose one person comes into your office and says, 'I cannot make our appointment. I am going to Brazil.' A second person comes into your office and says, 'Could you teach my class for me, I am going to Brazil.' You immediately ask the question, 'Why is everyone going to Brazil?'"

constants are italicized. There are five constants in productions P16, P17, and P18. Production P18 is an acceptable generalization from P16 and P17 because it only involves replacement of two of the constants. Production P20 is not an acceptable generalization from P16 and P19 because it involves replacement of 4 of the 5 constants.

Even with this restriction on the proportion of constants deleted it is likely that unacceptably many generalizations will be formed. A realistic simulation of an adult human's entire procedural knowledge would require hundreds of thousands of ACT productions. Under these circumstances it would be disastrous to attempt to generalize all possible pairs of productions. ACT only attempts to form generalizations when a new production has been designated. Although no potential generalizations would be missed if a generalization was attempted for each possible pairing of this newly-designed production with existing productions, an enormous computational cost is required even under this scheme. For this reason generalizations are attempted only for pairings of newly-designated productions with the productions on the APPLYLIST. Since a production is on the APPLYLIST only if the constants it references are active and it has met a strength criterion (see p. 3), this implies that attempts to generalize will be restricted to productions that are relevant to the current context and which have enough strength to indicate a history of past success.

Discrimination

Even with these restrictions placed on it, ACT's generalization mechanisms will produce productions that are overgeneralizations of the desired production. However, given our goal of a psychologically realistic simulation, such overgeneralizations on ACT's part are actually desirable since it can be shown that people make similar overgeneralizations. For example, children learning language (and, it appears, adults learning a second language - see Bailey, Madder, and Krashen, 1974) overgeneralize morphemic rules. Thus a child will generate *mans*, *gived*, etc. ACT will do the same. It is also possible that productions will be directly designated in overgeneral form. Thus, for instance, ACT might generate the following rule for predicting rice growing:

```
P21:       IF  the climate of LVplace is warm
               and there is ample rainfall in LVplace
         THEN  LVplace can grow rice
```

This rule is overgeneral in that it fails to specify that the terrain be flat.

To correct overgeneralizations ACT must create more discriminate productions. A production can be made more discriminate either by adding clauses to the condition or by replacing variables by constants. So production P22 serves as a discrimination of P21 by the addition of a clause:

```
P22:       IF  the climate of LVplace is warm
               and there is ample rainfall in LVplace
               and the terrain is flat in LVplace
         THEN  LVplace can grow rice
```

Such a discriminate production does not replace P21 but rather coexists with it. Because of the specificity principle described earlier (p. 5), P22 will apply rather than P21 if both are selected for application.

It is possible for ACT to directly designate such productions to correct overgeneral ones. However, just as in the case of designated generalizations, the existence of the required designating productions is plausible only for domains in which ACT already possesses some expertise. In such domains, ACT could possess the knowledge required to debug its own errors intelligently, but in the majority of cases it will rely on its automatic discrimination mechanism.

ACT's automatic discrimination mechanism requires that it have examples both of correct and incorrect application of a production. This raises the issue of how ACT can get feedback on the operation of its productions. Productions place new propositions into the data base and emit observable responses; either of these actions can be declared incorrect by a human observer or by ACT itself. In the absence of such a declaration an action is considered correct. That is, the only distinction made by the discrimination mechanism is between negative feedback and its absence. Since the way in which ACT declares that the action of a production is incorrect is to apply another production that makes such a declaration as part of its own action, arbitrarily complex ACT computations can be performed to decide the correctness of any particular action.

The discrimination mechanism will only attempt to discriminate a production when it has both a correct and an incorrect application of that production to compare. Basically, this algorithm remembers and compares the variable bindings in the correct and incorrect applications. By finding a variable that had different bindings in these two applications it is possible to place restrictions on that variable that would prevent the match that led to the unsuccessful application while still permitting the match that led to the successful application. Although we have explored other ways of restricting this variable, in the simulations of schema abstraction that will be discussed a new production was formed from the old production simply by replacing the variable by the constant it was bound to during the successful application.

As an example of a discrimination process, we will consider a categorization experiment from Medin and Schaffer (1978). We will focus on two instances they presented from category A. One was two large red triangles and the other was two large blue circles. From these two examples, ACT would designate the following categorization productions:

```
P23:      IF  a stimulus has two large red triangles
        THEN  it is in category A

P24:      IF  a stimulus has two large blue circles
        THEN  it is in category A
```

From these two ACT would form the following generalization:

```
P25:      IF  a stimulus has two large LVcolor LVshapes
        THEN  it is in category A
```

However, this turned out to be an overgeneralization. To be in category A the stimulus had to be either red or a circle or or both. Thus, the counter-example was presented of two large blue triangles which was a stimulus in category B. Generalization P25 misapplied in this circumstance. By noting what distinguished the circumstances of correct applications of generalization P25 from the circumstances of incorrect application, both of the following productions would eventually be formed by the discrimination mechanism. These productions will always produce correct classifications.

```
P26:        IF  a stimulus has two large red LVshapes
          THEN  it is in category A

P27:        IF  a stimulus has two large LVcolor circles
          THEN  it is in category A
```

These productions were formed from P25 by replacing one of its variables by the binding that variable had during a successful application -- (i.e. an application to a stimulus that was actually from category A. As an aside, these two productions illustrate how ACT can encode disjunctive concepts by the use of multiple productions).

C. Production Strength

When a new production is created by the designation process there is no assurance that its condition is really the best characterization of the circumstances in which its action is appropriate. For this reason, generalization and discrimination processes exist to give ACT the opportunity to evaluate alternative conditions for this action. It is the responsibility of ACT's strength mechanisms to perform the evaluation of these competing productions.

Through experience with the ACT system we have created a set of parameters that appear to yield human-like performance. The first time a production is created (by designation, generalization, or discrimination) it is given a strength of .1. Should that production be recreated its strength is incremented by .05. Furthermore, a production has its strength incremented by .025 every time it applies or a production consistent with it applies. (One production is considered consistent with another if its condition is more general and its action is identical.) Finally, whenever a production receives negative feedback its strength is reduced by a factor of 1/4 and the same happens to the strength of all productions consistent with it. Since a multiplicative adjustment produces a greater change in strength than an additive adjustment, a "punishment" is more effective than a "reinforcement".

Note that productions are created out of what might be considered a "reinforcing" event. That is, the designation of production occurs because for some reason ACT considers this to be a "good" rule. Generalization occurs in response to a designation event - that is, generalizations are found by comparing designated productions with productions on the APPLYLIST. Since, designation and generalization can lead to an increase in strength and negative feedback leads to a decrease in strength, the ACT strength mechanism can be seen to have a principle of reinforcement built into it. There is also a principle of exercise - a production gains strength just by applying. This principle is motivated by the observation that behaviors become more reliably evoked and rapidly executed by sheer exercise.

Both decrements and increments in strength generalize to more general productions. This means that if a more general production is created it can rapidly gain strength even if it does not apply nor is it recreated.

It is important to understand how production strength affects performance and how it interacts with specificity. Recall that a production's strength encodes the probability that it will apply. If s is the strength of a production and S the total strength of all productions selected, the probability of that production being chosen on a cycle for application is $1-e^{-bs/S}$ where b is a parameter currently set at 15. Of course, if it is not applied one cycle and the circumstances do not change, it can apply on a later cycle. Thus, strength affects both the latency and reliability of production application.

While selection rules based on strength can make some of the required choices among competing productions, it is clear that strength cannot be the sole criterion. For example, people reliably generate irregular plurals (e.g., *men*) under circumstances in which the "add *s*" rule for regular plurals is presumably also applicable. This reliable performance is obtained despite the fact that the productions responsible for generating regular plurals are applied much more frequently than those for irregulars and therefore should be much stronger. ACT's solution to the problem of exceptions to strong general rules relies on the specificity-ordering principle to decide which productions on the APPLYLIST should actually execute. This principle accounts for the execution of a production generating an irregular plural since its condition presumably contains all of the requirements for generating the regular plural and must, in addition, make reference to the specific noun to be pluralized.

The precedence of exceptions over much stronger general rules does not imply that exceptions always apply, however. In order to benefit from the specificity-ordering principle exceptions must first have achieved the amount of strength necessary to be placed on the APPLYLIST. Furthermore, because the amount of strength necessary depends on the strengths of the other productions that could apply, the stronger a general rule is, the more strength its exceptions need in order to apply reliably. This property of the ACT model is consistent with the fact that words with irregular inflections tend to have high frequencies of occurrence.

Production strength is an important way in which ACT differs from other computer-based learning systems (e.g., Anderson, 1977; Vere, 1977; Hayes-Roth & McDermott, 1976; Sussman, 1975; Winston, 1970; Waterman, 1974). The learning of all these systems has an all-or-none character that ACT would share if creating new productions was its only learning mechanism. Our hope is that strength mechanisms modulate the all-or-none character of production creation in a way that enables ACT to cope with the kind of world that people have to cope with -- a world where data is not perfectly reliable and contingencies change in such a way that even being as cautious as possible it is certain that occasional errors will be made.

