

A LEARNING SYSTEM AND ITS PSYCHOLOGICAL IMPLICATIONS

John R. Anderson
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Paul J. Kline
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Abstract

ACT is a computer simulation program that uses a propositional network to represent knowledge of 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: designation, strengthening, generalization, and discrimination. Designation refers to the ability of productions to call for the creation of new productions. Strengthening a production involves adjusting 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 prototype formation. ACT successfully accounts for the effects of distance of instances from a central tendency, frequency of individual instances, and the family resemblance structure of categories.

I Introduction

ACT is a theory that combines ideas from cognitive psychology and artificial intelligence. It is both a performance theory and a learning theory. As a performance theory ACT is concerned with the factors that determine how quickly, how reliably, and in what ways humans can perform a cognitive task. As a learning theory it is concerned with how the knowledge needed to perform such a task is acquired and properly integrated.

The behavioral domains we have tried to model with the performance system include memory, inference, and language. This work is described in Anderson (1976), Anderson, Kline, and Lewis (1977), Anderson and Kline (1977). Our learning research has concentrated in the past on language acquisition, but more recently we have also become interested in modelling learning of high school geometry and learning of LISP by programming novices. Preliminary reports of this work are Anderson, Kline, and Beasley (1977, 1979a, 1979b). This paper focuses on how the ACT theory would apply to prototype formation. There is a considerable body of empirical research on this topic in cognitive psychology.

The ACT system falls at the intersection between cognitive psychology and artificial intelligence. Many of the concepts described in this paper can already be found in the fields of knowledge representation and knowledge acquisition but there are some new concepts developed out of detailed consideration of human behavior. The ACT system architecture provides a novel synthesis of these individual concepts. With these concepts and architecture we are able to account in detail for some important psychological phenomena. Thus, we hope that this paper will provide researchers in AI with a few new concepts but mainly a new perspective on familiar concepts and their possible combination. We also hope that the paper will indicate to researchers in cognitive psychology new potentials for some of the concepts in artificial intelligence.

The ACT System

In ACT knowledge is divided into two categories: declarative and procedural. The declarative knowledge is represented in a propositional network, which is similar to other semantic network representations (Quillian, 1969; Anderson and Bower, 1973; Norman and Rumelhart, 1975; Shapiro, 1971; Simmons, 1973). 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, 1978). A production is a condition-action rule. The condition is an abstract specification of a set of propositions in the network. To be matched to the condition, the propositions must be *active*. ACT's activation mechanism is designed such that the only active propositions are those that have recently been added to the data base or that are closely associated to such propositions. 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 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 the productions whose conditions have all their constants active in the data base. 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 those productions which are the most likely to apply successfully.

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 the strengths of its 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 prototype formation. 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 added through the spontaneous restructuring of productions in response to experience. Our focus in this paper will be on spontaneous restructuring. The designation process plays only a small role in our modelling of the prototype formation literature.

Generalization

A very important component of the spontaneous learning process involves generalization. ACT uses an algorithm which finds the maximal common generalization of two productions in the sense developed by Vere (e.g. 1977) and Hayes-Roth (e.g. 1976). The basic idea behind this generalization technique is to compare pairs of similar productions P_1 and P_2 and to form new productions P_3 which have the following characteristics:

- (a) P_3 will apply in the circumstances that either P_1 or P_2 did (and possibly new circumstances)
- (b) P_3 will have the same effect as P_1 or P_2 in the circumstances that P_1 or P_2 applied
- (c) There is no production P_4 that satisfies (a) and (b) and only applies in a subset of the circumstances that P_3 does.

The maximal common generalization is not always unique; in which case ACT selects in a random fashion. These generalizations are formed basically by deleting clauses in the conditions of P_1 and P_2 and by replacing constants by variables.

We have a number of heuristics to help direct the generalization process, but obviously have no general solution to the NP-complete partial matching problem embedded in the process of generalization (see Hayes-Roth, 1977). Most of the examples we run into seem to be computationally tractable. Examples can be designed that push the computational limits of the system but they do not seem particularly easy generalization problems for humans either. We have been concerned in ACT with developing (a) principles for deciding when to attempt forming generalizations in a realistically large system with very many productions; and (b) principles of caution for integrating generalizations (there is always a danger of overgeneralization) into the operation of the system.

