1992

Resilience of prototypes to ambiguous feedback

Scott Sarnikowski
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Sarnikowski, Scott, "Resilience of prototypes to ambiguous feedback" (1992). *Master's Theses*. 504.
DOI: <https://doi.org/10.31979/etd.97n4-zebx>
https://scholarworks.sjsu.edu/etd_theses/504

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 1351075

Resilience of prototypes to ambiguous feedback

Sarnikowski, Scott, M.A.

San Jose State University, 1992

Copyright ©1992 by Sarnikowski, Scott. All rights reserved.

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

RESILIENCE OF PROTOTYPES TO AMBIGUOUS FEEDBACK

A Thesis

Presented to

the Faculty of the Department of Psychology

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Masters of Arts

by

Scott Sarnikowski

Advisor: Robert G. Cooper, Ph.D.

December, 1992

© All rights reserved 1992
Scott Sarnikowski

APPROVED FOR THE DEPARTMENT OF PSYCHOLOGY

Dr. Robert Cooper

Dr. Robert Fox

Dr. Mildred Alvarez

APPROVED FOR THE UNIVERSITY

ABSTRACT

RESILIENCE OF PROTOTYPES TO AMBIGUOUS FEEDBACK

by Scott Sarnikowski

Along with death and taxes, there are two additional truisms in life: The world is ambiguous and a large amount of our learning involves prototypes. Surprisingly, the effect of ambiguity on prototype formation has not been explored. The present study investigated this issue through a two-phase procedure. The subjects were 90 San Jose State University students. In the first phase (acquisition phase), the subjects were presented with stimuli (based on Franks & Bransford, 1971) which they sorted into two classes. Immediately following each presentation, the subject groups were given different levels of valid feedback on their judgment. In the second phase (recognition phase) the subjects stated whether they had seen the stimulus and stated their confidence. The key finding of the present study is that the formation of prototypes is independent of feedback. In addition, this study proposes a new memory store to address this feedback-independent behavior.

ACKNOWLEDGMENTS

I would like to thank the University Foundation of San Jose State for their support in procuring the equipment used in this experiment through the University Foundation Research Grant program. I would also like to thank my thesis committee, Robert Cooper, Robert Fox, and Mildred Alvarez, for their time and effort. I would like to especially thank Robert Cooper for shepherding me through my journey from engineer to psychologist over the last four years. To Peter Reynen, I wish you had lived long enough to see the profound effect you had on my life. Finally, to my wife Sue, thanks for all the edits and late nights, this was only possible because of your love and encouragement.

TABLE OF CONTENTS

SECTION	PAGE
INTRODUCTION	3
The Field of Prototype Learning.....	4
Prototype Learning Research and Theories.....	4
The Field of Probability Learning.....	5
Probability Learning Research and Theories.....	6
Regression Theory.....	7
Bayesian Theory.....	8
Hypothesis Sampling.....	10
Heuristics Approach.	12
Integrating Prototype and Probabilistic Learning	13
Research Strategy.....	15
Description of the Franks and Bransford Procedure.	16
General Predictions.....	17
METHOD	18
Subjects	18
Design	19
Acquisition Phase.	19
Recognition Phase.....	19
Stimuli	20
Stimulus Generation.	22
Stimulus Selection.....	24
Apparatus.....	27
Procedure	27

RESULTS.....	36
Acquisition.....	36
Recognition	53
Recognition Confidence Rating Analysis.	57
Recognition Correctness Analysis.	70
Debriefing.....	75
DISCUSSION.....	76
Effect for Feedback Found in the Acquisition Phase	77
Effect for Transform Found in the Acquisition Phase	77
No Effect for Feedback Found in the Recognition Phase	77
Effect for Transform Found in the Recognition Phase.....	78
Debriefing Data	78
Conclusion.....	79
REFERENCES	82
APPENDICES.....	84
Appendix A. Signed Approval Forms.....	84
Appendix B. Stimulus Selection Procedure.....	87
Selection of Stimuli Set.....	87
LISP Program Source.....	88
Appendix C. Stimuli.....	99
Acquisition Items.....	99
Recognition Items	102
Appendix D. Debriefing Responses.....	104
100%/0% Feedback Condition.....	104
80%/20% Feedback Condition.....	105

60%/40% Feedback Condition.....	107
---------------------------------	-----

LIST OF TABLES

TABLE	PAGE
1. Summary of Significant Effects for the Acquisition Correct Analysis with Both Bases Analyzed Together	41
2. Summary of Significant Effects for Acquisition Correct Analysis for Stimuli Generated from Base 1.....	42
3. Summary of Significant Effects for Acquisition Correct Analysis for Stimuli Generated from Base 2.....	43
4. Chance Analysis for Base 1 Mean Acquisition Correct Data for Base 1.....	51
5. Chance Analysis for Base 2 Mean Acquisition Correct Data for Base 2.....	52
6. Summary of Significant Effects for Recognition Confidence Rating Data	59
7. Chance Analysis of Base 1 Recognition Correctness Data.....	73
8. Chance Analysis of Base 2 Recognition Correctness Data.....	74

LIST OF FIGURES

FIGURE	PAGE
1. The symbol set used to generate the acquisition and recognition stimuli.....	21
2. The bases used to generate all stimuli for the acquisition and recognition stimuli. Base 1 consists of straight line figures, and Base 2 consists of curved line figures.	23
3. An example of a transform sequence that demonstrates a complete continuum from Base 1 to Base 2. After four transforms from each base, the resulting stimuli are identical.	25
4. A schematic representation of the random selection process for the acquisition stimuli.....	26
5. An example of the introductory screen seen by the subjects.....	28
6. An example of the screen presenting instructions for the acquisition phase of the procedure.....	29
7. An example of the screen presenting an acquisition stimulus.	30
8. An example of the screen presenting acquisition feedback to the subjects.....	32
9. An example of the screen presenting instructions for the recognition phase to the subjects.....	33
10. An example of the screen presenting a recognition stimulus.	34
11. An example of the screen requesting a recognition confidence rating from the subjects.	35
12. An example of the screen presented for debriefing.	37

13.	An example of the final screen presented at the end of the experiment.	38
14.	The mean number of correct responses for stimuli generated from Base 1 as a function of feedback group.....	44
15.	The mean number of correct responses for stimuli generated from Base 2 as a function of feedback group.....	45
16.	The mean number of correct responses for stimuli generated from Base 1 as a function of transformational distance from Base 1.	46
17.	The mean number of correct responses for stimuli generated from Base 2 as a function of transformational distance from Base 2.	47
18.	The mean number of correct responses for stimuli generated from Base 1 as a function of both feedback group and transformational distance from Base 1.	49
19.	The mean number of correct responses for stimuli generated from Base 2 as a function of both feedback group and transformational distance from Base 2.	50
20.	The mean number of correct responses for stimuli generated from Base 1 as a function of both feedback group and acquisition order.....	54
21.	The mean number of correct responses for stimuli generated from Base 2 as a function of acquisition order.....	55
22.	The mean number of correct responses for stimuli generated from Base 2 as a function of both transformational distance from Base 2 and acquisition order.	56

23.	The mean recognition confidence rating for all stimuli as a function of the new/old factor.	60
24.	The mean recognition confidence rating for all stimuli as a function of presentation order.....	61
25.	The mean recognition confidence rating for all stimuli as a function of both repetition and new/old factors.	63
26.	The mean recognition confidence rating for all stimuli as a function of transformational distance from base and the new/old factor.	64
27.	The mean recognition confidence rating for stimuli generated from Base 1 as a function of transformational distance from Base 1.	65
28.	The mean recognition confidence rating for stimuli generated from Base 2 as a function of transformational distance from Base 2.	66
29.	The mean recognition confidence rating for stimuli generated from Base 2 as a function of both transformation distance from Base 2 and recognition order.....	68
30.	The mean recognition confidence rating for stimuli generated from Base 2 as a function of both transformational distance from Base 2 and acquisition order.	69
31.	The mean number of correct responses during the recognition phase for stimuli generated from Base 1 as a function of transformational distance from Base 1.	71
32.	The mean number of correct responses during the recognition phase for stimuli generated from Base 2 as a function of transformational distance from Base 2.	72

Resilience of Prototypes to Ambiguous Feedback

Scott Sarnikowski

San Jose State University

Running head: PROTOTYPES AND AMBIGUOUS FEEDBACK

Requests for reprints should be sent to Scott Sarnikowski, Department of
Psychology, San Jose State University, San Jose, California 95192.

Abstract

Along with death and taxes, there are two additional truisms in life: The world is ambiguous and a large amount of our learning involves prototypes.

Surprisingly, the effect of ambiguity on prototype formation has not been explored. The present study investigated this issue through a two-phase procedure. The subjects were 90 San Jose State University students. In the first phase (acquisition phase), the subjects were presented with stimuli (based on Franks & Bransford, 1971) which they sorted into two classes. Immediately following each presentation, the subject groups were given different levels of valid feedback on their judgment. In the second phase (recognition phase) the subjects stated whether they had seen the stimulus and stated their confidence. The key finding of the present study is that the formation of prototypes is independent of feedback. In addition, this study proposes a new memory store to address this feedback-independent behavior.

Resilience of Mental Prototypes to Ambiguous Feedback

Most everyday reasoning and decision making is based on uncertain premises—most of our actions are based on guesses, often requiring explicit weighing of conflicting evidence. (Shafer & Pearl, 1990, p. 1)

In the real world (as opposed to the mostly artificial world of the laboratory) we are surrounded by uncertainty and ambiguity. We interact with friends who are not consistent. We face problems that have no “right” answer. We encounter pressing decisions prepared with only a fraction of the necessary information. At times our environment tells us we are right, at times it tells us we are wrong, and at times it doesn’t tell us either way. In spite of this, we still must interact with the world. We don’t have the luxury to opt out if we don’t have all the information or if we don’t get clear messages as to whether we are right or wrong. What’s a reasonable simian to do?

The primary way our environment provides us with information is through the feedback we receive in response to our behavior. When we’re children, our parents tell us “no” when we do something wrong. When we’re in school, the teacher tells us when we miss a problem. When we’re in high school, our friends tell us when we dress funny. When we’re on the job, our bosses tell us when we’re doing a good job (well, most of the time). In all of these examples, the quality of the feedback is never 100% accurate, certain, or timely; but nonetheless we have to deal with it as it is. An important question arises from the uncertainty of daily life: How does the quality of environmental feedback, or lack of it, affect what we take away from each encounter? This question is the focus of my thesis.

To investigate this question I have drawn from two fields of research within cognitive psychology: prototype learning and probabilistic learning. In the remainder of this overview, I will give a brief description of the history of

each field and discuss the relevant theories. Afterward, I will present an integration of the two fields as well as introduce two competing models for the effects of ambiguous feedback. Finally, I will outline the strategy and justification for the experimental procedure used in this thesis.

The Field of Prototype Learning

Prototype or schema learning is pervasive throughout human development. Prototype theory proposes that concepts are stored in long-term memory as a propositional representation, rather than as a pictorial representation. A propositional representation can take the form of assertions, such as "Has four legs" or "Barks," or any equivalent assertion of class membership. In addition, the propositions can take the form of component relationships or relative spatial arrangement. Although there are many theories about the exact organization of the propositional representation held in memory (Anderson, 1976; Anderson & Bower, 1973; Collins & Quillian, 1969; and Norman & Rumelhart, 1975), all of these models share the same fundamental belief in a structured set of propositions, frequently referred to as *prototypes*. The most important characteristic of prototypes is that a prototype may not exactly match any previously presented or encountered object. Our understanding for classes of objects such as dogs or furniture is based on a prototype that may not be exactly like any particular instance that we have ever seen.

Prototype Learning Research and Theories. Several researchers have demonstrated that schemas or prototypes provide a framework for the organization of long-term memory and form the basis of at least some of our reasoning (Bobrow & Winograd, 1977; Brewer & Treyens, 1981). Several prototype learning studies have investigated well-established schemata such as

the contents of a professor's office (Brewer & Treyens, 1981) or the sequencing of events in activities such as dining out (Shank & Abelson, 1977). Other researchers, such as Bartlett (1932), have investigated the characteristics of schema and schematic learning that affect the recall of ambiguous information.

The studies of schema for professor's offices and dining out depend on the prototypes and structures that the subject brings to the experiment.

Presumably, the subject developed the prototypes through experience with specific instances of professor's offices and dining out.

The Field of Probability Learning

In 1763, Reverend Thomas Bayes proposed a theory to determine the optimal decision when faced with uncertain information. His theory was based on the laws of probability and produced a spectrum of decisions based on their maximized utility. Although not originally proposed as a theory of human judgment, it was accepted as the standard by which human judgment should be made. This position remained the status quo until the early 1960s, partially because the computing resources needed to evaluate the conditional probabilities of the theory were not available to test its assertions efficiently.

Work in the field pursued in the 1960s was started by the research of Brunswik (1956, cited in Slovic & Lichtenstein, 1971), Hammond (1955, cited in Slovic & Lichtenstein, 1971), and Estes (1959, cited in Slovic & Lichtenstein, 1971). During the 1960s, research into learning with probabilistic information focused on both Bayesian and regression approaches. The work on Bayesian theory was based on the ideas and theories of Thomas Bayes, while the work on regression approaches used regression techniques for probabilistic judgment. As a model of human judgment, both the Bayesian and regression approaches assert that we

are all intuitive statisticians that systematically apply mathematical principles in all our judgments.

As evidence began to mount that human beings aren't very good innate statisticians, two new theories were proposed in the early 1970s. One of these new approaches was proposed by Kahneman and Tversky (1973), who postulated that we are not intuitive statisticians at all; instead, we use very simple heuristics when making judgments with probabilistic or uncertain information. Their approach was a direct consequence of the empirical data that showed neither regression nor Bayesian theories could predict performance in learning tasks with probabilistic information.

Another alternative to regression and Bayesian theories was proposed by Brehmer in 1974. Brehmer proposed a hypothesis sampling theory in which judgment is based on a hierarchy of hypotheses. Both the hypothesis sampling and heuristics theories of probabilistic learning continue to dominate the field today.

Probability Learning Research and Theories. There are four major theories in the field of probability learning: regression, Bayesian, hypothesis sampling, and heuristics. All four address the nature of probabilistic information in very distinct ways. To illustrate and examine the claims of each theory, I will apply all four theories to the same probability judgment task, auto insurance underwriting.