D. Review of Critical Assumptions

It is worthwhile, as a review, to state what the critical assumptions are which underlie the ACT learning model.

1. Productions can be designated by other productions.

2. When a production is designated an attempt will be made to generalize it with all the productions in the APPLYLIST.

3. Generalization occurs by replacing constants on which two productions differ by variables.

4. A generalization of two productions will be formed if.they have the same action and if no more than half of the constants in the production with the least constants are replaced by variables in forming a generalization.

5. If a production has a record of both a correct and incorrect application a discrimination will be formed.

6. A discrimination is formed by filling in one variable of the production with the value that variable had during its correct application but did not have during its incorrect application.

7. Upon creation productions are given strength of .1.

8. Upon an attempt to recreate a production its strength is increased by .05.

9. Everytime a production is applied its strength is increased by .025

10. When any of events 7, 8, or 9 occur a strength increment of .025 is inherited by all consistent productions.

11. If a production is found to misapply its strength is decreased by 1/4 as is the strength of all consistent productions.

12. If S is the total strength of all productions selected and s is the strength of a particular selected production, the probability of its being applied if it matches is $1\text{-}e^{-15s/S}$.

13. If two productions on the APPLYLIST both match the data and one is more specific, the

more specific production will apply.

III Applications to Schema Abstraction

There is a growing literature concerned with the process by which subjects form concepts by detecting regularities among stimuli (e.g., Franks & Bransford, 1971; Hayes-Roth & Hayes-Roth, 1977; Newmann, 1974; Posner & Keele, 1970; Reed, 1972; Reitman & Bower, 1973; Rosch & Mervis, 1975). This literature is often referred to as studying prototype formation, hut for various reasons we prefer to refer to it as studying schema abstraction.

There are a number of features of this research area that distinguish it from the related research area that is often called concept formation: In the concept formation literature the concept that is to be discovered is usually quite simple (e.g. *red and a triangle*) and subjects are often able to verbalize the hypotheses they are considering at any point. In contrast, the concepts used in the schema abstraction literature may be quite complex. For example, these concepts might be defined in terms of a linear discriminant function (e.g. Reed, 1972) or solely by a listing of the exemplars (e.g., Medin & Schaffer, 1978). Subjects will often emerge from such experiments without being able to verbalize the criteria they are using to correctly classify instances. Their instructions may even suggest that they should avoid formulating explicit hypotheses and should simply study the instances one-by-one. Within the ACT framework there is a corresponding distinction between forming a concept by the action of a general set of productions for hypothesis testing versus forming a concept by the action of the automatic learning mechanisms of generalization, discrimination, and strengthening.

Our intention in the rest of this paper is to show that ACT's automatic learning mechanisms have a straightforward application to schema abstraction. In outline, this application is as follows: For each instance presented ACT designates a production that recognizes and/or categorizes that instance alone. Generalizations occur through the comparison of pairs of these productions. If feedback about the correctness of these generalizations is provided then the discrimination process can be evoked. Our working definition of a concept will be this set of designations, generalizations, and discriminations. It turns out that such sets of productions nicely capture the family resemblance structure that has been claimed for natural categories (e.g. Rosch & Mervis, 1975). It also turns out that ACT simulations can account for the results of various experiments in the literature on schema abstraction.

A. Franks and Bransford: Illustration of Basic Phenomena

We have already introduced (Figure 1) the material used by Franks and Bransford in one of their experiments on schema abstraction. Subjects studied the 12 pictures on the left of Figure 1 twice and then were transferred to a recognition phase in which they had to give recognition ratings of the 16 figures on the right of Figure 2 plus 6 other figures, called non-cases, which violated the rules under which the cases were generated. The 16 test cases in Figure 1 were generated by applying 0, 1, 2, or 3 transformations to the base figures. Half of these 16 were actually studied and half were not. While Franks and Bransford do not report subjects' performance for each stimulus, they do report that confidence ratings for recognition generally decreased with the number of transformations and was lowest for the non-cases.

We attempted to simulate the Franks and Bransford experiment by having ACT go through propositional encodings of the items in the study set twice, designating a recognition production for each stimulus it saw.[5] Then at test ACT was again presented with propositional encodings of the stimuli and the production which applied (if any) was noted. Sufficient generalization had occurred so that most of the stimuli were recognized by at least one of the productions.

A critical question was how to map the production selected onto a confidence rating. We assumed that ACT's confidence would be a function of the number of constants in the stimulus (and therefore an inverse function of the number of variables). This procedure for assigning confidence will be used throughout this paper. This is a reasonable procedure for assigning

[5] The simulations were not performed with the general purpose ACT simulation program, but rather with a special purpose simulation which runs about 10 times faster. This special simulation does not have all the general computational features of ACT. Rather, it is especially designed to allow us to follow only the interaction of strengthening, discrimination, and generalization.

confidence, since the more constants in the recognizing production the closer it is to an encoding of an actual test item. In the extreme, if the stimulus is recognized by a production with no variables the subject can be sure that the item was studied since a non-variabilized production is an encoding of a study item.

Note that this procedure for assigning confidence implicitly weights the strength of productions as well as their number of constants. Since strength of productions determines whether a production is selected, the stronger the productions that can classify an instance the more of these productions that will be selected and, thus, the more likely it is that a production with many constants will be selected. This increased probability of selecting a production with many constants translates quite directly into an increase in the probability of a high confidence rating because of ACT's preference for applying the most specific productions that have been selected. We have given some thought to the possibility that strength should have more than an implicit role in assigning confidence. That is, confidence could be made a joint function of number of constants in a production that applies and the strength of that production. Considering a production's strength in assigning confidence could be justified by the fact that strength reflects the production's past success in classifying instances and therefore predicts how successful the current application will be. We have not gone to this more complex procedure for assigning confidence mainly because we have been able to account for all the results just using the number of constants.

Consider again production P16 (on p. 10) which encodes the first item in the stimulus set:

P16: IF a *triangle* is to the right of a *circle*
and a *square* is to the right of a *heart*
and the first pair is *above* the second pair
THEN this is an instance of the study material

The five constants that can be replaced by variables are italicized. If this production applied, ACT would assign a confidence rating of 5 to its recognition of that stimulus. If all five constants were replaced by variables we would have a production that would recognize anything and if this applied we would assign a confidence of 0. For shorthand, we will denote the production above as TCSHA where each letter is the first letter of one of the constants. Variables will be denoted by hyphens. Therefore, production P18 (on p. 11) would be denoted --SHA.

To obtain predictions for this experiment we ran ten ACT simulations. Each simulation involved giving ACT a study phase and then following this with five passes through the test material. Since the process of production selection is probabilistic, ACT's ratings varied from one test to another. Altogether we obtained fifty ratings for each test stimulus and the data we report will be based on averages of these fifty ratings. The practice of having five test trials for each study represents a departure form the Franks and Bransford experiment. However, since the study phase was relatively expensive in computational terms, it made sense to get as much data as possible from each study phase that was simulated.

The numbers that were obtained from these simulations depend on the rather arbitrary values for the strengthening parameters that were detailed earlier (p.p. 17, 18).[6] It is currently impractical and probably premature to perform a search of the parameter space to determine the best fitting parameters. For this reason, we used these arbitrary values for all of the simulations that will be reported and had to be content to predict the relative ordering of conditions rather than their exact values.

The test stimuli identified as base or 0-transformations (1, 9 in Figure 1) were given a mean rating of 1.66 (i.e. mean number of constants in matching productions); the test stimuli (2-5, 10-13) identified as one transformation away from the base were rated 1.24; the stimuli (6, 7, 14, 15) identified as two steps away were rated 1.11; the stimuli (8, 16) three steps away were value 1.13; and the non-cases were rated .65. This corresponds to Franks and Bransford's report of an overall correlation between closeness to base and rating. (Franks and Bransford do not report the actual ratings.)

[6] One additional parameter besides those discussed earlier is required. If ACT had all of the productions that would be needed to account for a subject's total procedural knowledge, some of these, although irrelevant to the schema abstraction task, would be selected anyway and their strengths would contribute to S in assumption 12 (p. 21). For all of the simulations reported in this paper the contribution of such irrelevant productions to S was set to 20.

Neumann (1974) performed a replication of Franks and Bransford and he did report mean ratings for each of the five categories of test stimuli. Subjects assigned ratings of +1 to +5 to the stimuli that they thought they recognized and assigned ratings of -1 to -5 to stimuli they did not recognize. Mean ratings were 2.79 for base stimuli, 2.18 for 1-transformation stimuli, .49 for 2-transformation stimuli, .90 for 3-transformation stimuli, and -.26 for non-case stimuli. While the ordering ACT scores corresponds perfectly to the ordering of these mean ratings, a comparison of the exact values is not meaningful because the scales are different. Some monotonic transformation is required to convert the ACT scores which are based on the number of constants in the recognizing production into the -5 to +5 confidence scale used by Neumann's subjects. If the transformation from ACT match score to confidence were linear there should be a strong correlation between the two measures. In fact, the correlation is .927 suggesting such a linear transformation might not be that far from the truth.

