UNIVERSITY OF ALBERTA

Computer Models of Analytic and Nonanalytic Strategies for Concept Learning

by

Kui Lin ©

A Thesis
Submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Science

Department of Computing Science

Edmonton, Alberta
Fall, 1990

ISBN 0-315-65088-5

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: **Kui Lin**

TITLE OF THESIS: **Computer Models of Analytic and Nonanalytic Strategies for Concept Learning**

DEGREE: **Master Of Science**

YEAR THIS DEGREE GRANTED: **1990**

(Signed) ....

Permanent Address:
Apt. 302, 220 Woodridge Cres.,
Nepean, Ontario, K2B 8G1

Date: October 10, 1990

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Computer Models of Analytic and Nonanalytic Strategies for Concept Learning** submitted by **Kui Lin** in partial fulfillment of the requirements for the degree of **Master Of Science.**

....
R. Elio (Supervisor)

....
J. Schaeffer

....
R. Goebel

....
J. Bisanz

Date:

Abstract

Learning concepts from studying concept exemplars is an important type of inductive learning that has been studied intensively from both cognitive science and artificial intelligence perspectives. Empirical studies have identified many factors that affect learning performance of human concept learning. These include the orders in which training exemplars are encountered during learning, the analytic versus nonanalytic learning strategies, and the interaction between the two. However, these issues have not been investigated much from a machine learning perspective, and there is no computer model that can account for these effects. This research presents a computational model called LANA that simulates the empirical results on analytic and nonanalytic learning strategies and their interaction with exemplar order. The LANA simulation has identified a small set of basic processes as critical for simulating the effects of different learning strategies. Key aspects of the cognitive simulation approach, examples of other concept learning simulations, and the behavior of general learning systems are also discussed.

Acknowledgements

I wish to express my gratitude and appreciation to my thesis supervisor, Dr. Renée Elio, for her guidance, suggestions, and support. Dr. Elio put in significant amount of time and effort during the critical stage of this research. Dr. Elio also provided encouragement and guidance for my first year graduate course work.

I would like to thank the members of my thesis examination committee, Dr. R. Goebel, Dr. J. Schaeffer, and Dr. J. Bisanz, for their valuable criticism and constructive suggestions. These suggestions have improved the quality of this thesis.

Thanks go to my peers in the department of computing science for their help and advice, especially Pavan Sikka, Charlse Truscott, and Peter Leung. Special thank goes to Daniel Lanovaz for his help in using the Xerox Lisp machines and for his work on an earlier program upon which LANA had been developed and evolved.

Thanks also go to the supporting staff Andy Nowacki, Rene Leiva, Carol Smith, and Rick Hughes, for keeping the A.I. lab in good working condition.

Most of all I would like to thank my wife, Hui Wang, for her love, support, and understanding, and for her help in editing part of the thesis.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

Learning from past experience is a fundamental element of human intelligence. The study of learning has a long history of research from both the cognitive psychology and artificial intelligence (A.I.) perspectives. These fields of studies attempt to understand the key mechanisms underlying basic intelligence behavior and both address a similar set of issues. One of the most studied learning behaviors is *concept learning*. The main idea of concept learning from examples is *induction*, i.e., the creation of general information from a set of specific instances that describes these instances [Michalski, Carbonell, and Mitchell, 1983].

1.1. Cognitive Psychology Perspective

In cognitive psychology, *concept learning* is often called *category learning*. The landmark of category learning research was best marked by the work of Bruner, Goodnow, and Austin [1956]. They pointed out that learning and utilizing categories represent one of the most elementary and general forms of cognition by which we adjust to our environment. They identified the following reasons as to why category learning is so fundamental to human thinking: (a) by categorizing discriminably different events as equivalent, we reduce the complexity of our environment; (b) categorizing is the means by which objects of the world about us are identified; identifying something is in fact an act of "placing" it in an equivalence class; (c) established categories provide abstract general properties that may cover future encounters and reduce the necessity of constant learning; (d) established categories allow inferences to be drawn about the properties of the individual members by virtue of knowing their membership; and (e) categorizing allows the description of relations among groups of things without referencing individual members. We map and give meaning to our world by relating classes of objects or events rather than by relating the individuals. Since human thinking can be viewed as a computational process, similar arguments about the utilization of categorization processes can be made with respect to machines.

From a cognitive psychology perspective, the key issues of category learning concern identifying the nature of the category information abstracted from studying individual category members and the processes that create and use this information. Experimental evidence has been reported that both supports and disconfirms existing models that differ fundamentally in their characterization of what is learned from exposure to category exemplars and how it is used to make judgements on novel items. So far, there is no single model that accounts for all the empirical results obtained. Besides, there are certain empirical findings about the factors affecting performance of human concept learning that have not yet been accounted for by any existing model. These factors include, for example, the orders in which training exemplars are encountered during learning, the analytic versus nonanalytic learning strategies, and the interaction between the two. Computational models are yet to be developed to account for these basic results.

As researchers gain greater understanding in both human and machine learning behaviors, they found that certain basic processing mechanisms and constraints believed to govern human learning also underlie many machine learning systems [Medin, Wattenmaker, and Michalski, 1987]. For instance, some researchers have discovered that humans have difficulty learning disjunctive concept descriptions [e.g., Bruner et al., 1956]; similarly, algorithms for learning disjunctive concepts are rarely as straightforward as their conjunctive counterparts [Michalski et al., 1983]. Common mechanisms believed to associate with human learning, such as similarity-based pattern matching, retrieval of past experiences, generalization, discrimination, and refinement of existing knowledge, are also found to underlie most machine learning systems in one form or another. This includes systems under the framework of production-based learning [Klahr, Langley, and Neches, 1987], case-based learning [Kolodner, 1983], automated knowledge acquisition [Michalski et al., 1983], and so on. Thus, advances in computational models of human concept learning have profound implications to theories of both machine and human learning.

1.2. Artificial Intelligence Perspective

Since the early days of A.I., learning has been held as a critical element of human intelligence to achieve for machines. From the historical view point, machine learning went through three major periods [Michalski et al., 1983]: neural modelling and decision-theoretic techniques, *symbolic concept learning*, and knowledge intensive learning.

The symbolic machine learning paradigm has intimate relations to the research on descriptive and process models of human concept learning. However, parallel and distributed processing and the connectionist approach present additional computational paradigms that bridge the research focuses of cognitive psychologists interested in process models, and artificial intelligence researchers interested in building learning systems that better match certain human learning abilities. In symbolic concept learning, systems learn by constructing a symbolic representation of a given set of concepts through the experience of individual examples and counter-examples of these concepts. Their representations are typically in the form of logical expressions, a decision tree, production rules, or a semantic network. The attributes or predicates relevant to the concept are explicitly provided for these systems. Some of the landmark works within symbolic concept learning paradigm include, Winston's [1975] influential ARCH system in learning structural descriptions, Michalski's [1980] inductive learning algorithm INDUCE, and Quinlan's [1979] discriminative learning program ID3.

Given any finite set of experiences, there are infinite inductive generalizations that may describe these experiences [Michalski et al., 1983]. The question of how to constrain this virtually unlimited set of generalizations is at the core of concept learning from examples. Computationally, concept learning is essentially a constrained search problem [Mitchell, 1982]. The search space contains all possible states representing alternative concept descriptions (or hypotheses). The initial state corresponds to the observed descriptions of a certain example. As examples are learned, inductive operators, such as generalization and discrimination, move the system from one state to another. The goal state of the search is the "best" concept to be learned with respect to certain predefined criteria, and the task of concept learning is to identify such a goal state in this space within certain computational resource constraints.

According to another scheme, machine learning research can also be organized around the following three paradigms: theoretical analysis, *cognitive simulation*, and domain-specific studies [Michalski et al., 1983]. Cognitive simulation involves developing computational models to investigate the underlying mechanisms contributing to human intelligence behaviors in order to build learning machines. A cognitive simulation is a learning program with a set of computational processes similar to those believed to underlie human learning. Although many research efforts primarily belong to one of the three areas, progress in one often benefits the research in the others. Specifically, to investigate the space of possible learning methods, a reasonable starting point may be to consider certain known examples of human learning behavior [Michalski et al., 1983]. Similarly, psychological investigations of human learning have been advanced by the development of A.I. models and algorithms.

1.3. Overview of LANA Simulation

This research proposes and verifies a computational model, LANA[1], that simulates certain empirical results of human category learning. Specifically, the research has

1 LANA stands for Learning Analytically and NonAnalytically.

explored and identified alternative models that account for the effects of exemplar order, learning strategy, and the interactions between the two on concept learning performance. Elio and Anderson [1984] found that category learning proceeds better if the learner first studies a sample with only typical exemplars, and gradually encounters the less typical category exemplars, than if a learner is trained on samples that are representative, i.e., containing typical as well as non-typical exemplars. However, this result interacts with the learning strategies; if the learner actively forms rules and tests hypotheses about category membership during learning, the opposite trend occurs, and learning is better with representative samples.

The LANA simulation has implications beyond accounting for these results. First, it proposes computational processes underlying analytic and nonanalytic learning, which to date have only descriptive accounts. Secondly, it investigates the exemplar order effects on concept learning. Thirdly, it addresses a number of issues related to rule-based generalization learning and case-based learning. Fourthly, it looks at the key aspects of cognitive modelling.

The LANA simulation approach is essentially a constrained search for valid model(s) in a space of all possible models. One effective way to constrain the search is to start by investigating a small but plausible set of processing assumptions. The model validation is based on matching behaviors and evaluating the psychological plausibility of model processes.

The preliminary search covered a space with a few hundred models. Identified through this exercise was a small set of architectural assumptions, a set of processes, and associated parameter settings that define a family of models that can account for the empirical results. LANA's analytic and nonanalytic learning strategies were simulated by alternating some critical aspects of a few processes. Specifically, nonanalytic learning trends were simulated through requiring: (a) partial match for all pattern matching; (b) a higher strength threshold as a generalization retrieval constraint; (c) items to compete with generalizations on an equal basis in the pattern selection process; and (d) frequently used items being "permanently remembered" in the form of generalizations. With almost the opposite parameter settings, analytic learning trends were achieved by requiring: (a) full match between a presented item and stored generalizations; (b) a relatively lower strength threshold as a generalization retrieval constraint; (c) items not to compete with generalizations in the pattern selection process; and (d) frequently-used items not being remembered permanently as generalizations. These results represent another discovery of this research.

Still another important aspect of this work, which has not been studied much in the literature, was the exemplar order effects. Different presentation orders of training examples have been found affecting the learning behavior of both humans [e.g., Elio and Anderson, 1981, 1984] and machines [e.g., Lebowitz, 1986], especially for incremental learning where examples encountered earlier may have significant impact on what can be learned next. Therefore, it is important to understand, through simulation, the effects of exemplar orders on the evolution of category knowledge. In this regard, this work has direct implications to computer assisted instruction (CAI) as well as human education.

Finally, to test LANA's generality, it has been applied to duplicating one other concept learning result by Hayes-Roth and Hayes-Roth [1977] that supported a feature-set model.

1.4. Thesis Organization

The remaining chapters are organized as follow. Chapter 2 reviews selected empirical results on category learning, A.I. research on concept learning, and cognitive simulation paradigm and systems. Chapter 3 outlines the LANA simulation methods and architectural assumptions. Chapter 4 presents the details of the LANA framework, followed by chapter

5 that provides information about implementation. Chapter 6 presents LANA's performance data in accounting for the effects of strategy, exemplar order, and their interactions. Chapter 7 discusses some remaining issues. The thesis closes by discussing relations to other work and areas for future research.

Chapter 2
Literature Review

2.1. Introduction

It is a ubiquitous phenomenon that humans group objects, events and experiences in the world into categories. To understand the underlying principles that govern category learning of natural as well as artificial categories, various empirical studies have been performed and many descriptive models have been proposed. These include: (a) Katz and Postal's [1964] classical theory of concepts specified by necessary and sufficient features, (b) Rosch and Mervis's [1975] *family resemblance* theory, (c) the notion of *category validity* [Rosch and Mervis, 1975] defined as the conditional probability that an objects has certain attribute(s), given that it belongs to the category, (d) the notion of *cue validity* [Tversky, 1977] defined as the conditional probability that an object is in a category, given that it has certain attribute(s) or cue(s) associated with the category, and (e) Murphy and Medin's [1985] *conceptual coherence*. Various category learning models have been constructed to account for empirical results, including prototype models [e.g., Posner and Keele, 1968], instance models [e.g., Medin and Schaffer, 1978], and feature-set models [e.g., Hayes-Roth and Hayes-Roth, 1977]. They differ chiefly on their assumptions about the nature of representations of category information abstracted and the processes acting on them. So far, empirical evidence has been found both for and against these models.

Parallel to these investigations, there has been concept learning research in machine learning and A.I., some of which have been inspired by the results from empirical studies. Research approaches such as the case-based learning [Kolodner, 1983], production-based learning [Klahr, Langley, and Neches, 1987], and knowledge-based expert problem-solving [Jackson, 1986] are closely related to their empirical counterparts.

Computational modelling of human learning behaviors through computer simulations combines the initiatives of both the empirical studies and the A.I. research. In the last few decades, cognitive simulation has led to the development of a wide range of approaches and systems, including EPAM [Feigenbaum, 1963], ACT [Anderson et al., 1979], and MINERVA and MINERVA 2 [Hintzman et al., 1980; Hintzman, 1986], and INDUCE and PATCH [Medin et al., 1987]. This research falls into the cognitive simulation category.

The rest of the chapter contains three sections. Section 2 introduces the empirical paradigm of human category learning. It reviews the key experimental results, alternative models, and the descriptive characteristics of analytic and nonanalytic concept learning strategies. It also describes the empirical results by Elio and Anderson [1984] that LANA simulates. Section 3 reviews the cognitive simulation methodology and describes five selected simulation systems and their applications to concept learning tasks. It also discusses the mechanisms common to other concept learning systems. The final section is a summary of the entire chapter.

2.2. Category Learning Theories and Empirical Findings

To facilitate understanding of the empirical research in the area, the general paradigm for empirical studies of concept learning is briefly introduced first.

2.2.1. Empirical Paradigms for Category Learning

Most empirical studies of human learning involve conducting controlled experiments with either carefully designed artificial or natural categories. Artificial categories may include instances ranging from simple geometric patterns to fairly complicated artificial grammars or human faces. Categories are constructed as meaningful collections of individual instances[1]. Some categories may contain subcategory structures, item types, or prototypes.

[1] We use the terms instance, exemplar, and item inter-changeably to mean category member.

Category members are often represented as a discrete set of attributes, normally with discrete values. For example, a typical member *canary* of the natural category *bird* may be represented as (flies, sings, has feathers, lays eggs, and builds nest in trees), which has five attributes[1], each having binary or multiple values.

A typical experiment involves a learning or training phase, followed (immediately or with a delay) by a classification or transfer test phase which contains some novel exemplars. During learning, learners are presented with category exemplars to study and to classify into available categories. They are given feedback as to whether a judgment is correct or incorrect. During the test phase, learners are given a classification or recognition task over both previously-seen exemplars and novel exemplars. Accuracy, confidence rating, typicality rating, reaction time, and written protocol are the usual dependent measures. It is the performance on the carefully constructed test items that supports hypotheses about the underlying representations and processes.

2.2.2. Theories of Category Learning

The primary objective of category learning theories is to determine, and hence differ in their assumptions about, the following: (a) the nature of internal category representations constructed during learning, (b) how this representation is stored, retrieved and applied to classify novel instances, and (c) what kind of underlying processes are responsible for (a) and (b). Category learning models can be distinguished into three major types [Anderson, Kline, and Beasley, 1979]: prototype, instance-based, and feature-set models.

Prototype models. Prototype models assert that a single representation of the category is formed that represents some kind of a central tendency or prototype of all the category exemplars seen. Other instances belong to the category with varying degrees depending on how similar they are to the category prototype. According to prototype

---

[1] We will interchangeably use the terms dimension, property, and feature to mean attribute.

models, classification of a new or old instance is based on the distance between the instance and the category prototype, where the distance is defined by Posner and Keele [1968] as the number of transformation necessary to convert the instance into the category prototype.

Posner and Keele [1968] conducted several experiments that provided evidence for prototype formation. They defined categories by randomly choosing their prototypes first, and then systematically distorted these prototypes to create other category members. After learning these categories, learners were given a transfer task that contained novel patterns. The main finding was that the prototype pattern, which was never presented, had the best classification and recognition ratings. Other results indicated that never-studied prototypes were sometimes better classified than much-studied instances [Posner and Keele, 1970] .

A similar line of research investigating the formation of a category prototype was conducted by Franks and Bransford [1971]. They found an inverse relationship between recognition ratings and the instances' transformational distance from the prototype, with the prototype receiving the highest recognition rating. Therefore, they concluded that the representation of visual pattern schema consisted of a prototype and a set of transformations.

Since Posner and Keele's work, the prototype theory spawned a considerable number of subsequent experiments aimed at confirming or extending the theory. These included studies by Posner and Keele [1970], Read [1972], and Rosch and Mervis [1975].

The notion of prototypicality is intuitive in that it captures the characteristics commonly seen in natural categories. This notion has been used in A.I. to build effective knowledge-based systems [Aikins, 1983]. However, this does not necessarily mean that people form prototypes as category representations. In fact, there exist some major shortcomings for prototype models. For example, prototype models cannot explain the effects of frequency of instances exposure demonstrated by Hayes-Roth and Hayes-Roth [1977], discussed in more detail below. Another shortcoming is that prototype models assume that all attributes are equally salient and attribute information is combined in an *additive* (*independent-cue*) fashion. This independent-cue assumption makes the complete learning of linearly-nonseparable categories impossible [Medin and Schaffer, 1978]. Anderson et al. [1979] pointed out that prototype models seem to over-simplify the complexity of natural categories in the sense that many natural categories have complicated subcategory structures which can not be captured by a single prototype.

Instance-based Model. In contrast to prototype models, instance-based models posit that specific exemplars are stored during learning and the judgments of novel instances are based on the similarity between the novel instance and the stored instances. Medin and Schaffer's [1978] context model was the first and the most significant effort of this theoretical line.