An example will help to illustrate ACT's automatic generalization mechanism. Figure 1 illustrates the stimuli studied by subjects in Experiments 3 and 4 of Franks and Bransford (1971). We will assume that subjects designate productions to recognize each stimulus. So for the first stimulus item subjects would designate the following production:

P1: 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:

P2: 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 generalization.

P3: 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

The local variables in this generalization are prefaced by LV. This generalization can be thought of as an attempt on ACT's part to arrive at a more general characterization of the study material. Note that ACT's generalization mechanism needs only two examples to propose a

generalization. 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:

P4: 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 (P_1) we would get a production that would accept any array of four geometric objects as an instance of the study material. While it is conceivable that any such 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 forming a generalization. Of the two productions from which the generalization is formed, the one with the least number of constants is used as a reference. The number of constants in the generalization can be no fewer than half this quantity. The terms that ACT considers constants are italicized. There are five constants in productions P_1 , P_2 , and P_3 . Production P_3 is an acceptable generalization from P_1 and P_2 because it only involves replacement of two of the constants. P_1 and P_4 are prevented from forming a generalization because this would require replacing four of the five constants with variables.

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 seems infeasible to attempt to generalize all possible pairs of productions. ACT only attempts to form generalizations when a new production has been designated. Also, 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, 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.

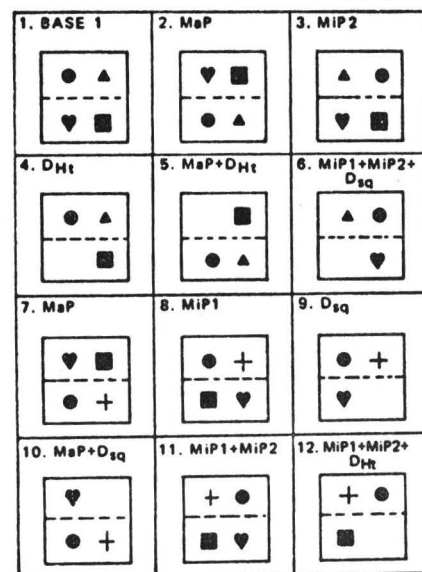


Figure 1. Examples of material from the Franks and Bransford study.

*The actual syntax of ACT productions is considerably more complex than this. We present them in a pseudo-English to facilitate readability.

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 (Brown, 1973). Thus a child will generate *mans*, *glvod*, etc. ACT will do the same.

Use of a discrimination mechanism seems relatively unique in the knowledge acquisition literature--Winston's (1972) use of "near misses" being the only related mechanism we are aware of. The use of discrimination is motivated by the fact that any system that tries to improve itself by making generalizations must, occasionally, make overgeneralizations. Discrimination provides an automatic means for "debugging" such errors. To correct overgeneralizations ACT must create more discriminate productions.

The discrimination mechanism will only attempt to discriminate an overgeneral production when it has both a correct and an incorrect application of that production to compare. (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.) Basically, this algorithm remembers and compares the variable bindings in the correct and incorrect applications. It attempts to find a variable that had different bindings in these two applications. In forming the discrimination, restrictions are placed on that variable to prevent the match that led to the unsuccessful application while still permitting the match that led to the successful application. 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. In some of our other work (e.g. Anderson, Kline, & Beasley, 1979a) the variable was kept but an additional clause was added to restrict what the variable could match.

As an example of the 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:

P6: IF a stimulus has two large red triangles
THEN it is in category A

P7: IF a stimulus has two large blue circles
THEN it is in category A

From these two ACT would form the following generalization:

P8: 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 both. Thus, the counter-example was presented of two large blue triangles which was a stimulus in category B. Generalization P8 misapplied in this circumstance. By noting what distinguished the circumstances of correct applications of generalization P8 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.

P9: IF a stimulus has two large red LVshapes
THEN it is in category A

P10: IF a stimulus has two large LVcolor circles
THEN it is in category A

These productions were formed from P8 by replacing one of its variables (either LVcolor or LV-shapes) 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).

Production Strength and Specificity

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 alternative productions.

Through experience with the ACT system we have created a set of parameters which, although somewhat arbitrary, 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. Productions consistent with the misapplied production also have their strength reduced by a factor of 1/4. Since a multiplicative adjustment produces a greater change in strength than an additive adjustment, a "punishment" is more effective than a "reinforcement". Since increments and decrements in strength propagate to more general productions and since general productions can have a number of more specific productions consistent with them, they tend to reflect quite rapidly and sensitively the weight of experience.