This experiment does not provide a particularly telling test of the ACT learning model, but it is a good introduction in that it serves to illustrate that ACT can account for one of the basic phenomena of schema abstraction -- namely that confidence falls off with distance from the stimuli that are the central tendency of the category. Subsequent experiments will deal with the issue of whether the details of ACT's abstraction process correspond to the details of human abstraction.

To help understand how ACT accounts for preference for central stimuli like 1 or 9, consider Figure 2 which compares the specificity network around test stimulus 1 during one of the ten simulations (Part a) with the specificity network around test stimulus 8 (Part b). In our notation, test stimulus 1 is ACTHS and test stimulus 8 is ATCBH. Both were presented twice during study and so have strength .15. However, ACTHS is more similar to other stimuli and so has entered into more generalizations. Hence, there is a denser network above ACTHS. (Actually, the network around ACTHS is even denser than Figure 2 but we have eliminated some of the generalizations to make the figure easier to read). ATCBH differs from all other stimuli on at least two dimensions. There are no 1-variable productions above

ATCBH. On the other hand there are two 1-variable productions (ACT-S and -CTHS) above ACTHS with a combined strength of .40. ACTBH does have two 2-variable productions above it (A-C-H and ATC--), but their combined strength of .325 is still much less than the combined strength of 1.475 possessed by the four 2-variable productions above ACTHS (A--HS, -CT-S, -C-HS, AC-H-; only three of these are illustrated). A similar picture is obtained when we look at the 3- and 4-variable generalizations: There are two 3-variable productions above ATCBH (A---H and A-C--) with strength 1.025; but there are six 3-variable productions above ACTHS (A---S, ---HS, -C--S, -C-H-, AC---, A--H-; only four of these are illustrated) with total strength 3.4. Finally, ATCBH was involved in no 4-variable generalizations while ACTHS is involved in three (----S, -C---, ---H-) with total strength 3.25. Table 3a summarizes these comparisons.

Under some approximating assumptions, it is possible to derive the expected match values from these strengths. Assume that if a n-variable production is selected which matches the stimulus, it will apply in preference to all n+1 variable productions. This assumption is an approximate realization of ACT's specificity ordering. Let $Q_{i,s}$ be the probability of at least one i-variable production being selected for stimulus S. The probability $P_{i,s}$ that one of the i-variable productions will be the one that applies to classify stimulus S is:

$$P_{i,s} = Q_{i,s} [1 - \sum_{j=0}^{i-1} P_{j,s}] \qquad (1)$$

That is, the probability that a i-variable production will be the one to apply is the probability that a i-variable production is selected times the probability that no more discriminate production is also selected. Then expected rating for stimulus *S* is:

$$R_s = \sum_{i=0}^{4} (5-i)P_{i,s} \qquad (2)$$

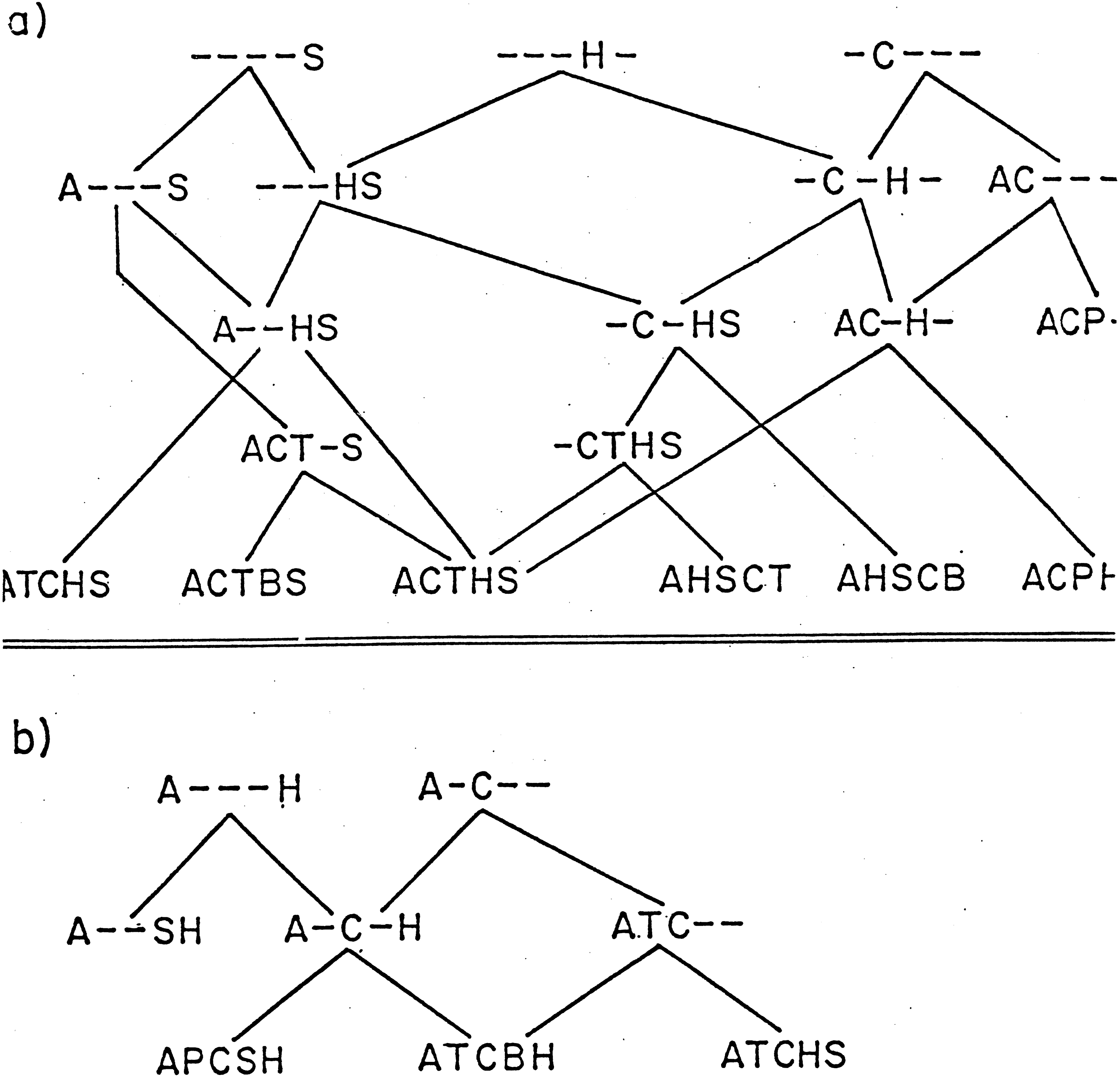

Figure 2: Part a illustrates the specificity network around stimulus 1 (ACTHS) and part b illustrates the specificity network around stimulus 8 (ATCHS).

By the end of this experiment, the total strength of all productions, relevant and irrelevant, was about 35. Therefore, according to assumption 12 in Section IID, if $t_{i,s}$ is the strength of all of the productions matching S that have i variables, then the probability of at least one being selected is:

$$Q_{i,s} = 1 - e^{-15t_{i,s}/35} \quad (3)$$

From equation (3) we can derive the probabilities of selecting productions with various numbers of variables and these are given in Part (b) of Table 3.

Insert Table 3 about here

From these values we can calculate by equation (1) the probabilities of applying an i-variable production, $P_{i,s}$, subject to the specificity restriction. These probabilities are given in Part c of Table 3.. Substituting these values into equation (2) yields the expected confidence ratings:

$$R_{ACTHS} = 2.730$$
$$R_{ATSBH} = 1.194$$

In actual fact, the rating difference between 0-transformation stimuli like ACTHS and 4-transformation stimuli like ATSBH is considerably less than this expected difference. This can be shown to be due to the following fact: If two productions are selected that match a stimulus and neither has a condition that is a subset of the other, the one to apply is determined probabilistically by relative strength and not number of non-variable condition elements. Thus, unlike our analysis, it is not always the production with the least number of variables that applies. For instance, if ---H- and ACT-S are both selected, the more variabilized ---H- may apply because neither production is above the other in the specificity network. Nonetheless, the above analysis does illustrate in approximate terms why 0-transformation stimuli get better ratings than the non-central stimuli.