The key idea of the context model is that a novel instance serves as a retrieval cue to access the stored instances similar to the cue. The major assumptions of the context model are: (a) only the specific exemplar information is stored; (b) classification of an item is an increasing function of its similarity to the items in a particular category and a decreasing function of its similarity to the items of other categories; similarity is the sole criterion for classification decision; (c) similarity between two instances is assumed to be the *multiplication* of the similarity along each dimension of the instances. This multiplication rule is referred to as *interactive-cue*, in contrast to the independent-cue approach which counts important features. As a result of (c), dimensions of an instance do not have to be equally salient as implied by prototype models, making it possible to learn linearly-nonseparable categories.

In their experiments, Medin and Schaffer [1978] found that transfer performance was better predicted by the context model than by a prototype model. They found that learning and transfer performance on novel items was not a function of their distance from the prototype, but a function of the interitem similarity among the exemplars. They argued that the evidence for a prototype was an artifact of how category members were typically generated as distortions of the prototype so that the prototype was most similar to members of its own category and least similar to items of the contrasting categories. The superior recognition and classification performance of the prototype after delay was actually due to the prototype's resemblance to the remembered exemplars of the category. Thus, they argued that one does not need to posit an abstraction process to account for the performance; the retrieval of stored exemplar information will do.

An instance-based model possesses the following merits: (a) it posits only the level of representation which leads to simplicity; (b) it is naturally related to similarity-based analogical learning and to the evidence about learners' memory for instances [Brooks, 1978; Reber, 1976]; (c) it makes it easier, relative to a prototype model, to update existing information when new information is acquired [Elio and Anderson, 1984].

One major point against instance models is that a mere collection of individual instances does not usually reflect the intention of a category. Therefore, some category level information seems necessary [Anderson et al., 1979]. Another difficulty for instance-based models is that learners frequently report verbally that they use rules to make category judgment [Elio and Anderson, 1984].

Feature-Set Models. Feature-set models posit that learners are sensitive to, and encode representations of, co-occurring feature patterns. These feature patterns are more general than specific instances and they carry predictive information about category memberships. In most feature-set models, each feature combination has an associated *strength* that serves as a measure of the pattern's relative successfulness during past experience. This is why such models are also called frequency or strength models. Models in this category include Hayes-Roth and Hayes-Roth's [1977] property-set model, Reitman and Bower's [1973] frequency model, Anderson et al.'s [1979] ACT model, and variations of the ACT model [Elio and Anderson, 1981; 1984].

Using category exemplars with multiple attributes and values, Hayes-Roth et al. [1977] compared a collection of models of all three types and demonstrated the advantages for feature-set models. They proposed a property-set model, which assumed that the elements of the power set of all individual properties, called property sets, form the representation of category knowledge. To associate frequency information with each property set, they defined an associative strength as the frequency with which a property set was encountered in all the exemplars seen in a category. They defined *diagnosticity* of a property set for a given category as an increasing function of its associative strength to that category and a decreasing function of its associative strength to other categories.

By carefully controlling an exemplar's presentation frequency and distance from a central tendency, Hayes-Roth et al. found that instances farther away from the prototype may have higher property-set diagnosticity than instances closer to the prototype. This evidence is inconsistent with the assumption that the distance from the prototypes is the only relevant factor influencing item recognition and classification. Instead, their results supported the notion that recognition and classification are influenced by a rather complex mixture of both frequency of exposure and distance from prototype. In addition, they found a rather low correlation between recognition and classification ratings, a result that the context model has difficult explaining. According to Hayes-Roth et al., the prototypes received the highest classification ratings but the frequently-studied non-prototypes received the highest recognition ratings.

Elio and Anderson [1981] also conducted a series of experiments which supported the feature-set models. Firstly, they found that the accuracy and confidence ratings on transfer items were better in a learning situation that was conductive to making generalization, than in one that was non-conductive to generalization, where conductive simply meant that items presented were classifiable by generalizations. However, they observed that for both learning situations, transfer was also a function of interitem similarity, a result consistent with the findings by Medin and Schaffer [1978]. Secondly, they discovered highly significant correlations among accuracy, confidence rating, and similarity for transfer items in both situations, which would not be predicted by an instance-based model. Thirdly, they found that learning speed and accuracy were better when generalizable items appeared in close temporal proximity than when they were randomly ordered. However, they also reported evidence that could be explained by an instance-based model but not by any existing feature-set model.

Based on these findings, Elio and Anderson suggested that the representation of instances be augmented with, rather than replaced by, category level information. They argued that it may be both unnecessary and inadequate for a general theory to choose between the generalization framework and the similarity-based analogical mechanism. However, it seems that no such theory has yet emerged from the current experimental literature, nor from the machine learning literature.

Compared to using prototypes, a feature set model seems less economical in representation because the set of possible feature patterns can be very large. However, it is claimed that this might be necessary in capturing the complex subcategory structures of many categories [Anderson et al., 1979]. Studies have suggested that most natural categories have neither a set of defining features nor a prototypical instance to which all other members should be compared. However, these categories do not appear to be unstructured as in the case of a collection of instances [Anderson et al., 1979].

Feature-set models can account for most results that can be explained by either a prototype model or an instance-based model. The power of feature-set models can be attributed to the following: (a) co-occurring feature patterns are a powerful concept which ...s the notion of interitem similarity as well as frequency of exposure; (b) feature ...n be seen as containing both exemplar level information and category level information; they can be viewed as segments of instances and can be matched the same way instances are matched; (c) feature patterns reflect the notion of correlated features, and the idea of predictive relations; (d) feature-set models do not assume independence of individual features (independent-cue assumption), rather they rely on feature relationships.

2.2.3. Analytic and Nonanalytic Learning Strategies

The fact that strategies affect human concept learning has been repeatedly reported and analyzed in a number of studies [Brooks, 1978; Elio and Anderson, 1984; Medin and Smith, 1981; Reber, 1976]. Strategies can be induced either by learning task features (e.g., learning material, instructions) or by learners' existing knowledge (e.g., learners' theories of the world) [Medin and Smith, 1981]. The formation of strategies does not appear to be an arbitrary phenomenon, rather, it is often found operative in association with certain learning conditions.

Researchers have frequently studied two type of learning strategies: *analytic* and *nonanalytic strategy* [Brooks, 1978; Elio and Anderson, 1984; Reber, 1976]. Some researchers prefer to use different terms to contrast the two strategies, such as explicit versus implicit learning [Reber, 1976], rule versus analogical learning, and deliberate versus intuitive learning [Brooks, 1978].

Analytic learning strategy has been characterized as the explicit encoding of category-level information (e.g., correlated attributes) and conscious hypothesis testing activities [Elio and Anderson, 1984; Kemler and Nelson, 1984; Medin and Murphy, 1987]. Learners using this strategy usually focus on the detection of regularities, salient attributes, and co-occurring feature patterns, and on testing explicit hypotheses [Elio and Anderson, 1984; Kemler and Nelson, 1984]. One characteristic of this strategy is that analytic learners have poor memory for instances they have actually studied, but good memory for hypotheses they tested [Reitman and Bower, 1973]. The protocol analysis by Elio and Anderson [1984] and Kemler and Nelson [1984] indicated that learners were generally successful in indicating which attribute(s) or feature relations were salient and used to make pre .ictions.

Nonanalytic learning strategy, on the other hand, has generally been characterized as the unconscious encoding of exemplar information and the use of similarity-based analogy [Brook 1978; Reber, 1976]. Learners under this strategy focus more on overall similarity of category exemplars [Kemler and Nelson, 1984]. There is evidence that learners under this strategy have better memory for instances because they rely on them more than analytic learners do. This naturally leads to their inability to discover category level features or their inter-relations [Kemler and Nelson, 1984]. The protocol analysis from several studies repeatedly demonstrated that learners adopting this strategy were generally unable to describe verbally what was responsible for their category judgments [Brooks, 1978], yet their performance were fairly good.

2.2.4. The Interaction of Exemplar Order and Learning Strategy

This subsection introduces the empirical findings by Elio and Anderson [1984], which have been the focus of the LANA simulation.

Elio and Anderson [1984] investigated the effects of exemplar variance order and concerning category learning. The categories were relatively large with several item types and with no defining feature or simple rule that could perfectly predict category memberships, reflecting their ill-defined nature.

To study the differential effect on the evolution of concepts over time from receiving representative and non-representative samples, they constructed two[1] sampling conditions: centered and representative conditions, which differed over how samples were drawn from the population and presented to learners. The centered condition began with a low variance sample and introduced more category variation in later samples, while the representative

[1] In fact, they used four different sample conditions, but only two are relevant here.

condition drew its samples in proportion to the item frequency in the whole population. Although all the category items were contained in the samples under both centered and representative conditions, the variance of each sample and the presentation orders were different for different conditions. The detailed experiment design, categories, and procedures are described in chapter 3.

In a series of experiments, Elio and Anderson [1984] found a main effect for exemplar order but this effect interacted with learning strategies induced by experiment instructions. Specifically, when learners were instructed that the category was too complex to look for rules and encouraged to simply memorize items, there was a transfer advantage for learners in the centered condition over learners in the representative condition. In other words, centered-condition learners gave significantly higher typicality ratings and had slightly higher classification accuracy than those given by representative-condition learners. However, subsequent experiments indicated that this result reversed completely if learners were encouraged to look for regularities, to form rules, and to actively test hypotheses. In this case, learners receiving representative samples had the superior transfer performance on classification typicality and accuracy.

2.3. Cognitive Simulation and Machine Learning

This section describes the cognitive simulation paradigm and its relation with other areas of machine learning research. It also presents five simulation systems in symbolic concept learning: EPAM, ACT, MINERVA and MINERVA 2, and INDUCE.

2.3.1. Cognitive Simulation Paradigm

Cognitive simulation, or computer modelling, aims at developing computational models to simulate and investigate the underlying mechanisms contributing to intelligence behaviors of humans and machines. A cognitive simulation is a computer learning system and contains a set of computational processes believed to underlie human learning behavior. Therefore, the differences between simulation systems and performance learning systems are mainly reflected in the objectives.

Normally, a model simplifies the mechanisms which it is meant to describe by specifying only the most important elements and ignoring the rest. In much the same way, a computer model of cognitive behavior, either for humans or machines, involves a substantial amount of simplifications based on some assumptions about the key processing principles and constraints [Feigenbaum and Feldman, 1963; Slatter, 1987]. Therefore, assumptions are the underlying keystones of a particular model architecture, and the testing of models can sometimes be viewed as the test of alternative assumption sets.

There are two levels at which one can specify a model: a descriptive level and a prescriptive (process) level [Murphy and Medin, 1985]. At the descriptive level, one normally describes "what is done", while at the prescriptive level one needs, in addition, to state "how it is done." For example, the family resemblance principle is basically descriptive while classification by retrieving matched instances is a prescriptive model.

One related distinction to the above is between product simulation and process simulation [Medin et al., 1987]. Product simulation only compares the end products (input and output) of a cognitive behavior without addressing the internal processes. On the other hand, process simulation deals with both the match of products and the match of processes responsible for the products. In other words, production simulation is based on behavior (product) validation while process simulation must be based on both product and process validation. Under this view, some A.I. programs might be regarded as only product level simulation to human behavior if they could duplicate the behavior, since they are not concerned with the (internal) processes responsible for the corresponding human behavior.

Cognitive modelling can be viewed as an inductive search process over a space of possible models. A computational model is a specific set of processing assumptions and constraints, realized through a specific algorithm. The space of different models can be viewed as a space of different parameter combinations (i.e., how assumptions are realized as processes), and the corresponding search has been called a parameter search [Anderson et al., 1979]. The search process involves repeatedly going through simulation "cycles," each of which involves specifying a model, implementing the model, and then comparing the model's results with the observed data. A set of criteria for evaluating a model must be specified, which may involve model assumptions, implementation of processes, and behavior. The manner in which the model results match the observed data may suggest further direction of search, and the cycle continues [Feigenbaum and Feldman, 1963]. The target (goal) state of the search is the model which best satisfies the specified criteria. Obviously different criteria may, therefore, lead to different target models.

2.3.2. Importance of Cognitive Simulation

It is important to appreciate the role that cognitive simulation plays with respect to advances in other areas of artificial intelligence and cognitive science. Cognitive simulation has been adopted not just in symbolic concept learning, but also in connectionist learning [Wisniewski and Anderson, 1988], statistical learning [Feigenbaum, 1963], and areas outside machine learning such as problem-solving [Newell and Simon, 1963].

In general, simulation systems can serve a number of purposes. Firstly, it may help to gain better understanding of human cognitive behavior by : (a) forcing theories to provide complete and explicit descriptions of the cognitive processes; (b) testing the theory's ability to accurately predict and explain observed data; (c) clarifying vague boundaries between theories by evaluating existing assumptions; (d) enabling detailed, directed, and controlled investigation of various mechanisms and allowing for substantive analysis of differential effects for interactive factors; and (e) contributing new clues and insights to further theoretical and empirical investigations of both human and machine intelligent behaviors [Feigenbaum et al., 1963; Slatter, 1987].

Secondly, it may serve to bridge the gap that often exists between theories and application systems (e.g., learning systems and expert systems) by: (a) illustrating characteristics of certain computational systems and the influence of the assumption set; (b) demonstrating performance of learning systems under the impacts of a set of factors such as exemplar order, bias, and input noise; and (c) providing evidence and techniques for application systems such as knowledge-based systems and, therefore, contributing to the application of artificial intelligence systems [Slatter, 1987].

There are, however, disadvantages and pitfalls associated with computer simulations [Feigenbaum et al., 1963; Slatter, 1987]. Firstly, there is a lack of general consensus as to the proper relationships between theories and simulation programs. Secondly, ad hoc assumptions may easily be introduced which may alter the original intended model; i.e., one may have a right theory but choose a wrong realization. Lastly, a process model in the form of an operatable system makes it easy for one to over-generalize from the model behavior.

2.3.3. The EPAM Simulation

As one of the earliest influential simulation program in symbolic learning, EPAM - Elementary Perceiver and Memorizer - successfully simulated a number of tasks in human verbal learning behavior [Feigenbaum, 1963]. One of the experiments EPAM was designed to simulate was rote memorization of nonsense syllables (items) in associated pairs or serial lists. For paired association learning tasks, the goal is to learn which response is paired with a particular cue. The cue item is presented first followed by the response item. In serial learning, the learner must learn the $(N+1)$th item in the list, given the Nth item as the cue.

EPAM is a process model with two major components: a learning component and a performance component [Feigenbaum, 1963]. During learning, EPAM discriminates and associates the paired items. During testing, EPAM produces responses to the cue items. Specifically, the performance component works as follows: when a cue item is presented, a *discriminator* sorts the cue item in a *discrimination net* (a tree of tests and branches) to find a stored representation for the cue item. A response cue (a link) associated with the representation of this cue item is accessed, and fed to the discriminator which sorts it in the net and finds the full response representation.

The learning component grows the discrimination net. Each time a pair is encountered, EPAM sorts, separately, cue and response items to find terminal nodes to store each of them. New branches may have to be built and new tests added to locate them. The discrimination of any new item depends on finding a difference between the terminal nodes to be discriminated. The response cues (links) are represented with partial information enough to retrieve the full response item from the net at the moment of association, while the response item contains full information. The net continues to grow as additional pairs are learned, and it is eventually used in the test phase to make associations.

EPAM has two additional features: generalization and forgetting. Since each paired-association consists of a cue and a response, there is cue generalization and a response generalization. Specifically, if X and X' are similar cue items, and Y is the correct response to the presentation of X, then if Y is given as association with X', this is called cue generalization. Likewise, response generalization is defined in a similar way.

EPAM's forgetting is a type of functional forgetting, i.e., information becomes lost or inaccessible in a large growing, associative network, but not physically destroyed. Forgetting occurs as a direct result of subsequent learning because the cue information stored at the moment of association may become insufficient at a later time as new items are added to the net. This forgetting is often temporary, i.e., a lost association can be reconstructed by storing more response cue information to differentially locate the intended full response item. As Feigenbaum indicated, EPAM was the first concrete demonstration of this type of (functional) forgetting in learning machines.

EPAM was tested on a large set of rote learning tasks and was proven to be fairly successful. The later versions of EPAM explored the different "sense mode" associated with human rote learning [Feigenbaum, 1963]. For example, "visual" input and "written" output, "auditory" input and "oral" output, and so on. Each mode corresponds a perceptual input coding scheme and a discrimination net, with easy internal transformation between them.

As a classic work of symbolic learning, EPAM has inspired various kinds of architectures. The idea of the discrimination net has been widely used in many learning systems, most recently evidenced in the MOP architecture used for case-based learning and memory system [Kolodner, 1983; Lebowitz, 1986].

2.3.4. The ACT System

ACT [Anderson et al., 1979] and its successors ACT* [Anderson, 1986] are theories of general human cognition, including memory, skill acquisition, problem-solving, and learning. The original ACT theory embodies the powerful thesis that a single set of learning processes - generalization, discrimination, and strengthening - underlies the general human learning.[1] The validity of ACT theory has been tested by evaluating its ability to simulate a wide range of empirical data and the plausibility of its learning mechanisms.

---

[1] The more recent ACT* theory realized that separation of generalization and discrimination processes was unnecessary and, it posited only one general learning mechanism: proceduralization [Anderson, 1983].

Knowledge in ACT is divided into two categories: declarative and procedural [Anderson et al., 1979]. The declarative knowledge is represented in a propositional network similar to the semantic network representations [Michalski, et al., 1983]. While the declarative knowledge representation aspect of the model is important, it is not particularly relevant for this discussion of their simulation work. In ACT, the procedural knowledge is represented as a set of productions. Specifically, a production is a condition-action rule, where the condition part is an abstract specification of a set of propositions. If a set of propositions that satisfy this specification is active in the knowledge base, the production will perform its action.

In ACT, the basic control structure iterates through successive cycles, where each cycle consists of a production selection phase followed by an execution phase. On each cycle, a probablistically-defined subset of all the productions whose conditions are satisfied by active propositions is computed. The probability that a production will be included in the subset depends on the ratio s/S, where s is the strength of that production and S is the sum of the strengths of all the productions whose conditions are active propositions. This ratio is meant to reflect how successful the past applications of a production have been. Another selection criterion for productions is called the specificity principle, which says that all other things being equal, productions with more constant conditions, as opposed to variable conditions, are preferred. Successful application of a production leads to an increment in its strength, while a failure reduces its strength. Further details about the general ACT framework are given by Anderson [1976, 1986].