You are an underwriter and it is your job to assess the accident risk for potential policyholders in order to set their insurance rates. Although you know that accidents are random events, you are given several pieces of information about a potential customer and asked to find a way to predict risk. The

information includes: driving record, claims history, age, sex, make of car, profession, home address, miles driven per week, marital and family status, and potentially other information as well. It is your job to take all of these variables into account and make a judgment about the person's insurability.

Regression Theory. Regression theory proposes that judgment with probabilistic information is based on correlational statistics. The basic concept is that subjects weigh the available variables and then combine them in some way to develop their judgments. Egon Brunswik proposed *probabilistic functionalism* (Brunswik, 1952, 1956, cited in Slovic & Lichtenstein, 1971) to describe this process. Within probabilistic functionalism, Brunswik developed a "lens model" to represent the probabilistic interactions between the subject and the environment that lead to the judgments.

More specifically, the main constructs of the model are: cues (information sources that define each piece of information); quantified cue dimension; cue dimension relevance to the real world; and the criterion (the actual state of the judgment). The correlation between the available information, the cue, and the true state of the world establishes the ecological validity of the cue. To develop a judgment, the subject linearly sums all the cues. This process leads to a prediction that is an optimal deviation between the judgment and the real world. Within the model, the subject's judgment can be adjusted based on the utility of the cue.

Regression theory would describe the underwriter's task as an exercise in correlational statistics. In arriving at a final determination of accident risk, the underwriter would take all of the available information and scale the importance of each according to the variable's ecological validity. For example, perhaps the

actuarial figures available indicate that 60% of all 25-year-olds have an accident every two years. This information, along with similar data, would be linearly summed, resulting in an index of insurability to be used in setting the insurance rate for the customer.

One of the problems with Brunswik's lens model is its difficulty with intercorrelation between variables. So in our example, if actuarial figures indicate that 21-year-old drivers with big cars are at greater risk than 21-year-old drivers with small cars, but 25-year-old drivers with small cars are a greater risk than 25-year-old drivers with big cars, this interaction between age and car size would cause problems for the model. To address this issue, Slovic and Rorer (1968, cited in Slovic & Lichtenstein, 1971) proposed the use of a multiple regression approach to address intercorrelation and non-linear relationships between cues and criterion.

Bayesian Theory. Bayesian theory, like regression theory, is based on quantitative statistical analysis. The difference between the two is that regression theory is based on correlational statistics, whereas Bayesian theory is based on conditional probabilities. Using the same terms from Brunswik's lens model, Bayesian theory describes probabilistic judgment by the conditional probabilities between cue and criterion. The Bayesian theory is best described as "a highly idealized theory of the behavior of a 'rational' person with respect to decisions" (Savage, 1954, cited in Slovic & Lichtenstein, 1971, p. 665).

Von Neumann and Morganstern (1947, cited in Slovic & Lichtenstein, 1971) started the modern interest in using Bayesian theory in human judgment. The Bayesian theory holds that judgment is expressed in terms of subjective or personal probabilities. Of particular interest is the process by which these

judgments are changed. The optimal strategy to change judgments is based on the laws of probability postulated by Bayes in 1763.

Applying Bayesian theory, the underwriter would compute the conditional probabilities between each variable and the likelihood of an accident. Then, the resulting conditional probabilities would be multiplied, yielding the final probability profile for the individual. The final profile would consist of a distribution of probabilities tied to a set of possible driving and life situations. Finally, this probability distribution would be combined with the associated cost of accidents and price of insurance to arrive at the final determination of the individual's premium.

Investigations with Bayesian theory have centered on failures in judgment that do not adhere to the strict predictions of the theory. The most commonly investigated phenomenon is *conservatism*. Conservatism describes judgment that is more cautious than warranted by the available conditional probabilities. Slovic and Liechtenstein (1971) have proposed three explanations for conservatism: misperception, misaggregation, and artifact hypothesis. Using the example of insurance underwriting, misperception could lead to conservatism because the underwriter might possibly have mis-read the conditional probabilities on the actuarial tables. With the wrong information, the judgment is made more conservative than is really necessary. Misaggregation would lead to conservatism because the underwriter may have failed to integrate the data in a way that could be evaluated. Finally, artifact hypothesis could lead to conservatism because people (the underwriter) can only deal with probabilities within a very narrow range. Probabilities greater than 10:1 or less than 1:10 are typically not handled correctly (Edwards, 1969, cited in Slovic & Lichtenstein, 1971).

Hypothesis Sampling. Hypothesis sampling represents a departure from the statistical models of both regression and Bayesian theory. In 1974, Brehmer proposed a two-stage model for learning with probabilistic information based on the theories of functional learning, in which subjects store pairs of cue and criterion values (Bjorkman, 1965b, cited in Brehmer, 1974, Carroll, 1963). Brehmer felt that functional learning theories were insufficient to describe performance when probabilistic information was available. Therefore, Brehmer believed that these theories had to be modified to account for tasks in which the relationship between cue and criterion is probabilistic. This type of task is referred to as a Cue Probability Learning task (CPL).

Brehmer's two-stage model proposes that the subject, in the first (or detection) stage, must discover the rule governing the relation between the cue and criterion. In the second (or testing) stage, after the rule has been discovered, the subject tests the rule and must learn how to use it. Within the detection stage, Brehmer defines two substages: hypothesis sampling and hypothesis construction. Hypothesis sampling proposes a selection from a set of preestablished hypotheses and is considered data independent. Data dependency differentiates hypothesis sampling from hypothesis construction because sampling does not require data, whereas construction must incorporate available data. Brehmer chose to concentrate only on the sampling alternative.

Hypothesis sampling assigns sampling probabilities to the different hypotheses. The basic assumption of the sampling substage is that a subject "has available [a] set of hypothesis about [the] possible functional relationships [between a cue and criterion] and that he samples from this set until he has found a satisfactory alternative or until he has exhausted the set" (Brehmer, 1974, p. 5).

The preestablished hypotheses are organized in a hierarchy, which, in turn, is organized around the sampling probability of each individual hypothesis. A strong hypothesis has a high sampling probability and is at the top of the hierarchy. As you travel down the hierarchy, the hypothesis strength becomes weaker. When a subject is presented with a set of cues, the strongest hypothesis in the hierarchy is chosen. If that hypothesis is found to be incorrect during the hypothesis testing stage, the hypothesis is abandoned and the next strongest hypothesis is chosen.

In the testing stage, the subject tests the selected hypothesis. However, in a CPL task in which the subject can not be 100% certain that the hypothesis is incorrect, the idea of testing runs into problems. How does the subject reject a bad hypothesis or retain a good hypothesis? Brehmer acknowledges that the nature of the CPL task leads to ambiguity in rejecting hypotheses.

For insurance underwriters utilizing the hypothesis sampling approach, the most important aspect of their judgment is the hypothesis hierarchy they have developed in dealing with similar situations. In an iterative process, the underwriter would first attempt to apply a simple linear relationship between the variables and the insurability of the client. If this relationship turned out to be wrong, the underwriter would discard this relationship and try a more complex approach. In general, underwriters would continue using a relationship until they were satisfied with their ability to predict a client's insurability. Unfortunately, an underwriter can never be 100% certain that a hypothesis is correctly rejected because the relationship between the insured's accident performance and the judgment is probabilistic. Under the theory there is no quantitative criterion for accepting a hypothesis, as there is in both the Bayesian

and regression theories.

Heuristics Approach. Like the hypothesis sampling theory, the heuristics approach does not depend on statistical analysis. Instead, Tversky and Kahneman (1974) proposed that we use simple heuristics in making judgments under uncertain conditions. The four heuristics they proposed were: representativeness, availability, adjustment, and anchoring.

Using the representativeness heuristic, a subject makes a judgment based on how representative a cue is of a class of criterion. This heuristic is used most frequently in the case in which we are asked to determine the probability or degree to which one item resembles another. For the underwriter, the representativeness heuristic would predict that it is the underwriter's notion of a representative accident victim that is most important, rather than the intrinsic probabilistic information. For example, the underwriter may believe that yellow cars are more representative of cars in accidents than the statistics of highway accidents would suggest. In this case, the underwriter may erroneously set a higher premium than would be appropriate for yellow cars.

The availability heuristic leads a subject to make a judgment based on the availability of information. The probability of an event is judged on the ease of retrieving examples of the class. The effectiveness of this heuristic is negatively impacted by factors such as familiarity and the ability to imagine. In the case of the underwriting example, perhaps the week before a potential customer is to be evaluated, the underwriter witnesses a car accident involving the same make of car as the one belonging to the customer under evaluation. Using the availability heuristic, the underwriter may judge the driver of a similar car as more at risk for an accident.

In the adjustment heuristic, we begin with an initial estimate and then adjust to give the final answer. With this heuristic, we do not look at the probability information of the cues but merely adjust to the availability of new information. A related phenomenon, anchoring, describes the process of determining our initial estimate. The adjustment heuristic would predict that the final determination of accident risk is very sensitive to the initial classification of the customer's insurability. Items such as non-smoking or safety devices in the car act to reduce the total cost, but only marginally. Two individuals with very similar profiles may end up with widely different rates because of the marginal adjustments.

Integrating Prototype and Probabilistic Learning

The fields of prototype and probabilistic learning provide two complementary approaches to investigate learning in the real world. Each provides a different focus that contributes to a deeper understanding of learning with probabilistic information. The field of prototype learning has focused on the underlying structure of the concepts. For example, Collins and Quillian's model explicitly describes the organization of memory structures. But the investigations into Collins and Quillian's model depend on structures developed prior to the procedure. This is also true of the Shank and Abelson study into the schema of a professor's office.

The field of probability learning has focused on judgment behavior and performance. In Brehmer's studies (1974), the subject's behavior was analyzed to find the strategies in dealing with probabilistic information. Kahneman and Tversky's work also focused on judgment behavior. In these studies the focus was the judgment behavior rather than the underlying structure of the concepts.

Integrating the two fields and combining their areas of focus leads to an interesting approach to real-world learning: using probabilistic manipulations to investigate the formation of and nature of prototypes. The question I investigated in this thesis was: Does ambiguous feedback affect the nature of prototypes? There are two possible models.

In the first model, which I call the *feedback-dependent model*, the development of a prototype structure is influenced mostly by environmental feedback. The only instances of a class that are incorporated into the prototype for the class are those that are explicitly reinforced by the environment. Alternatively, instances that are not reinforced have no effect on the development of the prototype. This model is similar to Thorndike's law of effect, in which correct responses are strengthened by reinforcement, and incorrect responses are weakened by nonreinforcement (Gleitman, 1991). Therefore, extending the law of effect to prototypes, the feedback manipulation should be instrumental in the formation of a prototype. If the feedback-dependent theory holds, the prototype developed should reflect the items that are reinforced.

The second model, or the *feedback-independent model*, proposes that some aspect of prototype development is independent of feedback. Unlike the feedback-dependent model, the second model proposes that subjects attend to and incorporate instances that are not explicitly reinforced as part of the class the prototype represents. The ramification of this model is that subjects could be asked to perform a classification task that divided the stimuli in a way unrelated to the prototype developed in observing those stimuli. This model is similar to a model of perceptual learning proposed by Gibson (1969). Gibson proposed that "Perceptual learning is self regulating, in the sense that modification occurs

without the necessity of external reinforcement” (p. 4). Extending this model to prototype learning, the model would predict that, although individuals receiving various levels of ambiguous feedback perform differently on the same learning task, they would all show similar performance in a recall task.

Research Strategy

In order to differentiate between the feedback-dependent and the feedback-independent models for prototype development, it is necessary to experimentally create a prototype under ambiguous conditions and then examine the structure of the prototype. The goal of the procedure in this thesis is to demonstrate that subjects are attentive to ambiguous feedback and explicitly examine the nature of the prototype formed. A good procedure would be a CPL task, combined with a prototype recall task. The studies of CPL tasks have demonstrated that performance is affected by feedback, while many studies in prototype learning have been successful in examining the structure of prototypes through recall.

A good candidate for this kind of hybrid study is a procedure by Franks and Bransford (1971). This procedure is ideal for two reasons. First, the procedure does not depend on any previously developed structure that a subject would bring to the experiment; the subjects form the prototype during the procedure. The conditions under which the prototype is formed can be manipulated to be a CPL task, allowing the response behavior to be evaluated as in probabilistic learning studies.

Second, the procedure explicitly examines the nature of the prototype. This is crucial to determine the effectiveness of the feedback manipulation. The way in which the Franks and Bransford procedure accomplishes this is described

in the following subsection.

Description of the Franks and Bransford Procedure. Franks and Bransford supported the idea that visual recognition of a certain class of visual stimuli relied on the development of a prototype. In addition to the prototype, which was seen as representing the class of all presented stimuli, each new member of the class was encoded based on the number of discrete transformations away from the prototype. This model accurately characterized classification performance for previously presented members.

Franks and Bransford divided their procedure into two phases, *acquisition* and *recognition*. During the acquisition phase, the subjects were shown several stimuli and were given the task of manually reproducing the stimuli. Franks and Bransford believed that this task facilitated the development of the prototype as a concept. They designed the replication exercise as an incidental learning task in which the concept for the prototype (referred to as a *base*) was learned even though the subjects did not know they needed to learn it. Following the acquisition phase, the subjects were presented a set of stimuli that they had not seen before. After each presentation, the subjects were asked two questions: Had they seen the stimulus? How confident were they in their response?

The most important part of the procedure was the construction of the stimuli presented to the subjects. The stimuli were related to the base by a set of predefined *transformations*. These transformations were simple spatial permutations of the elements in each stimulus, applied one after another to the base to develop a set of stimuli. The number of transformations applied was called the *transformational distance*. Therefore, the higher the number of transformations from the base, the more dissimilar the stimulus was from the

base.

The crucial result of the Franks and Bransford study was the establishment of a negative correlation between recognition confidence rating and transformational distance from the base. For example, stimuli patterns one transformation from the prototype were recognized with greater confidence than stimuli patterns two transformations from the prototype, and so on. Most interestingly, subjects showed the highest confidence rating for the prototype even though it was never actually presented during the acquisition phase. Franks and Bransford saw this finding as supporting the notion that the prototype was developed during the acquisition phase and used in the recognition phase.

General Predictions. Using the same structure as the Franks and Bransford procedure, the present experiment contained two phases: acquisition and recognition. The acquisition phase consisted of a CPL task that attempted to establish a response pattern to demonstrate that the subjects were attentive to the ambiguity of the feedback. There were three feedback groups: 100% valid, 80% valid, and 60% valid. In the 100% valid group, the subjects were always told correctly whether they were right or wrong. In the 80% valid group, the subjects only received the correct feedback 80% of the time. In the 20% of the time that they were given invalid feedback, the subjects were told they were wrong when they were right and vice versa. The same applies to the 60% valid group, except that they were given invalid feedback 40% of the time. The recognition phase was identical to the Franks and Bransford procedure and attempted to demonstrate the same phenomenon as in their procedure.