Table 3

Analysis of the Differences between the stimuli ACTHS and ATSBH

(a) Strengths of classifying productions with different numbers of variables

	ACTHS	ATSBH
0-variables	.150	.150
1-variable	.400	-
2-variables	1.475	.325
3-variables	3.400	1.025
4-variables	3.250	-

(b) Probabilities of selecting productions with different numbers of variables

	ACTHS	ATCBH
$Q_{0\cdot}$	.062	.062
$Q_{1\cdot}$	.158	-
$Q_{2\cdot}$	.469	.130
$Q_{3\cdot}$	.767	.356
$Q_{4\cdot}$	.752	-

(c) Probabilities of applying productions with different numbers of variables

	ACTHS	ATCBH
$P_{0\cdot}$	.062	.062
$P_{1\cdot}$	.148	-
$P_{2\cdot}$	.371	.122
$P_{3\cdot}$	.321	.290
$P_{4\cdot}$	.073	-

B. Hayes-Roth and Hayes-Roth: Variation of Instance Frequency

One of the interesting features of the ACT simulation of the Franks and Bransford experiment is that the ratings of the 3-transformation stimuli are predicted to have slightly higher ratings than the 2-transfromation stimuli and this prediction was confirmed in the data of Neumann. ACT makes this prediction because both of the 3-transformation stimuli were presented for study while only one of the four 2-transformation stimuli was studied. It is weak memory for the instances that were studied which gives the 3-transformation stimuli this slight advantage. The Franks and Bransford paradigm has not been systematically studied for instance memory, but the ACT simulation predicts a weak advantage for studied stimuli over comparable non-studied stimuli.

Hayes-Roth and Hayes-Roth (1977) report a study, one function of which was to obtain data relevant to the issue of memory for instances. They presented subjects with three-attribute descriptions of people. One attribute was age and could have values 30, 40, 50, and 60. Another was education and could have values junior high, high school, trade school, college. The third was marital status which could have values single, married, divorced, widowed. Subjects were also given proper name and hobby but these dimensions were not critical. Thus, a subject might hear the description "John Doe, 30 years old, junior high education, single, plays chess." Subjects' task was to learn to classify these individuals as members of club 1, members of club 2, or neither club.

The four values of each dimension will be represented symbolically by the numbers 1 - 4. The assignment of the symbolic values 1 - 4 to the values of each dimension was randomized for each subject. In our discussion we will refer to stimuli by these numbers. Thus "111" might refer to "40 years, high school, single." The rules determining assignment of individuals to clubs were as follows:

1. If one of values was a 4, the individual belonged to neither club.
2. If there were more 1's than 2's and no 4's the individual was assigned to club 1.
3. If there were more 2's than 1's and no 4's the individual was assigned to club 2.

4. If there were as many 1's as 2's the individual was assigned with a 50% probability to club 1 and with 50% probability to club 2.

Thus, 1's were diagnostic of club 1, 2's were diagnostic of club 2, 3's were don't cares, and 4's disqualified club membership. A prototypical member of club 1 would be 111 and a prototypical member of club 2 would be 222. These prototypes were never presented.

We will assume that for each individual encountered, subjects designated a production mapping that individual's features into a prediction about club membership. So, for instance, a subject might form the following production:

```
  If  a person is forty years old
      and he has gone to high school
      and he is single
Then  he is a member of club 1
```

Or, more symbolically, we will represent this production as 111→1.

Hayes-Roth and Hayes-Roth varied the frequency with which various exemplars were studied and Table 4 shows these frequencies. A study trial consisted of first presenting the subject with an exemplar, asking him to classify it, and then providing feedback as to the correctness of the classification. In the case of equivocal exemplars like 132 the subject was given feedback half the time specifying club 1 and half the time specifying club 2. The feedback aspect to this experiment is a significant difference from the Franks and Bransford experiment. Negative feedback will lead to the evocation of ACT's discrimination mechanism which was silent during the earlier simulation.

Insert Table 4 about here

Table 4 also indicates which items were tested. Subjects were first asked to categorize each of the stimuli and then they were asked to decide whether each of the stimuli had been

Table 4

Initial Classification Exemplars and Test Items in Hayes-Roth and Hayes-Roth (1977)

Exemplar	Club	Number of Initial classifications	Tested for recognition and final classification
112	1	10	Yes
121	1	10	Yes
211	1	10	Yes
113	1	1	Yes
131	1	1	Yes
311	1	1	Yes
133	1	1	Yes
313	1	1	Yes
331	1	1	Yes
221	2	10	Yes
212	2	10	Yes
122	2	10	Yes
223	2	1	Yes
232	2	1	Yes
322	2	1	Yes
233	2	1	Yes
323	2	1	Yes
332	2	1	Yes
132	Either	10	Yes
321	Either	10	Yes
213	Either	10	Yes
231	Either	0	Yes
123	Either	0	Yes
312	Either	0	Yes
111	1	0	Yes
222	2	0	Yes
333	Either	0	Yes
444	Neither	0	Yes
411	Neither	1	No
422	Neither	1	No
141	Neither	1	No
242	Neither	1	No
114	Neither	1	No
224	Neither	1	No
441	Neither	1	No
442	Neither	1	No
144	Neither	1	No
244	Neither	1	No
414	Neither	1	No
424	Neither	1	No
134	Neither	1	No

Table 4, continued

234	Neither	1	No
413	Neither	1	No
423	Neither	1	No
341	Neither	1	No
342	Neither	1	No
124	Neither	1	No
214	Neither	1	No
412	Neither	1	No
421	Neither	1	No
241	Neither	1	No
142	Neither	1	No
143	Neither	1	No
243	Neither	1	No
314	Neither	1	No
324	Neither	1	No
431	Neither	1	No
432	Neither	1	No

studied or not. The recognition judgment was assigned a confidence from 1 - 5 as was the categorization judgment.

Table 5 gives the mean recognition ratings as well as mean categorization ratings for seven different classes of stimuli. The recognition ratings were averages formed by weighting rejection confidences negatively and acceptance confidences positively. The categorization ratings were averages formed by weighting negatively the confidences ascribed to incorrect category assignments and weighting positively the confidences ascribed to correct category assignments.

Insert Table 5 about here

The first class in Table 5 is formed from two prototypes which were never in fact studied. They receive the highest categorization rating and a relatively high recognition rating, indicating that subjects have extracted the central tendency of this set. The second class consists of the non-prototypes which have received ten study trials each. They have the highest recognition ratings, reflecting their high degree of exposure, and the second highest categorization rating. They get higher recognition ratings than the third class which is closer to (or as close to) the prototype. This reflects some residual instance memory. The third class would perhaps be regarded as closer to the prototype than the second because its members have "don't-care" elements rather than an element that directly violates the category's prototype. The third class is clearly closer to the prototype than the fourth whose members have two don't care items. The third and fourth classes have one exposure of each member, but the third class receives a higher rating reflecting the fact it is closer to the prototypes. The fifth class is equivocal between the two categories and probably is further from either prototype than are classes 3 or 4. Still it is given higher recognition ratings than classes 1, 3, or 4 reflecting its greater exposure. However, it does get a lower rating than class 2 despite the fact that members have the same frequency of exposure. This may be due to distance from prototype or the equivocal response assignment in study.

Table 5

Recognition and classification from Hayes-Roth and Hayes-Roth compared to ACT's match scores

	Recognition		Classification	
	Subject's degree of confidence	ACT's degree of match	Subject's degree of confidence	ACT's degree of match
1. Non-Practiced Prototypes (111,222)	1.00	.94	2.61	.94
2. Much Practiced Non-Prototypes (112,121,211, 221,212,122)	2.53	1.46	2.34	.86
3. Little Practiced Close-to-Prototype (113, 131, 311, 223, 232, 322)	.03	.70	2.27	.70
4. Little Practiced Far-from-Prototype (133, 313, 331, 233, 323, 332)	-2.25	.42	2.01	.41
5. Much Practiced Equivocal (132, 321, 213)	1.34	1.25	-	-
6. Non-Practiced Equivocal (231, 123, 312)	-.93	.46	-	-
7. Non-Practiced Anti-Prototypes (333, 444)	-2.52	.07	-	-

Categorization ratings are not meaningful for class 5 nor are they for classes 6 or 7. Class 6 is just as equivocal as class 5 but was never studied so it receives lower recognition ratings. The lowest recognition ratings are reserved for class 7 which contains non-presented instances composed of all 3's or all 4's.

There two features to emphasize about this data. First, ratings are influenced by a rather complex mixture of frequency of exposure and closeness to prototype. Second, the rank orderings of the recognition and classification data are not identical. Therefore, these data should provide a challenging test for the ACT simulation program.

Simulation

This experiment was simulated with the same parameter settings as the Franks and Bransford experiment. The one significant difference was that ACT was given feedback about the correctness of its classifications. This meant that productions would not simply increase in strength with every application, but rather would either increase or decrease in strength depending on their success in classification. Providing feedback also meant that it was possible for ACT to compare variable bindings on successful applications in order to produce more discriminate versions of its overgeneral productions. A study session consisted of passing through 132 classify-then-feedback trials presented in random order. After this the 28 test stimuli were presented in random order five times. This whole procedure was repeated ten times. The data we will report is averaged from the fifty test trials given to each stimuli.

As in the Franks and Bransford experiment, confidence was based on the number of constants in the production that recognized the stimulus. In this experiment that number would vary from 1 to 3. A value of 0 was assigned if no production was evoked to categorize the stimulus. These mean match scores are reported in Table 5. The categorization scores were taken by weighting negatively the confidences of incorrect classifications and weighting positively the confidences of correct classifications and ignoring the confidences of classifications to the neither-club category. Class 2 received a classification rating that was much lower than its recognition rating. This reflects the application of productions assigning the stimuli to the wrong category. Such productions were formed through the generalization process. For example, generalizing 121→1 with 321→1 would yield the production -21→1 which would misclassify the instance 221.