When given a category learning task, for each instance presented, ACT designates a production that recognizes and classifies the instance. Automatic generalization occurs by comparing pairs of these productions, which produces a more general production. For example, *production 1* and *production 2* can be generalized to form *production 3* by making the *shape* attribute a variable condition:

Production 1: IF the object is *large*, *green*, and *triangle*, THEN it is drawn by Tom

Production 2: IF the object is *large*, *green*, and *circle*, THEN it is drawn by Tom

Production 3: IF the object is *large*, *green*, and *any shape*, THEN it is drawn by Tom

If feedback on the correctness of the production applications is provided, a discrimination process may be evoked. In ACT, the discrimination process serves the purpose of converting over-generalizations to more specific, and hence, discriminative productions. A production can be made more discriminative either by adding constant clauses as conditions or by replacing variables by constant conditions. For example below, *production 1* can be discriminated into a more specific version such as *production 2* by adding a constant clause [Anderson et al., 1979]:

Production 1: IF climate of a place is *warm*, and has *ample rainfall*, THEN the place can *grow rice*

Production 2: IF climate is *warm*, has *ample rainfall*, and the *terrain is flat*, THEN it can *grow rice*

Obviously, with the underlined condition, *production 2* is more useful than *production 1* by being more discriminative in this context.

In the simulation of recognition confidence rating, Anderson et al. used the total number of constant features (conditions) in the decision-making production. The classification confidence rating was based on the total number of constant features in the production weighted positively for correct classifications and negatively for incorrect ones.

Using a simplified version of the general ACT framework, Anderson et al. [1979] demonstrated that ACT's learning mechanism has straightforward applications to concept learning in ill-defined categories. The model successfully simulated the results by Franks and Bransford [1971], Hayes-Roth and Hayes-Roth [1977], and Medin and Schaffer [1978]. These three empirical results have been discussed earlier as the important results supporting prototype models, instance-based models, and feature-set models, respectively.

2.3.5. MINERVA and MINERVA 2

Hintzman has developed two simulation models, MINERVA [Hintzman and Ludlam, 1980] and MINERVA 2 [Hintzman, 1986], based on episodic memory theory and instance-based process models of concept learning. MINERVA is a specific simulation for accounting the differential forgetting of prototypes and old instances, while MINERVA 2 is a more general memory process model.

MINERVA is an instance-based model with the following assumptions: (a) classification of a new item is based on the exemplar most similar to the test item, and (b) individual properties are lost from the example trace over time in an all-or-none fashion. The category instances are represented as a propositional structure of both properties and the property relationships with separate strength. The initial encoding of the memory trace was simply a copy of the item description. A new item is matched for similarity against all traces in memory and the degree of match between the item and the trace is computed using certain nearest-neighbor formula. An arbitrary retrieval threshold was used to control the minimum degree of match and the size of the retrieval set. Two different schemes of the forgetting were used: (a) decrement the strength of the trace by a proportion of its present value on each cycle, and (b) delete a property from the trace in an all-or-none fashion with a given probability.

The version of MINERVA with all-or-none forgetting scheme reproduced the ordering of the confidence ratings reported by Medin and Schaffer [1978] with a high correlation between the simulated data and the observed data. The finding provided a process model for the claim that the evidence for prototypes is due to differential forgetting of prototypes and old instances.

Hintzman [1986] extended the original MINERVA into MINERVA 2 which became a general multiple-trace memory process model for concept learning. The memory structure of MINERVA 2 consists of a *primary memory (PM)* and a *secondary memory (SM)*.

Representation of an experience is a unique memory *trace*. A theoretical assumption is made (as in MINERVA) that no matter how similar traces may be they are separate entities (traces) and represent difference experiences. Every trace is represented as a list of primitive properties. The communication between the two memories is straightforward: PM sends a retrieval cue (instance) to *all* traces in SM, and SM sends a response to PM.

The process in which stored traces are matched and retrieved is referred to *trace activation* by Hintzman. The activation of stored traces is based on their similarity to a given cue. As the result of an activation, a response is returned which has two characteristics: *intensity* and *content*. The intensity of the response depends on the total amount of activations triggered by the cue, reflecting the degree of familiarity or judgment of frequency with respect to the cue. The content of the response is the summed pattern of activation among primitive properties contributed by the reactions of all stored traces to the cue, each responding to its similarity to the cue.

MINERVA 2 has accounted for the following empirical findings: (a) the classification of prototypes is more stable over time than the classification of old exemplars; (b) old exemplars are classified better than new exemplars on both the immediate test and the test after a delay; (c) transfer to classification from the old exemplars to new patterns is best for prototypes, intermediate for item similar to prototypes, and worst for items very different from prototypes; (d) transfer to new patterns improves with increasing category size, that is, the number of training exemplars in the category; this result has been taken as evidence for the hypothesis that the greater the number, and hence, the greater the variance of the category, the more effective the hypothetical prototype abstraction process becomes; (e) the tendency of erroneously assign patterns to a category increases with category size.

2.3.6. The INDUCE Algorithm

In their experimental study and comparison of human and machine behaviors, Medin, Wattenmaker, and Michalski [1987] examined the constraints and preferences employed by both human and machine in learning decision rules from preclassified examples. They chose the INDUCE program as their machine simulation of human behavior and developed, with the inspiration of INDUCE, a process model called *Patch*, to account for specific patterns in human results and for easy comparison with INDUCE.

Motivated in part by some psychological considerations, the INDUCE algorithm realizes the *STAR* method of induction [Michalski, 1983] which focuses on various single positive examples and contrasts them with negative examples. INDUCE starts with a set of descriptions of entities, then selects a target category to proceed as follow:

a. First randomly select a *seed* example from the target (positive) category.

b. Generalize the seed (*star*) in various general ways without describing counter-examples of the contrasting category. This includes using both *selective* generalization rules (e.g., turning constants into variables, dropping constant conditions, and closing intervals) and *constructive* generalization rules (e.g., counting rule, and generating chain properties).

c. Concept descriptions on a *candidate list* are evaluated according to a *preference criterion*. This criterion is predefined and contains *consistency* and *completeness* constraints. A description is consistent if it does not apply to any members of the contrasting category or has no counter-examples. A description is complete if it applies to all members of the target category. Descriptions that are both consistent and complete represent alternative solutions and are saved.

d. Alternative descriptions are ordered according to the preference criterion and the best description is selected as the final concept description.

e. If the description covers all the positive examples, then a solution has been found and the process stops. Otherwise, all positive examples covered by this description are removed from the original set. Then a new seed is selected from the remaining positive examples, and the process repeats from a.

Note that the solution is either a single conjunctive description or a disjunction of such descriptions, which happens when the above process repeats itself. Thus INDUCE has an inherent bias toward conjunctive descriptions; when it cannot find one, it creates a disjunctive description.

Medin et al. [1987] observed that, in general, the rules developed by INDUCE through inductive learning tasks were fairly similar to what people obtained. They found that both people and INDUCE preferred conjunctive rules much more than disjunctive rules. Another finding was the bias of both human and INDUCE towards positive features over negative features.

Medin et al. pointed out that people set out to find descriptors that span the target category without using examples from contrasting categories. The first possibility was that if an assertion was consistent (covered no counter-examples) but not complete (did not span the target category), it was retained, and the attention shifted to the members of the target category not covered by the original assertion. Then new assertions were sought that were consistent and complete with the reduced set. This was precisely how the main algorithms in INDUCE worked. The second possibility was that an assertion would be complete but not consistent. In this event, the Patch model assumed that people focused on eliminating the counter-examples through specializing their description by negating properties that were true for the counter-examples but not for the positive examples (i.e., opportunistic conjunctions). Unfortunately, the INDUCE algorithm does not allow for these opportunistic conjunctions.

Based on their findings, Medin et al. suggested that descriptions of constraints should be in terms of model processes rather than in terms of the products or outputs. Although many A.I. induction programs may become candidate psychological process models because they are prescriptive models, only a handful of them have the potential to account for existing results, since most are not intended to be models of human rule induction [Medin et al., 1987]. The key point is that inductive learning experiments with humans can suggest new algorithms or improvements of the existing ones. Medin et al.'s research is one such example.

2.3.7. Learning Mechanisms Shared by Machine Learning Systems

Some mechanisms believed to underlie human concept learning have been intensively studied and widely used in different types of machine learning systems. For example, pattern matching and retrieval, generalization and discrimination with a strength assignment mechanism, and forgetting underlie a broad range of machine learning systems, including the classic symbolic learning system EPAM [Feigenbaum, 1963], the statistical learning systems [Uhr and Vossler, 1963], most production-based learning systems such as ACT and ACT* [Anderson, 1979, 1986], genetic algorithm learning systems [Holland, 1975], case-based learning systems such as CYRUS [Kolodner, 1983] and UNIMEM [Lebowitz, 1986].

To see why statistical learning systems adopt essentially the same set of mechanisms, Uhr and Vossler's [1963] classic pattern recognition system is a good example. Briefly speaking, their program works as follows [Uhr and Vossler, 1963]: unknown patterns are presented to the computer in a matrix of ones and zeros with certain size. The program generates and composes operators to transform the unknown input matrix into a list of characteristics. These characteristics are then compared to each set of the stored characteristics in memory to determine match. The most similar characteristics in memory will be chosen as the response for the input. The characteristics are then examined on whether they individually contributed to the success or failure in identify the input. Their corresponding strength adjustment processes are then turned up and down in a way rather similar to the strengthening and weakening of rules in production systems. As a result, poor characteristics with low strength are eventually dropped out of system (or forgotten) and replaced by new ones.

It may be argued that one reason for these mechanisms to underlie a large number of systems is that they reflect the "intuitive idea" of learning, i.e., find good rules, descriptors, or operators, and reward them when they lead to correct decisions. When they lead to errors, punish them, remove them, or replace them by new ones. In doing so, systems have to depend on some kind of strength, weight, and score to distinguish promising candidates from the poor ones, and reward and punish them accordingly.

Genetic algorithm (or classifier) systems are also based on this idea [Holland, 1975]. Both production systems and genetic algorithms use the *condition->action* type of productions (or classifiers) and both require assignment (or apportionment) of strength to rules. The rule retrieval, generalization and discrimination, and strength assignment processes in a production system correspond to the rule discovery, rule generating with genetic operators (crossover, mutation, and inversion), and credit apportionment processes in a genetic algorithm system. One major difference is that any number of rules can be activated at the same time in a classifier system. However, the basic idea of encouraging competition among rules, rewarding useful rules, and punishing bad ones is clearly there.

The idea of building effective reconstructive memory and corresponding retrieval mechanisms to do case-based learning [Kolodner, 1983; Lebowitz, 1986] represents another large collection of systems. One common feature of many case-based learning systems is the possession of a powerful memory for past experiences (cases) coupled with a set of efficient processes in case indexing, memory reorganization, generalization, and retrieval. The idea of generalization may seem incompatible with case-based reasoning. But without some form of generalization, the training received by a system would not be easily transferable to a new problems or cases. In CYRUS, there are two types of generalizations. One is the initial generalization which compares old and new events to extract out commonalities and add to the norm of the new E-MOP. The other is generalization refinement which corrects under-generalization and over-generalization.

These characteristics of case-based systems are fairly similar in nature to the corresponding processes in rule-based systems.

2.4. Summary

Concept learning from exemplars is an important type of intelligent activity that has been intensively studied. Researchers have proposed many cognitive theories and models to answer the question of what is the representation of abstraction from exemplars and the nature of the processes operating on it. For many characteristics of human learning, there are only empirical results or descriptive models. However, process models are important to understand theories and to transfer theories into real systems. One such example is the analytic versus nonanalytic strategies on concept learning, which yields fundamentally different classification performance. Unfortunately, there is no process model for analytic and nonanalytic concept learning.

Artificial intelligence and machine learning have not been concerned much with these strategy differences for learning algorithms and with the exemplar order effects on concept learning. Yet some paradigms such as case-based learning are extremely order sensitive. Therefore, it is important to understand the mechanisms that underlie the effects of strategy and order influence. This research is such an attempt in achieving this goal.

Chapter 3
Overview of Approach

3.1. Introduction

The target of the LANA simulation was the interaction reported by Elio and Anderson [1984] about the exemplar order and learning strategy. Recall that exemplar order was the manner in which learners encountered category variance while learning strategy was the instructional bias to either passively memorize the items during learning or actively generate and test hypotheses. One primary goal of the simulation effort was to explore specific mechanisms that might account for this interaction of exemplar order and learning strategy. The approach focused on mechanisms that were psychologically plausible and consistent with the descriptive and prescriptive learning models proposed by other researchers. Simplicity, plausibility, and parsimony reflected LANA's basic design philosophy. However, the possible combinations of even a small set of processing assumptions and the various alternative ways to implement them created an extremely large space of potentially viable models. Through a systematic search in this model space, a small set of mechanisms was identified that seemed critical to account for the empirical results.

This chapter presents the LANA simulation approach. It first describes the simulation of the experimental categories, procedures, and learning strategies. It then describes the basic model assumptions and a few important aspects of the model specification and implementation related to this approach. Subsequently, it gives the criteria and procedure of the model validation for LANA simulation, followed by a brief summary of the chapter.

3.2. Method of Simulation

The simulation used the same categories, sample structures, learning and testing procedures that corresponded to those used in the original experiments. This section describes the method for simulating these components.

3.2.1. Categories

The system's task was to learn to correctly classify category items into either of two given categories. Table 1 below lists all members of the two categories.

Table 1
Category Items of Simulation

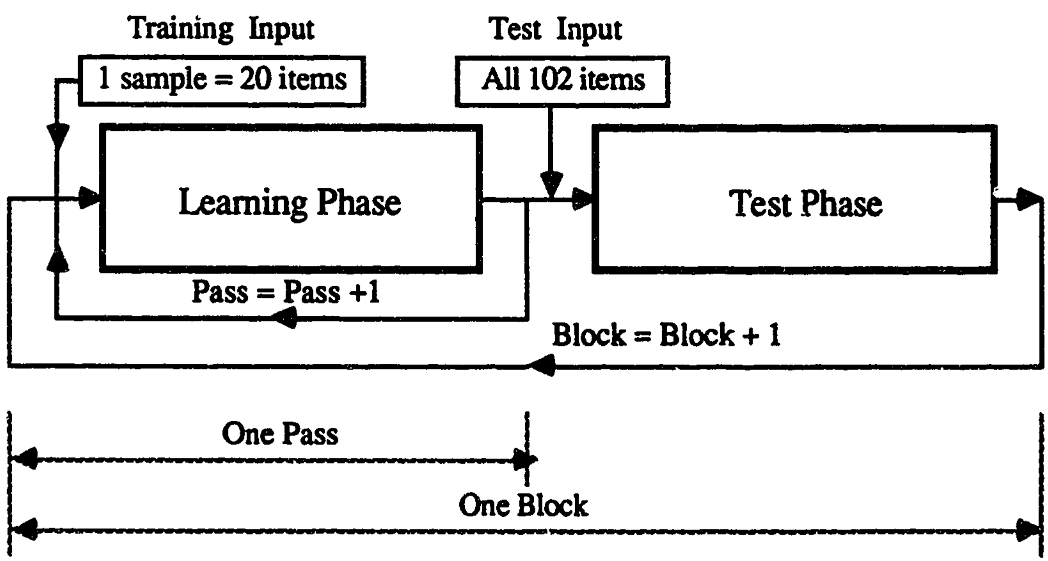
Category 1					Category 2				
Member Types					Member Types				
C	B	A	B'	C'	C	B	A	B'	C'
		Prototypes					Prototypes		
22211	22111	21111	11122	11222	33344	33444	34444	44433	44333
	12211	12111	11221			43344	43444	44334	
		11211					44344		
		11121					44434		
		11112					44443		
		Category Members					Category Members		
22231	22113	21113	31221	13222	33324	33442	34442	24334	42333
22241	22114	21114	41221	14222	33314	33441	34441	14334	41333
22213	22311	21131	13221	31222	33342	33244	34424	42334	24333
22214	22411	21141	14221	41222	33341	33144	34414	41334	14333
	12231	12311	11322			43324	43244	44233	
	12241	12411	11422			43314	43144	44133	
	12213	13211	31122			43342	42344	24433	
	12214	11241	41122			43341	44314	14433	
		31211					24344		
		11214					44341		
		11321					44234		
		11421					44134		
		31112					24443		
		41112					14443		
		13112					42443		
		14112					41443		

As shown in Table 1, each category had 51 members (including 11 prototypes and 40 ordinary category members) that could be characterized as belonging to one of five member types A, B, B', C, and C' defined by separate sets of generating rules. Each category member was notationally represented as a 5-dimensional and 4-valued tuple. Each dimension represented an attribute of a member and the corresponding value denoted the specific characteristic of that attribute. For example, a five-digit tuple, 11214, represented a member of category 1. Categories were defined such that one category (e.g., category 1) is dominated by attribute values 1's and 2's while the other (e.g., category 2) by 3's and 4's. Note that there was no defining feature or simple rule that perfectly determined category membership, reflecting the categories' ill-defined nature. However, some attribute values (e.g., 1's and 2's) are better predictors of one category (e.g., category 1) than the other, given particular combinations.

3.2.2. Simulating Experimental Procedures

The basic procedures of the simulation matched the original experiment procedure. A schematic description of the procedures is shown in Figure 1 below.

Figure 1. Experimental Procedure



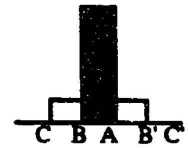
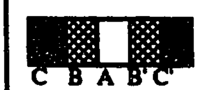
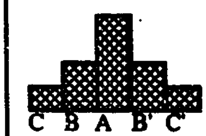
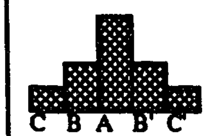
Each *simulation run* corresponded to one learner. A simulation run received samples of training items that were constructed according to either a centered or representative sampling condition. Each *experiment* contained 30 simulation runs corresponding to 30 learners, 15 for each sample condition and each run used a different random presentation of training items during learning.