In order to evaluate the two competing models (feedback-dependent and

feedback-independent), the data must reveal three essential findings. First, the feedback manipulation in the acquisition phase must be shown to affect the performance of the subjects. This finding establishes that the subjects were using the feedback in the judgment process. Second, in the case of the feedback-independent model, the recognition data must show that the subject's confidence is unaffected by the feedback manipulation. This finding would indicate that the prototype developed during the acquisition phase did not vary across the feedback groups. Third, since the evaluation of the feedback-independent model depends on accepting the null hypothesis for feedback during the recognition phase, it is essential that the recognition data demonstrate an effect for transform. A finding that there is a transform effect would demonstrate the type of behavior established in the Franks and Bransford study, indicating that the subjects used an underlying prototype for the classes.

Method

Subjects

The subjects were student volunteers from introductory psychology courses receiving credit for participation. There were a total of 90 subjects, 45 male and 45 female. During the acquisition phase, the 90 subjects were evenly distributed across three feedback groups. Within each feedback group, the subjects were evenly divided among three random presentation orders. During the recognition phase, the 30 subjects from each acquisition group were evenly divided between two random presentation orders.

Design

The experiment was divided into acquisition and recognition phases. The acquisition phase was designed to be a CPL task in which the subjects classified stimuli into one of two groups.

Acquisition Phase. There were five independent variables: feedback, stimulus transform, base, repetition, and acquisition order. There were three feedback levels: 100%/0% valid, 80%/20% valid, and 60%/40% valid. Stimulus transform had four levels that described the number of transforms from the base. The base factor had two levels corresponding to the two bases used to generate the stimuli set. There were 10 presentations within each base and transform leading to 10 levels for the repetition factor. Finally, there were three different acquisition orders.

Feedback and acquisition order were between-subject independent variables, while stimulus transform, base, and repetition were within-subject variables. The theoretically important variables of the acquisition phase were stimulus transform and feedback. The acquisition order independent variable was included as a control to monitor for presentation ordering effects.

The dependent variables collected were response time and response correctness. As in the Franks and Bransford procedure, the generation of the stimuli was crucial to the manipulation. The procedure for generating and selecting the stimuli for this procedure is presented in detail in the Stimuli subsection.

Recognition Phase. The recognition phase was similar to the Franks and Bransford procedure. During this phase, the subjects were asked to rate their confidence in recognition of items similar to those they saw in the acquisition

phase.

The independent variables for the recognition phase were: feedback, acquisition order, recognition order, base, stimulus transform, new/old, and repetition. Feedback (three levels), acquisition order (two levels), and base (two levels) factors are the same as in the acquisition phase. There were two random presentation orders leading to two levels of recognition order. The base factor had two levels corresponding to the two bases used to generate the stimuli set for the phase. Stimulus transform was like the stimulus transform in the acquisition phase except that there were five levels because the base itself was included during recognition. The five levels included the four possible transform levels and the base. The new/old factor had two levels, one level to describe new stimuli and one to describe stimuli seen during acquisition. Finally, there were two levels of repetition, corresponding to the number of presentations within base and stimulus transform.

Feedback, acquisition order, and recognition order were between-subject independent variables, while base, stimulus transform, new/old, and repetition were within-subject independent variables. As in the acquisition phase, the most theoretically important variables were feedback and stimulus transform. The acquisition and recognition order independent variables were included to monitor for presentation ordering effects.

Stimuli

The stimulus set for this experiment is similar to the Franks and Bransford study (a complete discussion of the stimuli generation and selection procedure is given in Appendix B). The elements used to construct each stimulus were chosen to be very simple and easily discernible. Figure 1 shows the set of all elements

Figure 1. The symbol set used to generate the acquisition and recognition stimuli.

used in the procedure. The symbols were chosen so that they were easily differentiated into two groups, figures composed entirely of straight lines and figures containing predominately curved lines.

Stimulus Generation. A base consisted of four of the five symbols chosen from each group, arranged in two rows of two (see Figure 2). The fifth symbol was used as a substitution for the manipulations to be described later. The bases were further defined to represent the central tendency for all members of a category or class. The importance of two bases will be explained later.

This experiment employed the same transformations as defined in the Franks and Bransford study. The four types of transforms were: Major Constituent Permutation (MaP), Minor Constituent Permutation (MiP), Deletion (De), and Substitution (Su). The MaP permutation exchanged groupings of two symbols, creating two valid permutations: MaP1, exchange top row for bottom; and MaP2, exchange left column for right column. The MiP permutation exchanged a single element for any other, for which there were six valid transformations: MiP1 through MiP6. The De permutation deleted a single element. Since there were four elements, there were four valid deletions: De1 through De4. Finally, the Su permutation substituted a single element in any of the four locations. There were six possible symbols for substitution; the one symbol unused in making the base from the same group plus five members of the other group. Any of these six could be substituted in each of four locations, so there were 24 valid substitutions, Su1 through Su24.

The stimuli presented during the procedure were created by applying the transformations to the bases for up to four sequentially applied transformations. The resulting large number of stimuli were then analyzed and the total was

Figure 2. The bases used to generate all stimuli for the acquisition and recognition stimuli. Base 1 consists of straight line figures, and Base 2 consists of curved line figures.

narrowed by a process discussed in the Stimuli Selection subsection.

Stimulus Selection. The goal of the stimulus selection process was to create a set of stimuli that formed a continuum. This was done by selecting sets of *transform sequences*, sequentially applied transformations that when applied to each base would lead to the same stimulus after four transformations. Figure 3 shows a sample continuum, where the transform sequence (MaP2:MiP5:Su11:Su20) from Base 1 and (MaP1:MiP3:Su14:Su9) from Base 2 achieve the same stimulus. There are several more transform sequences that achieve the same type of continuum. A more detailed discussion of the selection process is presented in Appendix B.

The most important characteristic of the continua, for this procedure, is that they present a range of stimuli that is bounded by the bases at each end. As you follow a continuum from a base, each additional transform makes each stimulus more dissimilar from the base. After four transformations, the resulting stimulus is ambiguous, belonging to neither category exclusively.

A total of 70 stimuli were selected to be presented during the acquisition phase of the procedure. The 70 stimuli were selected by randomly picking 20 of the 183 valid transform sequences. The 20 transform sequences were then randomly sampled so that each transform level contributed 10 items to the total of 70 (see Figure 4). The stimuli were sampled in this manner to provide additional stimuli for the recognition phase.

A total of 36 stimuli were selected to be presented during the recognition phase of the procedure. The 36 stimuli consisted of 4 instances of Base 1, 4 instances of Base 2, 14 stimuli randomly selected from the stimuli presented in the acquisition phase, and 14 novel stimuli, randomly sampled from those stimuli

Figure 3. An example of a transform sequence that demonstrates a complete continuum from Base 1 to Base 2. After four transforms from each base, the resulting stimuli are identical.

Figure 4. A schematic representation of the random selection process for the acquisition stimuli.

not chosen for the acquisition phase.

Apparatus

The procedure was implemented as a HyperCard stack running under HyperCard 2.0 on Apple Macintosh computers. The computer provided written instructions for the tasks, insuring that all subjects received the same information. The subjects used a mouse and a keyboard to participate in the experiment. The mouse enabled the subjects to select responses during the acquisition and recognition phases, while the keyboard let the subjects write their debriefing responses. The HyperCard stack was designed to collect the subjects' responses, measure their response times, and create data files to store these data.

Procedure

The subjects were lead into the room and seated in front of a Macintosh computer. The seating was arranged to insure that the subjects were not distracted during the experiment. The subjects were instructed to begin the experiment when they were seated and to proceed at their own pace. They were further instructed to ask for help if they had a problem with the computer. Figure 5 shows the instructions displayed on the computer screen when the subjects first sat down.

After clicking on "Continue," the stack displayed the screen that contained the instructions for the acquisition phase, identified to the subjects as Part 1. Figure 6 shows the screen with the instructions.

After clicking on "Continue," the subjects were presented with the first item of the acquisition phase. Figure 7 presents a sample stimulus displayed on the computer screen. After viewing the stimulus, the subjects classified the

Figure 5. An example of the introductory screen seen by the subjects.

Figure 6. An example of the screen presenting instructions for the acquisition phase of the procedure.

Figure 7. An example of the screen presenting an acquisition stimulus.

stimulus by clicking on "Category #1" or "Category #2." "Category #1" corresponded to Base 1 and "Category #2" corresponded to Base 2. In response to the subject's judgment, the stack presented the feedback as in Figure 8. The screen position of the feedback was chosen in order not to obscure the stimulus, thus allowing the subjects to view the feedback and the stimulus at the same time. There was no time limit for observing the stimulus with the feedback present. When the subjects were ready to proceed they clicked on "OK."

At this point, the stack recorded the subject's response time and response correctness and stored them in a data file. Response time was a measure of the time between the presentation of the stimulus and the selection of the category. Response correctness was based on the subject's actual classification of the stimulus, and in the case of the invalid groups, did not reflect the feedback given to the subjects. The stack then presented the next item to the subject. This same procedure continued until the subjects saw all 70 items. I designed the stack to randomize the presentation order and provide the proper feedback according to the group assignment of the subject.

After all of the stimuli of the acquisition phase were presented, the subjects moved on to the instruction for the recognition phase, identified as "Part II" (see Figure 9). After the subjects read the instructions and clicked on "Continue," they were presented the first of the 36 stimuli of the recognition phase. Figure 10 shows an example of the stimuli for the recognition phase. The subjects clicked on either "Yes" or "No" in response to the question "Have you seen this item?". After making a selection, the subject was asked to provide a confidence rating. Figure 11 shows an example of the confidence rating. The prompt for the confidence rating consisted of a 5-point selection, ranging from

Figure 8. An example of the screen presenting acquisition feedback to the subjects.

Part 2

In this part, your task will be to look at another set of symbols and determine whether you saw exactly that set during Part 1. Exactly means that the same symbols are in the same positions as in Part 1. Answer by clicking **Yes** or **No**.

After clicking your response, you will then be asked to rate your confidence in your judgement. You will be presented with a set of 5 buttons (click here for an example click OK when done). When you rate your confidence, 1 means no confidence at all, 5 means very confident, and 3 means somewhat confident. After you have selected your rating, click OK. You will then be shown the next group.

As in Part 1, I am also interested in the time it takes for you to make your judgement. ***Please concentrate on being correct rather than being fast.*** When you're ready, click Continue to start.

Continue

Figure 9. An example of the screen presenting instructions for the recognition phase to the subjects.

Figure 10. An example of the screen presenting a recognition stimulus.

The screen displays four symbols arranged in a 2x2 grid. The top-left symbol is a solid black square. The top-right symbol is a solid black club. The bottom-left symbol is a solid black five-pointed star. The bottom-right symbol is a solid black diamond. Below the symbols is a rectangular box containing a confidence rating scale. The scale consists of five numbered circles (1, 2, 3, 4, 5) arranged horizontally. Below the numbers are the labels 'Not at all' under '1', 'Somewhat' under '3', and 'Very' under '5'.

How confident are you in your choice?

1 2 3 4 5

Not at all Somewhat Very

Figure 11. An example of the screen requesting a recognition confidence rating from the subjects.

not at all confident (1), to *very confident* (5). The rating prompt was placed so that the recognition stimulus and the confidence rating could be seen at the same time.

At this point, as in the acquisition phase, the stack recorded the subject's response time, response correctness, and confidence rating and stored them in a data file. The response time measured the time between the presentation of the stimulus and the "yes/no" judgment. The response correctness was based on the subject's yes/no judgment. The stack then presented the next item to the subject. This same procedure continued until all 36 items were presented. I designed the stack to randomize the presentation order.

After all the recognition items were presented, the subjects moved on to the debriefing screen (see Figure 12) where they were asked to rate how difficult they found the task on a 5-point scale from *easy* to *difficult*. In addition, the subjects were also asked the following question: "...write 1 or 2 sentences on how you could tell the difference between the two categories in Part 1." The subjects were instructed to type in their responses and were allowed as much time as necessary to provide a complete answer. Once the subjects had completed their responses, they clicked on "Continue" and were then shown a final screen (see Figure 13).

After the experiment was complete, the subjects were allowed to ask questions about the procedure and were given the opportunity to request a further debriefing by mail.

Results

Acquisition

Response time and response correctness were collected during the

Figure 12. An example of the screen presented for debriefing.

Figure 13. An example of the final screen presented at the end of the experiment.

acquisition phase. The distribution of response time was examined and found to vary greatly across subjects. In one example, the response time for the fifth presentation of a first order transform in Base 2 varied across subjects from .22 seconds to 52 seconds. This was typical of the ranges of the subject's response time and seemed representative of the subject's computer skills. This conclusion was further supported by my observations of the subject's interaction with the computer. Therefore, further analysis of the response time data was abandoned.

Prior to analyzing the response correctness data, three predictions were proposed. First, analysis of the response correctness data was predicted to demonstrate a main effect for the feedback between-subject factor. More specifically, a significant linear trend was predicted in which the mean correct score for the 100%/0% feedback group would be greater than that of the 80%/20% group, which in turn would be greater than that of the 60%/40% group. Second, the data were expected to show a main effect for the transform within-subject factor. A significant linear trend was predicted for the transform factor in which the mean scores for the first order transform were expected to be higher than the second order transform pattern, which were expected to be higher than the third order transform. Finally, the data were expected to show a significant Transform x Feedback interaction. It was further predicted that the slope for the mean correct versus transform of the 100%/0% feedback group would be greater than the slope of the 80%/20% group, which would be greater than the slope of the 60%/40% group.

To analyze response correctness, the number of correct responses was tallied across the 10 repetitions within each transform and base, and analyzed

with a four-way MANOVA with two between-subject factors (feedback [three levels] and order [three levels]) and two within-subject factors (base [two levels] and transform [three levels]). Fourth order transforms were not analyzed because the configuration of symbols is identical for both bases, leaving no way to assign response correctness.

Table 1 shows a summary of all the significant main effects and interactions for the overall MANOVA. The existence of the base interaction in Acquisition Order (Acq. Order) $\times$ Base and Acq. Order $\times$ Base $\times$ Transform indicated that a separate analysis of each base was necessary.