The general hypothesis is that the ACT scores will be monotonically and perhaps linearly related to the obtained ratings. The monotonic hypothesis is clearly confirmed in that ACT perfectly predicts the rank ordering of the seven recognition scores and the rank ordering of the four classification scores. The linear hypothesis also fares quite well - a correlation of .968 is obtained for the recognition scores and of .948 for the classification scores.

Hayes-Roth and Hayes-Roth present a model for their data which is quite similar to the ACT model. (We will discuss similarities to other models at the end of the paper). They derive a set of pairwise comparisons among conditions which their model better predicts than any of a large class of categorization models. ACT's predictions correspond exactly with those of Hayes-Roth and Hayes-Roth on these pairwise conditions. However, the ACT model is more powerful than theirs, predicting the complete ordering of conditions and offers a possibility of assigning an interval scale to that ordering. They are unable to do this on the basis of their model, but it is something that falls out of a theory which has a computer simulation.

One important aspect of the ACT simulation of this experiment is its prediction of better performance on the class 5 stimuli than on the class 3 stimuli, despite the fact that both types of stimuli we presented equally frequently. The reason for this is the equivocal nature of the response assignment for class 5 which results in punishment of the productions that classify these stimuli and the consequent weakening of these productions. Most of the ACT predictions for the experiments under discussion rely on the generalization mechanism or discrimination and generalization in concert. This, however, is an instance of a result which depends solely on the discrimination mechanism.

C. Medin and Schaffer: Effects of Inter-item similarity

An interesting series of experiments has been performed by Medin and Schaffer (1978) who show that under some circumstances, how typical an instance is considered of a category depends, not on how close it is to the central tendency of the instances in the category, but rather how close it is to specific instances in the category. Particularly important is whether there are any category members which are very similar to this instance. Their experiments are also interesting because they report data on the time it takes to learn to make a classification.

They presented subjects with stimuli that took one of two values on four dimensions: color (red or blue), form (circle or triangle), size (large or small), and number (1 or 2). As in the Hayes-Roth and Hayes-Roth experiment these stimuli are best referred to abstractly with the numbers 0 and 1 for the values on each dimension. Values were randomly assigned to number for each subject. Thus, for one subject a 1101 might be a single small red circle. Subjects had to learn to classify these as members of category A or category B. The material was always designed so that 1111 was the central tendency for category A and 0000 was the central tendency for category 2.

1. Experiment 1

Table 6 illustrates the material for Experiment 1. The A training stimuli were designed so that for each dimension there are two training stimuli that have values of 1 on that dimension. The B training stimuli were similarly designed so that two 0 values can be found for each dimension. Thus the A prototype would be 1111 and the B prototype would be 0000. Subjects were trained in categorizing the material until they had correctly categorized all six twice in a row or until twenty trials through the six items expired. Then subjects were given transfer trials in which they saw the six old stimuli plus six new ones. Subjects' task was to indicate what category each stimulus came from. The categorization judgments were made on a 3 point scale varying from 1 = guess to 3 = high confidence. Medin and Schaffer transformed these scores to a 6 point scale where 1 = high confidence wrong and 6 = high confidence correct. Subjects made categorization judgments shortly after study and after a weeks' delay. The mean scores, averaged over immediate and delay as reported by Medin and Schaffer, are in Table 6. A value of 3.5 reflects chance performance.

Insert Table 6 about here .

Medin and Schaffer were particularly interested in transfer to the new stimuli. They predicted higher performance on the A transfer stimuli than on the B transfer stimuli even though the stimuli are all equally similar to their prototypes. They made this prediction because the A transfer stimuli agree in three positions with two of the study items (0111 with 1111 and 0101, 1101 with 1111 and 0101, 1110 with 1111 and 1010) while the B transfer stimuli agree in three positions with only one study item (all with the prototypical 0000). Moreover, each of the B transfer stimuli agree in three positions with an A study stimulus (1000 with 1010, 0010 with 1010, 0001 with 0101). The Medin and Schaffer predictions were verified.

ACT simulations of this experiment were performed with the same parameter settings as the previous experiments. Each simulation involved training ACT to criterion or until the twenty trials were up. Then, five test passes through the twelve items were administered to get classification ratings for each item. The strength of each production was then reduced by 50% to simulate the loss of strength with a week's delay and five more ratings were obtained for each stimuli. Ten such simulations were performed. Therefore, the ACT match ratings are calculated on 100 ratings per stimulus. The number of constants in the classifying production (weighted positively for correct classification and negatively for incorrect ones) was again taken to be ACT's confidence rating. Table 6 gives ACT results in terms of trials to criterion and mean match ratings. The ACT trials to criterion provide a good, but not perfect, rank order correlation (r=.89) with the actual data. Similarly, the ACT match scores provide a good, but not perfect, rank order correlation (r=.88) with the actual classification ratings. The

Table 6

Stimuli used in Experiment 1 of Medin and Schaffer (1978), number of errors on training stimuli, classification confidences, and ACT simulation

	Errors in Original Learning		Final Categorization	
	Data	ACT	Data	ACT's Match
A Training Stimuli				
1111	3.6	2.1	4.8	2.38
1010	4.7	3.8	4.6	2.28
0101	4.4	3.6	4.8	2.20
B Training Stimuli				
0000	3.1	3.3	5.2	2.79
1011	4.9	6.6	4.5	.81
0100	3.8	3.3	4.9	2.65
A Transfer Stimuli				
0111			4.3	1.22
1101			4.4	1.26
1110			3.6	1.57
B Transfer Stimuli				
1000			3.5	.00
0010			4.0	.00
0001			3.2	.00

linear correlation between the match scores and actual rating scores (r=.83) is again fairly high suggesting the possibility of a linear transformation of one into the other. Note that in simulating this experiment, unlike Franks and Bransford or Hayes-Roth and Hayes-Roth, ACT has the more demanding task of predicting the data obtained for individual stimuli. The less than perfect correlations may reflect this but they may also reflect that both the data points it is trying to predict and its own estimates of those data points tend to be less reliable than in previous simulations.

One consequence of the small number of stimuli in this experiment is that it is possible to consider the total set of classifying productions that are generated by ACT's automatic learning mechanisms. Figure 3 illustrates that conditions of both the A-response productions and the B-response productions arranged according to their specificity ordering. As for the A-response productions, the 1111 and 0101 productions generalize to form the -1-1 production. Also, the 1111 and 1010 productions generalize to produce a 1-1- production. This production can misapply in training and match the 1011 B stimulus. This mistake can evoke the discrimination process and so give rise to 1-10 and 111- productions which discriminate between the successful and unsuccessful contexts of application of the 1-1- generalization. These discriminations did not appear in all the simulation runs as they depended on a particular sequence of events happening and ACT sometimes reached learning criterion before this sequence was complete.

As for the B-response productions, there is only one generalization: 0000 and 0100 can combine to form 0-00. Note that a generalization could be formed from 0000 and 1011 which would be -0--. However, this would involve replacing more than 50% of the constants by variables. In other words, this generalization is not allowed because the productions it merges are just too dissimilar. Note that none of the productions in Figure 3 can match the B transfer stimuli. This accounts for their low rating. In contrast, at least one of the A generalizations match each of the A transfer stimuli: -1-1 matches 0111 and 1101, while 1-1-, 1-10, and 111- all match 1110. This accounts for the higher rating of the A transfer stimuli. Medin and Schaffer had constructed the material so that the A transfer stimuli would