Each experiment consisted of four experiment *blocks*; each block had a *learning phase* and a *classification* (*test*) *phase*. Each learning phase was further divided into three *passes* iterating through one sample of 20 items in different random orders. The processing of each item formed a *cycle*. On each learning cycle, an item from the sample was represented to the system for classification into either category 1 or category 2 and feedback was provided after each decision. Each learning phase was immediately followed by a test phase in which the system classified all category items without feedback. For each test item classified, an accuracy and a typicality rating was recorded. The typicality rating was computed based on the characteristics of the pattern making the classification decision. The exact way in which typicality rating was computed is described in chapter 4.

3.2.3. Manipulating Exemplar Order

Exemplar order was defined as the distribution of category items presented over learning blocks, which corresponded to using centered versus represented samples. The exemplar presentation orders within a block were completely randomized. The difference between the centered and representative sampling conditions was the amount of variation in the category members included in each sample. In the first block, centered sample over-represented the most frequently-occurring item type (i.e., type A) while it under-represented types B and C. Subsequent centered samples gradually introduce more variations by including more type B items in block 2 and type C items in block 3. All four representative samples correctly reflect the relative proportion of each item type. Despite the differences in sample variations in the first three blocks, all category items are eventually presented under each sampling condition. Thus, the only difference is the order in which LANA encounters them. Table 2 below gives the distributions of item types for both sampling conditions.

Table 2. Sample Structure

	Block 1	Block 2	Block 3	Block 4
Proportion	0 1 8 1 0	1 3 2 3 1	2 2 2 2 2	1 2 4 2 1
Centered Samples				
Proportion	1 2 4 2 1	1 2 4 2 1	1 2 4 2 1	1 2 4 2 1
Representa-tive Samples				

▩ Item types correctly-represented ■ Over-represented □ Under-represented

3.2.4. Simulating Learning Strategies

In specifying a computer model of strategy effects, one can take two different theoretical stances. One view of strategy believes that different strategies are associated with different category learning models, which implies that both representation and architecture are altered by strategies. In other words, a change in strategy implies a change in the basic model. Under this assumption, one needs, as Medin and Smith [1981] indicated, the following knowledge in order to understand categorization: (a) a list of possible strategies that might be used in a task, (b) a separate theory mapping each strategy onto performance, and (c) a meta-level theory specifying the factors governing strategy selection. Under this view, current theories of category learning are merely alternative procedures. In other words, all the models are correct and incorrect at least some of the time, depending on whether the strategies implied by a model are operative or not [Medin and Smith, 1981].

An alternative view has been proposed by Medin and Smith [1981], and supported either directly or indirectly by other researchers [e.g., Elio and Anderson, 1981; Medin, Dewey, and Murphy, 1983]. It argues that strategies induced by instructions alter the representation but not the fundamental nature of the underlying inductive processes. For example, a learning strategy may quantitatively change the criterion for pattern retrieval while leaving the nature of the basic retrieval process intact. In this thesis, it is assumed that learning strategies do not change the nature of the underlying architecture, nor do they add or remove processes.

Basically, the simulation of learning strategies was realized by alternating a few strategy parameters associated with a small set of basic processes. The last section in chapter 4 presents the mapping from behavioral and descriptive characterizations of analytic and nonanalytic learning to the changes in the processing mechanisms.

3.3. Basic Model Assumptions

This section describes the basic model assumptions and points out the key aspects of model architecture. Most models explored in the simulation were based on the following basic assumptions:

(a) A generalization process underlies the category learning. This process forms patterns of co-occurring features associated with a category.

(b) Feature patterns have an associated strength reflecting their proven "usefulness."

(c) Both the associated strength of a pattern and its similarity to an item influence whether the pattern is retrieved and used to classify the item or not.

(d) Learning strategies involve alternating strategy parameters associated with some basic processes, but they do not remove existing processes or add new ones.

In LANA, a limited item memory holds recently-seen items. To store category level abstractions, the model has a long-term memory for generalizations. The precise characteristics of these memories - size, retrieval process, and forgetting process -

constitute alternative models. The exact manners in which the generalization and discrimination processes operate and the schemes which reward and punish patterns also constitute different models.

An important theoretical assumption is whether item patterns and generalization patterns are distinguished in terms of processing assumptions. Under one view, the processes operating on them would be governed by different constraints, while an alternative view promotes the idea that both items and generalizations are simply patterns with equal processing status.

Alternative specifications of these assumptions and processes define the scope of the model space. Normally, a model's specification can be implemented in a number of different ways, and some implementations may implicitly correspond to additional assumptions which are undesirable or unintended. For example, to implement the forgetting of patterns, one could explicitly delete those that do not meet certain criteria (e.g., a pattern strength threshold). Alternatively, one could implement forgetting by specifying retrieval constraints so that those patterns failing to meet the constraints are not retrieved. However, these rules might still be available for other processes, which is not the case of explicit pattern-deletion. Thus, alternative ways of implementing a notion like "retain only useful patterns" may have profound effects.

Attempts have therefore been made to separate the assumptions from their specific implementations and to make sure that the implementations do not entail other unintended assumptions. The implementation itself is not important per se, the prior observation notwithstanding. The mapping between assumptions and their intended implementations is a difficult and challenging part of the simulation. Considering alternative implementations, the space of possible simulation models becomes extremely large and complex. Interactions frequently occur among various mechanisms in unforeseeable ways, making the isolation of effects from individual processes a difficult task.

3.4. Model Validation

Model validation procedure is a crucial aspect of any simulation work. In the LANA simulation, a model was evaluated by assessing its assumptions, its individual processes, and the degree of match between the model behavior and observed human data. The specific validation criteria were the following:

(a) Validation of assumptions and processes: A model's assumptions need to be consistent with empirical findings and psychological constraints; alternative implementations of assumptions as processes were verified.

(b) Validation of behavior: It was not important to match the exact numbers reported in the original experiments, but rather to simulate the main trends in the observed data. The first result to simulate was the interaction between exemplar order and learning strategy. When the system runs under a nonanalytic strategy, there should be a consistent advantage for centered exemplar order over that of the representative exemplar order. When the system runs under an analytic strategy, the reverse must occur. The second set of results to simulate was the interaction of performance with item types. Given centered samples, the system's block 1 classification performance should be better on type A items relative to types B and C items. This result was found for human learners and made intuitive sense, because block 1 included virtually no type B and C items. This advantage should decrease over blocks as the system encounters more B and C items. On the other hand, there should be a consistent improvement across blocks on all item types for the representative condition. Thus a model that simulated the interaction between learning strategy and exemplar order would have been rejected if these data trends were not observed.

Regarding model validation procedures, a rather informal approach was adopted which divided the evaluation into two steps; preliminary and final evaluation. During the preliminary evaluation, models whose results did not simulate the empirical data were rejected. This is very necessary for large scale preliminary exploration in order to cope with large model search space. Hence only those models that satisfied all the basic criteria in (b) were kept as candidates for final selection. During the final evaluation, plausible models were evaluated for their ability to simulate all the empirical trends as well as for the plausibility and parsimony of their assumptions. Compromises were often taken during the selection because some models simulated certain behavior aspects better than other models, but they did less well on certain other behavior aspects.

Preliminary Evaluation. This procedure considered the following:

1. the accuracy difference between the centered and the representative conditions on block 4 for correct direction and magnitude.
2. the typicality difference between the centered and the representative conditions over blocks 4 for correct direction and magnitude.
3. the accuracy difference between the centered and the representative conditions over blocks 1, 2, 3, and 4 for correct direction and magnitudes.
4. the typicality difference between the centered and the representative conditions over blocks 1, 2, 3, and 4 for correct direction and magnitudes.
5. the guess rates for both the centered and the representative conditions over all blocks (≤ 40% on block 4 was required). If LANA was making a large number of its decisions based on random guesses, this was taken as a signal that there was something fundamentally wrong with the model. Although we have no idea what the guess rate for humans was, we used 40% as the maximum acceptable guess rate for block 4 decisions.
6. the basic empirical trends presented in (b) listed above. For example, a basic upward trend (i.e., block 4 performance is better than block 1 performance) for both accuracy and typicality scores must be present.

These issues represented the informal guidelines used in the investigation and were not necessarily considered in order.

Final Evaluation. This procedure involved the following considerations:

1. the plausibility and necessity of the basic assumptions.
2. the total number of parameters involved in simulating the analytic and
3. the plausibility and significance of effect of parameters involved.
4. the degree to which a model simulated specific human data (discussed further in chapter 6).

It is clear that the final evaluation procedure is more subtle than the preliminary evaluation and involves a more comprehensive set of criteria. Also it emphasizes model comparison as opposed to model screening. It needs to be pointed out that the model validation was completely determined by the set of criterion used and the way in which they were applied.

3.5. Summary

This chapter has described the approach adopted for the LANA simulation, which is an instantiation of the general cognitive simulation approach outlined in chapter 2. However, there are a number of key simulation aspects that need to be summarized here.

Firstly, the target categories for learning were relatively large and ill-defined with various subcategory (item type) structures. Moreover, the approach must deal with the simulation of concept learning behavior over various learning "stages" (blocks), during which training samples were presented with different variance and order. The addition of this temporal dimension represented a further constraint on the part of the simulation models and the approach to be used.

Secondly, this particular approach must address the issue of how to simulate alternative learning strategies within a single architecture. This approach eliminated possibilities of simulating alternative learning strategies by adding extra processes or altering the existing ones in fundamental ways. As a results, the approach encouraged both simple and plausible ways to simulate learning strategies.

Chapter 4

LANA Framework

4.1. Overview

LANA's control structure has two components; a learning component and performance (classification) component, each defined by a set of processes. Adjustments in parameters associated with these processes correspond to the simulation of analytic and nonanalytic learning strategies. Since there are so many different processes and parameters explored in the simulation, this chapter describes only the set that constitutes the final models.

Section 2 describes the representation of patterns which include instances and generalizations. Section 3 explains how generalization patterns are formed and discriminated. Section 4 outlines LANA's control structure with learning and classification procedures. Section 5 describes the major processes and the associated parameters, followed by section 6 which characterizes the simulation of analytic and nonanalytic learning strategies. The last section summarizes the whole chapter.

4.2. Pattern Representation

The basic information-carrying structures in LANA are called patterns. There are two types of patterns: instance patterns and generalization patterns. Both instance patterns and generalization patterns consist of 5-dimension ordered tuples, with 4 possible values on each dimension. Each pattern also has an associated category membership tag, indicating the category this pattern belongs. Generalizations are distinguished from instances by the presence of a variable marker in place of a specific attribute value.

Both instances and generalizations have an associated strength. Pattern strength is a reflection of the pattern's past usefulness. The ways pattern strength are used and updated to influence processing can vary as a function of theoretical assumptions. In the models explored here, patterns with higher strength are generally preferred for making decisions over those with lower strength, other criteria being held constant.

Examples of typical instances and generalizations are shown in Table 3.

Table 3. Examples of Instances and Generalizations

	Feature Pattern	Strength	Remarks
Instance:	12213 -> cat.1	10	"->" separates condition-action
Generalization:	1-2- - -> cat.1	12	Variable "-" means *any value*

The semantics for these representations is: "when the feature pattern on the left-hand side of '->' is matched, classify the item into the category denoted by the category membership tag (either cat.1 or cat.2) on the right-hand side of '->'."

4.3. Generalization and Discrimination

A generalization represents a frequently co-occurring feature pattern. In most learning systems, generalizations are formed when two patterns occurring in close temporal proximity are compared. A new pattern is formed that retains the common elements of the two patterns and replaces the different attributes by variables. Generalization is often accompanied by a discrimination mechanism to adjust overly-general patterns that lead to incorrect judgments by reintroducing feature constant into the generalized pattern.

Pattern Generalization. Generalization occurs under two circumstances. The first is when a presented item is matched and correctly classified by a retrieved instance pattern in the same category. For example, if 11214 -> cat.1 is presented and 12231 -> cat.1 is the best pattern retrieved, then a generalization 1-2-- -> cat.1 would be formed and added to the generalization memory. This generalization is the ***maximum specific generalization*** (*MSG*) of the two patterns. In LANA, only the MSG is formed and added to memory when generalization occurs.

The second circumstance for generalization occurs when a generalization pattern partially matches a presented item and makes a correct classification. For example, when item 11214 -> cat.1 is partially matched by pattern 1-2-1 -> cat.1, then the classification of this item into category 1 is a correct decision. A new pattern, 1-2-- -> cat.1 is then formed, representing the portion of the retrieved generalization that is responsible for the correct decision.

Pattern Discrimination. If the pattern, --2-4 -> cat.2, is selected as the best match for item 11214 -> cat.1, the resulting decision would be wrong. Some learning mechanism should now operate to reduce the likelihood that this wrong decision will be made again. One such mechanism is discrimination, which is widely used in various learning systems. Discrimination tries to isolate what is responsible for the incorrect decision by focusing on the differences between the two patterns. Discrimination forms a new pattern more specific than the previously incorrect pattern by adding one or more features of the item to the incorrect generalization pattern. For the above example, a new generalization 1-2-4 -> cat.1 might be created by adding the first feature "1" and keeping the item's membership "cat.1". In models that prefer patterns with more constants, the new pattern will be selected over the original wrong pattern when similar situations arise, and therefore, reduce the chance of making the same mistake again.

Discrimination also operates when an item pattern leads to an incorrect decision. The result is a new pattern with the common attributes, plus one or more different attributes of the presented item. For example, if instance pattern 43244 -> cat.2 is retrieved from item memory and selected as the best match for the item 11214 -> cat.1, it will lead to an incorrect decision. Again, the goal is to create a pattern that would correctly classify the presented item by examining how the misclassified item differs from the retrieved item. In this case, a discrimination 1-2-4 -> cat.1 might be formed. This discrimination situation may be viewed as a kind of generalization, because it operates on specific items and generates a more general pattern in the same category as the presented item.

For discrimination, it is not always certain what aspect of the knowledge or what decision causes poor performance, hence it is not always clear how to assign "blame" in order to correct a wrong pattern. The difference between a presented item and an incorrectly applied pattern is the discriminative information, but there can be more than one difference. Suppose that the item 11214 -> cat.1 is incorrectly categorized by the pattern --2-4 -> cat.2. Table 4 below shows the set of differences between these two patterns and the possible discriminative feature sets.

Table 4. Components of Discrimination

Commonality	Difference	Discriminative Feature Set	Discrimination Set	
--2-4	11-1-	1----, -1---, ---1-	1-2-4 -> cat.1	-12-4 -> cat.1
		11---, -1-1-, 1--1-	--214 -> cat.1	112-4 -> cat.1
			-1214 -> cat.1	1-214 -> cat.1

One may choose to add any subset of these discriminations. In the model presented here, a simple random selection of a single discrimination was used because no justifiable ground has found in order to prefer one discrimination over another.

Generalization and discrimination processes are the primary sources of generalization pattern formation in LANA simulation. However, under the nonanalytic learning condition, LANA specifies that the frequently-used items (stored in the item memory) can be remembered permanently. This is implemented by transferring the items in item memory that are used twice or more to the generalization memory. However, some of the attributes of these items are dropped (at random) so that the items become item segments which resemble the form of a true generalization. As will be pointed out later, this scheme in fact provides LANA with a "permanent" memory for frequently-used items and it is related to the notion of analogy when these items are used.

4.4. Control Structure

LANA has a set of processes for learning and classification procedures operating under learning and testing phases, respectively. Learning consists of a classification judgment with feedback, followed by changes to the set of instance and generalization patterns that constitute the category knowledge. The control structure and processes used for classification are identical to those used for learning, except that no feedback is given and no change is made to the category knowledge.

4.4.1. Learning Procedure

Figure 2 on the next page presents a flow chart of the learning procedure based on one complete learning cycle.

Each learning cycle starts with the presentation of a training item to be classified to a category. Patterns that meet retrieval criteria are then retrieved. The retrieval criteria (described below) are also functions of theoretical assumptions, and the primary criteria used in LANA are pattern strength and similarity to the presented item. The precise definition of similarity is discussed in more detail later. The retrieved patterns may include both instance patterns and generalization patterns. These patterns are then scored using certain criteria and the "best" one is selected as the basis for classifying the training item. If no pattern is retrieved, a random classification is made. After feedback is given, the system may revise its generalizations in the memory, depending on what type of pattern (instance or generalization) was used for the judgment and what the outcome was. The revision could involve adding new patterns to generalization memory or increasing/decreasing a pattern's strength. The system updates its generalization and item memories, according to whatever forgetting schemes are in place. The system is now ready for the next learning cycle.