Recall that a production's strength determines the probability that it will apply. If s is the strength of a production and S the total strength of all active productions, 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. If a production passes this probabilistic hurdle it is placed on the APPLYLIST. Of course, if a production 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 conflict resolution. 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 has a specificity ordering on its productions which makes it possible to deal with exceptions to strong rules: If two productions both have enough strength to be placed on the APPLYLIST but the condition of one of them is a more specific version of the condition of the other (i.e. satisfied in a subset of the situations that satisfy the other), then the more specific production will apply and will block out the more general production. 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. This special case principal can be traced back to Waterman (1970) and variants of it can be found in many current production systems (e.g. Rychener & Newell, 1978).

The precedence of exceptions over much stronger general rules does not imply that exceptions always apply. 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. This property of the ACT model is consistent with the fact that words with irregular inflections have high frequencies of occurrence.

Production strength is another important way in which ACT differs from most other computer-based learning systems (e.g., Anderson, 1978; Vere, 1977; Hayes-Roth & McDermott, 1976; Sussman, 1975;

Winston, 1970; Waterman, 1970. The learning of all these systems has an all-or-none character that ACT would share if creating new productions were its only learning mechanism. The strength mechanisms modulate the all-or-none character of production creation enabling 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.

III Applications to Prototype Formation

There is a growing literature concerned with the process by which humans 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*, but for various reasons we also refer to it as studying *schema abstraction*.

In the remainder of this paper we describe how ACT's automatic learning mechanisms can be used to model 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.

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 in Figure 1 twice and then were transferred to a recognition phase in which they rated test pictures according to whether they had been studied or not. The test pictures could be classified according to how many transformations separated them from the central tendency of the study stimuli. There were test stimuli 0, 1, 2, or 3 transformations from the study stimuli and some "non-cases" which were still further removed. Some of the test figures were actually studied and some were not. Franks and Bransford report that confidence ratings for recognition generally decreased with the number of transformations from the base and was lowest for the non-cases.

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

Since most experiments in this literature report data on subjects confidence in their judgments, 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 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.

To obtain predictions for this experiment we ran a series of ACT simulations. Altogether we obtained fifty ratings for each test stimulus and the data we report will be based on averages of these fifty ratings.

O-transformation test stimuli were given a mean rating of 1.66 (i.e. mean number of constants in matching productions); the one-transformation test stimuli were rated 1.24; the two-transformation stimuli were rated 1.11; the three-transformation stimuli were rated 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. (They do not report the actual ratings.) Neumann (1974) performed a replication of Franks and Bransford and he did report mean ratings for each of the five categories of test stimuli. The confidences assigned to the stimuli by his subjects corresponds exactly in rank ordering to the results obtained from ACT.

This experiment serves to illustrate that ACT can account for one of the basic phenomena of schema abstraction -- namely that confidence falls off with distance of the stimuli from the central tendency (prototype) of the category. The generalizations ACT forms represent what various stimuli will have in common. Therefore, there will be more generalizations formed that match central stimuli than ones that match peripheral stimuli. It is this greater density of generalization that give central stimuli this advantage.

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

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 more than one hundred 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, and college. The third was marital status which could have values single, married, divorced, and widowed. The descriptions also included proper names and hobbies but this information was 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.

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

Hayes-Roth and Hayes-Roth varied the frequency with which various instances were studied. Some instances were presented 10 times while others only once. A study trial consisted of first presenting the subject with an instance, asking him to classify it, and then providing feedback as to which club the instance came from. Some instances received equivocal feedback in the sense that half the time the feedback for those instances specified club 1 and half the time club 2.

After studying a set of stimuli, subjects were shown a critical test set of 28 stimuli. Subjects were first asked to categorize each of the stimuli and then they were asked to decide whether each of the stimuli had been studied or not. The recognition judgment was assigned a confidence from 1-5 as was the categorization judgment.

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 discriminating versions of its overgeneral productions.

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. The categorization scores for a test stimulus were calculated by weighting negatively the confidences of incorrect classifications, weighting positively the confidences of correct classifications, and ignoring the confidences of classifications to the neither-club category. Figure 2 presents the actual and predicted recognition and classification confidences for the seven categories of test stimuli. The confidence scale used by subjects and the match scale used by ACT have been adjusted to cover approximately equal ranges. As can be seen the ACT predictions closely correspond to the data obtained by Hayes-Roth and Hayes-Roth.