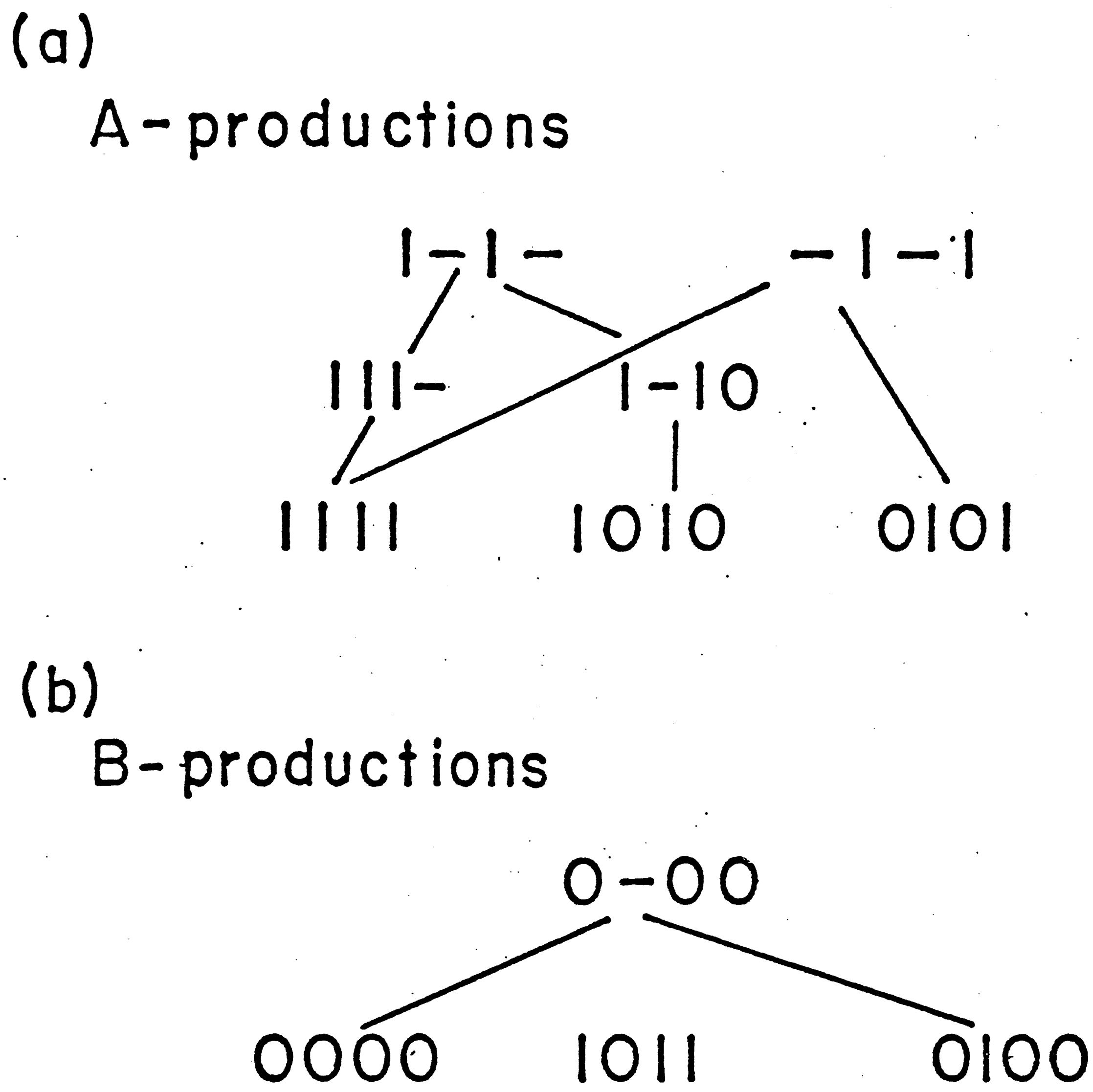

Figure 3: Part a illustrates the specificity network of A productions and part b illustrates the specificity network of B productions.

be closer to study items than the B transfer stimuli. The consequence in ACT is that the A transfer stimuli are closer to a number of the generalizations that arose from the study elements.

2. Experiments 2 and 3

Medin and Schaffer used very similar procedures for experiments 2 and 3. As in experiment 1 there were four dimensions with two values on each. However, in these experiments there were more study and test stimuli. Experiment 2 used the same geometric stimuli as Experiment 1 while Experiment 3 used Brunswik faces that varied in the dimensions of nose size, mouth height, eye separation, and eye height. There were two procedural differences between these two experiments and 1. First, the criterion for passing out of the study phase was one correct pass through all nine study stimuli or 16 total passes through the material (32 passes in experiment 3). The second procedural difference was that there was no delayed test at a week.

The ACT simulation was basically the same as for experiment 1 with two changes to reflect the procedural changes. First, we used the criterion of one correct pass or 20 total passes (a compromise between the 16 in Experiment 2 and the 32 in Experiment 3). Second, there was no attempt to simulate performance at a delay since Medin and Schaffer do not collect such data.

Table 7 presents the data from the two experiments and from the ACT simulation. Transfer stimuli were classified as A or B by Medin and Schaffer according to a linear discriminant function calculated to separate the A and B training stimuli. In general, subjects learned more slowly in Experiment 3 with the faces than Experiment 2 with the geometric stimuli. This may be due to the fact that the face material had distracting irrelevant dimensions. In any case, we used just one simulation run of ACT to fit both sets of data. As discussed earlier, our concern is to be able to reproduce the ordinal trends in the data, and not to perform the kind of parameter search required to get exact fits.

Again the prototype of Category A is 1111 and for Category B it is 0000. Medin and Schaffer were particularly interested in the contrast between the A training stimuli 1110 and 1010. While 1110 is closer to the A prototype than 1010, 1010 is closer to the A training instances. For example, the only A training stimulus that 1110 is one feature removed from is 1010, and it is this close to two of the B stimuli, 1100 and 0110. By contrast, 1010 is one feature removed from the two A training stimuli 1110 and 1011 and there are no B training stimuli one feature distant. As they predicted performance was higher on 1010 when measured either by the number of errors on training trials or by the subsequent classification ratings. ACT predicts this because a 1 - 10 generalization will be formed from the 1110 and 1010 combination and a 101- generalization will be formed from the 1010 and 1011 combination which will help classify 1010. In contrast, there is only one three-item generalization (101-) to classify 1011 and there is a B generalization (e.g., -1-0) that will misclassify the 1110 stimulus.

In general, ACT does a good job of predicting the rank orderings of the error data. ACT's rank ordering correlates .88 with the ordering in experiment 2 and .80 with experiment 3. It is worth noting that the rank orderings of experiments 2 and 3 only correlate .85 with each other. So ACT is doing about as well as could be expected without introducing a lot of additional machinery about the salience of individual dimensions. As for rank orderings of classification data, ACT's match scores correlate .79 with Experiment 2 and .89 with Experiment 3. The two experiments only correlate with each other .77. Another test was performed of the hypothesis that the ACT match scores were related to the confidence ratings by a linear transformation. The correlations between the actual ratings and ACT's match scores were .73 for Experiment 2 and .81 for Experiment 3.

Insert Table 7 about here

----

3. Experiment 4

The final experiment we simulated was Experiment 4 from Medin and Schaffer which used geometric stimuli again. The materials for this experiment are illustrated in Table 8. Subjects were given a maximum of 16 passes through the material to achieve the criterion of one perfect recall. ACT was run given the same 16 trial limit. Table 8 also presents the data from the experiment and from the ACT simulation.

Again a linear discriminant function was calculated to separate A from B training stimuli and then used to classify the transfer stimuli. Again 1111 would be regarded as the prototype for the A stimuli and 0000 for the B stimuli. Despite this, Medin and Schaffer predicted that subjects would display better performances on a number of A stimuli than on their B counterparts -- 0110 better than 1001, 0111 better than 1000, 1101 better than 0010, 1011 better than 0100, and 1111 than 0000. As can be seen, ACT makes these same predictions. Medin and Schaffer made these predictions on the basis of the number of other stimuli similar to the favored A instances. ACT makes these predictions because if there are a large number of similar stimuli generalizations will be made. These predictions are supported by the data except for the 0110 vs. 1001 contrast.

Insert Table 8 about here

The correlation between the rank order of ACT errors and the rank order of the data is fairly high (r=.62). The rank order correlation with classification ratings and ACT match scores is somewhat higher (r=.79). Again as a test of a linear relation we performed a correlation between the actual ratings and match scores. This correlation was even higher (r=.83).

4. Summing Up Medin and Schaffer Experiments

Medin and Schaffer designed their experiments to show the inadequacies of an independent cue theory which creates a prototype out of the modal values on each dimension and assigns rank orderings according to distance from these prototypes. Their data clearly refute such a model and indicate that subjects are sensitive to similarities among individual instances. Fortunately, ACT lines up with Medin and Schaffer in predicting this result. Medin and Schaffer's theory is that subjects only store instances and that ratings are particularly influenced by what instances are close to a test instance. ACT's ratings are also influenced by what instances are close to a test instance because these result in generalizations that will classify the test instance.

Medin and Schaffer derived predictions from their theory and compared these with predictions from an independent-cue-prototypes model. Rank order correlations were reported between these models and their data. It is interesting to compare the correlations of these two models with ACT. The three sets of rank order correlations are reported in Table 9 for Experiments 2, 3, and 4 (Medin and Schaffer do not report correlations for Experiment 1). There are two remarks that need to be made about interpreting these data. First, Medin and Schaffer's correlations concern percent-correct classification while ACT's previously reported classification correlations concerned confidence ratings. The ratings and percent correct are not perfectly correlated. We chose to report correlations with ratings because this measure tends to be more informative. For instance, if one compares two stimuli in the Medin and Schaffer experiments with identical percent-correct classification, one studied and the other not, the studied one will tend to receive higher mean confidence. Averaging over 10 non-studied stimuli and 17 comparable studied stimuli with mean correct identification of 81%, the non-studied stimuli were rated 4.60 and the studied stimuli 4.83. ACT predicts this because some of the studied stimulus judgments will result from the application of the production that was designated to classify just that stimulus. In contrast, all judgments for the non-studied stimuli result from the application of generalizations. Application of a designated production results in higher confidence than application of a generalization because the designated production has no variables. This dissociation between confidence and percent correct is not predicted by the other models.