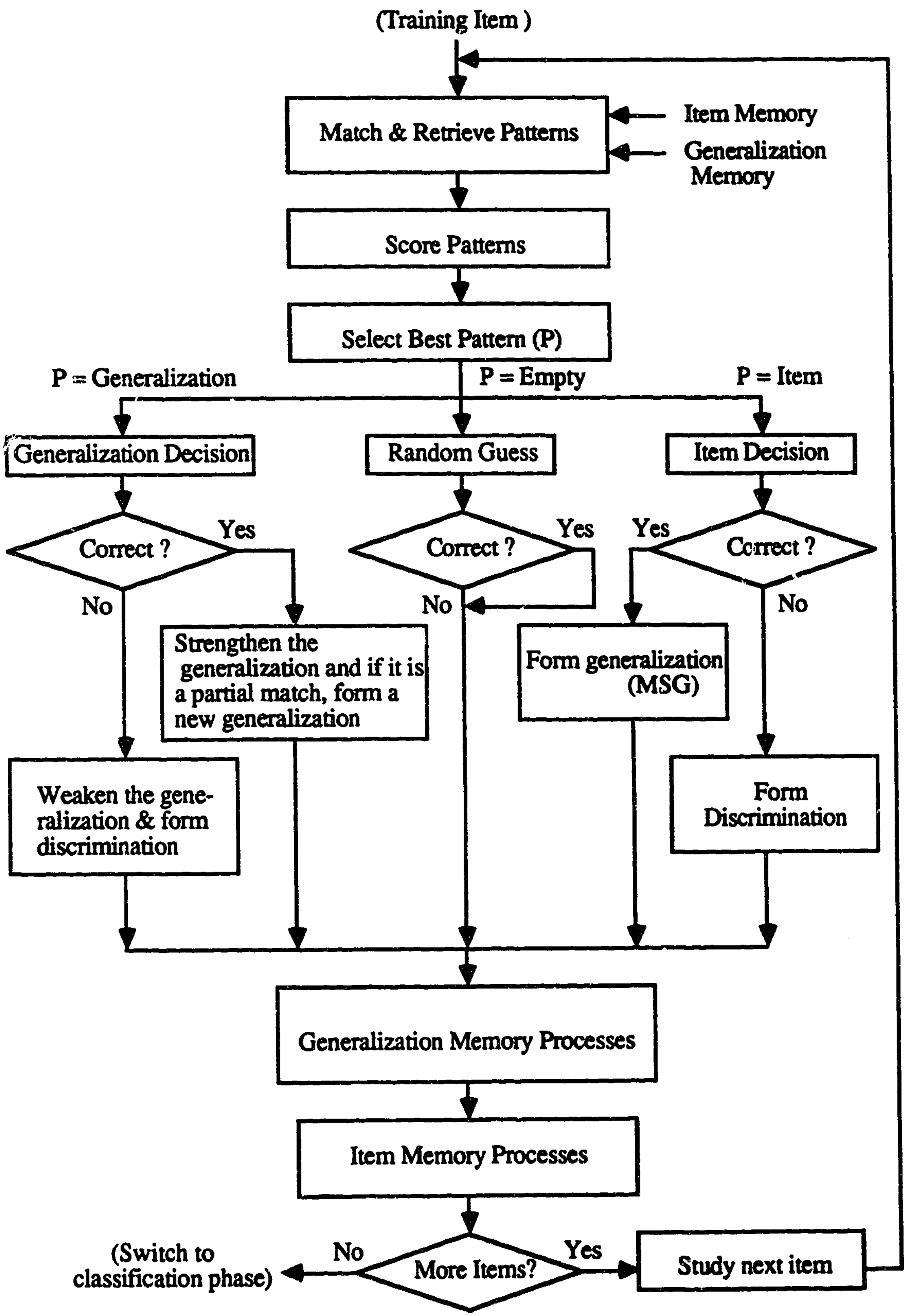


Figure 2. Flow Chart of Learning Procedures

4.4.2. Test Procedure

Classification decisions during the test phase are made exactly as they are in the learning phase. However, it is assumed that the test phase does not change category knowledge because the classification decisions are not followed by any feedback. There is one additional step required in classification but not in learning; a typicality decision is made by the system along with each classification decision, indicating how typical the test item is with respect to its assigned category.

Both confidence and typicality ratings have been measured in various experiments appearing in the literature [Anderson et al., 1979; Elio and Anderson, 1984; Hayes-Roth et al., 1977; Hintzman, 1986]. Generally speaking, confidence represents the degree of certainty about the learner's classification decision while typicality has to do with the representativeness of the item relative to its assigned category. For example, an instance such as *penguin* may be classified into the *bird* category with high certainty, but may receive a low typicality rating. Despite the differences, both measures are useful indications of learning performance, and both have been widely used in the same ways as has classification accuracy or error.

Although the distinction seems intuitive, it is difficult to know how confidence and typicality ratings can be accurately and differentially simulated. There have been experimental attempts to quantify one or the other, but there does not seem to be any experimental data that measure both. Intuitively, it would seem that confidence is more of a function of pattern strength while typicality must include some component of pattern similarity.

The LANA simulation was concerned only with typicality ratings, and various schemes were explored based on strength, similarity, and combinations of these two. Strength-based typicality was a computed score from the strength of the selected pattern. Similarity-based typicality was a measure of the similarity between the selected pattern and the test item. The combined typicality was defined as a function of both the pattern's strength and similarity. Typicality results based on the similarity scores and on the combined scores gave better fit to the observed data.

4.5. Major Processes

This section provides details on the basic processes governing forgetting, pattern matching, retrieval, generalization and discrimination, strengthening and weakening. In the descriptions of each process, we also discuss if and how they operate in quantitatively differently manners under analytic and nonanalytic learning strategies.

4.5.1. Forgetting Patterns from Memories

There are two memory structures in LANA: *item memory* (IM) and *generalization memory* (GM). Item memory is a temporary buffer of training items presented during learning. Generalization memory is a permanent store for holding generalizations formed during learning. No additional memory structure is assumed. Both item and generalization memories have corresponding forgetting mechanisms.

Forgetting Items. Item patterns in item memory obey the *first-in-first-out* rule. The oldest item is forgotten once the item memory capacity (typically set at 3) is reached. The memory size of 3 represents a plausible limit on the number of items that could be retained in the short term memory.

Item memory was modelled in a simple way because the focus was on learning mechanisms not on the short-term memory structures. However, the simulation tests did investigate the affects of different item forgetting schemes, that were motivated by considerations of analytic versus nonanalytic memory data. Both item memory capacity and forgetting schemes have implications for learning, because they define the item population on which learning mechanisms operate. These implications are discussed in subsequent chapters.

Forgetting Generalizations. Forgetting patterns from generalization memory may be modelled in two ways: (a) explicitly eliminating generalizations that fall beneath thresholds for some criteria metrics, and (b) constraining the retrieval process so that only generalizations that meet thresholds for the criteria metrics are retrieved and made available for subsequent decision making. This latter scheme has been referred to as functional forgetting [Feigenbaum, 1963].

All the forgetting schemes explored here have been based on the intuitive assumption that "useful" generalizations are more likely to be remembered and made available for use than "less useful" generalizations, where "useful" means that a pattern has led to correct decisions in the past. In LANA, this "usefulness" is reflected as the pattern strength. However, the usefulness is not to be confused with the "potential applicability" of a pattern to a particular item due to similarity.

In LANA, only the generalizations that exceed a strength threshold (if other criteria are also met) are available for retrieval during each cycle. Those that do not qualify for retrieval may still be available to other process such as strengthening recreated patterns. Each time a generalization participates in a correct decision, its strength increases, therefore increasing its likelihood of being retrieved again. As the strength of patterns increase over time (blocks), the strength threshold is adjusted accordingly. Therefore, the old "useful" patterns may not be retrievable if they do not lead to correct decisions in subsequent learning.

Although not in the final models, one forgetting scheme based on explicit elimination of forgotten patterns was explored. This scheme eliminated any generalization patterns whose strength was below threshold before moving from learning to testing. This scheme could be used in controlling pattern population and created an advantage for the centered sampling condition.

4.5.2. Retrieving Patterns

Both the learning and test phase require a classification judgment of some presented item. That judgment is based on a single "best" pattern retrieved from either item or generalization pattern memory.

The retrieval process can be divided into the following subprocesses:

- Match existing patterns to a given item, applying any matching constraints.
- Retrieve matched patterns, applying any retrieval constraints.
- Score patterns, applying the scoring procedure.
- Select the "best" pattern for decision, applying the selection criteria.

This division of the retrieval process into separate matching, retrieval, scoring, and selection steps is more for the sake of descriptive convenience than for theoretical reasons. They are implemented as sequential steps in LANA although they may well be parallel in human learning.

Similarity plays an important role in pattern retrieval. In LANA, similarity is defined as a function of *degree of match* and *degree of mismatch*.. The degree of match is defined as the number of corresponding attributes matched. For example, patterns 11214 -> cat.1 and 12213 -> cat.1 match on three attributes. The degree of mismatch is defined as the maximum number of specific features not matched. For example, the degree of mismatch between 11214 -> cat.1 and 31122 -> cat.1 is four, and that between 11214 -> cat.1 and 1-2-3 -> cat.1 is only one. The category membership tags (i.e., cat.1 or cat.2) and variables are not counted towards the degree of match or mismatch.

A model may require either a full match or allow a partial match, depending on whether it is the analytic or nonanalytic version of the model. Under full matching, all attributes in one pattern must completely match the corresponding attributes in the other pattern for the two patterns to match. Therefore, items will never fully match each other unless they are identical. In the case of a partial match, there may be specific attributes in the retrieved pattern that are not matched. In LANA, a minimum degree of match is specified as a parameter for controlling the similarity of matched patterns, and is currently set at 2 for both full and partial match.

Normally, there are multiple patterns that satisfy all matching and retrieval constraints and, therefore, are all retrieved. However, only one pattern is selected for classification decision in both learning and testing (although this could either be a frequency-based selection among conflict-set patterns or a group-voting type of decision making). To select the best pattern, a selection process scores patterns on the basis of their strength and their similarity to the presented item. Similarity score is an linearly increasing function of the degree of match (DM) and decreasing function of the degree of mismatch (DMM): similarity score = DM - DMM/3. Strength score is simply the pattern strength (scaled multiplying a constant). The inclusion of a strength component is equally important because it reflects how successful a particular pattern was in the past. Since there was no theoretical commitment to any complex weighing function, a summation function was chosen to combine them for simplicity so that: total score = (similarity score + strength score) / 2. The pattern with the highest total score is selected for the classification decision. When patterns tie for the highest score, LANA simply selects one at random.

However, the *select-best* criterion is sometimes criticized for over-emphasizing the importance of a single pattern. The argument is that a *group-vote* criterion may overcome this drawback by allowing multiple patterns to simultaneously participate in the final decision. Unfortunately, any group-vote scheme is computationally more expensive and likely to make the final decision more sensitive to the retrieval-set. Furthermore, it is more difficult to define a credibility of a group of patterns with highly variable individual credibilities. The LANA preliminary experiments showed that the select-best scheme is in fact quite powerful.

4.5.3. Pattern Strengthening and Weakening

Most generalization-based learning models include strength as an associated feature of abstracted patterns. Even some instance-based models borrowed the idea of using associative strength of memory instances to facilitate instance retrieval [Hintzman and Ludlam, 1980]. The key idea of strengthening patterns is that successful application of a pattern should increase its chance of being used again. In other words, pattern strength reflects the pattern's successfulness in leading to correct decisions in the past which, in turn, indicates its potential usefulness in the future.

All patterns receive an initial strength (e.g., 10) when they are created. Instance and generalization patterns may have different initial strength, depending on the strategies. Patterns are strengthened whenever they participate in a correct decision and weakened whenever they participate in an incorrect decision. However, the retrieval strength threshold might be set at 12. This means that patterns cannot participate in any decisions until their strength is at least 12. The other circumstance under which strengthening occurs is pattern recreation. If a pattern already exists, but it is formed again via the learning mechanisms, duplicate copies are not maintained. Instead, the original version of the generalization has its strength increased, which makes it more likely to be selected. This approach has profound implications for the simulation of information order effects, because the recreation of certain patterns will be influenced by the order in which items are presented.

Recreation effects can be viewed in two fundamentally different ways, either as resulting from some automatic, unconscious process, or as resulting from a conscious process by which a learner explicitly "notices" the reoccurrence of familiar patterns. The former can be realized as having a tight recreation requirement while the latter can be realized as having a loose recreation requirement. These two perspectives on pattern recreation seemed consistent with nonanalytic and analytic processing, respectively. The tighter/looser recreation requirements can be implemented by imposing a higher/lower strength thresholds for the corresponding generalization retrieval, respectively.

The function which assigns the strength increments and decrements has important effects on the learning curve. One type of strengthening method adds a small and absolute increment to the original strength as a reward and subtracts a decrement from the existing pattern strength as punishment. Alternatively, the amount of increment/decrement can be proportional to the pattern's current strength (e.g., 10%). The ACT model has used an absolute strength increment (+0.02) and a relative strength decrement (-25%). Note the behavioral implications of this approach: generalizations gain strength slowly by fixed increments for each reward. But each error has increasingly large and immediate effects on the strength of the pattern responsible for the error, and hence immediate affects on the pattern's likelihood of being used again. In other words, a pattern proves itself slowly, but if it has accumulated a very high strength that leads to its constant selection, one error will serve to temper this high strength.

In LANA, various methods regarding strength reward have been explored and compared including strengthening and/or weakening by an absolute amount, by a relative amount, by an amount proportional to the specificity of the pattern, and so on. In the end, it was found that a simple version worked adequately, which was to strengthen or weaken by one basic strength unit each time a generalization was to be rewarded or punished.

4.6. Simulating Analytic and Nonanalytic Strategies

The simulation of analytic and nonanalytic learning tried to identify processes or features of processes that might be consistent with previous empirical results, descriptive strategies, and theoretical assumptions about these two strategies. The following sections identify empirical results or theoretical assumptions presented in the literature and how they were realized in the LANA simulation.

A. Nonanalytic learning is more analogical in nature than analytic learning. Analytic learning is induced by stressing explicit rule-formation and hypothesis-testing [Brooks, 1978; Reber, 1976; Medin and Schaffer, 1978].

LANA's interpretation of other researchers' characterization of nonanalytic learning as more analogical and analytic learning as less analogical rests primarily on how pattern matching was done. In the LANA simulation, the "more analogical" aspect of nonanalytic learning was realized by allowing partial matching in all pattern matching circumstances. The "less analogical" aspect of analytic learning was realized by insisting on full matches between stored generalization patterns and a given item.

In analytic learning, the learner is generating and testing specific hypotheses about category membership rules. One interpretation of this perspective is that a hypothesized category assignment rule either matches an item or not. Hence, the full-match retrieval process further distinguishes the generalizations as explicit hypotheses. Nonanalytic learners, even though they are not consciously testing and generating hypotheses, nonetheless learn and make correct classification decisions. Its "more analogical" process may be based on varying degrees of similarity by permitting a less-than-perfect match to stored patterns.

B. Conscious hypotheses-testing in analytic learning may mean a greater likelihood of noticing when their hypotheses re-occur. Nonanalytic learners, by concentrating on memorizing items, may be less likely to notice these re-occurrences.

There is no direct empirical evidence for this conjecture, but it seems a plausible extension of Reitman and Bower's [1973] finding that analytic learners have better memory for hypotheses they tested than for items they saw. This conjecture was implemented in the model by specifying a lower strength threshold for generalization pattern retrieval under the analytic strategy than under the nonanalytic strategy. Recall that strength can increase through recreation. Thus, if the retrieval strength threshold is 12 and the initial pattern strength is 10, a pattern must re-occur twice before it can even participate in any decisions. This was the setting for nonanalytic, which meant that classification decisions were based primarily on item patterns in the buffer. This is consistent with a more analogic view of nonanalytic learning. In the analytic versions, any generalization created was immediately available for retrieval decision making, consistent with the notion that generalization represent consciously-formed rules the learner is interested in verifying.

C. Analytic learners are actively testing rules during learning and have poorer memory for items studied during learning [Reitman and Bower, 1973]. Nonanalytic learners have better memory for instances and rely more on instances to make judgments.

This suggested that, in analytic learning, the representation of studied training instances and the representation of generalizations are somehow distinguished in their representation or their processing. The mapping of this notion into an analytic model was done by instituting a preference to use generalizations, if they existed, over instance patterns, even if the generalizations did not match as well to the item presented. This was accomplished by giving instance patterns a lower initial strength than generalization patterns, making it unlikely for instances to compete with generalizations.

Under nonanalytic learning, LANA's pattern selection process treated all the patterns the same, whether they represented an instance or a generalization. Instance patterns received the same initial strength as generalizations and hence could compete on an equal footing in the pattern selection process.

The better memory for instances was implemented by transferring frequently-used items (in item memory) to the memory for generalizations, after randomly dropping some attributes. Therefore, if an item in item memory was used, say twice, in making a classification decision, part of it was remember as a pattern in the generalization memory permanently.

The parameter settings for the final LANA model and an alternative model were included in Appendix 3 and Appendix 4 for reference.

4.7. Summary

There are two components to indicate in the LANA simulation. The first is the process that constitute the general architecture and the second is how analytic and nonanalytic learning strategies were simulated as small, theoretically-motivated adjustments to the parameters associated with a small set of these processes.

The LANA architecture is basically an instantiation of a feature-set model of category learning. Hence, it includes generalization, discrimination, and strengthening mechanisms that are sensitive to frequently co-occurring sets of features across exemplars.

The following characteristics constitute the general architecture that underlies both the analytic and nonanalytic versions of LANA: the representation of instances and generalizations; how the generalization, discrimination, and strengthening mechanisms operate; how a pattern is selected to make a classification decision; and how instance and generalization patterns are memorized and forgotten.

Table 5 below presents a summary of the important parameters for simulating the analytic and nonanalytic strategies. The plural "models" indicates that there is a set of models, with slight differences among them, that are reasonable fits to both the analytic and non-analytic data.

Table 5. Summary of Analytic/Nonanalytic Model Differences

Model Aspect	Analytic	Nonanalytic
Pattern Matching	Full	Partial
Strength Threshold for Retrieval	Low	High
Items Compete with Generalizations	No	Yes
Frequently-Used Items Become Generalizations	No	Yes

The full versus partial match distinction is straightforward. The presence or absence of a processing distinction between instance and generalization patterns is realized under the analytic strategy primarily as a retrieval preference for generalization over items. If there are generalizations available, they are always preferred over any available instance pattern. Furthermore, the only way generalization patterns emerge in the analytic models is via generalization/discrimination mechanisms. In the nonanalytic models, they may emerge via generalization/discrimination processes, as well as imperfect permanent memory for frequently-used instance segments. Although both versions of LANA model employ a generalization process, the context in which generalization operates in the nonanalytic case is a more implicit, unconscious detection of co-occurring patterns: it takes longer both for abstracted generalization patterns to emerge and longer for them to have an impact on decisions. Finally, the full versus partial match difference further distinguishes how generalized patterns are treated under nonanalytic strategy as compared with their treatment under the analytic strategy: they are not intended to be rules, but rather patterns that imperfectly match.

Chapter 5
Implementation Details

5.1. Overview

From the implementation view point, LANA can be seen as a narrowly defined simulation environment. By proper manipulation of a set of parameters provided in LANA interface, one may simulate a relatively broad range of models with different characteristics. including models of different architectures (e.g., certain instance-based models). Within this simulation environment, various tools have been developed to facilitate the analysis and understanding the simulation behavior, and to allow flexible use of the system (e.g., experimental data collection, analysis, and plotting; batch simulation). The implementation of the key functions such as assessment of similarity, scoring, and strength assignment are implemented either by parameters or by separate functions. The environment provides a convenient interface through which users set the parameters for the simulation. The simulation output can be selectively displayed through this interface.

5.2. Simulation Details

The system is implemented using Xerox CommonLisp and runs on Xerox Lisp workstations running the Lyric operating system. The LANA simulation program contains around 5000 lines of Lisp code, organized into about 150 functions. The average (real) execution time for a standard experiment (i.e., 15 centered and 15 representative data files) ranges roughly from one to two hours, depending on the particular parameter settings and system resources.