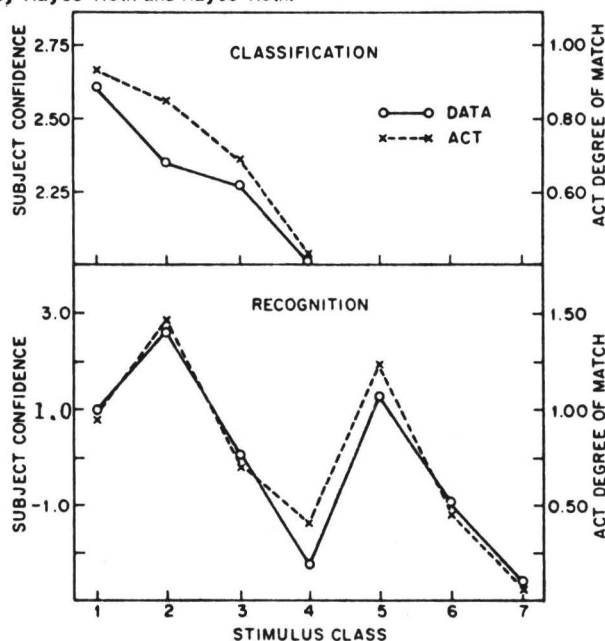


Figure 2. Comparison of ACT's predictions with the results from Hayes-Roth and Hayes-Roth.

Class 1 in Figure 2 is formed from two prototypes which, in fact, were never studied. However, they received the highest categorization rating and a relatively high recognition rating indicating that subjects have extracted the central tendencies of the categories. Class 2 items, while not as similar to the prototypes as Class 3 items, had categorization ratings second only to the prototypes themselves. Since each item in Class 2 was studied ten times while items in Class 3 were only studied once, it appears that frequency of exposure has an effect on categorization that can compensate for lack of similarity to the prototypes. A comparison of categorization ratings for Class 3 versus Class 4 holds frequency of exposure constant and the effect of similarity to prototypes re-emerges: Class 3 is closer to the prototypes and receives higher ratings.

Items in the last three classes were neutral with regard to category membership so that correct categorization is undefined and only recognition confidences can be reported. Whereas the data reviewed earlier from Classes 1-4 showed that frequency of exposure affects categorization ratings when we might have expected it only to affect recognition ratings; the data from Classes 5-7 shows that similarity to prototypes affects recognition when we might have expected it only to affect categorization. Class 5 has the same exposure as Class 2 so only its distance from the prototypes can explain its lower recognition ratings. Similarly, no instances of either Class 6 or Class 7 was ever studied by subjects, but Class 6 received higher recognition ratings and this must be due to the fact that, although very far from the

prototypes, Class 6 is not as far from them as is Class 7.

In summary, this experiment serves to illustrate how ACT correctly models the combined effects of frequency of exposure and closeness to prototype. Stimuli that are frequently studied will have fairly strong specific productions to recognize them. Stimuli which are close to the prototype will have many generalizations that can recognize them.

Comparison of ACT with Other Prototype Theories

There are three basic types of theories for prototype formation. 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 theories 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 theories to accommodate the recent results of Medin and Schaffer (1978) that indicate that subjects are sensitive to similarities among individual instances. On the other hand ACT is able to simulate this data (Anderson, Kline, & Beasley, 1979b).

A second class of theories (e.g. Medin & Schaffer, 1978) 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. 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. The instance theory could always be made to go 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 theory is that subjects frequently report having abstract characterizations or prototypes (e.g., Reed, 1972).

The third class of theories proposes that subjects store co-occurrence information about feature combinations. ACT is an instance of such a theory 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.

It is very difficult to find empirical predictions that distinguish ACT from the various other feature-set theories. 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 of feature-sets. However, there are a number of reasons for preferring ACT's version of the feature-set theories. First, it is a fully specified process model. It is often difficult to see in any detail how some of the feature-set theories 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 there are empirical consequences of these different ways of storing feature-sets, the differences are so subtle that existing experiments have failed to test them. 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 acquisition of 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 general 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, yet having a set of productions for recognizing instances of a category still seems far less economical than having a single prototype. However, two remarks need to be made here. The first remark is that the complexity of the representation of a category depends on one's production system implementation. This can be quite complex if each production is represented separately. However, the representation can be quite economical in a data-flow system like Forgy's (1977). Here a single discrimination net is constructed to encode the conditions of all productions. This allows overlapping productions to share the same net tests in encoding their conditions.