A second remark is that the independent cue model and the Medin-Schaffer context model estimated separate parameters for the salience of each dimension. This allows them to account for variation among dimensions -- both real and random. The impact of this is clear in Experiment 2 vs. 3. These two experiments have the same structure. The independent-cue and context theories display rank order correlations of about .8 with the data of Experiment 2 and about .9 with Experiment 3. However, the two experiments only correlate with each other .69 in rank order of percent correct classification.

ACT's correlations are uniformly below those of the Medin and Schaffer context model.

They are also below the independent-cue model except for Experiment 4 which was explicitly designed to discriminate maximally between the independent-cue model and the Medin and Schaffer theory. It needs to be emphasized, however, that ACT's predictions were done without any parameter search and without any parameters for cue salience. Thus, in ACT we are using a 0-parameter model to fit the data while the context model had 4 parameters and independent-cue model had 5 parameters.

One atheoretical way to give ACT four degrees of freedom is to identify for it the best four conditions and only require it to predict the ordering of the remaining 12 conditions. This was done in the last column of Table 9. Now ACT correlates better than either model in Experiments 3 and 4 and is only slightly worse than the other models in Experiment 2. Given that ACT did this well with the addition of four totally atheoretical parameters we suspect that an ACT model that estimated separate parameters for the salience of each of the four dimensions would do at least as well as the Medin and Schaffer model in accounting for the data.

Insert Table 9 about here

D. Comparison of ACT with Other Models

There are three basic types of models for schema abstraction. One type proposes that subjects form a single characterization of the central tendency of the category. A frequent suggestion is that they distinguish a particular instance (it need not be one they have actually seen) as the prototype for the concept. Other instances are members of the category to the extent that they are similar to this prototype. This class of models would include Franks and Bransford (1971), Bransford and Franks (1972), Rosch and Mervis (1975), Posner and Keele (1968), and Reed (1972). In order to account for the effects of instance frequency demonstrated by Hayes-Roth and Hayes-Roth the prototypes would have to be augmented by some memory for the individual instances studied. However, it is much more difficult for prototype models to accomodate the results of Medin and Schaffer that indicate that subjects are sensitive to similarities among individual instances.

A second class of theories are those that propose subjects store individual instances only, and make their category judgments on the basis of the similarity between the test instance and the stored instances. Among the theories in this class is the Medin and Schaffer theory. A difficulty for the Medin and Schaffer version of the store-instances-only model was the decorrelation found in Hayes-Roth and Hayes-Roth between recognition and classification. They found that the prototypes received the highest classification ratings but the frequently-presented non-prototypes had the highest recognition ratings. This suggests that information is acquired both about the instances and about their more abstract characteristics.

In a certain sense, any results that can be accounted for by a theory that says that subjects store abstractions can also be accounted for by a theory that says subjects only store instances. A store-instance-only theory could always be proposed that went through a test process equivalent to calculating an abstraction from the stored instances and making a judgment on the basis of the abstraction. However, a difficulty for the instance model is the frequent phenomena of subjects reporting verbally the existence of abstract characterizations or prototypes (e.g., Reed, 1972).

The third class of models is that which proposes that subjects store co-occurrence information about feature combinations. ACT is an instance of such a model as are those proposed by Reitman and Bower (1973), Hayes-Roth and Hayes-Roth (1977), and one aspect of Neumann's (1974) model. These models can potentially store all subsets of feature combinations. Thus, they store instances as a special case. The Hayes-Roth and Hayes-Roth experiment showed this model has advantages over many versions of the instance-only or prototype models. However, the Medin and Schaffer version of the instance-only model can accomodate their results.

It is very difficult to find empirical predictions that distinguish ACT from the various other feature-set models. Perhaps, it would be best to regard them as equivalent given the current state of our knowledge and simply conclude that subjects respond in terms feature-sets. However, there are a number of reasons for preferring ACT's version of the feature-set model. First, it is a fully specified process model. As Medin and Schaffer argue, it is often difficult to see in any detail how some of the feature-set models apply to particular paradigms or produce particular results.

Second, ACT has a reasonably efficient way of storing feature-sets. It only stores those subsets of properties and features that have arisen because of generalization or discrimination rather than attempting to store all possible subsets of features from all observed instances. While it seems as if there should be empirical consequences of these different ways of storing feature-sets, our efforts to find them have not been successful. However, if there is very little difference in behavior, that would seem to be all the more reason to prefer the more efficient storage requirements of ACT.

Third, it needs to be emphasized that the ACT learning mechanisms were not fashioned to account for schema abstraction. Rather they were designed in light of more general considerations about the nature of the rules that need to be acquired and the information typically available to acquisition mechanisms in real world situations. We were particularly concerned that our mechanisms should be capable of dealing with language acquisition and rules for making inferences and predictions about one's environment. The mechanisms were designed to both be robust (in being able to deal with many different rules in many different situations) and to be efficient. Their success in accounting for schema abstraction represents an independent confirmation of the learning theory.

Before concluding, we would like to discuss one characteristic of feature-set models which may seem unappealing on first encounter. This is the fact that they store so many different characterizations of the category. ACT may not be so bad as some of the other theories, but still having a set of productions for recognizing instances of a category seems far less economical than having a single prototype. However, the remark that needs to be made is that natural categories defy economical representations. This has been stressed in discussions of their family resemblance structure by Wittgenstein (e.g. Wittgenstein, 1953) and more recently by Rosch (e.g. Rosch & Mervis, 1975). The important fact about many natural categories (e.g., games, dogs) is that there is no set of features that define the category nor is there a prototypical instance that functions as a standard to which all other category members must be compared. On the other hand, these categories do not seem to be unstructured; they are not merely a list of instances. The introspections of one of us (JA) suggest that for him the category of dogs has subclasses that include the following:

(a) *The very large dogs, with short noses, and floppy ears* that include the St. Bernards, Newfoundlands, and Mastiffs.

(b) *The medium to large dogs with relatively long hair, and floppy ears* that include the spaniels, setters, and some of the other retrievers.

(c) *The shnrt and hairy dogs* which include breeds like the pekinese and toy terriers.

(d) *The large, multi-colored dogs, with medium hair, and pointed ears* which include the German Shepherds and Huskies.

The italicized portions of each description gives the physical features that seem to characterize that subclass. There are several things to notice about these feature-set descriptions. First is that certain features are left unspecified; for example, subclass (a) make no reference to coloration or hair. The implication is that these subclasses of the larger dog category are not defined by prototypes either. A second observation is that the feature-set descriptions overlap in complex and relatively unsystematic ways. For example, while there is a tendency for size to distinguish the subclasses, subclass *b* overlaps with subclass *d* on this feature so that large dogs are in both subclasses. Other features, like ear-type serve to distinguish some subclasses (viz., subclass *d* from subclasses *a* and *b*), fail to distinguish others (viz., subclass *a* from subclass *b*) and are irrelevant for still others (viz., subclass *c*). Feature-set models like ACT seem uniquely suited to explain the complex, overlapping, and only partially-specified feature structures of natural categories.

IV References

Anderson, J.R. *Language, memory and thought.* Hillsdale, N.J.: Erlbaum Assoc., 1976.

Anderson, J.R. & Bower, G.H. *Human Associative Memory.* Washington: Winston & Sons, 1973.

Anderson, J.R., Kline, P.J., & Beasley, C.M. Complex learning processes. In R.E.Snow, P.A. Federico, & W.E. Montague (Eds.), *Aptitude, Learning, and Instruction: Cognitive Process Analyses.* In press.

Anderson, J.R., Kline, P.J., & Beasley, C.M. A theory of the acquisition of cognitive skills. ONR Technical Report 77-1, Yale University, 1977.

Anderson, J.R., Kline, P.J., & Lewis, C. A production system model for language processing. In P. Carpenter & M. Just (Eds.), *Cognitive Processes in Comprehension.* Hillsdale, N.J.: Lawrence Erlbaum Assoc., 1977.

Bailey, N., Madden, D., & Krashen, S.D. Is there a "natural sequence" in adult second language learning. English Language Institute and Linguistics Department, Queens College and the Graduate Center, City University of New York, 1974.

Bransford, J.D. & Franks, J.J. The abstraction of linguistic ideas. *Cognitive Psychology,* 1971, *2,* 331-350.

Franks, J.J. & Bransford, J.D. Abstraction of visual patterns.

Journal of Experimental Psychology, 1971, *90*, 65-74.

Hayes-Roth, B. & Hayes-Roth, F. Concept learning and the recognition and classification of exemplars. *Journal of Verbal Learning and Verbal Behavior*, 1977, *16*, 321-338.

Hayes-Roth,F. & McDermott, J. Learning structures patterns from examples. Proceedings of the Third International Joint Conference on Pattern Recognition, 1976, 419-423.

Medin, D.L. & Schaffer, M.M. A context theory of classification learning. *Psychological Review*, 1978, in press.