5.3. Parameters and Their Settings

In LANA, models are defined by setting parameters to particular values. There are total of about 48 parameters that can be set by an experimenter, 32 of which are actual model parameters that define the characteristics of a model. The remaining parameters control other aspects of the experiment such as simulation input, output, and trace, experimental control, and the environment. Each parameter has a set of values ranging from two values up to 20 values. This gives at least a rough estimate of the size of the search space, not counting the major architectural changes not parameterized (e.g., calculation of similarity score between two patterns). For each model, however, often only a subset of the 32 parameters need to be set explicitly each time because many parameters can take default values. Figure 3 shows a subset of these parameters that became the focus of the most vigorous experimental testing.

Figure 3. Model Parameters and Their Defaults
(A subset of the total 32 parameters)

Parameter Name	Default Value
Maximum-item-memory	*3*
Maximum-generalization-memory	*1000*
Item-memory-scheme	*forget oldest pattern per cycle*
Functional-forgetting	*no*
Initial-generalization-strength	*10*
Pattern-matching	*full match*
#-of-generalizations-formed	*1*
#-of-item-discriminations-formed	*1*
Item-similarity-threshold	*2*
Discrimination-strength	*initial strength*
Generalization-similarity-threshold	*.2*
Strength-threshold	*10*
Threshold-adjusted	*yes*
Strength-increment	*1*
Strength-decrement	*1*
Item-compete-with-generalization	*yes*
Item-become-generalization	*no*
#-of-dimensions-to-drop	*3*
Pattern-scoring-method	*sum*

This list of parameters are briefly explained below.

Maximum-item-memory:	The maximum number of items retained in item memory.
Max-generalization-memory:	The maximum number of generalizations retained in the generalization memory. The default value of 1000 does not mean that this many will actually be available for processing; it is determined by other retrieval parameters.
Item-memory-scheme:	The manner in which items are forgotten from the item memory at the end of each cycle. An example of schemes explored includes *forgetting the given item if a generalization retrieved successfully classified it.*
Functional-forgetting:	Determines if below threshold generalization patterns will be removed from the permanent memory or not.
Initial-generalization-strength:	The initial strength given to newly-formed-generalizations.
Pattern-matching:	The manner in which patterns match each other. Possible values are *full match* and *partial match.*
Form-generalizations:	Total number of generalizations to form as a result of comparing two similar instances.
#-item-discrimination-formed:	Specifies the number of discriminations to form when items are compared and discriminated.
#-gen-discrimination-formed:	Specifies the number of discriminations to form when a generalization is compared with a given item and discriminated.
Item-similarity-threshold:	Specifies the minimum number of matched attributes before eligible for the retrieved item to register a match.
Discrimination-strength:	Specifies what strength a newly created discrimination should take. Possible values include values given by

initial-strength, or the strength of the parent patterns, i.e., the patterns from which the discrimination is formed.

Gen-similarity-threshold: Specifies the minimum number of matched attributes before eligible for retrieved generalization to claim a match.

Strength-threshold: Specifies the threshold above which generalization will become retrievable and useable.

Threshold-adjusted: Specifies whether strength-threshold will be adjusted over blocks. If the value is *no*, strength threshold is fixed.

Strength-increment: The amount of strength to be rewarded to useful patterns.

Strength-decrement: The amount of strength to remove from unuseful patterns.

Item-compete-with-gen: Whether items compete with generalizations during pattern selection process or not.

Item-become-generalization: Frequently-used items are transfered as generalization and remembered permanently. Certain dimensions are dropped as specified by #-of-dimensions-to-drop.

#-of-dimensions-to-drop: Effective only when item-become-generalization is set to *yes* and it controls how many dimensions to drop before frequently-used items become generalizations.

Pattern-scoring-method: Controls the ways the similarity and strength scores are combined.

5.4. Key Functions

In order to ensure that the LANA simulation results can be duplicated, the following key functions are provided to indicate how certain calculations are done.

(a) *Strength score = (pattern strength - strength threshold) + 1*

(b) *Similarity score = DM - DMM / 3*

where, DM is the degree of match for the patterns involved, and DMM is the degree of mismatch between the patterns. (Both (a) and (b) are used to rank retrieved patterns.)

(c) Typicality scores:

- *Strength typicality = ((pattern strength - strength threshold) / 5) - 1*
- *Similarity typicality = similarity score*
- *Combined typicality = (strength typicality + similarity typicality) / 2*

(d) Average typicality scores for a particular item type:

Average typicality score = (accumulated-correct - accumulated-incorrect) / total

where, accumulated-correct is the accumulated individual typicality score for all the *correct* decisions; accumulated-incorrect is the accumulated individual typicality for the *incorrect* decision.

Chapter 6
Simulation Results

6.1. Overview

The entire simulation effort investigated a search space of more than 500 models. Through this investigation, a family of models have been identified that can account for the basic results to varying extents. What seems important is that a small set of processes and associated parameters have been discovered as critical for simulating the effects of exemplar order, learning strategy, and their interaction. Although there were a large number of models that could simulate the interaction of order and strategy on the final block, only a subset reproduced the intended behavior across learning blocks and item types. It is interesting to note that a small number of parameters associated with some basic processes underlain most "successful" models. It turned out that these parameters produced the intended behaviors by properly influencing LANA's category knowledge development during learning. The sensitivity analysis provided evidence that these parameter settings were necessary to reproduce the behavior at least for the given assumption.

It must be pointed out that matching exact numbers were not important per se, rather the simulation concentrated on important behavior trends in the observed data. Models that seemed to match the numbers more closely but did not simulate the basic behavior trends were rejected. The generality of LANA models were tested by simulating one other empirical result by Hayes-Roth and Hayes-Roth [1977], which had been simulated by Anderson et al.'s ACT model [Anderson et al., 1979].

The rest of the chapter is organized as follows. Section 2 presents the results of LANA in matching the human data trends. This include comparisons of performances on the final block as well as across blocks and item types. Section 3 analyzes and explains LANA's behavior in terms of its knowledge structure and evolution. Aspects of the results not covered in section 2 are also discussed here. Section 4 discusses findings related to the major strategy parameters. Section 5 describes LANA's application to simulating one of Hayes-Roth and Hayes-Roth's [1977] results. Section 6 summarizes the entire chapter.

6.2. Basic Results

Before presenting the data, several clarifications must be made here. During test phase, both accuracy and typicality scores were recorded for each block as a function of item types. Typicality scores were based on either similarity or the average of pattern strength and similarity. The mean typicality score for an item type[1] was computed using the method by Elio and Anderson [1984]. It was the mean of the sum of typicality scores from correct classifications minus the sum of the typicality scores from the incorrect classifications, i.e., (summed typicality on correct decisions - summed typicality on incorrect decisions) / total number of item in the item type. The mean typicality scores were within the range of (-5, +5), same as the observed typicality data.

All the observed data presented here are taken from the results of Elio and Anderson's [1984] Experiment 3. The simulated data presented in this section correspond to the results from a particular analytic version and a particular nonanalytic version of the final LANA model, which was described in chapter 4. The parameter settings for the analytic and nonanalytic versions of this final model can be found in appendix 3. Table 6 on the next page presents the observed and simulated Block 4 accuracy and typicality data as a function of sampling condition and learning strategy. It shows the basic interaction between sample condition and learning strategy on block 4. Despite the minor discrepancies among the scales of these numbers, the exact ordering of the observed data for all conditions are correctly simulated. As indicated earlier, there were many models that simulated this reversal (interaction) when only Block 4 data were examined.

[1] There were five item types A, B, B', C, and C', but B and B' types and C and C' types were grouped as type B (B + B') and type C (C + C') for presentation convenience.

Table 6. Observed and Simulated Mean Block4 Accuracy and Typicality as a Function of Sample Condition and Learning Strategy

		Nonanalytic		Analytic	
		Observed	Simulated	Observed	Simulated
Centered Condition	Accuracy	.85	.81	.78	.83
	Typicality 1	2.60	1.11	2.11	1.38
	Typicality 2	2.60	2.24	2.11	1.85
Representative Condition	Accuracy	.81	.77	.87	.88
	Typicality 1	2.14	1.01	2.80	1.63
	Typicality 2	2.14	2.07	2.80	2.37

Note: Typicality 1: typicality scores based on similarity only
Typicality 2: typicality scores based on both strength and similarity

However, it seemed important to simulate the trends in performance on item types as a function of blocks. The ways in which accuracy and typicality scores changed as human learners encountered additional variance differed depending on the learning strategy and the sample condition in the observed data. Therefore, it was important to reject models that did not simulate some aspect of these learning trends. Table 7 on the next page presents the observed and simulated mean typicality and accuracy as a function of learning block, item type[1], and strategy. Since the typicality based on strength did not simulate some data trends as well as the other two typicality scores, it is not presented in Table 7.

In Table 7, consider first the observed accuracy trends in nonanalytic learning (top left). Performances in both centered and representative conditions improve over blocks.

[1] Elio and Anderson combined the data for the types B and C as "noncentered" item type, in contrast to A type items as "centered" type.

Table 7.
Observed and Simulated Mean Accuracy and Typicality as a Function of Block, Item Type, and Sample Condition

	Nonanalytic Strategy				Analytic Strategy			
	Accuracy							
	Observed		Simulated		Observed		Simulated	
Centered	A	B + C	A	B + C	A	B + C	A	B + C
Block 1	.76	.71	.82	.68	.80	.59	.90	.69
Block 2	.81	.80	.81	.70	.74	.74	.88	.74
Block 3	.82	.82	.81	.74	.74	.76	.89	.77
Block 4	.85	.84	.83	.79	.76	.80	.88	.80
Representative								
Block 1	.77	.73	.73	.69	.79	.75	.82	.86
Block 2	.79	.77	.75	.74	.84	.86	.86	.90
Block 3	.81	.76	.74	.71	.84	.85	.88	.91
Block 4	.80	.82	.75	.78	.83	.90	.86	.90
	Typicality (Similarity Only)							
	Observed		Simulated		Observed		Simulated	
Centered	A	B + C	A	B + C	A	B + C	A	B + C
Block 1	2.08	1.56	1.32	0.63	2.13	1.63	1.92	0.78
Block 2	2.32	2.19	1.21	0.76	1.78	1.74	1.95	1.26
Block 3	2.39	2.58	1.23	1.02	1.65	1.90	1.89	1.36
Block 4	2.79	2.57	1.29	1.00	1.92	2.24	1.68	1.18
Representative								
Block 1	1.91	1.53	0.77	0.75	2.18	1.94	1.41	1.65
Block 2	2.01	1.97	0.90	0.92	2.51	2.62	1.63	1.80
Block 3	2.16	1.90	0.95	1.00	2.63	2.60	1.56	1.76
Block 4	2.05	2.23	1.01	1.03	2.55	2.97	1.47	1.72
	Typicality (Strength and Similarity)							
	Observed		Simulated		Observed		Simulated	
Centered	A	B + C	A	B + C	A	B + C	A	B + C
Block 1	2.08	1.56	1.27	0.64	2.13	1.63	1.85	0.76
Block 2	2.32	2.19	1.64	1.04	1.78	1.74	2.06	1.22
Block 3	2.39	2.58	2.05	1.60	1.65	1.90	2.19	1.47
Block 4	2.79	2.57	2.55	2.05	1.92	2.24	2.27	1.53
Representative								
Block 1	1.91	1.53	0.81	0.75	2.18	1.94	1.32	1.55
Block 2	2.01	1.97	1.23	1.22	2.51	2.62	1.74	1.97
Block 3	2.16	1.90	1.59	1.59	2.63	2.60	1.95	2.28
Block 4	2.05	2.23	2.06	2.06	2.55	2.97	2.10	2.55

The representative performance increases slightly and nonsignificantly while centered performance shows a fairly large increase from blocks 1 to block 2 and levels off after after block 2. These sets of trends are well matched by the corresponding simulation data, except that type A simulated accuracy is high on block 1. In other words, the performance advantage of centered over representative is small on block 1 but larger on subsequent blocks for the observed data, but the LANA's performance shows steady differences even on the first the learning block. This phenomenon occurs in most cases for the observed data, where the differences in performance under alternative strategies is considerably small on block 1 compared to other blocks.

Now consider accuracy in the analytic learning case. In contrast to the above nonanalytic counterparts, the representative performance shows a large gain from block 1 to block 2 on both A and B+C type items. However, the accuracy for the centered condition on type A items shows a dip from block 1 to block 2 and levels off after that. The simulated accuracy data shows a similar trend, but the dip is considerably smaller. Similar to the nonanalytic accuracy data, the simulated analytic accuracy shows a larger difference between the centered and the representative for type A on block 1, but the the observed data has only a small difference. This can perhaps be explained by the fact that during the experiments, human learners did not start to get fully biased towards analytic strategy until the *end* of block 1 while LANA's corresponding strategy became operative at the very beginning of block 1. In general, although these accuracy trends in the observed data did not reach statistical significance, it seemed important to simulate the increase in accuracy as a function of block, and a differential trend in accuracy across blocks as a function of sample condition and learning strategy.

Table 7 indicates that all the observed B+C typicality ratings are simulated well by the corresponding simulation data. In fact, the improvement of type B+C over blocks holds true for both observed and simulated data, for both strategies, and for both sample conditions. For centered sample condition, these improvements should be even more notable because the samples introduce type B+C items mostly after the first block. This makes intuitive sense because as more items are studied, learners should perform better on these items. Now consider the observed analytic typicality data (middle and bottom right). As the corresponding observed accuracy data, there is a significant block by item type interaction for analytic typicality ratings: note how the observed typicality ratings drop on type A items and rise on type B+C under. This contrasts to the corresponding trends for nonanalytic learning typicality (middle and bottom left), in which both the observed and simulated results for type A items improve even as additional category variance is encountered in subsequent blocks. In this sense, the simulated analytic typicality scores (middle and bottom left) do not quite match the type A observed data. However, the similarity-based typicality does show a decline after block 2, but the shape of the curve is convex as opposed to concave for the observed data. This observed performance dip on type A items is not contradictory to intuition when considering the possible interference among different item types. This issues is addressed again at a later point.

In general, Table 7 shows that performances of B+C type are better simulated than performances of type A items for most situations. The concave accuracy curve for the analytic learning with centered sample condition is roughly simulated while the corresponding concave typicality curve is not adequately simulated. One quick explanation for these concave curves would be that interference among knowledge for different item types occurs and causes performance drop as one item type is intensively studied, followed by studying very different types of sample items. Since B+C type items are never presented with as high concentration as with A type items in the first centered sample, the gradual improvement seems intuitive. The fact that the last centered sample is representative explains the final boost in type A performance in this case.

As pointed out earlier, the data presented here is from a particular version of a LANA model. There are several competing models with slight differences that could account for these basic results. However, certain trends were better simulated by some models while certain other trends were simulated by a different set of models. The simulation results presented in appendix 1 is one such example, in which the concave curves of the type A centered accuracy and typicality under analytic learning and the nonanalytic type A centered accuracy, seem to be better simulated. Appendix 4 includes the parameter settings for both the analytic and nonanalytic strategy versions of this alternative model.

The results in Table 6 and Table 7 are based on nonprototype items, not including category prototypes. As expected, the performance on prototype types are consistently higher than the corresponding performance on the item types. One important reason is that prototypes have the highest intra-category similarity and least inter-category similarity, making them easier to be classified correctly. This is consistent with the findings by Medin and Schaffer [1978] that item classification is an increasing function of the intra-category similarity and decreasing function of inter-category similarity. However, performance on prototypes behaved rather differently, at least from the data reported by Elio and Anderson [1984]. Specifically, the observed block 4 accuracy and typicality for prototypes had no interaction between sample condition and learning strategy. However, LANA's prototype data did show interactions in the way similar to those presented in Table 6.

One performance related measure of the quality of the LANA simulation was the classification guess rate (not available for the observed data). When LANA can not retrieve category knowledge to classify an presented item, it has to guess. Therefore, the guess rate associated with any simulation can be considered as an indicator for the availability of category knowledge, and hence performance. Although LANA's guess rates could not be compared with the corresponding human data, they were used to detect progress in learning. For example, several models were rejected because their corresponding guess rates were too high (say, ≥ 50% at the end of block 4). Normally, guess rates formed a inverse relation with respect to classification accuracy. A flat learning curve would imply a slow reduction of guess rates. For the models presented here, guess rates were in the range of 20% to about 40% for both analytic and nonanalytic learning.

6.3. Behavior Analysis and Explanation

Given the results in Table 6 and 7, it is important to also understand, in terms of underlying processes, how and why LANA simulated the observed results. The following discussion explains the basic interaction simulated by LANA.

Under the nonanalytic learning strategy, the centered condition begins with a very low variance sample with mostly type A items. The high concentration of similar items makes it easier to generalize. However, since nonanalytic strategy, many generalizations are formed and used. In addition, under this strategy, many classification decisions are in fact based on the retrieval of instances instead of generalizations, showing similarities to the notion of classification by analogy. However, the few newly-formed generalizations do get recreated more often and then used during learning because they are, by definition, similar to most type A items. As a results, some of them (mostly from type A items) do become relatively strong in terms of strength. Since there is barely any type B items and no type C items in the first centered sample, almost no generalizations are formed, creating a highly biased knowledge structure.

Due to the tight retrieval constraint under this strategy, only those highly strong type A generalizations can be retrieved and used during test phase. This leads to good performance on type A items, and relatively poor performance on type B and C items (see Table 7). The partial match here allows these type A generalizations to match some type B or even type C items, contributing to the otherwise very poor performance on types B and

C items. Overall, the centered seems to get a "good start." (In fact the simulated data for this block is higher than it should be. See Table 7.)

In the second block for centered sampling, there is a higher proportion of type B items and fewer type A items. The partial match scheme permits the "surviving" generalizations (mostly formed from type A patterns on block 1) to match some of the type B items. The generalization and discrimination processes provide the centered the opportunity to discriminate some type A generalizations in order to cover some type B items without deteriorating type A performance. The concentration of type B items helps to form and recreate some new type B generalizations. As a results, the type A and some type B generalizations have relatively high strength due to the concentration of strength rewards. At the end of block 2, a centered run has less biased knowledge which leads to a notable performance increase for type B items and at least a steady performance for type A items.

For the third centered sample, which over-represents type C items a similar scenario happens. The big improvement is now on type C items and those on types A and B items remain steady or slightly increased, keeping centered runs in an clearly advantaged position (see Table 7).