Separate three-way MANOVAs were performed for each base. The independent variables for the MANOVAs were two between-subject factors (feedback [three levels] and order [three levels]) and one within-subject factor (transform [three levels]). Tables 2 and 3 summarize the two analyses.

The first prediction stated that there should be a main effect for feedback. This prediction is supported by the significant main effects for feedback found in the analyses of Base 1 and Base 2. Figures 14 and 15 show the mean number of correct responses at each level of feedback for each base. Visual inspection of the data supports the prediction. In addition, the existence of significant linear trends for Base 1 ($F(1,81) = 14.62, p < .001$) and Base 2 ($F(1,81) = 16.38, p < .001$) also support the prediction that the performance of the 100%/0% feedback group is better than that of the 80%/20% group, which in turn is better than that of the 60%/40% feedback group.

The second prediction stated that there should be a significant main effect for transform. The analyses of Base 1 and Base 2 support the prediction by showing significant main effects for transform. Figures 16 and 17 show the

Table 1

Summary of Significant Effects for the Acquisition Correct Analysis with Both Bases Analyzed Together

Effect Type	Effect	df	F	p
Main Effect	AOrder	(2,81)	4.43	.015
	Feedback	(2,81)	13.98	<.001
	Transform	(2,162)	13.26	<.001
Two-Way Interaction	Acq. Order x Base	(2,81)	5.12	.008
	Feedback x Transform	(4,162)	4.99	.001
Three-Way Interaction	Acq. Order x Base x Transform	(4,162)	2.60	.038

Table 2

Summary of Significant Effects for Acquisition Correct Analysis for Stimuli Generated from Base 1

Effect Type	Effect	<i>df</i>	<i>F</i>	<i>p</i>
Main Effect	Feedback	(2,81)	7.40	.001
	Transform	(2,162)	5.81	.004
Two-Way Interaction	Acq. Order x Feedback	(4,81)	3.29	.015
	Feedback x Transform	(4,162)	3.79	.006

Table 3

Summary of Significant Effects for Acquisition Correct Analysis for Stimuli Generated from Base 2

Effect Type	Effect	<i>df</i>	<i>F</i>	<i>p</i>
Main Effect	Acq. Order	(2,81)	10.08	<.001
	Feedback	(2,81)	8.40	<.001
	Transform	(2,162)	8.59	<.001
Two-Way Interaction	Acq. Order x Transform	(4,162)	2.82	.027
	Feedback x Transform	(4,162)	2.31	.060

Figure 14. The mean number of correct responses for stimuli generated from Base 1 as a function of feedback group.

Figure 15. The mean number of correct responses for stimuli generated from Base 2 as a function of feedback group.

Figure 16. The mean number of correct responses for stimuli generated from Base 1 as a function of transformational distance from Base 1.

Figure 17. The mean number of correct responses for stimuli generated from Base 2 as a function of transformational distance from Base 2.

mean number of correct responses at each of the three transform levels. Visual inspection of the data supports this prediction, with the exception that Base 1 did not show a significant difference between Transform 1 and Transform 2 ($F(1,81) = 6.89, p = 0.82$). The existence of significant linear trends for Base 1 ($F(1,81) = 7.95, p < .01$) and Base 2 ($F(1,81) = 15.43, p < .001$) also supports the prediction that the response correctness of first order transforms should be higher than that of second order transforms, and in turn, second order transforms should be higher than that of third order transforms.

The third prediction stated that there should be a significant Feedback x Transform interaction. The analysis did find a significant Feedback x Transform interaction for Base 1 ($F(4,162) = 3.79, p < .01$) and a moderately significant feedback interaction for Base 2 ($F(4,162) = 2.31, p = .06$). The data for the analysis are shown graphically in Figures 18 and 19. The main characteristic of the data shows that the 100%/0% group performed better than either the 80%/20% or the 60%/40% groups at all transform levels except Transform 3, where the mean correct for each group converge. A simple effects analysis for each of the three transforms confirms this observation. For Base 1, the differences among the three feedback groups are significant at Transform 1 ($F(2,81) = 10.28, p < .001$) and Transform 2 ($F(2,81) = 6.48, p = .002$). For Base 2, the difference among the three feedback groups are significant at Transform 1 ($F(2,81) = 9.13, p < .001$) and Transform 2 ($F(2,81) = 3.67, p = .002$).

As a further analysis, each data point in Figures 18 and 19 was tested against the chance level of responding (number correct = 5) with a paired *t* test. The two-tail probabilities are shown in Tables 4 and 5. The 100%/0% and 80%/20% groups for both bases show significant mean correct scores for

Figure 18. The mean number of correct responses for stimuli generated from Base 1 as a function of both feedback group and transformational distance from Base 1.

Figure 19. The mean number of correct responses for stimuli generated from Base 2 as a function of both feedback group and transformational distance from Base 2.

Table 4

Chance Analysis for Base 1 Mean Acquisition Correct Data for Base 1

Feedback	Transform	Mean	SD	df	t	p
100% / 0%	1	6.93	1.66	29	6.38	<.001
	2	6.63	1.63	29	5.49	<.001
	3	5.33	1.56	29	1.17	.252
80% / 20%	1	5.80	2.12	29	2.06	.048
	2	5.97	1.92	29	2.76	.010
	3	5.03	1.67	29	0.11	.914
60% / 40%	1	4.97	1.43	29	-0.13	.899
	2	5.07	1.07	29	0.21	.831
	3	5.33	1.90	29	0.96	.344

Table 5

Chance Analysis for Base 2 Mean Acquisition Correct Data for Base 2

Feedback	Transform	Mean	SD	df	t	p
100% / 0%	1	7.33	1.73	29	7.39	<.001
	2	6.40	1.77	29	4.32	<.001
	3	5.53	1.65	29	1.76	.088
80% / 20%	1	6.13	2.00	29	3.11	.004
	2	6.20	1.16	29	5.68	<.001
	3	5.73	1.51	29	2.67	.012
60% / 40%	1	5.50	1.76	29	1.56	.130
	2	5.40	1.61	29	1.36	.184
	3	5.03	1.77	29	.10	.919

Transforms 1 and 2. In all cases, except for the 80%/20% group of Base 2, the third transform mean correct score was not significantly different from chance. In addition, the 60%/40% group performed no better than chance for all transforms and bases. These results indicate that both the transform manipulation and the feedback manipulation were successful in altering the subject's response. Specifically, subjects in the 60%/40% group and all subjects as a whole with third order transforms, performed no better than chance.

The analysis uncovered three additional effects that were not predicted. Base 1 revealed a significant Feedback $\times$ Acq. Order interaction ($F(4,81) = 3.29$, $p < .05$). Figure 20 displays the data for this analysis. Base 2 revealed a significant main effect for acquisition order ($F(2,81) = 10.08$, $p < .001$) and a significant Transform $\times$ Acq. Order interaction ($F(4,162) = 2.82$, $p < .05$). Figures 21 and 22 display the data for these analyses. These findings suggest that there are some carry-over effects from previously presented items that affects performance on new items. The phenomenon may be similar to proactive inhibition.

Recognition

The ability to differentiate between the feedback-dependent and feedback-independent models of prototype learning hinges on the data gathered in the recognition phase. The prototypes that the subjects learned in the acquisition phase were explicitly examined in the recognition phase data. Based on the work of Franks and Bransford, the existence of a prototype should lead to a main effect for transform, with the base having the highest recognition score. This characteristic should be true regardless of the feedback manipulation because it is an implicit result of the prototype itself. Therefore, the difference between the two models depends on the existence of a significant main effect for

Figure 20. The mean number of correct responses for stimuli generated from Base 1 as a function of both feedback group and acquisition order.

Figure 21. The mean number of correct responses for stimuli generated from Base 2 as a function of acquisition order.

Figure 22. The mean number of correct responses for stimuli generated from Base 2 as a function of both transformational distance from Base 2 and acquisition order.

feedback, which would indicate that the prototype developed was indeed affected by feedback. Under the feedback-dependent model, there should be a main effect for feedback and transform, while, in the feedback-independent model, there should only be a significant main effect for transform. The key to the significant main effect for transform in the feedback-independent model is that it would indicate that the underlying prototype was developed as a central tendency of the base in spite of the ambiguous feedback.

Data were collected for each of the three variables: response time, confidence rating, and response correctness. Response time was a measure of the time between the presentation of the stimulus and the subject's judgment response. The confidence rating indicated, on a scale from 1 to 5, the subject's confidence in the judgment. The response correctness measured the accuracy of the subject's judgment. As in the acquisition phase, the response time was found to vary widely across subjects and further analysis was abandoned.

Recognition Confidence Rating Analysis. The confidence rating was computed by taking the 1 to 5 confidence rating and multiplying by -1 for "no" responses and +1 for "yes" responses. This yielded a -5 to +5 range for confidence rating, as in the Franks and Bransford procedure.

The first analysis of the confidence data was performed using a seven-way MANOVA, with three between-subject and four within-subject independent variables. The between-subject independent variables were: feedback (three levels), acq. order (three levels), and recognition order (two levels). The within-subject independent variables were: base (two levels), transform (three levels), new/old (two levels), and repetition (two levels). Although there were a total of five levels of transform (base and four transforms), only three could be

analyzed because the base and the fourth transform had empty cells. Since there were no "old" presentations of the base, the observations did not have a filled cell for the new/old within-subjects variable. In the case of fourth order transforms, as in the acquisition phase, the items were the same for both bases, leaving an empty cell for the base within-subjects variable. Table 6 summarizes the significant main effects and interactions for the confidence rating analysis.

There are five types of significant effects shown in Table 6 that warrant further discussion: interactions between the base and other factors; main effect for new/old; main effect for repetition; interaction effect for New/Old x Repetition; and interaction effect for New/Old x Transform. Each of these effects will be discussed separately.

First, as in the acquisition phase, the existence of significant interactions with the base factor indicates that it is necessary to analyze the bases separately. Second, the main effect for new/old (see Figure 23) indicates that the subject had a significantly higher confidence rating for items previously seen (old) than those not seen before (new). This result indicates that the subjects, on average, were able to accurately judge whether they had seen the stimulus. However, the magnitude of the mean confidence ratings indicates that the subjects were biased towards a "yes" response, even for items they hadn't seen.

Third, the main effect for repetition (see Figure 24) indicates that the mean confidence rating of the first presentation was significantly higher than that of the following presentation of the same class of item. The higher confidence rating on the first presentation indicates that the subjects were significantly more likely to answer "yes" than on the second presentation. This was possibly due to the subjects' belief that there was a 50% base rate in the yes/no judgment of the

Table 6

Summary of Significant Effects for Recognition Confidence Rating Data

Effect Type	Effect	df	F	p
Main Effect	New/Old	(1,72)	50.89	<.001
	Reps	(1,72)	15.03	<.001
Two-Way Interaction	Acq. Order x Reps	(2,72)	3.78	.028
	Base x Reps	(1,72)	9.20	.003
	Base x New/Old	(1,72)	15.41	<.001
	Feedback x New/Old	(2,72)	3.28	.043
	Rec. Order x Transform	(2,144)	6.80	.002
	New/Old x Reps	(1,72)	5.33	.024
	New/Old x Transform	(2,144)	6.22	.003
Three-Way Interaction	Base x New/Old x Reps	(1,72)	9.62	.003
	Base x New/Old x Transform	(2,144)	3.11	.048
	Acq. Order x Feedback x Reps	(4,72)	4.58	.002
	Acq. Order x Feedback x Transform	(8,144)	2.33	.002
	Base x New/Old x Rec. Order	(1,72)	7.61	.007
Four-Way Interaction	Base x Rec. Order x Reps x Transform	(2,144)	3.42	.035
	New/Old x Rec. Order x Reps x Transform	(2,144)	5.47	.005
Five-Way Interaction	Acq. Order x Base x Feedback x New/Old x Rec. Order	(4,72)	2.55	.047
Six-Way Interaction	Acq. Order x Feedback x New/Old x Rec. Order x Transform x Reps	(8,144)	3.12	.003

Figure 23. The mean recognition confidence rating for all stimuli as a function of the new/old factor.

Figure 24. The mean recognition confidence rating for all stimuli as a function of presentation order.

items, and may have lead the subjects to compensate for a perceived large number of “yes” responses for the first presentation with a large number of “no” responses for the second presentation. In order to avoid the confounding effect of this response set, only the first repetition was analyzed. Fourth, further examination of the New/Old x Repetition effect (see Figure 25) indicates that the interaction does not compromise the main effect. Finally, examining the New/Old x Transform interaction (see Figure 26) indicates that the subject’s differentiation among the transforms was more pronounced for items seen previously.

A further simplification was made in the analysis of the recognition confidence data. In order to discuss the results in the context of the work of Franks and Bransford, only items that were not presented during the acquisition phase were analyzed. This approach has two advantages. First, this approach, combined with analyzing each base separately, increases the number of levels in the transform within-subject independent variable to five, now including the base and the completely ambiguous fourth transform. Second, this design closely matches that of Franks and Bransford, allowing for direct comparison with their findings.

Both bases were analyzed separately with 2 four-way MANOVAs with three between-subject and one within-subject independent variable. The between-subject independent variables were feedback (three levels), acq. order (three levels), and recognition order (two levels). The within-subject variable was transform (five levels).

The analysis showed significant main effects for Base 1 ($F(4, 288) = 6.47$, $p < .001$) and Base 2 ($F(4, 288) = 7.54$, $p < .001$). Figures 27 and 28 show the mean

Figure 25. The mean recognition confidence rating for all stimuli as a function of both repetition and new/old factors.

Figure 26. The mean recognition confidence rating for all stimuli as a function of transformational distance from base and the new/old factor.

Figure 27. The mean recognition confidence rating for stimuli generated from Base 1 as a function of transformational distance from Base 1.

Figure 28. The mean recognition confidence rating for stimuli generated from Base 2 as a function of transformational distance from Base 2.

confidence rating by transform level. Further analysis was performed to understand the nature of the relationship between confidence rating and transform level. Base 1 data had a significant quadratic trend ($F(1, 72) = 22.68, p < .001$). In addition, an analytical comparison between the mean for the base and mean for the fourth order transform found the base to be significantly higher ($F(1, 72) = 141.68, p < .001$). Base 2 was found to have a significant linear ($F(1, 72) = 50.41, p < .01$) and a significant quadratic trend ($F(1, 72) = 33.35, p < .05$). An analytical comparison of the base against the mean of Transforms 1 through 4 was significant ($F(1, 72) = 162.60, p < .001$). The nature of the relationships shown in Figures 27 and 28, along with the analytical comparisons, indicates that the base confidence rating was the highest confidence rating of all the transforms for both bases. This result replicates the findings of the Franks and Bransford study, but the overall nature of the quadratic trend does not. The Franks and Bransford study found a linear relationship for transform.