Neumann, P.G. An attribute frequency model for the abstraction of prototypes. *Memory and Cognition*, 1974, *2*, 241-248.

Neumann, P.G. Visual prototype formulation with discontinuous representation of dimensions of variability. *Memory and Cognition*, 1977, *5*, 187-197.

Newell, A. A theoretical exploration of mechanisms for coding the stimulus. In A.W. Melton & E. Martin (Eds.), *Coding processes in human memory*, Washington, D.C.: Winston, 1972.

Newell, A. Production Systems: Models of control structures. In W.G. Chase (Ed.), *Visual Information Processing*. New York: Academic Press, 1973.

Norman, D.A. & Rumelhart, D.E. *Explorations in cognition.* San Francisco: Freeman, 1975.

Posner, M.I. & Keele, S.W. Retention of abstract ideas. *Journal of*

Experimental Psychology, 1970, *83*, 304-308.

Quillian, M.R. The teachable language comprehender. *Communications of the ACM*, .969, *12*, 459-476.

Reed, S. Pattern recognition and categorization. *Cognitive Psychology*, 1972, *3*, 382-407.

Reitman, J.S. & Bower, G.H. Structure and later recognition of exemplars of concepts. *Cognitive Psychology*, 1973, *4*, 194-206.

Rosch, E. & Mervis, C.B. Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 1975, *7*, 573-605.

Rychener, M.D. & Newell, A. An instructible production system: Basic design issues. In D.A. Waterman & F. Hayes-Roth (Eds.), *Pattern-Directed Inference Systems*, New York, N.Y.: Academic Press, 1978.

Sussman, G.J. *A Computer Model of Skill Acquisition*. New York: American Elsevier Publishing Co., 1975.

Vere, S.A. Induction of relational productions in the presence of background information. Proceedings of Fifth International Joint Conference on Artificial Intelligence, Boston, Mass., 1977, 349-355.

Waterman, D.A. Adaptive productic⁻ⁿ systems, CᵀP working paper #285, Psychology Department, Carnegie-Mellon University, 1974.

Weissman, C. *LISP 1.5 Primer*. Belmont, Ca.: Dickenson, 1967.

Winston, P.H. Learning structural descriptions from examples. MIT Artificial Intelligence Laboratory Project AI-TR-231, 1970.

Wittgenstein, L. *Philosophical Investigations*. New York: Macmillan, 1953.

Navy

1 Dr. Ed Aiken
Navy Personnel R&D Center
San Diego, CA 92152

1 Dr. Jack R. Borsting
Provost & Academic Dean
U.S. Naval Postgraduate School
Monterey, CA 93940

1 Dr. Robert Breaux
Code N-71
NAVTRAEQUIPCEN
Orlando, FL 32813

1 MR. MAURICE CALLAHAN
Pers 23a
Bureau of Naval Personnel
Washington, DC 20370

1 Mr. James S. Duva
Chief, Human Factors Laboratory
Naval Training Equipment Center
(Code N-215)
Orlando, Florida 32813

1 Dr. Richard Elster
Department of Administrative Sciences
Naval Postgraduate School
Monterey, CA 93940

1 DR. PAT FEDERICO
NAVY PERSONNEL R&D CENTER
SAN DIEGO, CA 92152

1 CDR John Ferguson, MSC, USN
Naval Medical R&D Command (Code 44)
National Naval Medical Center
Bethesda, MD 20014

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

1 Dr. Richard Gibson
Bureau of Medecine and Surgery
Code 513
Navy Department
Washington, DC 20372

Navy

1 Dr. Steve Harris
Code L522
NAMRL
Pensacola FL 32508

1 LCDR Charles W. Hutchins
Naval Air Systems Command
444 Jefferson Plaza # 1
1411 Jefferson Davis Highway
Arlington, VA 20360

1 CDR Robert S. Kennedy
Naval Aerospace Medical and
Research Lab
Box 29407
New Orleans, LA 70189

1 Dr. Norman J. Kerr
Chief of Naval Technical Training
Naval Air Station Memphis (75)
Millington, TN 38054

1 Dr. Leonard Kroeker
Navy Personnel R&D Center
San Diego, CA 92152

1 Dr. William L. Maloy
Principal Civilian Advisor for
Education and Training
Naval Training Command, Code 00A
Pensacola, FL 32508

1 CAPT Richard L. Martin
USS Francis Marion (LPA-249)
FPO New York, NY 09501

2 Dr. James McGrath
Navy Personnel R&D Center
Code 306
San Diego, CA 92152

1 DR. WILLIAM MONTAGUE
LRDC
UNIVERSITY OF PITTSBURGH
3939 O'HARA STREET
PITTSBURGH, PA 15213

Navy

1 Commanding Officer
Naval Health Research
Center
Attn: Library
San Diego, CA 92152

Navy

1 Office of Naval Research
Code 437
800 N. Quincy SStreet
Arlington, VA 22217

1 Naval Medical R&D Command
Code 44.
National Naval Medical Center
Bethesda, MD 20014

1 CAPT Paul Nelson, USN
Chief, Medical Service Corps
Code 7
Bureau of Medicine & Surgery
U. S. Department of the Navy
Washington, DC 20372

1 Library
Navy Personnel R&D Center
San Diego, CA 92152

6 Commanding Officer
Naval Research Laboratory
Code 2627
Washington, DC 20390

1 JOHN OLSEN
CHIEF OF NAVAL EDUCATION &
TRAINING SUPPORT
PENSACOLA, FL 32509

1 Psychologist
ONR Branch Office
495 Summer Street
Boston, MA 02210

1 Psychologist
ONR Branch Office
536 S. Clark Street
Chicago, IL 60605

1 Office of Naval Research
Code 200
Arlington, VA 22217

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

1 Psychologist
OFFICE OF NAVAL RESEARCH BRANCH
223 OLD MARYLEBONE ROAD
LONDON, NW, 15TH ENGLAND

1 Psychologist
ONR Branch Office
1030 East Green Street
Pasadena, CA 91101

1 Scientific Director
Office of Naval Research
Scientific Liaison Group/Tokyo
American Embassy
APO San Francisco, CA 96503

1 Head, Research, Development, and Studies
(OP102X)
Office of the Chief of Naval Operations
Washington, DC 20370

1 Scientific Advisor to the Chief of
Naval Personnel (Pers-Or)
Naval Bureau of Personnel
Room 4410, Arlington Annex
Washington, DC 20370

1 DR. RICHARD A. POLLAK
ACADEMIC COMPUTING CENTER
U.S. NAVAL ACADEMY
ANNAPOLIS, MD 21402

1 Mr. Arnold Rubenstein
Naval Personnel Support Technology
Naval Material Command (08T244)
Room 1044, Crystal Plaza #5
2221 Jefferson Davis Highway
Arlington, VA 20360

Navy

1 Dr. Worth Scanland
Chief of Naval Education and Training
Code N-5
NAS, Pensacola, FL 32508

1 A. A. SJOHOLM
TECH. SUPPORT, CODE 201
NAVY PERSONNEL R& D CENTER
SAN DIEGO, CA 92152

1 Mr. Robert Smith
Office of Chief of Naval Operations
OP-987E
Washington, DC 20350

Army

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

1 DR. RALPH DUSEK
U.S. ARMY RESEARCH INSTITUTE
5001 EISENHOWER AVENUE
ALEXANDRIA, VA 22333

1 Dr. Ed Johnson
Army Research Institute
5001 Eisenhower Blvd.
Alexandria, VA 22333

1 Dr. Alfred F. Smode
Training Analysis & Evaluation Group
(TAEG)
Dept. of the Navy
Orlando, FL 32813

1 W. Gary Thomson
Naval Ocean Systems Center
Code 7132
San Diego, CA 92152

1 Dr. Michael Kaplan
U.S. ARMY RESEARCH INSTITUTE
5001 EISENHOWER AVENUE
ALEXANDRIA, VA 22333

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

1 Dr. Harold F. O'Neil, Jr.
ATTN: PERI-OK
5001 EISENHOWER AVENUE
ALEXANDRIA, VA 22333

1 Director, Training Development
U.S. Army Administration Center
ATTN: Dr. Sherrill
Ft. Benjamin Harrison, IN 46218

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

Air Force

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

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

1 Research Branch
AFMPC/DPMYP
Randolph AFB, TX 78148

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

1 Jack A. Thorpe, Capt, USAF
Program Manager
Life Sciences Directorate
AFOSR
Bolling AFB, DC 20332

Marines

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

1 MCDEC
Quantico Marine Corps Base
Quantico, VA 22134

1 DR. A.L. SLAFKOSKY
SCIENTIFIC ADVISOR (CODE RD-1)
HQ, U.S. MARINE CORPS
WASHINGTON, DC 20380

Other DoD

1 Dr. Stephen Andriole
ADVANCED RESEARCH PROJECTS AGENCY
1400 WILSON BLVD.
ARLINGTON, VA 22209

Civil Govt

1 Dr. Susan Chipman
Basic Skills Program
National Institute of Education
1200 19th Street NW