Under the representative condition for nonanalytic learning, the situation is very different because there is no over-concentration of particular item types in any sample. For example, on block 1, the high variance representative sample creates a wide range of generalizations with relatively smaller chance of pattern recreation. Therefore, few of the generalizations are sufficiently strong to be retrieved under the tight generalization retrieval constraint for this strategy. The high variance sample also makes it hard for any qualified generalization to match enough items to gain strength. The shear number of generalizations creates a highly competitive situation in which the strength rewards become highly distributed. The partial match scheme makes this situation even worse. As a result, representative runs have a balanced but generally weak pattern distribution. Under the

LANA's forgetting scheme, most of the generalizations are in fact functionally forgotten for the subsequent blocks. The same scenario happens again and again on the subsequent learning blocks, keeping the overall performances at a relatively lower level compared to those under the centered condition (refer to Table 7).

The key point here is that the sample structure should match the strategy of learning in order to maximize performance [Elio and Anderson, 1984]. As they observed, when learning is basically passive and inaccurate, it is better off for learners (human or machine) to start with narrow but representative material as the first centered sample. The excessive representativeness or frequent re-occurring of similar instances allows these "passive" learners to get hold of something to reply and to improve on. On the other hand, representative samples, though reflective of the true category variabilities, provide nonanalytic learners relatively fewer opportunities to pick up regularities from examples and to reinforce on the abstracted regularities. If they continue to be nonanalytic, they would likely end up with a confused picture of the categories because any regularities abstracted at the beginning encounter counter-examples, which may discourage them from pursuing further.

However, the above effects reverse themselves when LANA models an analytic learning strategy. To explain this, consider the representative condition first. Under representative sampling, the high-variance samples lead to generalization that can cover wide range of items. These generalizations are used immediately under the analytic strategy. As a result, most of these representative generalizations (which are forgotten under the nonanalytic strategy) are retained and frequently used. The fact that items are not used for decisions under the analytic strategy makes it easier for generalizations to be used more often and to gain strength. The large population of generalizations implies that only highly useful patterns could become strong and remembered for later blocks. As a result, the classification performance for the representative is fairly good over all item types.

Under the analytic strategy, the centered condition begins with a biased category knowledge. The hypothesis testing strategy promotes full match as oppose to partial match. As a result, the set of biased generalization (mostly formed from type A items) can match few items other than those of type A, causing very poor performance on types B and C items (see Table 7). The average performance for the centered, therefore, is relatively low, creating a reversed situation as compared with the case under the nonanalytic strategy.

During the subsequent blocks, the centered starts to see more variance on types B and C items but with fewer type A items. The previously-developed type A generalizations, under the full match scheme, are not successful in matching or correctly classifying these type B and C items. As a result, fewer of them become strong. The subsequently abstracted generalizations are relatively weak because they have smaller item population to which they can fully match. Thus fewer generalizations are of high strength and they are mostly from type A. However, the initial advantage on the type A items gets lost (the beginning dip in type A performance in Table 7) due to the fact that some type A generalizations are forgotten because they cannot match enough items and gain strength when new variance is encountered. However, for types B and C items, the improvements are steady because both types are introduced relatively more gradually over time.

The intuition for this analytic learning situation is simply the reverse of the nonanalytic situation. Analytic learners are better off when they are presented with category material representative of the true category variance, because their strategy allows them to hypothesize and test various category contingencies and obtain a non-biased knowledge at the very beginning. In this case, the low variance centered samples cannot help them but provide harmful initial biases. The full match constraint prevents a revision of knowledge.

The basic model behavior can be considered as the result of LANA's knowledge structure development, i.e., the generalization base formed over time. At any given point in time, LANA's knowledge structure determines not only the its behavior at the current time but also influences what may be learned in the near future.

The following two types of generalization distributions are useful in characterize LANA's evolving knowledge: (a) *coverage distribution*, and (b) *strength distribution*. The coverage is defined in terms of the total number of items that can be matched by generalization(s) from a certain population. According to this definition, rule coverage reflects the category types which can be covered by generalizations and, therefore, it is particularly useful in examining the impact of similarity-related processes. The strength distribution is defined as the relation between pattern strength and total number of patterns having that strength. It is more useful in monitoring how strength-related processes influence knowledge structure.

Not surprisingly, the coverage distribution mirrors the sample variance distribution, i.e., when additional variance in certain item type is encountered, that part of the coverage distribution starts to grow. This explains in part the performance distributions over item types (see Table 7.). The coverage distribution also helps to understand the effect of using partial versus full match in overcoming type-sensitivity of generalizations. Note that the level of coverage tends to predict, but is different from, the level of performance because (a) generalizations make mistakes, (b) generalizations overlap on their covered territories.

The investigation of strength distribution revealed that LANA's functional forgetting scheme play the role as intended. This distribution makes it easy to see what part of the generalization population is being remember for any given level of retrieval threshold and how is population changes over time as retrieval threshold is adjusted over blocks.

6.4. Related Findings

One important finding that emerged from the exploration of potential analytic and nonanalytic learning was that it was difficult to simulate nonanalytic learning without allowing partial matches. In other words, most nonanalytic models that showed reasonable benefits for beginning with a low-variance allowed partial matching. If full match was used, the advantage either disappeared or went in the opposite direction. Similarly, it was difficult to model an analytic advantage for the representative condition over the centered condition without constraining the matching between retrieved generalizations and the test items to only full match.

This finding seems intuitively reasonable. First, it is not possible for an instance pattern to fully match a presented item (unless they are identical). Second, generalizations that must be fully matched to be retrieved correspond more closely to what one might think of as conscious "rules" about category membership. This fits the analytic learning strategy, where a representative sample yields "rules" that are more likely to fully match on subsequent blocks due to their coverages. This accounts for the accuracy and typicality disadvantage found for analytic learners, given a starting centered sample, who performed relatively poorly as they encountered more category variance.

The differential forgetting under different learning strategy is the second most important parameter for simulating the results. Recall that the functional forgetting of generalizations is implemented as pattern retrieval strength threshold, and it is set to different levels according to different strategies. A large number of simulation experiments indicated that it was critical to set a higher (i.e., 12) retrieval strength threshold for simulating a performance advantage for the centered condition, and it was equally important to set it at a lower level (i.e., 10) or base level to simulate the performance advantage for representation condition. The key to getting a "good start" for the centered condition is to set a higher strength threshold so that only frequently-appearing feature patterns can become strong enough to be retrieved. The strength distribution indicated that under centered condition, the first sample created quite a few very strong generalizations while the representative condition had a more even distribution of pattern strength.

The above two parameters were sufficient to roughly simulate the basic interaction of sampling condition and learning strategy. Two additional parameters (the third and the fourth parameters in Table 5 at the end of chapter 4) were important in simulating the key trends and the acceptable performance levels, i.e., items-compete-with-generalizations and frequently-used-items-become-generalizations. In general, the configuration of analytic parameter settings fosters a processing distinction in the formation and use of "rules". The configuration of nonanalytic parameter settings fosters a more similarity-driven approach that treats instance representations and generalization representations in similar manners in pattern selection process. It is interesting to note that the interaction was easily simulated for the first or even the second block. It was difficult to maintain an initial wrong bias in the centered-analytic simulation as new information was encountered over later blocks.

The above discussion regarding importance of parameters was further supported by the data from a set of experiments aimed at evaluating the relative importance of specific parameters to simulating the observed data. These experiments systematically changed individual parameters while holding the other parameters constant. It was found that by changing anyone in the set of four parameters in Table 5, the results were significantly deteriorated. For example, to test the relative importance of partial match, the remaining parameters were set to final values but used full match instead. The results were compared with those from the final nonanalytic model to examine significance of impact. This same procedure was repeated for all four parameters one at a time under both the analytic and nonanalytic learning strategies. Therefore, the results from the total of eight sensitivity experiments were sufficient to test the impact of each of the four final parameters under the two strategies. These results are summarized below.

Analytic Model Tests. Under the analytic strategy and using the final LANA model reported here, the following percentage differences (i.e., difference of percentages. e.g., 86% - 83.5% = 2.5) were observed and they were based on *average* accuracy performance over all item types.

(a) *Pattern-matching*: when pattern matching was set to partial match instead of full match, the representative advantages for blocks 1 - 4 were greatly reduced, eliminated, or even reversed. By block 4, there was a slight or no real advantage for the representative condition (0.5%).

(b) *Strength-threshold-for-retrieval:* when strength threshold was set to high (12) instead of low (10), the representative advantages over all blocks again disappeared or reversed. By block 4, there was a clear advantage for the centered condition instead of the representative condition (-4.0%). The guess rate for all blocks and under both sampling conditions were too high to accept (≥ 70% guess rate on block 4).

(c) *Items-compete-with-generalizations*: when items were allowed to compete with generalizations in the pattern selection process, the representative advantages for blocks 1 - 4 were significantly reduced. Block 4 showed an advantage for the representative condition (2.4%).

(d) *Frequently-used-items-become-generalizations*: when frequently-used items became generalizations, the representative advantages over blocks 1 - 4 were again greatly reduced. By block 4 there was an advantage for the representative condition (2.5%).

Nonanalytic Model Tests. Under the nonanalytic learning strategy, the same measurement was used and the same procedure was followed.

(a) *Pattern-matching*: when pattern matching was set to full match instead of partial match, the centered advantages for blocks 1 - 4 disappeared or even reversed. Block 4 showed an advantage for the representative instead of the centered condition (2.0%).

(b) *Strength-threshold-for-retrieval:* when strength threshold was set to low (10) instead of high (12), the centered advantages over all blocks disappeared or reversed. By block 4, there was again an advantage for the representative condition (2.5%).

(c) *Items-compete-with-generalizations*: when items were not allowed to compete with generalizations in the pattern selection process, i.e., when generalizations patterns, and not items patterns, were used for decisions, the centered advantages for blocks 1 - 4 were significantly reduced or eliminated; and block 4 showed an advantage for the centered condition (3.0%).

(d) *Frequently-used-items-become-generalizations*: when frequently-used items were not allowed to become generalizations, the centered advantages over blocks 1 - 4 were maintained. However, the guess rate became unacceptablely high for both conditions and over all blocks (≥ 60% on block 4). By block 4, the centered condition maintained a wide margin over the representative condition (8.0%).

The above sensitivity data were based on the accuracy differences only. In fact, similar deteriorations were found on other dependent variables such as typicality scores and data trends when the settings for the four parameters were different from what were proposed. These facts seemed to support the discussions made earlier about the validity and necessity of these four parameters in their proposed settings.

Furthermore, above experiments varied only one parameter at a time. Naturally, parameters interact and one could investigate all possible combinations of all parameters even within this constrained set. However, it was the previous experimentation that lead us to this particular set of parameters and we had a fairly good intuition that these parameters, in their final combination, constituted viable models. The only questionable parameter is the one that controls whether processing distinctions are made between item and generalization patterns. We claim that an analytic strategy is best modeled if only rule patterns are the basis for a decision, and several parameters in combination implemented that aspect of the model. Conversely, we claim that non-analytic learning is more analytic in nature and uses exemplar patterns primarily, "noticing" generalizations only with some difficulty. The parameter that prohibits exemplar patterns from being used under analytic strategy may not be as important as the others. Changing this parameter to allow exemplar pattern decisions reduced the magnitude of the desired effect, but not reversed the desired nonanalytic trends.

6.5. Simulating Other Results Using LANA

As a way of evaluating the generality of these models, both the analytic and nonanalytic versions of LANA model have been applied to Hayes-Roth and Hayes-Roth's [1977] experiments.

Specifically, their data on confidence ratings of classification decisions were simulated by using LANA's three types of typicality ratings based on strength, similarity, and both combined. The simulation of the experimental procedure followed the same procedure reported by Hayes-Roth and Hayes-Roth [1977]. A total of 30 simulation runs (corresponding to 30 learners) were conducted per experiment, each run receiving a different random input order. For further details on the experiment categories, design, and procedure, refer to either Hayes-Roth and Hayes-Roth [1977] or Anderson et al. [1979].

Table 8 below shows that under both analytic and nonanalytic strategies, LANA's simulations match the exact ordering of the observed data under all three confidence measures. Since Anderson et al. [1979] also simulated the same results using the ACT model, their data is also included here. It was discovered that this set of data was relatively easy to simulate and a number of the analytic and nonanalytic versions of LANA could simulate the exact ordering. The factors this thesis identified as central for analytic and nonanalytic processing are not relevant here. This was somewhat surprising, given the performance differences these parameters yield for the Elio and Anderson's [1984] task. One reason may be that Hayes-Roth and Hayes-Roth manipulated the frequency with which particular patterns may make certain generalizations salient even for our nonanalytic model.

Table 8. Observed and Simulated Mean
Confidence Ratings Over Four Types of Items

	Hayes-Roth Data	ACT Simulation	LANA Simulation: Analytic Version Strength	LANA Simulation: Analytic Version Similarity	LANA Simulation: Analytic Version Overall	LANA Simulation: Nonanalytic Version Strength	LANA Simulation: Nonanalytic Version Similarity	LANA Simulation: Nonanalytic Version Overall
Nonpracticed Prototypes	2.61	0.94	1.53	1.34	1.44	1.23	1.52	1.38
Much practiced Nonprototypes	2.34	0.86	1.31	1.30	1.30	1.08	1.05	1.06
Little practiced close-to-prototype	2.27	0.70	1.27	1.11	1.19	0.96	1.01	0.98
Little practiced far-from-prototype	2.01	0.41	0.97	1.02	0.99	0.77	0.94	0.86

Note: Overall confidence ratings are based on the combination of both strength and similarity

6.6. Summary

This chapter has presented the basic set of results produced from certain versions of LANA model. However, there was a family of competing models which could account for the results to different extents. The selected LANA model did a reasonable job as indicated in Table 7. The results from one competing model also simulated the results reasonably well as shown in the Appendix 1. The analysis of LANA's behavior in terms of its underlying mechanism and knowledge evolution formed a coherent picture with respect to the psychological explanation by Elio and Anderson [1984] for their empirical findings. The sensitivity analysis regarding the importance of all the four parameters confirmed that they are key components for this simulation.

The fact that a large number of models have been able to simulate the set of empirical results suggests that exemplar order, strategy, and their interactions in concept learning is computational in nature. In other words, there is at least one set of assumptions under which a computer model can explain these effects. Therefore, it is a phenomenon not just happening to human learners but to learning machines as well.

Chapter 7
General Discussions

This chapter discusses remaining issues related to LANA simulation. It highlights relations between this research and other areas of research including machine learning and cognitive science. The thesis closes by pointing out directions for future research.

7.1. Simulation Using Instance-Based Models

The preliminary exploration with LANA has led to some success. The interaction between information order and learning strategy has been simulated by alternating only a handful of strategy parameters. A variety of observed data trends were also simulated, including behavior patterns with respect to sample structure, learning block, learning strategy, and their interactions. The explanation of LANA's behavior based on its knowledge structure is basically consistent with the psychological explanations for the corresponding human behavior.

However, it would be interesting to know how other models other than LANA might perform on the same learning task and possibly account for some results. To investigate this possibility, three different instance-based models were constructed and tested. The first model was based on Hintzman's [1980] assumptions, but used LANA's existing memory settings and pattern matching scheme. The other two models were based on Medin's assumptions for instance-based models; one (Model B) stored identical instances as a single case and the other (Model C) stored them as separate cases. Forgetting schemes on all three model were based on all-or-none attribute forgetting with a given probability. The typicality ratings were computed and collected in the same way as in the LANA simulation. However, since Medin's assumptions did not include a strength component, the strength-plus-similarity typicality was not available for these models.

The block 4 mean classification accuracy and typicality for both the observed and simulated are presented in Appendix 2 (a). The mean accuracy data as a function of block and item type are given in Appendix 2 (b). The data indicate that in most cases there are slightly advantages for the representative condition over the centered condition across blocks and item types. However, as one would expect that an instance-based model should be able to simulate the nonanalytic learning more easily, the results suggested otherwise. This discovery led to the conjecture that an instance-based model may not be able to simulate the interaction results as easily as a feature-set mc[illegible], if at all. The simplest reason may be that, in LANA simulation, nonanalytic learning is best simulated by promoting certain characteristics associated with instance-based models within a generalization-based architecture.

7.2. Relation to Other Research

In addition to accounting for certain empirical results, LANA has implications to various areas of research in both machine learning and cognitive science. The LANA framework embodies a set of computational mechanisms for investigating human analytic and nonanalytic concept learning strategy, which has not yet been done with computer models. LANA simulation can also be viewed as an empirical study of a rule-based incremental concept learning system which is sensitive to a particular type of exemplar order and can be adjusted with alternative learning strategies. The learning mechanisms of LANA have characteristics of many other concept learning systems. The assessment of similarity, pattern matching, conflict resolution, knowledge retrieval, generalization, discrimination, strength assignment, and information forgetting processes are shared by many machine learning systems across various research paradigms. LANA's results are relevant to better understanding the effects of exemplar order for production-based learning and problem-solving systems.

As indicated in the literature review, the effects of exemplar variance and presentation order in learning samples have not been studied much with machine learning systems. Yet many concept learning systems are inherently order dependent, such as those in case-based learning (e.g., UNIMEM [Lebowitz, 1986]), rule-based learning (e.g., ACT [Anderson, 1979, 1986]), and other symbolic learning systems (e.g., Winston's system [1975]). This is simply because early experience affects the learning of later examples. Generally, most incremental concept learning systems are more likely to be order sensitive than nonincremental systems, although there are incremental learning systems that are not very sensitive to example order (e.g. version space [Mitchell, 1982]). The point is that if all the information from studied examples is remembered, a learning system tends to be less sensitive to order. Order effects often become a concern for systems that do not store all the information. Therefore, understanding the computational nature of order sensitivity may have direct influence on the understanding of the behavior of these learning systems. Although there are various "types" of exemplar orders, the one studied in this research has certain generality, especially when a relatively large set of training examples is learned incrementally.

The results of LANA simulation and related findings also have implications to the studies in computer-aided instruction (CAI) or intelligent tutoring systems (ITS). Having a model of human learning is important to designing such systems, which must infer the internal knowledge state of the learner from his or her behavior. In this regard, the effect of using different training materials and learners adopting different strategies may have profound influence on how a student model might be built for a CAI / ITS system.