Neither Base 1 nor Base 2 showed any significant main or interaction effects for feedback. This finding is consistent with the feedback-independent model. Only Base 2 showed any additional significant effects beyond transform: significant Transform $\times$ Recognition Order interaction ($F(4,288) = 32.88, p < .01$) shown in Figure 29; Acq. Order $\times$ Transform interaction ($F(8,288) = 14.09, p = .050$) shown in Figure 30; and Acq. Order $\times$ Recognition Order $\times$ Transform interaction ($F(8,288) = 14.84, p < .05$). A simple effects analysis of the Acq. Order $\times$ Transform interaction, at each level of transform, found no significant simple effects. The Recognition Order $\times$ Transform effect was also tested with a simple effects analysis at each level of transform, with only the simple effect at the third order transform found to be significant ($F(1,72) = 12.84, p = .001$).

Figure 29. The mean recognition confidence rating for stimuli generated from Base 2 as a function of both transformation distance from Base 2 and recognition order.

Figure 30. The mean recognition confidence rating for stimuli generated from Base 2 as a function of both transformational distance from Base 2 and acquisition order.

Recognition Correctness Analysis. The correctness of the response was recorded for each yes/no judgment response. These were tallied across both repetitions and analyzed for those items not presented in the acquisition phase. The data were analyzed with a four-way MANOVA with three between-subject variables (feedback [three levels], acq. order [three levels], and recognition order [three levels]), and one within-subject variable (transform [five levels]). In keeping with the analysis of the recognition confidence rating data, the data were restricted to the first repetition for new items, with the bases analyzed separately.

As with the analysis of the recognition confidence data, no main effect for feedback was found. This is consistent with the feedback-independent model. The analysis revealed significant main effects for the transform within-subject variable for both Base 1 ($F(4,288) = 5.20, p < .001$) and Base 2 ($F(4,288) = 3.65, p < .001$). Figures 31 and 32 show the mean number of correct recognition judgments for both bases and at each transform. Examination of the data shows that the mean number correct for the base was lower than the mean correct score for the four transforms. This result is similar to the one found in the recognition confidence data, except that the -1 and +1 weighting of the confidence rating has been removed. The mean correctness rating represents a purer measure of the accuracy of the judgment.

Each mean correctness value was further tested against the chance value of one with a paired t test. Tables 7 and 8 list the results of this analysis. All mean correct values significantly differ from chance, with the overall shape of the curves tending toward chance as transform order increases. The important feature of the figures is that the mean number correct for the base is the lowest

Figure 31. The mean number of correct responses during the recognition phase for stimuli generated from Base 1 as a function of transformational distance from Base 1.

Figure 32. The mean number of correct responses during the recognition phase for stimuli generated from Base 2 as a function of transformational distance from Base 2.

Table 7

Chance Analysis of Base 1 Recognition Correctness Data

Transform	Mean	SD	df	t	p
Base	.46	.66	89	-7.87	<.001
1	.77	.78	89	-2.84	.006
2	.66	.64	89	-5.12	<.001
3	.83	.66	89	-2.40	.018
4	.64	.72	89	-4.66	<.001

Table 8

Chance Analysis of Base 2 Recognition Correctness Data

Transform	Mean	SD	df	<i>t</i>	<i>p</i>
Base	.23	.48	89	-15.30	<.001
1	.76	.72	89	-3.20	.002
2	.60	.63	89	-6.00	<.001
3	.46	.66	89	-7.87	<.001
4	.64	.72	89	-4.66	<.001

of all transforms. This remains consistent with the nature of the base being the most “typical” and therefore most mistaken item in the recognition phase.

The mean correct analysis revealed two additional significant effects. For Base 1 there was a significant four-way interaction of Feedback x Acq. Order x Recognition Order x Transform ($F(16,288) = 2.02, p = .012$). For Base 2 there was a significant Recognition Order x Transform, two-way interaction ($F(4,288) = 3.54, p = .008$).

Debriefing

Two pieces of data were collected during the debriefing. The first was a 5-point scale indicating the subject's perceived difficulty with the task. Because the task was not mandatory, not all subjects responded. This made analysis of the data impossible.

The second piece of data collected during the debriefing was the subject's freeform description of the classification judgment. The unedited text of all responses is in Appendix D. References to statements in the following discussion are marked with the statement number in square brackets “[]”.

Although not statistically analyzed, the data provide an insight into the thinking of the subjects. One concern in the development of the procedure was whether the subjects would differentiate the two bases according to the straight lines/curved lines characteristics use to develop the bases. It is clear from examining the responses that at least some of the subjects were making their judgment along this intended dimension. Furthermore, the subjects in the 100%/0% valid feedback group demonstrated more competence in identifying the categories.

For the subjects in the 100%/0% valid feedback group, 7 of the 30 subjects

wrote statements that described Category 1 as containing “corners” [4, 16, 19], “edges/edgy” [7, 27], “angles” [3], and “points” [1]. The statements also contained references to Category 2 as having “round” [3, 4, 7, 16, 27] or “curved” [19] shapes. For these subjects there was a clear demonstration that they found a global feature by which to differentiate the two categories.

For the subjects in the 80%/20% valid feedback group, 8 of the 30 subjects wrote statements that described Category 1 as containing “lines” [31], “edged” [34], “straight” [37], “angular” [40, 44], or “sharp edges” [42] shapes. The same statements also contained references to Category 2 as having “rounded” [34, 40, 42, 43, 44] or “curved” [37, 54] shapes. There were several responses that seemed to indicate that the subjects were aware of the invalid feedback, such as Response 34. This subject’s response demonstrates that the subject had the idea of the bases but that the invalid feedback had caused concern.

The pattern of response of the 60%/40% group varied greatly from either the 100%/0% or the 80%/20% groups. In the 60%/40% group, only two subjects [59, 74] described the categories in terms of the criteria used to define the bases. In fact, a large group of subjects expressed total confusion about the categories, as evidenced by the use of words such as “guessed” [57, 63], “couldn’t tell” [58, 62, 71, 81], “have no idea” [61], or “don’t know” [76]. The most representative response of the confusion exhibited by the subjects was Response 82. This subject indicated that there was a relationship, but it wasn't consistent.

Discussion

The feedback-independent model for prototype learning was supported by the results. The results will be discussed in more detail by examining the

following findings: a main effect for feedback in the acquisition phase; a main effect for transform in the acquisition phase; no main effect for feedback in the recognition phase; a main effect for transform in the recognition phase; and positive evidence from the debriefing data that the subjects could accurately describe the classification.

Effect for Feedback Found in the Acquisition Phase

Assuming that judgment during the acquisition phase is based on an acquired notion of the categories, the main effect for feedback during the acquisition phase for the mean correct data indicates that the feedback groups differed in their ease of learning the categories. This conclusion is further reinforced by the fact that the categories were artificial and the subjects' experience with the categories was limited to the experimental setting. All knowledge of the categories was acquired during the procedure. The pattern of results was as predicted with the 100%/0% feedback group performing the best, and the 60%/40% group performing the worst.

Effect for Transform Found in the Acquisition Phase

The main effect for transform establishes the importance of the transform manipulation. As predicted, the subjects' performance generally deteriorated as the numbers of transforms increased until it reached no better than chance after three transforms. This result is an important indication that the transform manipulation significantly predicts performance and validates that the Franks and Bransford methodology applies to the judgment task.

No Effect for Feedback Found in the Recognition Phase

The data analysis for the recognition phase followed the same methodology as Franks and Bransford, but also included recognition correctness.

The analysis of the recognition confidence data failed to show a main effect for feedback. Additional analysis of the mean correct data also failed to show a main effect for feedback. Because of the lack of a significant difference between the groups with either recognition confidence or recognition correctness, I accepted the null hypothesis, asserting that the three feedback groups formed similar prototypes as measured during the recognition phase. This result supports the feedback-independent model of prototype learning.

Effect for Transform Found in the Recognition Phase

Because the null hypothesis was accepted for the feedback analysis, it was important to establish the reliability of the data set. Further analysis for both the recognition confidence and recognition correctness found main effects for transform. Although the exact linear relationship of the Franks and Bransford study was not replicated, the main finding that the base was rated higher than the other transforms was replicated. This was further supported by a similar finding for the recognition correctness. These patterns were predicted and indicate that failure to reject the null hypothesis in the case of feedback cannot be explained in terms of unreliability of the data. Hence, this result adds to the credibility of accepting the null hypothesis for feedback.

Debriefing Data

Finally, the data collected during the debriefing provides anecdotal evidence that supports the empirical findings from the acquisition phase. In line with the analysis of the acquisition correctness data, a larger number of subjects in the 100%/0% feedback group was able to accurately describe the categories than in the 60%/40% feedback group. This evidence provides an indication that the task, although not easy, was doable as designed. In general, the remarks in

the debriefing were more congruent with the statistical results for the acquisition phase than the recognition phase. The fact that the debriefing statements indicated a feedback effect, while the recognition data showed no feedback effect appears to be contradictory. In effect, the metacognitive knowledge of the subjects was inconsistent with the knowledge demonstrated in the recognition phase.

Conclusion

At first examination, the results of the acquisition phase and recognition phase appear to contradict each other. The pattern of the acquisition data argues for the feedback-dependent model for prototype learning. As in other CPL tasks, the nature of the feedback appeared to significantly affect the subjects' behavior. But the pattern of the recognition data argues that the underlying prototype is the same for each group. What does this apparent contradiction mean?

One possible explanation is that the nature of the two tasks (judgment and recognition) lead subjects to utilize different memory structures. During the acquisition phase, the subjects may actually be developing two memory structures: a *prototype store* and an *exception store*. In the prototype store, the prototype is being developed during the procedure independent of feedback as a set of propositions about the stimuli. At the same time, the exception store accumulates the exceptions brought about by the feedback on missed trials. The process that generates the exceptions is called *reconciliation*. During the reconciliation process, each trial that yields an incorrect judgment generates an exception that is stored in the exception store.

In the case of the 100%/0% valid group, very few exceptions are stored in

the exception store because there is very little negative feedback. In the 80%/20% and 60%/40% groups, there is a larger number of contradictory feedback trials, leading to the storage of a larger number of exceptions. The growing exception store causes the subject to perform poorly on the judgment task by acting as a proactive inhibitor. The larger the number of exceptions stored, the greater the proactive inhibition effect. The proactive inhibition effect should be at maximum strength for a group that receives 50%/50% feedback.

In the case of the recognition task, the data indicate that a exception store is irrelevant. Independent of the feedback, the recognition confidence and correctness data demonstrate the transform effect indicative of a prototype. Either the exception store is present and just not accessed, or it is in fact unavailable during the recognition task.

One possible explanation for the exception store's unavailability during recognition is that the exception store is similar to a short-term memory store that has a limited lifetime. If this is the case, then the effects of the exception store could be mediated by manipulating the time between trials. One possible approach to investigate the nature of the exception could be to structure the acquisition phase of a similar experiment into two stages.

In the first stage, the subjects would receive a feedback manipulation that is similar to the manipulation in this experiment. The second stage would then consist of additional trials presented without feedback. The time between the two stages would be manipulated. If the exception store is a short-term memory store, then, as time between the two stages increases, the performance on the classification task during the second stage should improve. In addition, the feedback manipulation should act to increase the time between the stages

necessary to mediate the effects of the proactive inhibition. I would expect that the time between stages for 50%/50% groups would be longer than for an 80%/20% group.

References

- Anderson, J. R. (1976). *Language, memory, and thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Anderson, J. R., & Bower, G. H. (1973). *Human associative learning*. Washington, DC: Winston.
- Bartlett, J. C. (1932). *Remembering: A study in experimental and social psychology*. New York and London: Cambridge University Press.
- Bobrow, D. G., & Winograd, T. (1977). An overview of KRL, a knowledge representation language. *Cognitive Science*, 1, 3–46.
- Brehmer, B. (1974). Hypotheses about relations between scaled variables in the learning of probabilistic inference tasks. *Organizational Behavior and Human Performance*, 11, 1–27.
- Brewer, W. F., & Treyens, J. C. (1981). Role of schemata in memory for places. *Cognitive Psychology*, 13, 207–230.
- Collins, A. M., & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8, 240–247.
- Franks, J. J., & Bransford, J. D. (1971). Abstraction of visual prototypes. *Journal of Experimental Psychology*, 90, 1, 65–74.
- Gibson, E. J. (1969). *Principles of perceptual learning and development*. New York: Appleton-Century-Crofts.
- Gleitman, H. (1991). *Psychology*. New York: W W Norton and Company.
- Norman, D. A., & Rumelhart, D. E. (1975). *Explorations in cognition*. New York: W. H. Freeman and Company
- Shafer, G., & Pearl, J. (1990). *Uncertain reasoning*. San Mateo, California: Morgan Kaufman Publishers, Inc.

- Shank, R. C., & Abelson, R. (1977). *Scripts, plans, goals, and understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Slovic, P., & Lichtenstein, S. (1971). Comparison of bayesian and regression approaches to the study of information processing in judgment, organizational. *Behavior and Human Performance*, 6, 649–744.
- Tversky, A., & Kahneman, D. (1973). Availability: A heuristic for judging frequency and probability. *Cognitive Psychology*, 5, 207–232.
- Tversky, A. & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science*, 125, 1124–1131.

Appendix A

Prototypes and Ambiguous Feedback

2

APPROVED BY THE MASTERS THESIS COMMITTEE

Dr. Robert Cooper

Dr. Robert Fox

Dr. Mildred Alvarez

APPROVED BY THE DESIGN AND ANALYSIS COMMITTEE

A campus of The California State University

College of Social Sciences • Department of Psychology
One Washington Square • San José, California 95192-0120 • 408/924-5600 • FAX 408/924-5605

January 6, 1992

TO: Scott Sarnikowski, MA candidate

FROM: Kevin Jordan, MA Coordinator

A handwritten signature in dark ink, appearing to read 'Kevin Jordan', written over the printed name.

RE: Design and analysis review

I apologize for the late date of this letter but the intervening holidays prevented an earlier response. Drs. Hawkins and Wise have read your thesis proposal for the Design and Analysis Committee. Dr. Hawkins' comments are attached and, based on his extensive editorial comments, I have enclosed Dr. Wise's copy of your proposal. Based on their comments, the thesis proposal is approved. As you can see, however, both reviewers claim to have difficulty comprehending the material; as such, they are placing a great deal of responsibility for the test of your hypotheses in yours and your thesis chairs hands. For the final version of the thesis, you might consider including more background material to make the entire manuscript more accessible to the educated reader.