A second remark is that natural categories defy economical representation. 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. It is interesting to note here that Dreyfus (1972) has claimed that computer implementations are not capable of representing the family resemblance structure of categories. ACT shows just the opposite to be true.

IV References

- [1] Anderson, J.R. *Language, memory and thought*. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1976.
- [2] Anderson, J. R. Induction of Augmented Transition Networks. *Cognitive Science*, 1977, 1, 125-157.
- [3] Anderson, J.R. & Bower, G.H. *Human Associative Memory*. Washington: Winston & Sons, 1973.
- [4] 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*, 1979a, in press.
- [5] Anderson, J.R., Kline, P.J., and Beasley, C.M. A general learning theory and its application to Schema Abstraction. In G.H. Bower (Ed.) *The Psychology of Learning and Motivation*, Vol. 13, 1979b, in press.
- [6] 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.
- [7] 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 Associates, 1977.
- [8] 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.
- [9] Brown, R. *A first language*. Cambridge, Mass.: Harvard University Press, 1973.
- [10] Bransford, J.D. & Franks, J.J. The abstraction of linguistic ideas. *Cognitive Psychology*, 1971, 2, 331-350.
- [11] Dreyfus, H. L. *What Computers Can't Do*. New York: Harper and Row, 1972.
- [12] Forgy, C. L. A production system monitor for parallel computers. Technical Report, Computer Science Department, Carnegie-Mellon University, 1977.
- [13] Franks, J.J. & Bransford, J.D. Abstraction of visual patterns. *Journal of Experimental Psychology*, 1971, 90, 65-74.
- [14] 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.
- [15] Hayes-Roth, F. The Role of Partial and Best Matches in Knowledge Systems, Rand Corporation, P-5802, 1977.
- [16] Hayes-Roth, F. & McDermott, J. Learning structured patterns from examples. Proceedings of the Third International Joint Conference on Pattern Recognition, 1976, 419-423.
- [17] Medin, D.L. & Schaffer, M.M. A context theory of classification learning. *Psychological Review*, 1978, in press.
- [18] Neumann, P.G. An attribute frequency model for the abstraction of prototypes. *Memory and Cognition*, 1974, 2, 241-248.
- [19] Neumann, P.G. Visual prototype formulation with discontinuous representation of dimensions of variability. *Memory and Cognition*, 1977, 5, 187-197.
- [20] 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.
- [21] Newell, A. Production Systems: Models of control structures. In W.G. Chase (Ed.), *Visual Information Processing*. New York: Academic Press, 1973.
- [22] Norman, D.A. & Rumelhart, D.E. *Explorations in cognition*. San Francisco: Freeman, 1976.
- [23] Posner, M.I. & Keele, S.W. Retention of abstract ideas. *Journal of Experimental Psychology*, 1970, 83, 304-308.
- [24] Quillian, M.R. The teachable language comprehender. *Communications of the ACM*, 1969, 12, 459-476.
- [25] Reed, S. Pattern recognition and categorization. *Cognitive Psychology*, 1972, 3, 382-407.
- [26] Reitman, J.S. & Bower, G.H. Structure and later recognition of exemplars of concepts. *Cognitive Psychology*, 1973, 4, 194-206.
- [27] Rosch, E. & Mervis, C.B. Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 1975, 7, 573-605.
- [28] 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.
- [29] Schapiro, S.C. A net structure for semantic information storage, deduction, and retrieval. Proceedings of the Second International Joint Conference on Artificial Intelligence, 1971, 512-523.
- [30] Simmons, R.F. Semantic networks: Their computation and use for understanding English sentences. In R.C. Schank and K.M. Colby (Eds.), *Computer models of thought and language*. San Francisco: Freeman, 1973.
- [31] Sussman, G.J. *A Computer Model of Skill Acquisition*. New York, N.Y.: American Elsevier Publishing Co., 1975.
- [32] 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.
- [33] Waterman, D.A. Generalization learning techniques for automating the learning of heuristics. *Artificial Intelligence*, 1970, 1, 121-170.
- [34] Winston, P.H. Learning structural descriptions from examples. MIT Artificial Intelligence Laboratory Project AI-TR-231, 1970.
- [35] Wittgenstein, L. *Philosophical Investigations*. New York: Macmillan, 1953.