The impact of different sample variance and orders on the abstracted concept descriptions also has direct relations to the automatic knowledge acquisition [Quinlan, 1983]. Automatic knowledge acquisition tries to effectively automate the process of knowledge engineering which involves acquiring large amount of facts and knowledge.

Presumably, the manner in which thi· large set of input is acquired portion by portion may influence the effectiveness of this acquisition process.

In cognitive science, this research helped to clarify and adjust the previous empirical explanations on the effects of exemplar order, strategy, and their interactions. The simulation model has not only supported the basic assumptions underlying feature-set models, but also tried to incorporate components of an instance-based model into feature-set model framework. Therefore, LANA simulation is related to those efforts trying to combine salient elements of different types of models.

7.3. Directions for Future Research

There are a number of improvements to be made to the current LANA framework. One area of improvement regarding the simulation of learning strategies is to define a "neutral" strategy (according to any empirical results) and to map it to LANA computational processes and strategy-parameter settings. Although the analytic and nonanalytic learning strategies were simulated by LANA as two "extreme" cases, there is no reason to believe that these strategies are "all-or-none" phenomenon. Currently, it is not clear what the exact behavior characteristics are for a neutral learning strategy; it seems that it must be a strategy compromising the two extremes. Once defined, a neutral strategy provides a "logical" case to which both strategies can and should be compared.

It is certainly important to continue testing the model presented here on other empirical results generated for different type of category learning tasks and with different type of categories. For example, it would be interesting to see how well LANA would perform on categories tasks for investigating analytic or nonanalytic learnings, such as those used by Reber [1976] and Brooks [1978].

Similarly, it is perhaps also interesting to see how the LANA framework compares with other existing machine learning systems on common learning tasks. One possibility is to choose a well studied machine learning system and make it perform the category learning task assigned to LANA. The results from such comparison may shed light on the possible applications of LANA's strategy simulation to performance systems. They may also suggest further improvement to the current LANA framework.

BIBLIOGRAPHY

Aikins, J. S. (1983). Prototypical knowledge for expert systems. *Artificial Intelligence*, **20**, 163-210.

Anderson, J. R. (1976). *Language, Memory and Thought*. Hillsdale, N. J.: Lawrence Erlbaum Associates.

Anderson, J. R. (1986). Knowledge compilation: the general learning mechanism. In Michalski, R. S., Carbonell, J. G. and Mitchell, T. M. (Eds.), *Machine Learning: An Artificial Intelligence Approach*, Vol. **2**. Tioga: Palo Alto.

Anderson, J. R., Kline, P. J., & Beasley, C. M. (1979). A general learning theory and its application to schema abstraction. In G. H. Bower (Ed.), *The Psychology of Learning and Motivation*, Vol. **13**. New York: Academic Press.

Anderson, J. R., & Bower, G. H. (1973). *Human Associative Memory*, Washington, D.C.: Winston.

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

Brooks, L. (1978). Nonanalytic concept formation and memory for instances. In E. Rosch & B. Lloyd (Eds.), *Cognition and Categorization*. Hillsdale, N.J.: Erlbaum.

Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1956). *A Study of Thinking*. New York: Wiley.

Elio, R., & Anderson, J. R. (1981). The effects of category generalizations and instance similarity on schema abstraction. *Journal of Experimental Psychology: Human Learning and Memory*, **7**, 397-417.

Elio, R., & Anderson, J. R. (1984). The effects of information order and learning mode on schema abstraction. *Memory & Cognition*, **12**(1), 20-30.

Feigenbaum, E. A., & Feldman, J. (1963). *Computers and Thought*. New York: McGraw - Hill Inc..

Feigenbaum, E. A. (1963). The simulation of verbal learning behavior. *In Computers and Thought*, Feigenbaum, E. A., & Feldman, J., 297-309, New York: McGraw - Hill Inc..

Franks, J. J., & Bransford, J. D. (1971). Abstraction of visual patterns. *Journal of Experimental Psychology*, Vol. **90**, No. 1, 65-74.

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

Hintzman, D. L. (1986). "Schema abstraction" in a multiple-trace memory model

Psychological Review, Vol. 93, No. 4, 411-428.

Hintzman, D. L., & Ludlam, G. (1980). Differential forgetting of prototypes and old instances: simulation by an exemplar-based classification model. *Memory & Cognition*, 8, 378-382.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.

Jackson, P. (1986). *Introduction to Expert Systems*. London, U.K.: Addison-Wesley Publishing Company.

Katz, J. J., & Postal, P. M. (1964). *An Integrated Theory of Linguistic Descriptions*. *Cambridge*, MA: MIT Press.

Kearsley, G. (1987). *Artificial Intelligence and Instruction: application and methods*. Addison-Wesley Publishing Company.

Kemler-Nelson, D. G. (1984). The effect of intention on what concepts are acquired. *Journal of Verbal Learning and Verbal Behavior*, 23, 734-759.

Klahr, D., Langley, P., & Neches, R. (1987). *Production System Methods of Learning and Development*. MA.: MIT Press.

Kolodner, J. L. (1983). Maintaining organization in a dynamic long-term memory, *Cognitive Science*, 7, 243-280.

Lebowitz, M. (1986). Concept learning in a rich input domain: generalization-based memory. *In Machine Learning: An Artificial Intelligence Approach*, Vol. 1, in Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. Morgan Kaufman: Los Altos.

Medin, D. L., Dewey, G. I., & Murphy, T. D. (1983). Relationships between item and category learning: evidence that abstraction is not automatic. *Journal of Experimental Psychology: Learning Memory and Cognition*, Vol. 9, No. 4, 607-625.

Medin, D. L., & Schaffer, M. M. (1978). A context theory of classification learning. *Psychological Review*, 85, 207-238.

Medin, D. L., & Smith, E. E. (1981). Strategies and classification learning. *Journal of Experimental Psychology: Human Learning and Memory*, 7, 241-253.

Medin, D. L., Wattenmaker, W. D., & Michalski, R. S. (1987). Constraints and preferences in inductive learning: an experimental study of human and machine performance. *Cognitive Science*, 299-399.

Michalski, R. S. (1986). Understanding the nature of learning: issues and research directions. In Michalski, R. S., Carbonell, J.G. & Mitchell, T. M. (Eds.), *Machine Learning: An Artificial Intelligence Approach*, Vol. 1. Tioga: Palo Alto.

Michalski, R. S. (1983). A theory of methodology of inductive learning. *Artificial Intelligence*, Vol. 2, 111-161.

Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (1983). *Machine Learning: An*

Artificial Intelligence Approach, Vol. 1, Morgan Kaufmann: Los Altos.

Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (1986). *Machine Learning: An Artificial Intelligence Approach*, Vol. 2, Morgan Kaufmann: Los Altos.

Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence*, Vol. **18**, 203-226.

Murphy, G. L., & Medin, D. L. (1985). The role of theories in conceptual coherence. *Psychological Review*, Vol. **92**, No. 3, 289-316.

Newell, A., & Simon, H. A. (1963). GPS, a program that simulates human thought. In *Computers and Thought*, Feigenbaum, E. A. & Feldman, J.(Eds.), 279-293, New York: McGraw - Hill Inc..

Posner, M. I., & Keele, S. W. (1968). On the genesis of abstract ideas. *Journal of Experimental Psychology*, Vol. 77, No. 3, Part 1, 353-363.

Posner, M. I., & Keele, S. W. (1970). Retention of abstract ideas. *Journal of Experimental Psychology*, **83**, 304-308.

Quinlan, J. R. (1979). Induction over large data bases. *Technical Report HPP-79-14*, Stanford University.

Reber, A. S. (1976). Implicit learning of synthetic languages: The role of instructional set. *Journal of Experimental Psychology: Human Learning and Memory*, Vol. **2**, No. 1, 88-94.

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

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

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

Slatter, P. E. (1987). *Building Expert Systems*. London, U.K.: Ellis Horwood Limited.

Smith, E. R. (1988). Category accessibility effects in a simulated exemplar-based memory. *Journal of Experimental Social Psychology*, **24**, 448-463.

Tversky, A. (1977). Features of similarity. *Psychological Review*, **84**, 327-352.

Uhr, L. & Vossler, C. (1963). A pattern-recognition program that generates, evaluates and adjusts its own operators. In *Computers and Thought*, Feigenbaum, E. A. & Feldman, J. (Eds.), 251- 268, New York: McGraw-Hill Inc..

Utgoff, P. E. (1986). Shift of bias for inductive concept learning. In R. S. Michalski, J. G. Carbonell and T. M. Mitchell(Eds), *Machine Learning: An Artificial Intelligence Approach*, Vol. 2. Morgan Kaufmann: Los Altos.

Winston, P. H. (1975). Learning structural descriptions from examples. In P. H. Winston(Ed.), The psychology of computer vision. New York: McGraw - Hill Inc..

Wisniewski, E. J., & Anderson, J. A. (1988). Some interesting properties of a connectionist inductive learning system, Proceedings of the Fifth International Conference on Machine Learning.

Appendix 1. Simulation Results of Alternative Model

Appendix 1 (a). Observed and Simulated Mean Block4 Accuracy and Typicality as a Function of Sample Condition and Learning Strategy

		Nonanalytic		Analytic	
		Observed	Simulated	Observed	Simulated
Centered Condition	Accuracy	.85	.86	.78	.83
	Typicality 1	2.60	1.44	2.11	1.49
	Typicality 2	2.60	2.41	2.11	1.81
Representative Condition	Accuracy	.81	.81	.87	.87
	Typicality 1	2.14	1.27	2.80	1.68
	Typicality 2	2.14	1.84	2.80	2.39

Note: Typicality 1: typicality scores are based on similarity only
Typicality 2: typicality scores are based on both strength and similarity

Appendix 1 (b). Observed and Simulated Mean Accuracy and Typicality as a Function of Block, Item Type, Sample Condition, and Strategy

Accuracy

	Nonanalytic Strategy				Analytic Strategy			
	Observed		Simulated		Observed		Simulated	
Centered	A	B + C	A	B + C	A	B + C	A	B + C
Block 1	.76	.71	.81	.70	.80	.69	.90	.68
Block 2	.81	.80	.84	.76	.74	.74	.89	.74
Block 3	.82	.82	.82	.77	.74	.76	.88	.82
Block 4	.85	.84	.89	.84	.76	.80	.90	.79
Representative								
Block 1	.77	.73	.62	.68	.79	.75	.81	.85
Block 2	.79	.77	.74	.79	.84	.86	.85	.92
Block 3	.81	.76	.76	.77	.84	.85	.88	.92
Block 4	.80	.82	.77	.83	.83	.90	.84	.89

Typicality (Similarity Only)

	Nonanalytic Strategy				Analytic Strategy			
	Observed		Simulated		Observed		Simulated	
Centered	A	B + C	A	B + C	A	B + C	A	B + C
Block 1	2.08	1.56	1.24	0.62	2.13	1.63	1.92	0.79
Block 2	2.32	2.19	1.38	0.95	1.78	1.74	1.94	1.29
Block 3	2.39	2.58	1.47	1.26	1.65	1.90	1.85	1.42
Block 4	2.79	2.57	1.55	1.38	1.92	2.24	1.77	1.29
Representative								
Block 1	1.91	1.53	0.43	0.58	2.18	1.94	1.41	1.69
Block 2	2.01	1.97	0.79	0.93	2.51	2.62	1.67	1.87
Block 3	2.16	1.90	1.04	1.19	2.63	2.60	1.59	1.82
Block 4	2.05	2.23	1.16	1.35	2.55	2.97	1.51	1.79

Typicality (Strength and Similarity)

	Nonanalytic Strategy				Analytic Strategy			
	Observed		Simulated		Observed		Simulated	
Centered	A	B + C	A	B + C	A	B + C	A	B + C
Block 1	2.08	1.56	1.05	0.53	2.13	1.63	1.79	0.75
Block 2	2.32	2.19	1.61	1.10	1.78	1.74	1.93	1.20
Block 3	2.39	2.58	2.04	1.70	1.65	1.90	2.05	1.49
Block 4	2.79	2.57	2.59	2.29	1.92	2.24	2.19	1.56
Representative								
Block 1	1.91	1.53	0.45	0.53	2.18	1.94	1.29	1.57
Block 2	2.01	1.97	0.88	1.05	2.51	2.62	1.77	1.96
Block 3	2.16	1.90	1.29	1.49	2.63	2.60	1.97	2.31
Block 4	2.05	2.23	1.66	1.95	2.55	2.97	2.10	2.57

Appendix 2. Simulation Results using Instance Models

Appendix 2 (a). Observed and Simulated Mean Block 4 Accuracy and Typicality as a Function of Sample Condition and Learning Strategy

		Observed	Simulated		
		Human Data	Model A	Model B	Model C
Centered Condition	Accuracy	.85	.95	.90	.90
	Typicality 1	2.60	1.79	1.71	1.09
	Typicality 2	2.60	2.96	-	-
Representative Condition	Accuracy	.81	.94	.93	.94
	Typicality 1	2.14	1.76	1.61	1.14
	Typicality 2	2.14	3.07	-	-

Note: Typicality 1: typicality scores based on similarity only
Typicality 2: typicality scores based on both strength and similarity

Appendix 2 (b). Observed and Simulated
Mean Accuracy and Typicality as a Function of Block,
Item Type, and Sample Condition for Three Instance-Based Models

	Accuracy							
	Observed Human Data		Simulated Model A		Simulated Model B		Simulated Model C	
Centered	A	B + C	A	B + C	A	B + C	A	B + C
Block 1	.76	.71	.82	.78	.89	.81	.81	[illegible]
Block 2	.81	.80	.87	.87	.90	.87	.86	[illegible]
Block 3	.82	.82	.90	.94	.89	.89	.88	[illegible]
Block 4	.85	.84	.93	.96	.89	.90	.89	.91
Representative								
Block 1	.77	.73	.79	.83	.86	.82	.86	.84
Block 2	.79	.77	.88	.93	.85	.86	.88	.88
Block 3	.81	.76	.91	.94	.92	.90	.91	.93
Block 4	.80	.82	.93	.95	.93	.93	.93	.95
	Typicality (Similarity Only)							
Centered	A	B + C	A	B + C	A	B + C	A	B + C
Block 1	2.08	1.56	1.37	1.06	1.51	1.28	1.05	1.03
Block 2	2.32	2.19	1.59	1.48	1.63	1.61	1.06	1.03
Block 3	2.39	2.58	1.65	1.73	1.92	1.52	1.07	1.51
Block 4	2.79	2.57	1.77	1.83	1.98	1.50	1.08	1.10
Representative								
Block 1	1.91	1.53	1.15	1.26	1.49	1.25	1.05	1.05
Block 2	2.01	1.97	1.57	1.70	1.71	1.51	1.07	1.08
Block 3	2.16	1.90	1.67	1.74	1.80	1.71	1.10	1.14
Block 4	2.05	2.23	1.75	1.80	1.66	1.58	1.11	1.17

Appendix 3. Final Model Parameter Settings

A: Parameter Setting for Nonanalytic Strategy

Maximum-item-memory	*3*
Maximum-generalization-memory	*1000*
Item-memory-scheme	*forget oldest pattern per cycle*
Functional-forgetting	*yes*
Initial-generalization-strength	*10*
Pattern-matching	***full match***
#-of-generalizations-formed	*1*
#-of-item-discriminations-formed	*1*
Item-similarity-threshold	*2*
Discrimination-strength	*initial strength*
Generalization-similarity-threshold	*2*
Strength-threshold	***10 (low)***
Threshold-adjusted	*yes*
Strength-increment	*1*
Strength-decrement	*1*
Item-compete-with-generalization	***no***
Item-become-generalization	***no***
#-of-dimensions-to-drop	*3*
Pattern-scoring-method	*add*

B: Parameter Setting for Nonanalytic Strategy

Maximum-item-memory	*3*
Maximum-generalization-memory	*1000*
Item-memory-scheme	*forget oldest pattern per cycle*
Functional-forgetting	*yes*
Initial-generalization-strength	*10*
Pattern-matching	***Patial match***
#-of-generalizations-formed	*1*
#-of-item-discriminations-formed	*1*
Item-similarity-threshold	*2*
Discrimination-strength	*initial strength*
Generalization-similarity-threshold	*2*
Strength-threshold	***12 (high)***
Threshold-adjusted	*yes*
Strength-increment	*1*
Strength-decrement	*1*
Item-compete-with-generalization	***yes***
Item-become-generalization	***yes***
#-of-dimensions-to-drop	*3*
Pattern-scoring-method	*add*

Note: Highlighted parameters are the ones that require different parameter settings for the alternative strategies.

Appendix 4. Alternative Model Parameter Settings

A: Parameter Setting for Analytic Strategy

Maximum-item-memory	*3*
Maximum-generalization-memory	*1000*
Item-memory-scheme	*forget oldest pattern per cycle*
Functional-forgetting	*yes*
Initial-generalization-strength	*10*
Pattern-matching	***Patial match***
#-of-generalizations-formed	*1*
#-of-item-discriminations-formed	*1*
Item-similarity-threshold	*2*
Discrimination-strength	*initial strength*
Generalization-similarity-threshold	*2*
Strength-threshold	***12 (high)***
Threshold-adjusted	***yes***
Strength-increment	*1*
Strength-decrement	***1***
Item-compete-with-generalization	***yes***
Item-become-generalization	***yes***
#-of-dimensions-to-drop	*3*
Pattern-scoring-method	*add*

B: Parameter Setting for Nonanalytic Strategy

Maximum-item-memory	*3*
Maximum-generalization-memory	*1000*
Item-memory-scheme	*forget oldest pattern per cycle*
Functional-forgetting	*yes*
Initial-generalization-strength	*10*
Pattern-matching	***Patial match***
#-of-generalizations-formed	*1*
#-of-item-discriminations-formed	*1*
Item-similarity-threshold	*2*
Discrimination-strength	*initial strength*
Generalization-similarity-threshold	*2*
Strength-threshold	***12 (high)***
Threshold-adjusted	***no (i.e., fixed at 12)***
Strength-increment	*1*
Strength-decrement	*0*
Item-compete-with-generalization	***yes***
Item-become-generalization	***yes***
#-of-dimensions-to-drop	*3*
Pattern-scoring-method	*add*

Note: Highlighted parameters are the ones that require different parameter settings for the alternative strategies.