Based on this committee's approval, the collection of data for your thesis is approved contingent on documentation of compliance with university policy regarding the use of human subjects in research. University policy requires approval of your project by the Human Subjects Institutional Review Board. Please provide me with a file copy documenting such approval as soon as you receive it. After that copy is part of your file, you may begin collecting data.

Congratulations on your progress to date! We look forward to the continuation of your fine performance in the program.

cc:

Alvarez
Cooper
Fox
Hawkins
Wise
file

A campus of The California State University

Office of the Academic Vice President • Associate Academic Vice President • Graduate Studies and Research
One Washington Square • San José, California 95192-0025 • 408/924-2480

To: Scott Sarnikowski, Psychology
900 Pepper Tree Lane #1121
Santa Clara, CA 95051

From: Serena W. Stanford *Serena W. Stanford*
AAVP, Graduate Studies and Research

Date: January 23, 1992

The Human Subjects-Institutional Review Board has approved
your request to use human subjects in the study entitled:

"Resilience of Prototypes to Ambiguous Feedback"

This approval is contingent upon the subjects participating in your research project being appropriately protected from risk. This includes the protection of the anonymity of the subjects' identity when they participate in your research project, and with regard to any and all data that may be collected from the subjects. The Board's approval includes continued monitoring of your research by the Board to assure that the subjects are being adequately and properly protected from such risks. If at any time a subject becomes injured or complains of injury, you must notify Dr. Serena Stanford immediately. Injury includes but is not limited to bodily harm, psychological trauma and release of potentially damaging personal information.

Please also be advised that each subject needs to be fully informed and aware that their participation in your research project is voluntary, and that he or she may withdraw from the project at any time. Further, a subject's participation, refusal to participate or withdrawal will not affect any services the subject is receiving or will receive at the institution in which the research is being conducted.

If you have questions, please contact me at 408-924-2480.

CC: Robert G. Cooper, Ph.D.

Appendix B

Selection of Stimuli Set

Five symbols were chosen for each base, four to be used for the base and one for a congruent substitution. Considering there are two bases, this yielded a total of 10 symbols that were chosen to be included in the stimuli set.

The transformations defined for this experiment were similar to the Franks and Bransford study with the exception that the substitutions were defined to include the symbols in the alternate base. This definition was crucial to the development of the stimuli continuum, which was an important manipulation in the experiment. Twenty-four substitutions were defined with the restriction that there could be no repeated symbols in any stimulus. The number 24 derives from the fact that there were six valid symbols for substitution (one congruent with the base and the five symbols that make up the other base) that could be substituted in the four locations, yielding 6×4 , or 24 possible substitutions.

Another important restriction for the substitutions was to limit the number of substitutions to two. This restriction guaranteed that any transform sequence would not “cross-over” the midpoint. If the items had been allowed to “cross-over,” the concept of a continuum would be violated. The final restriction for the transforms was to limit the number of deletions to a single deletion. This restriction was also used in the Franks and Bransford study.

Taking into account that there were 11 symbols (the 10 defined for the bases and the empty symbol from a deletion) and four locations in which to place them, there were 11^4 or 14,641 possible stimuli to select from. With 36 possible transforms (MaP1:2, MiP1:6, De1:4, and Su1:24), with up to 4 levels of

transformations from both bases, there were approximately 3.4 million possible transforms sequences.

To select the potential transform sequences, I used two restrictions: The ambiguous fourth order transform stimulus must not have any duplicated symbols, and the ambiguous stimulus must only be attainable through four consecutive transforms (similar to a restriction imposed in the Franks and Bransford study). I wrote several LISP programs in Macintosh Common LISP (version 2.01) and ran them on a Macintosh IIfx to yield 183 ambiguous stimuli out of a possible 14,641. These resulting stimuli were then exhaustively processed to determine the total number of transform sequences that yielded these ambiguous stimulus (it was expected that there would be many transform sequences that lead to each stimulus). There were approximately 20,000 valid transform sequences, with an average of 200 per stimulus (100 from each base).

LISP Program Source

```

;;;
;;; Name:      triples
;;;
;;; function:  check a list of 4 items for the existence of three similar items
;;;
;;; arguments: list - list of 4 items
;;;
;;; returns:   t - the supplied list contains 3 identical items.
;;;           nil - the supplied list does not contain 3 identical items.
;;;
(defun triples (list)
  (if (or (< (length (remove (nth 0 list) list)) 2)
        (< (length (remove (nth 1 list) list)) 2))
      (setq triples t)
      (setq triples nil))
  )

;;;
;;; name:      create_results_table
;;;
;;; function:  Create the table that will hold the results of the operations. This
;;;            function assumes a fourth dimensional table which corresponds to the
;;;            number of elements in a stimulus set.
;;;
;;; arguments: limit - the limit to the transforms. This is equal to the number of
;;;                   of all transforms (36). Used in performing the transforms.
;;;            dim    - the number of elements for the transform set. Set at 11.
;;;
(defun create_results_table (limit dim)
  (setq results_table (make-array (list dim dim dim dim))) ; allocate table
  (dotimes (first dim) ; loop through the table, with the dimensions
    (dotimes (second dim) ; representing the elements.
      (dotimes (third dim)

```

```

(dotimes (fourth dim)

;;
;;      Check for duplicate elements. This is used to screen out invalid item
;;      configurations. This covers duplicate deletions. The net effect of the
;;      operation is to pre-validate the table. Valid entries contain a t.
;;      In later operations. Only locations with a t will be filled in
;;

      (if (not (duplicates (list first second third fourth)))
          (setf (aref results_table first second third fourth) t)
          )
      )
    )
  )

;;
;; Generate and fill in the table for the base 1, first order transforms.
;;
(format t "~&Doing Base 1, one transform...")           ; Status indicator.
(do_one_transform 1 base1 sub1 limit)

;;
;; Generate and fill in the table for the base 2, first order transforms.
;;
(format t "~&Doing Base 2, one transform...")           ; Status indicator.
(do_one_transform 2 base2 sub2 limit)

;;
;; Generate and fill in the table for the base 1, second order transforms.
;;
(format t "~&Doing Base 1, two transforms...")         ; Status indicator.
(do_two_transforms 1 base1 sub1 limit)

;;
;; Generate and fill in the table for the base 2, second order transforms.
;;
(format t "~&Doing Base 2, two transforms...")         ; Status indicator.
(do_two_transforms 2 base2 sub2 limit)

;;
;; Generate and fill in the table for the base 1, third order transforms.
;;
(format t "~&Doing Base 1, three transforms...")       ; Status indicator.
(do_three_transforms 1 base1 sub1 limit)

;;
;; Generate and fill in the table for the base 2, third order transforms.
;;
(format t "~&Doing Base 2, three transforms...")       ; Status indicator.
(do_three_transforms 2 base2 sub2 limit)

;;
;; Generate and fill in the table for the base 1, fourth order transforms.
;;
(format t "~&Doing Base 1, four transforms...")        ; Status indicator.
(do_four_transforms 1 base1 sub1 limit t 4)

;;
;; Generate and fill in the table for the base 2, fourth order transforms.
;;
(format t "~&Doing Base 2, four transforms...")        ; Status indicator.
(do_four_transforms 2 base2 sub2 limit 4 '*')

)

;;;
;;; Name:      duplicates
;;;
;;; function:  Take in a list with 4 elements and determine if there are items
;;;            that are duplicated. If duplicates are found, return t.
;;;
;;; arguments: list - list of elements to be checked.
;;;
;;; returns:  t    - the list contains two identical items.
;;;           nil   - the list does not contain two identical items.
;;;

(defun duplicates (list)

  (if (or
      (equal (nth 0 list) (nth 1 list))
      (equal (nth 0 list) (nth 2 list))
      (equal (nth 0 list) (nth 3 list))
      (equal (nth 1 list) (nth 2 list))
      (equal (nth 1 list) (nth 3 list))
  )

```

```

(equal (nth 2 list) (nth 3 list)))

(setq duplicates t)
(setq duplicates nil))
)

;;;
;;; Set list of final transforms.
;;;

(setq final_transform_patterns_base1 (list (list su13 mip2 mip1 su12)
                                           (list map1 map2 su18 su13)
                                           (list su21 su11 mip6 mip5)
                                           (list mip4 su21 su12 mip1)
                                           (list su9 mip1 su15 mip3)
                                           (list mip1 su7 su20 mip4)
                                           (list mip2 mip4 su21 su16)
                                           (list mip3 mip5 su17 su15)
                                           (list su23 map2 de3 mip5)
                                           (list mip5 mip3 de2 su23)))

(setq final_transform_patterns_base2 (list (list su11 map1 su14 mip6)
                                           (list su12 su5 mip2 map2)
                                           (list mip5 mip1 su12 su19)
                                           (list su15 map1 su12 mip6)
                                           (list map2 su19 mip2 su12)
                                           (list su4 su10 mip3 map2)
                                           (list mip6 su9 map1 su6 )
                                           (list su8 map1 mip6 su12)
                                           (list su10 mip1 mip5 de3 )
                                           (list de3 su10 mip2 map2)))

;;;
;;; Assign values to the elements
;;;

(setq a 0)
(setq b 1)
(setq c 2)
(setq d 3)
(setq e 4)
(setq empty 5)
(setq v 6)
(setq w 7)
(setq x 8)
(setq y 9)
(setq z 10)

;;;
;;; Assign values to the transforms so they can be referenced
;;;

(setq map1 0)
(setq map2 1)
(setq mip1 2)
(setq mip2 3)
(setq mip3 4)
(setq mip4 5)
(setq mip5 6)
(setq mip6 7)
(setq de1 8)
(setq de2 9)
(setq de3 10)
(setq de4 11)
(setq su1 12)
(setq su2 13)
(setq su3 14)
(setq su4 15)
(setq su5 16)
(setq su6 17)
(setq su7 18)
(setq su8 19)
(setq su9 20)
(setq su10 21)
(setq su11 22)
(setq su12 23)
(setq su13 24)
(setq su14 25)
(setq su15 26)
(setq su16 27)
(setq su17 28)
(setq su18 29)
(setq su19 30)
(setq su20 31)
(setq su21 32)
(setq su22 33)
(setq su23 34)

```

```

(setq su24 35)

;;;
;;; Define a little short cut for evaluating valid transform patterns.
;;;
(setq subs 8)
(setq dels 8)

;;;
;;; List for base 1 manipulations.
;;;
(setq base1 (list a b c d))

;;;
;;; Substitutions for base 1 manipulations.
;;;
(setq sub1 (list e v w x y z))

;;;
;;; List for base 2 manipulations
;;;
(setq base2 (list w x y z))

;;;
;;; Substitutions for base 2 manipulations
;;;
(setq sub2 (list v a b c d e))

;;;
;;; List of string values for reconstructing the item patterns after they have
;;; been added to the results table.
;;;
(setq item_list (list 'a 'b 'c 'd 'e 'empty 'v 'w 'x 'y 'z))

;;;
;;; List of all transforms in strings.
;;;
(setq transform_list (list 'map1 'map2
                           'mip1 'mip2 'mip3 'mip4 'mip5 'mip6
                           'del1 'del2 'del3 'del4
                           'su1 'su2 'su3 'su4
                           'su5 'su6 'su7 'su8
                           'su9 'su10 'su11 'su12
                           'su13 'su14 'su15 'su16
                           'su17 'su18 'su19 'su20
                           'su21 'su22 'su23 'su24))

;;;
;;; List of valid transforms that is used to validate the transforms strings. This list
;;; is used to insure that there are no more than 2 deletions or substitutions. Deletions
;;; are limited to two for obvious reasons. Substitutions are limited to two so that the
;;; continuum is not violated.
;;;
(setq valid_transform_list (list map1 map2
                                  mip1 mip2 mip3 mip4 mip5 mip6
                                  dels dels dels dels
                                  subs subs subs subs
                                  subs subs subs subs
                                  subs subs subs subs
                                  subs subs subs subs
                                  subs subs subs subs
                                  subs subs subs subs))

(setq base_list (list 'a 'b 'c 'd 'e '*' 'v 'w 'x 'y 'z))

;;;
;;; Name:      find-value
;;;
;;; function:  Search the results table and pull out all transform patterns that reach
;;; the supplied value. This is used after the results table has been
;;; processed to get all the transform patterns.
;;;
;;; arguments: value - value to be looked up.
;;;             dim  - dimension of the each axis.
;;;
;;; returns:   find_list - list containing the transform patterns for the value.
;;;
(defun find_value (value dim)

```

```

(setq find_list nil) ; initialize the find_list
(dotimes (first dim)      ; cycle through the results_table
  (dotimes (second dim)
    (dotimes (third dim)
      (dotimes (fourth dim)

;;          Check the value against the location of the results table.
;;
;;          (if (equal (aref results_table first second third fourth) value)

;;
;;          If the value matches, add the item pattern to the list.
;;
;;          (setq find_list (append find_list (list (list (nth first base_list)
                                                         (nth second base_list)
                                                         (nth third base_list)
                                                         (nth fourth base_list))))))
      )
    )
  )
)

;;;
;;; Name:      apply_transform
;;;
;;; function:  Apply the defined transforms.
;;;
;;; arguments: transform - the transform to be applied.
;;;            base      - the base to use.
;;;            sub       - the substitution list to use.
;;;
;;; returns:   function is evaluated as a list that is properly manipulated.
;;;
(defun apply_transform (transform base sub)
  (cond ((equal transform map1) (list (nth 2 base)
                                       (nth 3 base)
                                       (nth 0 base)
                                       (nth 1 base)))
        ((equal transform map2) (list (nth 1 base)
                                       (nth 0 base)
                                       (nth 3 base)
                                       (nth 2 base)))
        ((equal transform mip1) (list (nth 1 base)
                                       (nth 0 base)
                                       (nth 2 base)
                                       (nth 3 base)))
        ((equal transform mip2) (list (nth 2 base)
                                       (nth 1 base)
                                       (nth 0 base)
                                       (nth 3 base)))
        ((equal transform mip3) (list (nth 3 base)
                                       (nth 1 base)
                                       (nth 2 base)
                                       (nth 0 base)))
        ((equal transform mip4) (list (nth 0 base)
                                       (nth 2 base)
                                       (nth 1 base)
                                       (nth 3 base)))
        ((equal transform mip5) (list (nth 0 base)
                                       (nth 3 base)
                                       (nth 2 base)
                                       (nth 1 base)))
        ((equal transform mip6) (list (nth 0 base)
                                       (nth 1 base)
                                       (nth 3 base)
                                       (nth 2 base)))
        ((equal transform del1) (list empty
                                       (nth 1 base)
                                       (nth 2 base)
                                       (nth 3 base)))
        ((equal transform de2)  (list (nth 0 base)
                                       empty)))

```

		(nth 2 base) (nth 3 base))
((equal transform de3)	(list	(nth 0 base) (nth 1 base) empty (nth 3 base))
((equal transform de4)	(list	(nth 0 base) (nth 1 base) (nth 2 base) empty))
((equal transform su1)	(list	(nth 0 sub) (nth 1 base) (nth 2 base) (nth 3 base))
((equal transform su2)	(list	(nth 0 base) (nth 0 sub) (nth 2 base) (nth 3 base))
((equal transform su3)	(list	(nth 0 base) (nth 1 base) (nth 0 sub) (nth 3 base))
((equal transform su4)	(list	(nth 0 base) (nth 1 base) (nth 2 base) (nth 0 sub))
((equal transform su5)	(list	(nth 1 sub) (nth 1 base) (nth 2 base) (nth 3 base))
((equal transform su6)	(list	(nth 0 base) (nth 1 sub) (nth 2 base) (nth 3 base))
((equal transform su7)	(list	(nth 0 base) (nth 1 base) (nth 1 sub) (nth 3 base))
((equal transform su8)	(list	(nth 0 base) (nth 1 base) (nth 2 base) (nth 1 sub))
((equal transform su9)	(list	(nth 2 sub) (nth 1 base) (nth 2 base) (nth 3 base))
((equal transform su10)	(list	(nth 0 base1) (nth 2 sub) (nth 2 base) (nth 3 base))
((equal transform su11)	(list	(nth 0 base) (nth 1 base) (nth 2 sub) (nth 3 base))
((equal transform su12)	(list	(nth 0 base) (nth 1 base) (nth 2 base) (nth 2 sub))
((equal transform su13)	(list	(nth 3 sub) (nth 1 base) (nth 2 base) (nth 3 base))
((equal transform su14)	(list	(nth 0 base) (nth 3 sub) (nth 2 base) (nth 3 base))
((equal transform su15)	(list	(nth 0 base) (nth 1 base) (nth 3 sub) (nth 3 base))

```

    ((equal transform su16) (list (nth 0 base)
                                  (nth 1 base)
                                  (nth 2 base)
                                  (nth 3 sub)))

    ((equal transform su17) (list (nth 4 sub)
                                  (nth 1 base)
                                  (nth 2 base)
                                  (nth 3 base)))

    ((equal transform su18) (list (nth 0 base)
                                  (nth 4 sub)
                                  (nth 2 base)
                                  (nth 3 base)))

    ((equal transform su19) (list (nth 0 base)
                                  (nth 1 base)
                                  (nth 4 sub)
                                  (nth 3 base)))

    ((equal transform su20) (list (nth 0 base)
                                  (nth 1 base)
                                  (nth 2 base)
                                  (nth 4 sub)))

    ((equal transform su21) (list (nth 5 sub)
                                  (nth 1 base)
                                  (nth 2 base)
                                  (nth 3 base)))

    ((equal transform su22) (list (nth 0 base)
                                  (nth 5 sub)
                                  (nth 2 base)
                                  (nth 3 base)))

    ((equal transform su23) (list (nth 0 base)
                                  (nth 1 base)
                                  (nth 5 sub)
                                  (nth 3 base)))

    ((equal transform su24) (list (nth 0 base)
                                  (nth 1 base)
                                  (nth 2 base)
                                  (nth 5 sub)))
  )
)

;;;
;;; Name:      do_one_transform
;;;
;;; function: Exhaustively generate all first order transforms and add them to the
;;;           results table.
;;;
;;; arguments: base_no - Number of base.
;;;           sub_list - List to use for substitutions.
;;;           limit    - Limit of the loops. This is equal to the number of elements.
;;;
(defun do_one_transform (base_no base sub_list limit)
  (dotimes (first (+ limit 1)) ; loop through all transforms

    ;;
    ;; generate the stimuli set and assign it to the stimuli variable.
    ;;
    (setf stimuli (apply_transform first base sub_list))

    ;;
    ;; check the location in the results table pointed to by the stimuli list.
    ;;

    ;;
    ;; if the location pointed to by the stimulus pattern is set to nil, this is an illegal
    ;; set of items and nothing is done with this item.
    ;;

    (cond ((equal (aref results_table (nth 0 stimuli)
                                      (nth 1 stimuli)
                                      (nth 2 stimuli)
                                      (nth 3 stimuli))
                  nil)

          )

    ;;
    ;; if the location pointed to by the stimulus patten is t, then this is a legal set of items

```

```

;; so 1 is stored at the location, marking it as a first order transform.
;;
;;      ((equal (aref results_table      (nth 0 stimuli)
;;                                         (nth 1 stimuli)
;;                                         (nth 2 stimuli)
;;                                         (nth 3 stimuli))
;;
;;                                         t)
;;
;;      (setf (aref results_table      (nth 0 stimuli)
;;                                         (nth 1 stimuli)
;;                                         (nth 2 stimuli)
;;                                         (nth 3 stimuli))
;;
;;          1)
;;
;;      )
;;
;;  )
;;
;;  )
;;
;;;
;;; name:      do_two_transforms
;;;
;;; function:  Exhaustively generate second order transforms and update the results table.
;;;
;;; arguments: base_no - number of the base
;;;             base    - base to use
;;;             sub_list - list to use for substitutions
;;;             limit   - limit of the loops
;;;
(defun do_two_transforms (base_no base sub_list limit)
  (dotimes (first (+ limit 1))      ; generate loop
    (dotimes (second (+ limit 1))
      ;;
      ;; generate the second order transforms and assign the item pattern to stimuli.
      ;;
      (setf stimuli (apply_transform second
                                     (apply_transform first base sub_list) sub_list))

      ;;
      ;; check the results table pointed to by the stimuli pattern
      ;;
      ;; If the location is nil then this is an invalid stimuli pattern, and ignore it.
      ;;
      (cond ((equal (aref results_table      (nth 0 stimuli)
                                              (nth 1 stimuli)
                                              (nth 2 stimuli)
                                              (nth 3 stimuli))
                    nil)
              )

            )

      ;;
      ;; If the location is t then this is a legal stimuli pattern and a 2 is stored at the location
      ;; to mark this a second order transform. If the location was reached by a first order
      ;; transform then this location is ignored, insuring that it remains a first order transform.
      ;;
      ((equal (aref results_table      (nth 0 stimuli)
                                      (nth 1 stimuli)
                                      (nth 2 stimuli)
                                      (nth 3 stimuli))
              t)
        (setf (aref results_table      (nth 0 stimuli)
                                      (nth 1 stimuli)
                                      (nth 2 stimuli)
                                      (nth 3 stimuli))
              2)
        )

      )

    )

  )

)

;;;
;;; Name:      do_three_transforms
;;;
;;; function:  Exhaustively generate all third order transforms and update the results

```



```

;;;      table.
;;;
;;; arguments: base_no - Number of the base
;;;             base    - base list
;;;             sub_list - list for substitutions list
;;;             limit    - limit of looping.
;;;
(defun do_three_transforms (base_no base sub_list limit)
  (dotimes (first (+ limit 1))
    (format t "~&~& 3 Transforms Doing ~s..." (nth first transform_list)) ; status check
    (dotimes (second (+ limit 1))
      (dotimes (third (+ limit 1))
        ;;
        ;; Generate the third order transform and assign the result to stimuli.
        ;;
        (setf stimuli (apply_transform third
                                       (apply_transform second
                                         (apply_transform first base sub_list) sub_list) sub_list))

        ;;
        ;; Check the transform pattern to see if there are three substitutions or deletions. If
        ;; there is don't do anything.
        ;;
        (cond ((and (equal (nth first valid_transform_list)
                           (nth second valid_transform_list))
                    (equal (nth second valid_transform_list)
                           (nth third valid_transform_list)))
              )

        ;;
        ;; Check the results table to see if the stimuli pattern is valid. If the entry is nil
        ;; then the stimuli pattern is illegal and is skipped.
        ;;
        ((equal (aref results_table
                     (nth 0 stimuli)
                     (nth 1 stimuli)
                     (nth 2 stimuli)
                     (nth 3 stimuli))
              nil)
         )

        ;;
        ;; If the location is set to t, then this is a valid stimuli set and has not been
        ;; reached by either a first or second order transform. Set the location to 3 to mark it
        ;; as a third order transform.
        ;;
        ((equal (aref results_table
                     (nth 0 stimuli)
                     (nth 1 stimuli)
                     (nth 2 stimuli)
                     (nth 3 stimuli))
              t)
         (setf (aref results_table
                     (nth 0 stimuli)
                     (nth 1 stimuli)
                     (nth 2 stimuli)
                     (nth 3 stimuli))
               3)
         )
      )
    )
  )

;;;
;;; Name:      do_four_transforms
;;;
;;; function: Exhaustively generate all fourth order transforms and update the
;;;            results table.
;;;
;;; arguments: base_no - Number of the base
;;;             base    - base list
;;;             sub_list - substitution list
;;;             limit    - limit for loops
;;;
(defun do_four_transforms (base_no base sub_list limit)

```

```

(dotimes (first (+ limit 1))
  (format t "~&~& 4 transforms Doing ~s..." (nth first transform_list)) ; Status check

  (dotimes (second (+ limit 1))
    (format t "~&~& 4 transforms Doing ~s..." (nth second transform_list)) ; Status check

    (dotimes (third (+ limit 1))
      (dotimes (fourth (+ limit 1))

        ;;
        ;; Generate the items for the fourth order transform
        ;;

        (setf stimuli (apply_transform fourth
          (apply_transform third
            (apply_transform second
              (apply_transform first base sub_list) sub_list) sub_list) sub_list))

        ;;
        ;; Check the transform pattern to insure that there are not 3 substitutions or
        ;; deletions.
        ;;

        (cond ((triples (list (nth first valid_transform_list)
                              (nth second valid_transform_list)
                              (nth third valid_transform_list)
                              (nth fourth valid_transform_list)))

              )

        ;;
        ;; Check the location in the results table to determine if the item is legal. If the location is
        ;; nil then the item is illegal and should be ignored.
        ;;

        ((equal (aref results_table (nth 0 stimuli)
                                   (nth 1 stimuli)
                                   (nth 2 stimuli)
                                   (nth 3 stimuli))
              nil)

        ;;
        ;; If the location is t then the location is valid and the transform should be marked. A 4 is placed
        ;; at the location and guarantees that the item is only reachable through a fourth order transform.
        ;;

        ((equal (aref results_table (nth 0 stimuli)
                                   (nth 1 stimuli)
                                   (nth 2 stimuli)
                                   (nth 3 stimuli))
              t)

          (setf (aref results_table (nth 0 stimuli)
                                   (nth 1 stimuli)
                                   (nth 2 stimuli)
                                   (nth 3 stimuli))
                4)

        )

      )

    )

  )

)

;;
;; Name: generate_transform
;;
;; function: Go through the complete results table and print out the transforms. This function
;; is used to go through the complete results table and print out the final item
;; configurations. In generating the items, this final results table is culled for 4's,
;; the transform pattern that reaches the cell with a 4, is saved. Later the 4 transforms
;; are used to generate the intermediate item configurations by pulling them apart and
;; applying them in order. For example: Sul2, Sul2:Map1, Sul2:Map1:Mip3,
;; Sul2:Map1:Mip3:Del, are all generated from a single fourth order transform.
;;
;; This function is run twice, once for each base to get the transform patterns from
;; each base.
;;
;; arguments: base_no - Number of the base
;; base - base list
;; sub_list - substitution list
;; limit - limit to looping
;; search_char - character to look for

```

```

;;
(defun generate_transforms (base_no base sub_list limit search_char)

;;
;; Open output file
;;
    (with-open-file (stream "Thesis Project:Final output - Base 2" :direction :output)
      (dotimes (first (+ limit 1))
        (format t "~&~& 4 transforms Doing ~s..." (nth first transform_list))

        (dotimes (second (+ limit 1))
          (format t "~&~& 4 transforms Doing ~s..." (nth second transform_list))

          (dotimes (third (+ limit 1))

            (dotimes (fourth (+ limit 1))

              ;;
              ;; Generate the fourth order transforms
              ;;
              (setf stimuli (apply_transform fourth
                (apply_transform third
                  (apply_transform second
                    (apply_transform first base sub_list)
                    sub_list) sub_list) sub_list))

              ;;
              ;; Check the transform pattern to insure that no more than two substitutions or deletions
              ;; are present.
              ;;
              (cond ((triples (list (nth first valid_transform_list)
                (nth second valid_transform_list)
                (nth third valid_transform_list)
                (nth fourth valid_transform_list)))

                )

              ;;
              ;; If the location is equal to the specified search character then print out the base number,
              ;; transform pattern and item. This will be used to derive the final set of items.
              ;;
              ((equal (aref results_table (nth 0 stimuli)
                (nth 1 stimuli)
                (nth 2 stimuli)
                (nth 3 stimuli))
                search_char)

                (format stream "~&~&S, ~S, ~S, ~S, ~S, ~S, ~S, ~S"
                  base_no
                  (nth (nth 0 stimuli) item_list)
                  (nth (nth 1 stimuli) item_list)
                  (nth (nth 2 stimuli) item_list)
                  (nth (nth 3 stimuli) item_list)
                  (nth first transform_list)
                  (nth second transform_list)
                  (nth third transform_list)
                  (nth fourth transform_list))

                )

              )

            )

          )

        )

      )

    )

)

```

Appendix C

The following tables list all the stimuli presented in the acquisition and recognition phases of the experiment.

Acquisition Items

Stimuli	Base	Transform Sequence	Stimuli	Base	Transform Sequence
1	MIP1		1 2	SU24:MIP3:SU3:MAP2 SU23:SU10:MAP1:MIP3
1	SU16		1 2	SU18:SU24:MIP6:MIP3 MIP5:SU13:SU8:MIP4
1	SU20		1 2	MIP6:SU17:MIP4:SU23 SU18:MIP2:SU15:MIP6
1	SU9		1 2	SU17:SU15:MIP5:MIP3 SU12:MIP5:SU17:MAP2
1	SU15		1 2	MAP1:SU20:MIP4:SU11 SU6:MIP6:SU15:MIP2
1	DE4		2	SU23:SU10:MAP1
1	SU18		2	SU7:MIP1:SU12
1	SU24		2	SU11:SU8:MAP1
1	SU10		2	MIP6:MIP4:SU8
1	MIP2		2	MIP2:MIP1:SU18
1	SU16:MIP3		2	MAP1:SU6:MIP6
1	MAP2:SU10		2	MIP4:SU10:MIP1
1	MIP3:SU22		2	SU18:MIP2:SU15
1	MIP2:SU21		2	SU5:SU11:MIP3

Stimuli	Base	Transform Sequence	Stimuli	Base	Transform Sequence
	1	SU24:MIP3		2	MIP5:SU13:SU8
	1	SU19:MIP2		2	SU19:MIP5
	1	SU13:MIP3		2	SU9:SU19
	1	MIP1:SU16		2	SU7:MIP1
	1	MIP6:SU17		2	SU13:MAP1
	1	SU20:SU15		2	MIP4:SU10
	1	SU24:MIP3:SU3		2	MAP1:SU6
	1	MIP6:SU17:MIP4		2	DE3:SU10
	1	SU13:MIP3:SU22		2	MIP6:MIP4
	1	MIP2:SU21:SU16		2	SU11:SU8
	1	SU22:SU16:MIP6		2	SU6:MIP6
	1	MIP3:SU22:SU13		2	DE3
	1	SU10:MAP1:SU23		2	SU23
	1	MAP2:SU10:SU15		2	MAP1
	1	MIP5:MIP3:SU23		2	SU11
	1	SU16:MIP3:SU23		2	SU12

Stimuli	Base	Transform Sequence
1 2	SU20:SU15:MIP5:MIP1 MAP1:SU6:MIP6:SU12
1 2	MAP2:SU10:SU15:MIP3 MIP1:SU15:MIP2:SU8
1 2	DE4:SU19:MIP2:MIP1 DE4:SU10:MIP4:MIP2
1 2	SU15:MAP2:MIP3:SU11 SU11:SU8:MAP1:MIP3
1 2	SU19:MIP2:DE4:MAP2 SU10:MIP2:MAP2:DE3

Stimuli	Base	Transform Sequence
2	MIP2
2	SU6
2	SU19
2	MIP4
2	SU5

Recognition Items

Stimuli	Base	Transform Sequence	Stimuli	Base	Transform Sequence
' ♦ ★ ▲	1	SU9	♣ ♦ ■	1	SU19:MIP2:DE4
■ ♦ ★ ⊖	1	SU24	♣ ▲ ♠ ♦	1	SU17:SU15:MIP5
■ ' ▲ ★	1	MAP2:SU10	♠ ● ■ ★	1 2	SU22:SU16:MIP6:MIP2 MIP3:SU16:SU7:MIP1
♠ ♦ ★ ■	1	SU16:MIP3	▲ ● ♠ ★	1 2	SU13:MIP3:SU22:MIP6 SU13:MAP1:SU17:MIP6
● ♦ ■ ♠	1	MIP2:SU21:SU16	● ♠ ■ ★	2	MIP3:SU16:SU7
● ♦ × ■	1	SU24:MIP3:SU3	' ■ ★ ♣	2	SU6:MIP6:SU15
♦ ● ■ ×	1 2	SU24:MIP3:SU3:MAP2 SU23:SU10:MAP1:MIP3	♣ ▲ ' ●	2	SU18:MIP2
♦ ▲ ♠ ♣	1 2	SU17:SU15:MIP5:MIP3 SU12:MIP5:SU17:MAP2	■ ♠ ♦ ●	2	SU5:SU11
★ ● ♣ ■	2	MIP5:SU13:SU8	' ♠ ♣	2	DE4
♠ ▲ ' ●	2	MIP2:MIP1:SU18	♠ ' ♣ ●	2	MIP1
' ■ ● ♣	2	SU6:MIP6	■ ♦ ★ ▲	1	Base
♦ ♠ ▲ ●	2	SU9:SU19	■ ♦ ★ ▲	1	Base
■ ♠ ♣ ●	2	SU5	■ ♦ ★ ▲	1	Base
♣ ♠ ' ●	2	MIP2	■ ♦ ★ ▲	1	Base
■ ▲ ★ ♦	1	MIP5	' ♠ ♣ ●	2	Base

Stimuli	Base	Transform Sequence	Stimuli	Base	Transform Sequence
1	MIP4		2	Base
1	SU9:MAP2		2	Base
1	DE4:SU19		2	Base

Appendix D

This appendix lists all debriefing statements as typed by the subjects and are grouped by feedback condition. These statements are unedited and may contain both grammar and spelling errors.

100%/0% Feedback

1. *"In category one it usually contained the group with half circles. Category two usually contained the group with pointed shapes such as triangle or diamond."*
2. *"I COULD TELL THE DIFFERENCE BY WANT I GOT CORRECT ON THE FIRST PART."*
3. *"the first catagory usually consisted of shapes that had angles. The second catagory were rounded shapes."*
4. *"category 1 had mostly corners and category 2 had mostly rounded sides."*
5. *"i tried to memorize the symbols in each category . Then to tell which category it was I counted how many symbols were in that category and chose that one."*
6. *"I can tell by the club symbol, the comma, and the star. Usually, when there is a star present, it is the first group, and when there is the comma, or the club, it is usually the second group."*
7. *"if rounded shapes exceeded the number of edgy shapes ,it was category 2"*
8. *"From the complete shape if it was a original shape that was not alter ."*
9. *"there were some with organic shapes and half-cut polygons, while some were all polygonial"*
10. *"I tried to remember the figures and specially the squares and the circles."*
11. *"I tried to memorize the configuration and the sequence of the figures"*
12. *"I couldn't."*
13. *"TO ME THE DIFFERENCE BETWEEN THE TWO CATEGORIES WAS THE POSITION OF THE BLACK SPADE AND THE QUOTE MARK."*
14. *"catigory 1 I thought was all four or three had to have at least one side to them. catigory 2 was anything that did not have a side to it."*
15. *"1. Same symbols were used.
2. Partial symbols + full symbols were used together in the same way."*
16. *"Category one contained stars, traingles, and other sharp cornered objects. On the other*

hand, cat. two contained round objects which were easier to recognize and remember. I also realized when two shapes were from each category the lower ones were dominant."

17. *"I could tell the difference by what shapes were in the first row or the second row."*
18. *"I couldn't tell the difference between the two categories. I just guessed"*
19. *"In one category there was a majority of shapes with corners whereas in the other category, there was a majority of curves,"*
20. *"It seems that the figures were already set in my mind. There was no difficulty in trying to pattern the categories."*
21. *"by circle symbol"*
22. *"I could tell the difference by placing the star, circle and semi circle in a group."*
23. *"by the circles and stars in certain positions"*
24. *"One category had shapes of a star, triangle, tree, diamond, and a square. The second category had shapes of a sphere, half circle, and a comma."*
25. *"It was harder than I thought. The part 1 was ok, but the part 2 was hard. I didn't pay much attention to the pictures on part 1. Also, if I did pay attention to part 1, it will still be hard to tell if I have seen those pictures before in exact order. It was fun and interesting project."*
26. *"I could tell by how the geometric shapes related to each other."*
27. *"I could tell by looking at the side which the larger shape is on (whether it was on the 1 side or on the 2 side). also by the way the sharp edge and round edge shapes are arranged."*
28. *"A half a circle or an empty spot meant category 2. However, if there was a combination of a square, a circle or a triangle (2 items) with a half circle or the empty spot, it meant category 1."*

80%/20% Feedback

29. *"I ATTEMPTED TO ASSOCIATE THE GROUPS WITH STREET SIGNS AND PLAYING CARDS. GROUP 2 WAS EVERYTHING ELSE."*
30. *"I just guessed!"*
31. *"IT WAS HARD, BUT THE ONLY THING I COULD NOTICE WAS THAT CATEGORY 1 HAD MORE FIGURES WITH MORE LINES. FOR EXAMPLE, THE "STAR", THE "X", AND THE " , ". BASICALLY IT WAS A FIFTY, FIFTY CHANCE. I GUESS IT WAS BECAUSE I HAVE A LOT OF THINGS ON MY MIND AND MY MIND WAS DRIFTING AND NOT CONCENTRATING ON WHAT I WAS DOING."*
32. *"THE CATEGORIES WERE MATCHES AND AFTER SEEING THEM A COUPLE TIMES I STARTED TO OR THOUGHT I RECOGNIZED THEM."*

33. "I COULD TELL , BECAUSE ONE PART HAD FOUR SHAPES AS THE OTHER PART THREE SHAPES."
34. "I DETERMINED WHICH CATEGORY BY SEEING IF THE SHAPES WERE ROUNDED OR EDGED. OF COURSE, FOR SOME REASON, THIS DIDN'T ALWAYS WORK, WHICH IS WHY I GOT SOME OF THEM INCORRECT."
35. "WHEN BOTH TOP SYMBOLS CAN COVER OR BE COVERED BY THE BOTTOM ONES, THEY ARE BELONG TO GROUP 2. IF A TOP SYMBOL CAN COVER ITS BOTTOM IN A COLUMN, BUT THE OTHER TOP CAN'T, THEN THEY BELONG TO GROUP 1."
36. "NOT REALLY SURE BUT BY THE FIRST TWO SYMBOLS ON TOP"
37. "I COULD TELL THE DIFFERENCE BECAUSE SOME OF THE SHAPES HAD CURVED LINES WHILE OTHERS ONLY HAD STRAIGHT LINES"
38. "I CONCENTRATED ON THE POSITION OF THE STAR AND THE CLUB MOSTLY."
39. "I looked at the symbols, what they represented and I tried to figure it out which categories the symbols fit in. Still, I couldn't get it right!!"
40. "I think that most of shapes in category 1 were sharp angular shapes, and most of the shapes in categories 2 were more rounded."
41. "I think that one of the category was for complete shapes, with no half circles and the other was of complete shapes."
42. "The way I basically did it was just by seeing which category dominated- roundness or sharp edges."
43. "when the figure is rounded it most likely to belong to cat. 2."
44. "the round shapes and the angular shapes"
45. "I used the more unique symbols to distinguish with ones I had seen before."
46. "sometimes I would concentrate only one half of the symbols and then vaguely remember the other symbols. Sometimes I would try to concentrate on the whole set of symbols, but after a while it got confusing since the same symbols kept appearing although rearrange in different sets."
47. "I looked at the half circle and wi"
48. "In part one, I categorized it by seeing the half circle. In part two I categorized it by seeing the club."
49. "I never really felt like I picked up on the difference between the two categories. At first I thought I understood what they were, then I would start getting many wrong when I applied my ideas. I think it may have been the way the figures were set up. (e.g. basic geometrical

shapes vs. stars, commas, etc.)"

50. *"The signs are similar and so difficult to remember their combinations."*
51. *"At first I tried to focus on specific symbols then I tried to focus on their position. The ones with only three symbols threw me off because I did not expect them."*
52. *"You couldn't tell the difference between the categories. The only way I can see it was by the shape of the given figures."*
53. *"I could distinguish the difference by the pattern of the shapes, and by the combination and placement of the shapes. Also, I made my choice based on the shapes themselves."*
54. *"By seeing which one the figures turned up in which category the most and The ones with a lot of curves seemed to be in category #2 most of the time."*
55. *"I THINK IT IS SOMEHOW DIFFICULT BECAUSE IT SEEMS LIKE ALL THE FIGURES ARE THE SAME."*
56. *"I don't think that I ever really got it. It seemed like the side with the circle on it was usually right. A star over an ' usually was right too. Often times it seemed that the side with the most total black area was right."*

60%/40% Feedback

57. *"THE FIRST CATAGORY WAS HARD. I MOSTLY GUESSED BUT THE SECOND PART WAS HARD BECAUSE I HAD TO RECALL THE INFORMATION."*
58. *"I couldn't tell the difference at all. I could not see any evident pattern in the way the different shapes were placed, and to what category they belonged."*
59. *"I separated the categories in Part 1 by round or not round."*
60. *"LOOKED FOR A PATTREN OF FREQUINCIES THAT WOULD AND WAS ABLE TO LOOK AT THE COLEN AND SOME WHAT DETERMINE THAT THE GROUP WOULD BE IN CAT. 1. BUT RAN OUT EXAMPLES AND TIME TO MAKE ANY MORE CONCLUSION."*
61. *"I have no idea!!!!"*
62. *"I FOUND THAT I COULDN'T MAKE OUT ANY CORRELATION BETWEEN THE TWO CATEGORIES. I FIGURED THAT IT WAS BASED ON THE SHAPES, BUT THAT DIDN'T SEEM TO WORK."*
63. *"I LOOK AT THE DIAMOND SHAPE. MOST OF CATERGORY 2 CONTAINS DIAMONDS. MOST OF THE OTHERS , I GUESS"*
64. *"The first category had distinct shapes such as the whole circle. The second category had odd symbols like the comma."*
65. *"If one side had only one picture, I choose the other side. I choose the star and circle side if I*

saw them."

66. *"if it looks weird by first impression"*
67. *"Mainly you have to differentiate the way the objects were placed at different places."*
68. *"It seemed that the first category was asking for symmetrical and rounded objects, also for non-symmetrical items. the second category was looking for sharp corned objects that were symmetrical in two ways."*
69. *"Category 1 contained the triangle and the diamond or the square and the diamond. Category 2 contained the comma and the circle or half-square and the comma."*
70. *"By the frequency a symbol appeared in one category and if it was correct or incorrect."*
71. *"I could not tell the difference between the two categories. I try to group them according to the side/edge, but it seem not right."*
72. *"part I was easier than part II."*
73. *"I concentrated on the similarity between the answer I got correct and the one I got wrong."*
74. *"I could tell the difference between the 2 categories by the way the shapes looked (round or straight), the specific shapes I remembered, and the number of shapes."*
75. *"I could tell the differences because in part one there were a lot of clubs in the shapes that were shown but in part two there were a lot of stars."*
76. *"don't know the difference"*
77. *"The way they were positioned."*
78. *"I JUST WENT FROM MEMORY.I TRIED TO REMEMBER WHAT I SAW."*
79. *"CERTAIN OBJECTS IN A GROUP WERE IN EITHER CATEGORIES, AND CERTAIN PATTERNS CAME UP."*
80. *"PART ONE IS SIMILAR TO PART TWO, THE DIRRERENT ARE THE THE WAY IS SET UP ."*
81. *"I COULDN'T TELL A PART EASILY."*
82. *"Unfortunately I never established a relationship that seemed consistant although I experimented with quite a few possibilities."*
83. *"I am not too sure, but I related the symbols in certain order."*